

https://dspace.org/

A tree grammar-based visual password scheme

Okundaye, Benjamin

2016
http://hdl.handle.net/10539/19294

Downloaded from a DSpace repository. https://dspace.org/

A Tree Grammar-based Visual Password Scheme



Benjamin Okundaye

School of Computer Science

University of the Witwatersrand, Johannesburg

A PhD Research Thesis

Supervisors

Prof. S. Ewert and Prof. I. Sanders

August 31, 2015

Dedication

I dedicate this thesis to my loving family.

Declaration

I hereby declare that except where specific reference is made to the work of others, the contents of this dissertation are original and have not been submitted in whole or in part for consideration for any other degree or qualification in this, or any other university. This dissertation is my own work and contains nothing which is the outcome of work done in collaboration with others, except as specified in the text and Acknowledgements.



Benjamin Okundaye
August 31, 2015

Acknowledgements

I would like to thank my supervisors, Prof. Sigrid Ewert and Prof. Ian Sanders, for their intellectually stimulating suggestions and encouragement throughout the writing of this research thesis. I would also like to acknowledge the contributions of my co-authors, who are incidentally my supervisors, in the papers presented at PRASA 2013 (Okundaye *et al.*, 2013), SAICSIT 2014 (Okundaye *et al.*, 2014b), PRASA 2014 (Okundaye *et al.*, 2014a) and others yet to be published. I would also like to say a big thank you to Frank Drewes for his open source software, Treebag and its associated tree grammars used in this study.

Abstract

Visual password schemes can be considered as an alternative to alphanumeric passwords. Studies have shown that alphanumeric passwords can, amongst others, be eavesdropped, shoulder surfed, or guessed, and are susceptible to brute force automated attacks. Visual password schemes use images, in place of alphanumeric characters, for authentication. For example, users of visual password schemes either select images (Cognometric) or points on an image (Locimetric) or attempt to redraw their password image (Drawmetric), in order to gain authentication. Visual passwords are limited by the so-called password space, i.e., by the size of the alphabet from which users can draw to create a password and by susceptibility to stealing of pass-images by someone looking over your shoulders, referred to as shoulder surfing in the literature. The use of automatically generated highly similar abstract images defeats shoulder surfing and means that an almost unlimited pool of images is available for use in a visual password scheme, thus also overcoming the issue of limited potential password space.

This research investigated visual password schemes. In particular, this study looked at the possibility of using tree picture grammars to generate abstract graphics for use in a visual password scheme. In this work, we also took a look at how humans determine similarity of abstract computer generated images, referred to as perceptual similarity in the literature. We drew on the psychological idea of similarity and matched that as closely as possible with a mathematical measure of image similarity, using Content Based Image Retrieval (CBIR) and tree edit distance measures. To this end, an online similarity survey was conducted with respondents ordering answer images in order

of similarity to question images, involving 661 respondents and 50 images. The survey images were also compared with eight, state of the art, computer based similarity measures to determine how closely they model perceptual similarity. Since all the images were generated with tree grammars, the most popular measure of tree similarity, the tree edit distance, was also used to compare the images. Eight different types of tree edit distance measures were used in order to cover the broad range of tree edit distance and tree edit distance approximation methods. All the computer based similarity methods were then correlated with the online similarity survey results, to determine which ones more closely model perceptual similarity. The results were then analysed in the light of some modern psychological theories of perceptual similarity.

This work represents a novel approach to the Passfaces type of visual password schemes using dynamically generated pass-images and their highly similar distractors, instead of static pictures stored in an online database. The results of the online survey were then accurately modelled using the most suitable tree edit distance measure, in order to automate the determination of similarity of our generated distractor images. The information gathered from our various experiments was then used in the design of a prototype visual password scheme. The generated images were similar, but not identical, in order to defeat shoulder surfing. This approach overcomes the following problems with this category of visual password schemes: shoulder surfing, bias in image selection, selection of easy to guess pictures and infrastructural limitations like large picture databases, network speed and database security issues. The resulting prototype developed is highly secure, resilient to shoulder surfing and easy for humans to use, and overcomes the aforementioned limitations in this category of visual password schemes.

Contents

Contents vii

List of Figures xxi

List of Tables xxxi

Nomenclature xxxv

1 Introduction 1

1.1 Introduction . . . 1
1.2 Research aims and objectives . . . 2
1.3 Research questions . . . 3
1.4 Research method . . . 4
1.5 Results . . . 4
1.6 Scope and limitations . . . 5
1.7 Significance . . . 5
1.8 Structure of the document . . . 6

2 Literature review 9

2.1 Introduction . . . 9
2.2 Authentication methods . . . 10
2.2.1 The three major authentication methods . . . 11
2.2.1.1 Biometric authentication . . . 11
2.2.1.2 Token based authentication . . . 11
2.2.1.3 Knowledge based authentication . . . 12
2.2.2 Alphanumeric and visual password schemes . . . 13

2.3 Recent developments . . . 21
2.4 Mathematical models of similarity . . . 27
2.4.1 Feature extraction methods . . . 27
2.4.1.1 Colour features . . . 27
2.4.1.2 Texture features . . . 28
2.4.1.3 Shape features . . . 29
2.4.1.4 Salient points . . . 29
2.4.1.5 Compound or composite descriptors . . . 29
2.4.2 Core techniques used in CBIR . . . 30
2.4.2.1 Visual signatures . . . 30
2.4.2.2 Similarity measures . . . 32
2.5 Tree grammars and syntactic picture generation . . . 33
2.5.1 Signatures, trees and notations . . . 33
2.5.2 Tree grammar definitions . . . 34
2.6 Perceptual similarity . . . 37
2.6.1 Theories of perceptual similarity . . . 40
2.6.1.1 Common element approach or alignment based models . . . 40
2.6.1.2 Template or transformational models . . . 40
2.6.1.3 Geometric models . . . 41
2.6.1.4 Feature-contrast models . . . 42
2.6.1.5 Geon model . . . 44
2.6.2 Experimental measurement and computation of similarity 45
2.6.2.1 Stimulus generalization . . . 46
2.6.2.2 Multidimensional scaling . . . 46
2.6.2.3 Cluster analysis . . . 48
2.6.2.4 Summary of perceptual similarity . . . 50
2.7 Summary of literature review . . . 51

3 Research Method 53
3.1 Introduction . . . 53
3.2 Tree Grammar-based picture generation . . . 54
3.2.1 Trees . . . 54

3.2.2 Tree transducers . . . 55
3.2.3 Algebras . . . 55
3.2.4 Display . . . 56
3.2.5 Tree picture grammars implemented in this study . . . 57
3.2.5.1 Regular tree grammars . . . 57
3.2.5.2 Parallel deterministic total tree grammars (pdt) . . . 58
3.2.5.3 ET0L tree grammars . . . 59
3.2.5.4 Branching tree grammars . . . 60
3.3 Perceptual similarity . . . 62
3.4 Similarity methods used in this study . . . 63
3.4.1 Tree edit distances . . . 63
3.4.1.1 General tree edit distance (ED) . . . 64
3.4.1.2 Fanout weighted (FW), pq-gram (PQ) and Windowed pq-gram (WPG) distance measures . . . 65
3.4.1.3 Tree embedding distance (TE) . . . 67
3.4.1.4 Binary branch distance (BB) . . . 67
3.4.1.5 Bottom-up tree edit distance (BU) . . . 68
3.4.1.6 Fullpath tree edit distance (FP) . . . 68
3.4.2 CBIR measures used in this study . . . 68
3.4.2.1 Colour Histogram . . . 70
3.4.2.2 Auto-Correlograms . . . 71
3.4.2.3 Tamura Texture Features . . . 71
3.4.2.4 Edge Histogram Descriptor . . . 73
3.4.2.5 Colour Layout Descriptor . . . 74
3.4.2.6 Colour and Edge Directivity Descriptor and Fuzzy Colour and Texture Histogram . . . 75
3.4.2.7 Spatial Colour Distribution Descriptor . . . 76
3.5 Visual password scheme . . . 77
3.6 Data collection method . . . 77
3.6.1 Sample size . . . 77
3.6.2 Research instruments . . . 78
3.7 Summary of research methods . . . 79

4 Similarity Measures **81**
4.1 Experimental Determination of Appropriate Similarity Measures . 81
4.2 Background to the study 83
4.2.1 Picture grammars 83
4.2.2 Tree edit distance 83
4.2.3 Visual descriptors 84
4.3 Experiments and Results 85
4.3.1 Experimental Studies 85
4.3.1.1 Experiment on first set of data 85
4.3.1.2 Experiment on second set of data 86
4.3.1.3 Experiment on third set of data 87
4.3.2 Results 87
4.3.2.1 Results for data set 1 87
4.3.2.2 Results for data set 2 92
4.3.2.3 Results for data set 3 94
4.3.2.4 Correlation studies 95
4.4 Summary 96

5 Perceptual Similarity and Computer-based Similarity Measures **99**
5.1 Introduction 99
5.2 Background to the study 100
5.3 Experiments 101
5.3.1 Perceptual similarity study 103
5.3.2 Cluster analysis 105
5.3.3 CBIR similarity experiments 105
5.3.4 Tree edit distances experiments 107
5.4 Results 107
5.4.1 Results of perceptual similarity study 107
5.4.2 Results of CBIR similarity experiments for perceptual similarity images 109
5.4.3 Results of tree edit distance measurements experiment for perceptual similarity images 113
5.5 Correlation studies 115

5.5.1 Correlation of perceptual similarity, tree edit distances and CBIR similarity measures . . . 117
5.5.2 Tversky's feature-contrast model and tree edit distance . . 117
5.6 Summary . . . 119

6 Design and Implementation of a Novel Visual Password Scheme 121
6.1 Introduction . . . 121
6.2 Background to this study . . . 122
6.3 Design and implementation . . . 123
6.3.1 Determination of similarity . . . 123
6.3.2 Grammar design . . . 125
6.3.3 Design and implementation of the system . . . 126
6.3.4 Scalability and timing . . . 129
6.4 Summary . . . 134

7 Experimental characterisation of tree picture grammars used in a visual password scheme 139
7.1 Introduction . . . 139
7.2 Similarity threshold value experiments . . . 139
7.3 Results . . . 141
7.3.1 Symmetrical tree picture grammars . . . 147
7.3.2 Non-symmetrical tree picture grammars . . . 148
7.3.3 Hybrid tree picture grammars . . . 152
7.3.4 Goodness of fit plot for normalised tree edit distance values of grammars . . . 152
7.3.5 Grammar type limitations . . . 154
7.3.5.1 Symmetrical grammars . . . 154
7.3.5.2 Non-symmetrical grammars . . . 158
7.3.6 General observations for all the grammars . . . 158
7.4 Summary . . . 159

8 Research Discussion 163
8.1 Introduction . . . 163
8.1.1 Shoulder surfing . . . 164

8.1.2 Description attack . . . 165
8.1.3 Bias in image selection . . . 165
8.1.4 Password space . . . 165
8.1.5 Infrastructural and network limitations . . . 166
8.1.6 Internet and large client/server environment . . . 166
8.2 Security . . . 167
8.2.1 Shoulder surfing . . . 167
8.2.1.1 Shoulder surfing by simple observation . . . 167
8.2.1.2 Shoulder surfing through the use of image capture devices . . . 168
8.2.2 Social Engineering . . . 169
8.2.2.1 Divulging pass-images through description . . . 169
8.2.2.2 Phishing . . . 171
8.2.2.3 User input simulation . . . 172
8.2.3 Guessing . . . 172
8.2.4 Operating System and infrastructural vulnerabilities . . . 174
8.2.4.1 Local access . . . 174
8.2.4.2 Internet access . . . 174
8.2.5 Other attack vectors . . . 175
8.3 User Experience . . . 175
8.3.1 Satisfaction . . . 175
8.3.1.1 Ease of use . . . 175
8.3.1.2 Ease of creation . . . 176
8.3.1.3 Memorability . . . 176
8.3.1.4 Ease of execution . . . 177
8.3.1.5 Execution speed . . . 177
8.3.1.6 User interface . . . 178
8.3.2 Effectiveness - Reliability . . . 178
8.3.3 Efficiency - Acceptability . . . 179
8.4 Prototype Overview . . . 179
8.4.1 Grammar design . . . 180
8.5 Research Overview . . . 181
8.5.1 The use of tree grammars . . . 181

8.5.2 Production rule granularity . . . 183
8.5.2.1 Granularity using nested tables . . . 183
8.5.2.2 Granularity using direct modification of production rules . . . 186
8.5.3 Choice of similarity measures . . . 187
8.5.4 Good and bad images . . . 188
8.5.5 Undone tasks . . . 189
8.5.6 Research weakness . . . 190
8.5.7 Research strengths . . . 191
8.5.8 What could have been done differently? . . . 191
8.6 Summary . . . 192

9 Further Studies **193**
9.1 Introduction . . . 193
9.2 The Passfaces type visual password scheme . . . 193
9.2.1 Modifications to the prototype . . . 193
9.2.2 Alternative design prototype . . . 194
9.3 Further studies in related visual password systems . . . 196
9.3.1 Passpoint type visual password system . . . 196
9.3.2 Hybrid visual password systems . . . 196
9.3.2.1 Partially opaque sub-images . . . 196
9.3.2.2 Multiple pass-image selection . . . 196
9.4 Summary . . . 197

10 Summary and Conclusion **199**
10.1 Contributions . . . 200
10.2 Conclusion . . . 201

References **205**

A Sample Tree Grammars, Images and Trees **227**
A.1 Sample Grid Picture Grammar (1) . . . 227
A.1.1 Grammar . . . 227
A.1.2 Algebra . . . 229

A.2 Another Sample Grid Picture Grammar 230
A.2.1 Grammar 230
A.2.2 Algebra 231
A.2.3 Generated grammar 232
A.2.4 Generated transducer 233
A.2.5 Sample trees for previous grammar 238
A.2.6 Sample images for previous grammar 240
A.3 Sample Branching Synchronization Grammar (16) 242
A.3.1 Grammar 242
A.3.2 Algebra 242
A.3.3 Generated Grammar 244
A.3.4 Generated Transducer 245

B Sample On-line Survey 247

C Tree Grammar-based Generated Pictures 257
C.1 Sample Barnsley Images 258
C.2 Sample Mandelbrot Curves 261

D Images for Execution of Prototype in Order to get Similarity Threshold Values 265
D.1 First Sixteen Images For Each Grammar 265
D.2 Threshold run values 278
D.2.1 Normalised similarity values for Grammar 0 278
D.2.1.1 After Sorting 278
D.2.1.2 After Shuffling 278
D.2.1.3 After Sorting 278
D.2.1.4 After Shuffling 279
D.2.1.5 After Sorting 279
D.2.1.6 After Shuffling 279
D.2.2 Normalised similarity values for Grammar 1 280
D.2.2.1 After Sorting 280
D.2.2.2 After Shuffling 280
D.2.2.3 After Sorting 280

D.2.2.4 After Shuffling . . . 281
D.2.2.5 After Sorting . . . 281
D.2.2.6 After Shuffling . . . 281
D.2.3 Normalised similarity values for Grammar 2 . . . 282
D.2.3.1 After Sorting . . . 282
D.2.3.2 After Shuffling . . . 282
D.2.3.3 After Sorting . . . 282
D.2.3.4 After Shuffling . . . 283
D.2.3.5 After Sorting . . . 283
D.2.3.6 After Shuffling . . . 283
D.2.4 Normalised similarity values for Grammar 3 . . . 284
D.2.4.1 After Sorting . . . 284
D.2.4.2 After Shuffling . . . 284
D.2.4.3 After Sorting . . . 284
D.2.4.4 After Shuffling . . . 285
D.2.4.5 After Sorting . . . 285
D.2.4.6 After Shuffling . . . 285
D.2.5 Normalised similarity values for Grammar 4 . . . 286
D.2.5.1 After Sorting . . . 286
D.2.5.2 After Shuffling . . . 286
D.2.5.3 After Sorting . . . 286
D.2.5.4 After Shuffling . . . 287
D.2.5.5 After Sorting . . . 287
D.2.5.6 After Shuffling . . . 287
D.2.6 Normalised similarity values for Grammar 5 . . . 288
D.2.6.1 After Sorting . . . 288
D.2.6.2 After Shuffling . . . 288
D.2.6.3 After Sorting . . . 288
D.2.6.4 After Shuffling . . . 289
D.2.6.5 After Sorting . . . 289
D.2.6.6 After Shuffling . . . 289
D.2.7 Normalised similarity values for Grammar 6 . . . 290
D.2.7.1 After Sorting . . . 290

D.2.7.2 After Shuffling . . . 290
D.2.7.3 After Sorting . . . 290
D.2.7.4 After Shuffling . . . 291
D.2.7.5 After Sorting . . . 291
D.2.7.6 After Shuffling . . . 291
D.2.8 Normalised similarity values for Grammar 7 . . . 292
D.2.8.1 After Sorting . . . 292
D.2.8.2 After Shuffling . . . 292
D.2.8.3 After Sorting . . . 292
D.2.8.4 After Shuffling . . . 293
D.2.8.5 After Sorting . . . 293
D.2.8.6 After Shuffling . . . 293
D.2.9 Normalised similarity values for Grammar 8 . . . 294
D.2.9.1 After Sorting . . . 294
D.2.9.2 After Shuffling . . . 294
D.2.9.3 After Sorting . . . 294
D.2.9.4 After Shuffling . . . 295
D.2.9.5 After Sorting . . . 295
D.2.9.6 After Shuffling . . . 295
D.2.10 Normalised similarity values for Grammar 9 . . . 296
D.2.10.1 After Sorting . . . 296
D.2.10.2 After Shuffling . . . 296
D.2.10.3 After Sorting . . . 296
D.2.10.4 After Shuffling . . . 297
D.2.10.5 After Sorting . . . 297
D.2.10.6 After Shuffling . . . 297
D.2.11 Normalised similarity values for Grammar 10 . . . 298
D.2.11.1 After Sorting . . . 298
D.2.11.2 After Shuffling . . . 298
D.2.11.3 After Sorting . . . 298
D.2.11.4 After Shuffling . . . 299
D.2.11.5 After Sorting . . . 299
D.2.11.6 After Shuffling . . . 299

D.2.12 Normalised similarity values for Grammar 11 300
D.2.12.1 After Sorting 300
D.2.12.2 After Shuffling 300
D.2.12.3 After Sorting 300
D.2.12.4 After Shuffling 301
D.2.12.5 After Sorting 301
D.2.12.6 After Shuffling 301
D.2.13 Normalised similarity values for Grammar 12 302
D.2.13.1 After Sorting 302
D.2.13.2 After Shuffling 302
D.2.13.3 After Sorting 302
D.2.13.4 After Shuffling 303
D.2.13.5 After Sorting 303
D.2.13.6 After Shuffling 303
D.2.14 Normalised similarity values for Grammar 13 304
D.2.14.1 After Sorting 304
D.2.14.2 After Shuffling 304
D.2.14.3 After Sorting 304
D.2.14.4 After Shuffling 305
D.2.14.5 After Sorting 305
D.2.14.6 After Shuffling 305
D.2.15 Normalised similarity values for Grammar 14 306
D.2.15.1 After Sorting 306
D.2.15.2 After Shuffling 306
D.2.15.3 After Sorting 306
D.2.15.4 After Shuffling 307
D.2.15.5 After Sorting 307
D.2.15.6 After Shuffling 307
D.2.16 Normalised similarity values for Grammar 15 308
D.2.16.1 After Shuffling 308
D.2.16.2 After Sorting 308
D.2.16.3 After Shuffling 309
D.2.16.4 After Sorting 309

D.2.16.5 After Shuffling . . . 309
D.2.17 Normalised similarity values for Grammar 16 . . . 310
D.2.17.1 After Sorting . . . 310
D.2.17.2 After Shuffling . . . 310
D.2.17.3 After Sorting . . . 310
D.2.17.4 After Shuffling . . . 311
D.2.17.5 After Sorting . . . 311
D.2.17.6 After Shuffling . . . 311
D.2.18 Normalised similarity values for Grammar 17 . . . 312
D.2.18.1 After Sorting . . . 312
D.2.18.2 After Shuffling . . . 312
D.2.18.3 After Sorting . . . 312
D.2.18.4 After Shuffling . . . 313
D.2.18.5 After Sorting . . . 313
D.2.18.6 After Shuffling . . . 313
D.2.19 Normalised similarity values for Grammar 18 . . . 314
D.2.19.1 After Sorting . . . 314
D.2.19.2 After Shuffling . . . 314
D.2.19.3 After Sorting . . . 314
D.2.19.4 After Shuffling . . . 315
D.2.19.5 After Sorting . . . 315
D.2.19.6 After Shuffling . . . 315
D.2.20 Normalised similarity values for Grammar 19 . . . 316
D.2.20.1 After Sorting . . . 316
D.2.20.2 After Shuffling . . . 316
D.2.20.3 After Sorting . . . 316
D.2.20.4 After Shuffling . . . 317
D.2.20.5 After Sorting . . . 317
D.2.20.6 After Shuffling . . . 317
D.2.21 Normalised similarity values for Grammar 20 . . . 318
D.2.21.1 After Sorting . . . 318
D.2.21.2 After Shuffling . . . 318
D.2.21.3 After Sorting . . . 318

D.2.21.4 After Shuffling . . . 319
D.2.21.5 After Sorting . . . 319
D.2.21.6 After Shuffling . . . 319
D.2.22 Normalised similarity values for Grammar 21 . . . 320
D.2.22.1 After Sorting . . . 320
D.2.22.2 After Shuffling . . . 320
D.2.22.3 After Sorting . . . 320
D.2.22.4 After Shuffling . . . 321
D.2.22.5 After Sorting . . . 321
D.2.22.6 After Shuffling . . . 321
D.2.23 Normalised similarity values for Grammar 22 . . . 322
D.2.23.1 After Sorting . . . 322
D.2.23.2 After Shuffling . . . 322
D.2.23.3 After Sorting . . . 322
D.2.23.4 After Shuffling . . . 323
D.2.23.5 After Sorting . . . 323
D.2.23.6 After Shuffling . . . 323
D.2.24 Normalised similarity values for Grammar 23 . . . 324
D.2.24.1 After Sorting . . . 324
D.2.24.2 After Shuffling . . . 324
D.2.24.3 After Sorting . . . 324
D.2.24.4 After Shuffling . . . 325
D.2.24.5 After Sorting . . . 325
D.2.24.6 After Shuffling . . . 325
D.3 Threshold run images . . . 326

E Clustering of Online similarity Data **345**
E.1 I Images Clusters . . . 345
E.2 E to T Images Clusters . . . 352
E.3 Fern Images Clusters . . . 353
E.4 Spiral Images Clusters . . . 359
E.5 Mosaic Images Clusters . . . 363
E.6 Squares Images Clusters . . . 367

F Graphs of goodness test fit of displayed images for all grammars 372

F.1 Bottom-up tree edit distance values for all grammars 372

F.2 Similarity threshold and bottom-up tree edit distance values for grammars 397

F.3 Goodness of fit graphs for all grammars 422

F.4 Table of goodness of fit values for all grammars 456

List of Figures

2.1 Pass-image selection types (Jebriel & Poet, 2011) 16
2.2 Pressure-Grid finger positions (Kim *et al.*, 2010) 17
2.3 Pressure-Grid study context (Kim *et al.*, 2010) 18
2.4 Viewport of persuasive cued click points (Chiasson *et al.*, 2008) . 21
2.5 An overview of image signature formulation (Datta *et al.*, 2008) . 31
2.6 Two lines of the same length appear different 38
2.7 Geons and related objects (Biederman, 1987) 45

3.1 Abstract syntax tree form 55
3.2 Tree transformation by transducer 56
3.3 TREEBAG system flow 56
3.4 Tree edit operations 65
3.5 Recursive formula for tree edit distance 66
3.6 Tree edit distance with unit cost ($dist_{unit}$) and Fanout Weighted ($dist_{ed}$) cost model (Pawlik & Augsten, 2011). 66
3.7 CEDD and FCTH flowchart 75

4.1 Production rules for letter I 86
4.2 Various versions of I used as set 1 in this study 86
4.3 Various letters used as set 2 in this study 86
4.4 Series 1: Barnsley fern 87
4.5 Series 2: Spiral 87
4.6 Series 3: Rotating squares 88
4.7 Series 4: Colour operations 88
4.8 Series 5: Spirals2 88

4.9 Series 6: Circles . . . 88
4.10 Series 7: Mosaic1 . . . 88
4.11 Series 8: Mosaic2 . . . 89
4.12 Series 9: Parametric Colour Operation . . . 89
4.13 Series 10: Triangles . . . 89
4.14 Graph for data set 1 . . . 91
4.15 Graph for data set 2 (F, L, I) . . . 92
4.16 Graph for data set 2 (T and H) . . . 93

5.1 Production rules for letter I . . . 102
5.2 Various versions of I used as set 1 in this study . . . 102
5.3 Various letters used as set 2 in this study . . . 102
5.4 Barnsley fern . . . 103
5.5 Spiral . . . 103
5.6 Mosaic . . . 103
5.7 Rotating squares . . . 103
5.8 Graph of survey results versus CBIR distance measures for Table 5.5 (Barnsley fern) . . . 110
5.9 Graph of survey results and tree edit distance measures for Table 5.11 (Barnsley fern) . . . 111
5.10 Graph of survey results and CBIR measures for Table 5.6 (Spiral) 114
5.11 Graph of survey results and tree edit distance measures for Table 5.12 (Spiral) . . . 114
5.12 Tree edit distance as a special case of Tversky's feature-contrast model . . . 118

6.1 Login / Add Pass-images / Registration screen for prototype . . . 124
6.2 Addition of authentication rounds dialog box . . . 124
6.3 Parallel threads/processes . . . 125
6.4 UML class diagram for class vps . . . 130
6.5 UML class diagram for class term . . . 131
6.6 UML class diagram for class matchtrees . . . 132
6.7 Grammar gallery and a three step authentication process . . . 136

7.1 Column graph for threshold values 142
7.2 First twelve grammars gallery 143
7.3 Second twelve grammars gallery 144
7.4 Similarity threshold values for grammar 0 149
7.5 Bottom-up tree edit distance values for grammar 0 for Pass-images 0 to 16 149
7.6 Similarity threshold values for grammar 21 150
7.7 Bottom-up tree edit distance values for grammar 21 for pass-images 0 to 16 150
7.8 Similarity threshold values for grammar 18 151
7.9 Bottom-up tree edit distance values for grammar 18 for pass-images 0 to 16 151
7.10 Goodness of fit graph for displayed images for grammar 0 for pass-image 5 155
7.11 Goodness of fit graph for displayed images for grammar 0 for pass-image 8 155
7.12 Goodness of fit graph for displayed images for grammar 0 for pass-image 8 and polynomial order 6 155
7.13 Goodness of fit graph for displayed images for grammar 0 for pass-images 9 to 16 156
7.14 Goodness of fit graph for displayed images for grammar 21 for pass-image 5 156
7.15 Goodness of fit graph for displayed images for grammar 21 for pass-image 8 157
7.16 Goodness of fit graph for displayed images for grammar 21 for pass-images 9 to 16 157

A.1 Sample trees for previous grammar 239
A.2 ET0L Grammar images 241

D.1 First sixteen images for grammar 0 266
D.2 First sixteen images for grammar 1 266
D.3 First sixteen images for grammar 2 267
D.4 First sixteen images for grammar 3 267

D.5 First sixteen images for grammar 4 268
D.6 First sixteen images for grammar 5 268
D.7 First sixteen images for grammar 6 269
D.8 First sixteen images for grammar 7 269
D.9 First sixteen images for grammar 8 270
D.10 First sixteen images for grammar 9 270
D.11 First sixteen images for grammar 10 271
D.12 First sixteen images for grammar 11 271
D.13 First sixteen images for grammar 12 272
D.14 First sixteen images for grammar 13 272
D.15 First sixteen images for grammar 14 273
D.16 First sixteen images for grammar 15 273
D.17 First sixteen images for grammar 16 274
D.18 First sixteen images for grammar 17 274
D.19 First sixteen images for grammar 18 275
D.20 First sixteen images for grammar 19 275
D.21 First sixteen images for grammar 20 276
D.22 First sixteen images for grammar 21 276
D.23 First sixteen images for grammar 22 277
D.24 First sixteen images for grammar 23 277
D.25 Run of prototype to get similarity threshold values 327
D.26 Run of prototype to get similarity threshold values 328
D.27 Run of prototype to get similarity threshold values 329
D.28 Run of prototype to get similarity threshold values 330
D.29 Run of prototype to get similarity threshold values 331
D.30 Run of prototype to get similarity threshold values 332
D.31 Run of prototype to get similarity threshold values 333
D.32 Run of prototype to get similarity threshold values 334
D.33 Run of prototype to get similarity threshold values 335
D.34 Run of prototype to get similarity threshold values 336
D.35 Run of prototype to get similarity threshold values 337
D.36 Run of prototype to get similarity threshold values 338
D.37 Run of prototype to get similarity threshold values 339

D.38 Run of prototype to get similarity threshold values 340
D.39 Run of prototype to get similarity threshold values 341
D.40 Run of prototype to get similarity threshold values 342
D.41 Run of prototype to get similarity threshold values 343
D.42 Run of prototype to get similarity threshold values 344

F.1 Similarity threshold values for grammar 0 398
F.2 Bottom-up tree edit distance values for grammar 0 for pass-images 0 to 16 399
F.3 Similarity threshold values for grammar 1 400
F.4 Bottom-up tree edit distance values for grammar 1 for pass-images 0 to 16 401
F.5 Similarity threshold values for grammar 2 402
F.6 Bottom-up tree edit distance values for grammar 2 for pass-images 0 to 16 402
F.7 Similarity threshold values for grammar 3 403
F.8 Bottom-up tree edit distance values for grammar 3 for pass-images 0 to 16 403
F.9 Similarity threshold values for grammar 4 404
F.10 Bottom-up tree edit distance values for grammar 4 for pass-images 0 to 16 404
F.11 Similarity threshold values for grammar 5 405
F.12 Bottom-up tree edit distance values for grammar 5 for pass-images 0 to 16 405
F.13 Similarity threshold values for grammar 6 406
F.14 Bottom-up tree edit distance values for grammar 6 for pass-images 0 to 16 406
F.15 Similarity threshold values for grammar 7 407
F.16 Bottom-up tree edit distance values for grammar 7 for pass-images 0 to 16 407
F.17 Similarity threshold values for grammar 8 408
F.18 Bottom-up tree edit distance values for grammar 8 for pass-images 0 to 16 408

F.19 Similarity threshold values for grammar 9 409
F.20 Bottom-up tree edit distance values for grammar 9 for pass-images 0 to 16 409
F.21 Similarity threshold values for grammar 10 410
F.22 Bottom-up tree edit distance values for grammar 10 for pass-images 0 to 16 410
F.23 Similarity threshold values for grammar 11 411
F.24 Bottom-up tree edit distance values for grammar 11 for pass-images 0 to 16 411
F.25 Similarity threshold values for grammar 12 412
F.26 Bottom-up tree edit distance values for grammar 12 for pass-images 0 to 16 412
F.27 Similarity threshold values for grammar 13 413
F.28 Bottom-up tree edit distance values for grammar 13 for pass-images 0 to 16 413
F.29 Similarity threshold values for grammar 14 414
F.30 Bottom-up tree edit distance values for grammar 14 for pass-images 0 to 16 415
F.31 Similarity threshold values for grammar 15 416
F.32 Bottom-up tree edit distance values for grammar 15 for pass-images 0 to 16 417
F.33 Similarity threshold values for grammar 16 418
F.34 Bottom-up tree edit distance values for grammar 16 for pass-images 0 to 16 419
F.35 Similarity threshold values for grammar 17 420
F.36 Bottom-up tree edit distance values for grammar 17 for pass-images 0 to 16 421
F.37 Similarity threshold values for grammar 18 422
F.38 Bottom-up tree edit distance values for grammar 18 for pass-images 0 to 16 423
F.39 Similarity threshold values for grammar 19 424
F.40 Bottom-up tree edit distance values for grammar 19 for pass-images 0 to 16 425

F.41 Similarity threshold values for grammar 20 426
F.42 Bottom-up tree edit distance values for grammar 20 for pass-images 0 to 16 427
F.43 Similarity threshold values for grammar 21 427
F.44 Bottom-up tree edit distance values for grammar 21 for pass-images 0 to 16 428
F.45 Similarity threshold values for grammar 22 428
F.46 Bottom-up tree edit distance values for grammar 22 for pass-images 0 to 16 429
F.47 Similarity threshold values for grammar 23 430
F.48 Bottom-up tree edit distance values for grammar 23 for pass-images 0 to 16 431
F.49 Goodness of linear fit graph for displayed images for grammar 0 for pass-images 9 to 16 432
F.50 Goodness of quadratic fit graph for displayed images for grammar 0 for pass-images 9 to 16 432
F.51 Goodness of linear fit graph for displayed images for grammar 1 for pass-images 9 to 16 433
F.52 Goodness of quadratic fit graph for displayed images for grammar 1 for pass-images 9 to 16 433
F.53 Goodness of linear fit graph for displayed images for grammar 2 for pass-images 9 to 16 434
F.54 Goodness of quadratic fit graph for displayed images for grammar 2 for pass-images 9 to 16 434
F.55 Goodness of linear fit graph for displayed images for grammar 3 for pass-images 9 to 16 435
F.56 Goodness of quadratic fit graph for displayed images for grammar 3 for pass-images 9 to 16 435
F.57 Goodness of linear fit graph for displayed images for grammar 4 for pass-images 9 to 16 436
F.58 Goodness of quadratic fit graph for displayed images for grammar 4 for pass-images 9 to 16 436

F.59 Goodness of linear fit graph for displayed images for grammar 5 for pass-images 9 to 16 437
F.60 Goodness of quadratic fit graph for displayed images for grammar 5 for pass-images 9 to 16 437
F.61 Goodness of linear fit graph for displayed images for grammar 6 for pass-images 9 to 16 438
F.62 Goodness of quadratic fit graph for displayed images for grammar 6 for pass-images 9 to 16 438
F.63 Goodness of linear fit graph for displayed images for grammar 7 for pass-images 9 to 16 439
F.64 Goodness of quadratic fit graph for displayed images for grammar 7 for pass-images 9 to 16 439
F.65 Goodness of linear fit graph for displayed images for grammar 8 for pass-images 9 to 16 440
F.66 Goodness of quadratic fit graph for displayed images for grammar 8 for pass-images 9 to 16 440
F.67 Goodness of linear fit graph for displayed images for grammar 9 for pass-images 9 to 16 441
F.68 Goodness of quadratic fit graph for displayed images for grammar 9 for pass-images 9 to 16 441
F.69 Goodness of linear fit graph for displayed images for grammar 10 for pass-images 9 to 16 442
F.70 Goodness of quadratic fit graph for displayed images for grammar 10 for pass-images 9 to 16 442
F.71 Goodness of linear fit graph for displayed images for grammar 11 for pass-images 9 to 16 443
F.72 Goodness of quadratic fit graph for displayed images for grammar 11 for pass-images 9 to 16 443
F.73 Goodness of linear fit graph for displayed images for grammar12 for pass-images 9 to 16 444
F.74 Goodness of quadratic fit graph for displayed images for grammar 12 for pass-images 9 to 16 444

F.75 Goodness of linear fit graph for displayed images for grammar 13 for pass-images 9 to 16 445
F.76 Goodness of quadratic fit graph for displayed images for grammar 13 for pass-images 9 to 16 445
F.77 Goodness of linear fit graph for displayed images for grammar 14 for pass-images 9 to 16 446
F.78 Goodness of quadratic fit graph for displayed images for grammar 14 for pass-images 9 to 16 446
F.79 Goodness of linear fit graph for displayed images for grammar 15 for pass-images 9 to 16 447
F.80 Goodness of quadratic fit graph for displayed images for grammar 15 for pass-images 9 to 16 447
F.81 Goodness of linear fit graph for displayed images for grammar 16 for pass-images 9 to 16 448
F.82 Goodness of quadratic fit graph for displayed images for grammar 16 for pass-images 9 to 16 448
F.83 Goodness of linear fit graph for displayed images for grammar 17 for pass-images 9 to 16 449
F.84 Goodness of quadratic fit graph for displayed images for grammar 17 for pass-images 9 to 16 449
F.85 Goodness of linear fit graph for displayed images for grammar 18 for pass-images 9 to 16 450
F.86 Goodness of quadratic fit graph for displayed images for grammar 18 for pass-images 9 to 16 450
F.87 Goodness of linear fit graph for displayed images for grammar 19 for pass-images 9 to 16 451
F.88 Goodness of quadratic fit graph for displayed images for grammar 19 for pass-images 9 to 16 451
F.89 Goodness of linear fit graph for displayed images for grammar 20 for pass-images 9 to 16 452
F.90 Goodness of quadratic fit graph for displayed images for grammar 20 for pass-images 9 to 16 452

F.91 Goodness of linear fit graph for displayed images for grammar 21 for pass-images 9 to 16 453
F.92 Goodness of quadratic fit graph for displayed images for grammar 21 for pass-images 9 to 16 453
F.93 Goodness of linear fit graph for displayed images for grammar 22 for pass-images 9 to 16 454
F.94 Goodness of quadratic fit graph for displayed images for grammar 22 for pass-images 9 to 16 454
F.95 Goodness of linear fit graph for displayed images for grammar23 for pass-images 9 to 16 455
F.96 Goodness of quadratic fit graph for displayed images for grammar 23 for pass-images 9 to 16 455

List of Tables

2.1 Results of Pressure-Grid experiments (Kim *et al.*, 2010) 18
2.2 Summary of authentication methods 22
2.3 Chomsky grammars 35
2.4 Summary of perceptual similarity theories 51

4.1 Comparison of I_1 to other I's 89
4.2 Comparison of I_1 to other I's (Normalised) 89
4.3 Comparison of I_1 to other I's (Similarity ordering) 90
4.4 Comparison of E to other letters 90
4.5 Comparison of E to other letters (Normalised) 90
4.6 Comparison of E to other letters (Similarity ordering) 90
4.7 Comparison of Treediff (T) and SpCD (S) 94
4.8 Similarity ordering of Treediff (T) and SpCD (S) 95
4.9 Correlation studies using Pearson and Kendall Tau 96

5.1 Background information 104
5.2 Online similarity survey data 104
5.3 Online survey respondents clusters 106
5.4 CBIR comparison of I_1 to other I's 111
5.5 CBIR comparison of E to other letters 111
5.6 CBIR distance measure for Figure 5.4 on page 103 (Barnsley fern) 112
5.7 CBIR distance measure for Figure 5.5 on page 103 (Spiral) 112
5.8 CBIR distance measure for Figure 5.6 on page 103 (Mosaic) . . . 112
5.9 CBIR distance measure for Figure 5.7 on page 103 (Rotating squares) 112
5.10 Tree edit distance measures (I) 115

5.11 Tree edit distance measures (E to I) . . . 115
5.12 Tree edit distance measures (Barnsley fern) . . . 115
5.13 Tree edit distance measures (Spiral) . . . 116
5.14 Tree edit distance measures (Mosaic) . . . 116
5.15 Tree edit distance measures (Rotating squares) . . . 116
5.16 Correlation of perceptual similarity, tree edit distances and CBIR similarity measures . . . 117

7.1 Similarity threshold value for the 24 grammars . . . 145
7.2 Similarity threshold value for the 24 grammars - continued . . . 146
7.3 Quadratic goodness of fit test for symmetrical grammars . . . 152
7.4 Goodness of fit values for various polynomial order values for grammar 0 and pass-mage 8 . . . 153
7.5 Grammar characterisation summary. . . . 161

E.1 Image I . . . 345
E.2 Images E to T . . . 352
E.3 Image Fern . . . 353
E.4 Image Spiral . . . 359
E.5 Image Mosaic . . . 363
E.6 Image Squares . . . 367

F.1 Bottom-up tree edit distance values for grammar 0 for pass-images 0 to 16 . . . 373
F.2 Bottom-up tree edit distance values for grammar 1 for pass-images 0 to 16 . . . 374
F.3 Bottom-up tree edit distance values for grammar 2 for pass-images 0 to 16 . . . 375
F.4 Bottom-up tree edit distance values for grammar 3 for pass-images 0 to 16 . . . 376
F.5 Bottom-up tree edit distance values for grammar 4 for pass-images 0 to 16 . . . 377
F.6 Bottom-up tree edit distance values for grammar 5 for pass-images 0 to 16 . . . 378

F.7 Bottom-up tree edit distance values for grammar 6 for pass-images 0 to 16 . . . 379
F.8 Bottom-up tree edit distance values for grammar 7 for pass-images 0 to 16 . . . 380
F.9 Bottom-up tree edit distance values for grammar 8 for pass-images 0 to 16 . . . 381
F.10 Bottom-up tree edit distance values for grammar 9 for pass-images 0 to 16 . . . 382
F.11 Bottom-up tree edit distance values for grammar 10 for pass-images 0 to 16 . . . 383
F.12 Bottom-up tree edit distance values for grammar 11 for pass-images 0 to 16 . . . 384
F.13 Bottom-up tree edit distance values for grammar 12 for pass-images 0 to 16 . . . 385
F.14 Bottom-up tree edit distance values for grammar 13 for pass-images 0 to 16 . . . 386
F.15 Bottom-up tree edit distance values for grammar 14 for pass-images 0 to 16 . . . 387
F.16 Bottom-up tree edit distance values for grammar 15 for pass-images 0 to 16 . . . 388
F.17 Bottom-up tree edit distance values for grammar 16 for pass-images 0 to 16 . . . 389
F.18 Bottom-up tree edit distance values for grammar 17 for pass-images 0 to 16 . . . 390
F.19 Bottom-up tree edit distance values for grammar 18 for pass-images 0 to 16 . . . 391
F.20 Bottom-up tree edit distance values for grammar 19 for pass-images 0 to 16 . . . 392
F.21 Bottom-up tree edit distance values for grammar 20 for pass-images 0 to 16 . . . 393
F.22 Bottom-up tree edit distance values for grammar 21 for pass-images 0 to 16 . . . 394

F.23 Bottom-up tree edit distance values for grammar 22 for pass-images 0 to 16 395
F.24 Bottom-up tree edit distance values for grammar 23 for pass-images 0 to 16 396
F.25 Goodness of fit values for all grammars 456

Nomenclature

Greek Symbols

$\Sigma^{(k)}$ Set of symbols of rank k in Σ 36

T_{Σ} Set of trees over Σ 36

$T_{\Sigma}(T)$ Set of trees over Σ with subtrees in T 36

$T_{\Sigma \cup N}$ Set of trees over the union of Σ and N 36

$val(t)$ Value of tree t 36

$T_{\Sigma \setminus N}$ Set of trees over symbols in Σ but not in N 37

Unique Definitions

Symmetrical grammars are grammars that generate images whose tree edit distance values alternate between the tree edit distance values of distractor images immediately before and after the pass-image and form quadratic graphs. 123

Non-symmetrical grammars have tree edit distance values of distractors that oscillate around a mean value and form straight line graphs. 123

Granularity of production rules refers to changing the number and or order of production rules applied at any one time in order to cause the smallest observable change to an image. 123

Similarity convergence point This is the point, after successive generation of images, where the images become indistinguishable to the human eye. 123

Similarity convergence region is the region before the similarity convergence point of progressively similar images. 123

Similarity threshold value is the maximum bottom-up tree edit distance value of the nine displayed images in our prototype visual password scheme.127

Chapter 1

Introduction

1.1 Introduction

Authentication is what takes place when users log on to their websites or email accounts or use automatic teller machines (ATMs). It is considered a very necessary aspect of computer usage that affects our everyday lives.

The most popular authentication method is when users log in by typing in their usernames and passwords. These are referred to as alphanumeric passwords, because of the use of digits and alphabet characters or letters. The perceived problem of alphanumeric passwords is that the system can be easily attacked because of the human tendency to use short and easy-to-remember passwords. This has led to the development of visual passwords as an alternative to alphanumeric passwords.

In one type of visual password scheme, called Passfaces, nine pictures of human faces are displayed on the screen at once in a three-by-three grid. One of the nine pictures is the user's password, while the others are decoys or distractors. After the user selects his [1] password, another set of nine pictures is presented for him to select his second password. This process, called the challenge-response cycle, is repeated a total of four times before the user either passes or fails authentication. A visual password scheme was first introduced by Blonder (1996).

Using the Passfaces type of visual passwords, overcomes the tendency to share

[1] The use of his, him or he in this text is for brevity and refers to both genders

passwords with colleagues and friends as they may be difficult to describe. Furthermore, automated brute force attacks do not work with this type of visual password scheme as there is no easily accessible alphabet to use for such attacks.

There are, however, some major problems with the Passfaces type of visual password schemes. The password space, which is the number of possible password combinations, is smaller than for alphanumeric based password schemes. For example, in a numeric password scheme of four digits, the password space will be 10^4, since there are only ten digits $(0 \ldots 9)$ typed four times. The password space for Passfaces mentioned previously is 9^4. The Passfaces type of visual password schemes has limitations on both the number of images that can be displayed on the screen at any given time and the number of pictures that can be stored in a picture database to use for authentication. Another limitation is shoulder surfing, as the fact that images are easy to remember also means they are easy to shoulder-surf. The third limitation has to do with people choosing their picture passwords based on racial profiles, family relationships and even attractiveness of faces in pictures. This is similar to the type of problems faced in alphanumeric password schemes where people choose easy to remember names of family members and friends as passwords, making it easy to guess and attack using a small dictionary of common names and words (Jermyn *et al.*, 1999). Another limitation is that passing pictures back and forth over the network has infrastructural and security implications.

The purpose of this research is to overcome these problems by automatically generating abstract graphics to use as visual passwords using formal grammars. This means that the number of passwords that can be generated is infinite. This will resolve the problem of small potential password space for this type of visual password scheme. The problem of bias in picture selection will also be eliminated by using abstract graphics as passwords.

1.2 Research aims and objectives

The aims of this research were as follows:

1. Find an appropriate generic model of tree grammars to use for syntactic

picture generation.

2. Conduct both a pilot and a full study to determine the human idea of similarity with respect to generated abstract images.

3. Conduct an experimental study to correlate mathematical similarity measures with the human one.

4. Incorporate this similarity measure into syntactic picture generation.

5. Use the above findings to create a prototype visual password scheme that eliminates, as much as possible, the limitations of this category of visual password scheme.

1.3 Research questions

Research questions.

1. To what extent is it possible to use picture grammars to generate visual passwords?

2. To what extent can a mathematical model of similarity match the human idea of similarity?

3. To what extent can a mathematical model of similarity be included in syntactic picture generation, more specifically, tree picture grammars?

4. To what extent can perceptual similarity be modelled in a visual password scheme to generate effective distractor pass-images?

5. To what extent is it possible to use tree picture grammars and a mathematical similarity measure to develop a Passfaces type of visual password scheme that eliminates most of the problems in this type of scheme?

1.4 Research method

The following steps were followed in this research work.

1. We used an effective tree picture grammar model to generate images.
2. We researched and implemented an appropriate test of similarity measures for the type of images described in (1) above.
3. We conducted both a pilot and full study using test subjects to measure the human idea of similarity as follows.
   (a) Fifty images were presented to the test subjects.
   (b) The respondents were expected to arrange the answer images in order of similarity to the question image displayed on the screen.
4. We compared computer test of similarity results with that of the test subjects in order to determine which similarity ranking method most closely matched that of human subjects.
5. We incorporated the ranking method determined in (4) into the syntactic picture generation process using tree grammars.
6. A prototype visual password scheme was then developed using all the results of our experiments.
7. We further characterised the tree grammars used in this study, in order to better understand the attributes of grammars suitable for this scheme. This involved the determination and analysis of the maximum cut-off similarity values of distractor images of tree grammars referred to as similarity threshold values in this study.

1.5 Results

The results showed that:

1. It is possible to use tree picture grammars to generate similar graphic images.

2. A mathematical image similarity measure that is especially suited to computer generated graphic images can be used to model the human idea of similarity.

3. Some of the generated graphics from many of the grammars are suitable for use as visual passwords.

4. A prototype visual password scheme can be built, that eliminates most of the problems in this category of visual password scheme, using results obtained from studies of tree picture grammar generated images, perceptual similarity and mathematical similarity measures.

1.6 Scope and limitations

The most important limitation of this work is the subjectivity of the human idea of similarity. Humans use different subjective criteria to determine similarity and as such our work in this area cannot be absolute. However, since the purpose of our similarity ranking is to produce decoy images to counteract shoulder surfing in visual password schemes, the perceptual similarity study is sufficient for our purpose. Increasing the number and variety of images in the online perceptual similarity survey may make this study more significant.

The second issue is the choice of the number of similarity measures to evaluate. There are literally hundreds, if not thousands, of visual descriptors in literature. This work decided to pick, in the author's opinion, the most popular and successful ones. The reason for this kind of subjective decision is the absence of standard evaluation benchmarks for image similarity measures to date.

1.7 Significance

This study used tree picture grammars as a viable means of graphic password generation. This is significant because it is the first time a tree picture grammar

is being applied to not only the generation of visual passwords, but for their automatic generation. A major criticism of visual password schemes is their low entropy or limited potential password space. Using the proposed method of picture generation opens up the potential password space, as an almost infinite number of graphic passwords can be generated.

This study also contributes significantly to the growing body of research on bridging the gap between the mathematical notion of similarity and the human psychological idea of similarity, with special reference to generated graphic or abstract images.

The following are some of the problems with the Passfaces type of visual password scheme that were addressed in this study.

1. Limited password space.
2. Shoulder surfing.
3. Poor choice of pass-image selection.
4. Infrastructural issues and their associated security problems.
5. Description attack. This is the ability to compromise system security by divulging or sharing pass-images by describing it to others.

A more detailed list of contributions can be found in the concluding chapter (Section 8.1).

1.8 Structure of the document

The rest of the document is structured as follows. A literature review is done in Chapter 2 in order to place the research work in context. The chapter discusses the various types of authentication systems, which include the three major types of password schemes, and covers in more detail the Passfaces type of visual password scheme, which is the focus of this study. The chapter also reviews the tree grammars used to generate the images for our visual password scheme. CBIR and various tree edit distance measures used to ensure that generated images are

close to that of humans are covered, along with the design of the online perceptual similarity survey used in this research. This makes it easy to place this work in the appropriate context of research in the Passfaces type of visual password scheme, using tree grammars for image generation and various state of the art similarity measures to select distractor images from the pool of generated images.

Chapter 3 discusses the research method that was used in this study. The research approach was to first experiment with a number of similarity measures, in order to determine which ones were best suited to the type of abstract computer generated images used in this work. This was done in Chapter 4. After discovering a number of measures suitable for determining similarity of abstract images in Chapter 4, the study then used an online perceptual similarity survey to determine how humans determine similarity of tree grammar-based generated images in order to find the similarity measure that best models perceptual similarity in Chapter 5. In Chapter 6, the best similarity measure was then used to pick distractor images in our prototype visual password scheme. Another series of experiments was conducted in Chapter 7 to determine how best to control the similarity of distractor images and the characteristics of the tree grammars suitable for use in a Passfaces type of visual password scheme.

Chapter 8 discusses the strengths and weaknesses of both the research and the prototype. Chapter 9 considers further studies arising from the results of this work and Chapter 10 summarises and concludes the major aspects of this research study, highlighting the contributions of this work to the computer science literature.

Chapter 2

Literature review

2.1 Introduction

This chapter discusses the three major types of authentication schemes and later discusses in some detail the most popular type of password scheme: the alphanumeric password scheme. This is to place the research work in context. The reasons for the increasing popularity of visual passwords for authentication purposes are explained, as well as the most popular visual password schemes in current use. A discussion of computer models of similarity is then undertaken, followed by a review of the image features and distance measures used for similarity matching. A brief survey of formal grammars, along with the definitions for the tree grammars used in this study are provided. This chapter delves into how humans determine similarity, referred to in the literature as perceptual similarity, and the various theories and experimental procedures that have been developed for perceptual similarity studies. The motivation for the online perceptual similarity study was to develop a system that generates images from a human point of view and to also ensure that the similarity measure used in the prototype visual password system was the one that most closely matched the human idea of similarity. This chapter concludes with a summary of the literature survey findings.

2.2 Authentication methods

Authentication systems have become part of our everyday lives. We need authentication to access some web sites, our emails, ATMs and our computer systems to name a few. Information can be remembered in one of two ways: recall and recognition (Suo *et al.*, 2005). Authentication schemes can be broadly categorised into three main types (Nicholson, 2009). "What you have" or token based authentication schemes rely on physical objects in your possession to gain access, for example ATM cards. "Who you are" or biometric authentication relies on physiological or biological uniqueness, like fingerprints or iris scan, to grant access to a resource or system. The above two methods require additional hardware and software which may be quite expensive. Also, individuals have been found to be resistant to giving out their bio-data, as, once stolen, they are stolen for life. The third category of authentication system depends on "What you know" and is usually dependent on the ability of the user to recall or recognize the information. Examples of this class of authentication are PINs (Personal Identification Numbers) and alphanumeric based passwords.

The problem with alphanumeric based passwords has always been the need to balance security and ease of use (Wiedenbeck *et al.*, 2005). Most users choose simple passwords that are easy to remember but are not secure (Adams & Sasse, 1999). Secure alphanumeric passwords should be long and contain a mixture of digits, special characters, capital and small letters. However, the more secure such passwords are, the more difficult they are to remember and the less easy they are to use. Visual or graphic passwords are an attempt to solve this problem. It has been shown that recognition based authentication systems, like the Passfaces type visual password schemes, are easier to use (Elftmann, 2006; Madigan, 1983). Alphanumeric based schemes are based on recollection and as such can be difficult to use (Sobrado & Birget, 2002). As a result of these perceived difficulties of traditional authentication systems, visual password schemes were developed.

The earliest approach to visual password schemes depends on a user selecting graphic objects from among other graphic objects in a specified order. Visual password schemes are fast becoming accepted as a better alternative to alphanumeric passwords. The first known mention of a graphical password scheme was

in a patent by Blonder (1996).

This work aims at using formal grammars, specifically tree picture grammars, in the generation of similar graphic images for visual password authentication in order to defeat shoulder surfing. In doing this, the idea of similarity of graphic images will be examined along with attributes possessed by images that can be said to be mathematically similar.

2.2.1 The three major authentication methods

This section reviews the different types of authentication methods and their effectiveness over the years.

2.2.1.1 Biometric authentication

This method relies on the physiological and biological characteristics of the user. There are two kinds, namely contact metric technology and contactless metric technology (Ray, 2012). The most popular contact metric technology types are hand/finger geometry, dynamic signature verification and fingerprint scan. Examples of contactless metric technology are iris scan, facial recognition and voice recognition. Collection and storage of biometric data indicated that these methods may only work with a relatively small section of the population (O'Gorman, 2003). They require expensive equipment and, with the exception of iris scan and fingerprint technology, may not work on a population of more than one thousand (Hunter, 2004). Another drawback of biometric identification is that identities, once stolen, are stolen for life (Hunter, 2004). Also, once compromised, the identities are usually very expensive to replace.

2.2.1.2 Token based authentication

In this method, the user presents an object or token as a means of authentication. Examples are credit/debit cards, smartcards, Automated Teller Machine cards and Integrated Campus Management (ICAM) cards. These tokens can be duplicated or stolen.

2.2.1.3 Knowledge based authentication

This class of authentication is based on what you know. There are basically four types, which are discussed here. The most representative examples mentioned under each type are discussed in the next section.

1. Recall based passwords.

   These depend on the ability to recall or remember a previously memorized password. Examples of these are alphanumeric passwords and the use of PINs. Others are:

   - Draw a secret (DAS) graphical passwords (Jermyn *et al.*, 1999).
   - PassShapes (Weiss & De Luca, 2008).
   - Passdoodle (Goldberg *et al.*, 2002; Govindarajulu & Madhvanath, 2007; Martinez-Diaz *et al.*, 2010; Varenhorst, 2004).
   - Pass-Go (Tao, 2006; Tao & Adams, 2008).

2. Recognition based passwords.

   These depend on humans' ability to recognize patterns or objects they are familiar with. These passwords have been shown to be easier to use than recall based passwords. Examples of these are:

   - Passfaces (Realuser, 2012).
   - Déjà vu (Dhamija & Perrig, 2000).
   - Story (Davis *et al.*, 2004).
   - Recall-a-story (Maetz *et al.*, 2009).
   - Use your illusions (Hayashi *et al.*, 2008).
   - Convex hull click (CHC) (Wiedenbeck *et al.*, 2006).

3. Cued recall based passwords.

   These include cues or pointers to aid recall. Examples are:

   - Passpoints (Wiedenbeck *et al.*, 2005).

- Cued Passpoints (Chiasson *et al.*, 2007; Subha, 2011) or variations of it (Chiasson *et al.*, 2008; Forget *et al.*, 2010; Liu *et al.*, 2011).
- 3D scheme (Alsulaiman & Saddik, 2006; Sonkar & SB, 2011).
- Suo scheme (Suo *et al.*, 2005).
- Inkblot Authentication (Stubblefield & Simon, 2004).

4. Hybrid systems.

   These are a combination of two or more of the above methods.

   - CDS (Come from DAS and Story) (Gao *et al.*, 2010).
   - TwoStep (Oorschot & Wan, 2009).
   - Ray's Scheme (Ray, 2012).

2.2.2 Alphanumeric and visual password schemes

Alphanumeric password schemes are the most widely used type of knowledge based authentication methods. Secure or strong alphanumeric passwords are not easy to remember. There is thus a conflict between security and ease of use. A number of practices that lessen the security of alphanumeric password schemes have been observed. For example, it was observed that users give away their passwords to colleagues to help cover for their absence from the office (Wiedenbeck *et al.*, 2006). Also, a large percentage of users decide to write down their passwords (Wiedenbeck *et al.*, 2006). This is especially so when passwords to remember are many and are not used on a regular basis (Wiedenbeck *et al.*, 2005). A study also shows that some users use pets' and family members' names (Benko *et al.*, 2006). In addition, colleagues find it difficult to hide their passwords when authenticating in the presence of their colleagues, as it might give the impression of distrusting them.

Alphanumeric passwords can be eavesdropped, written down and stolen, shoulder surfed and hacked both through dictionary attacks and brute force attacks. A case study has showed that 25% of alphanumeric passwords can be guessed by using a relatively small dictionary (Jermyn *et al.*, 1999). The reasons men-

tioned above led to the development of visual passwords as a better alternative to alphanumeric passwords.

The major types of visual password schemes will be reviewed next. Most of them have undergone modifications by different researchers, who have given their adaptations new names.

1. Passfaces.

   This method originated with Realuser (2012). In this method, the user selects some faces (e.g. four) from a large number of available faces stored on a server. On log-in, the user is presented with a three-by-three matrix of faces. One of the faces is part of the user's password, while the others act as decoys. The user touches a face to select it and the system displays the next set of faces. The challenge-response cycle continues until the user has selected all (four) faces and it is at this point that the user passes or fails authentication.

   The main concerns regarding Passfaces have to do with usability and security. Usability has to do with the network speed required to load pictures and is becoming less of a problem as network speed increases.

   There are a number of security issues with the Passfaces type of authentication, one of which has to do with the fact that the user has to physically touch or indicate the face that he is selecting, which may make it vulnerable to shoulder surfing (Lashkari *et al.*, 2009). However, a number of methods to counter shoulder surfing in this and other visual password schemes have been proposed (Farmand & Bin Zakaria, 2010; Jebriel & Poet, 2011; Kim *et al.*, 2010; Li *et al.*, 2005; Miyachi *et al.*, 2010; Seng *et al.*, 2011; Wiedenbeck *et al.*, 2006).

   For example, Farmand & Bin Zakaria (2010) proposed Recognition-Based Sequence Reproduction in four ways (RBSR4). One unique idea suggested by this approach is the addition of four scanning directions (4-ways) and dates in the choice of pass-images. The user types a number to represent the position of the pass-image randomly placed on the screen, along with the decoy images, counting from any of the four directions provided during the registration phase: left to right, right to left, bottom to top and top

to bottom. An additional level of security is provided by including dates, for example, choosing to use right to left on even days (i.e., Tuesday) and left to right on odd days (i.e., Wednesday). This information is kept on the server.

Jebriel & Poet (2011) found that using keyboards to select pass-images is more secure than using the mouse. They conducted an experiment where doodles were displayed in a four-by-four matrix, using four different methods of pass-image selection: numeric, numeric and alphabetic, column and row (matrix) type and clicking type as shown in Figure 2.1 on the following page. An interesting observation was that the matrix method was not better than any of the other keyboard entry methods, even though observing a single key press still left the observer with one in four guesses. They speculated that a possible explanation was that the user was unconsciously indicating which keys were needed to determine the pass-image, perhaps by hovering their fingers over both keys (for selecting the Pass-image row and column), before deciding which one to press. An indication that this might be the case was the observation that the time needed to enter data in the matrix case was five seconds longer than any other method. They suggested further experiments where users were given time to practice the various forms of choice indication before being observed and observers were given the chance to explain how they worked out the chosen image.

Kim *et al.* (2010) came up with the Pressure-Grid system for multi-touch tabletop environments. This method relies heavily on the technology being used in vision based multi-touch systems, which can obtain the size of the finger contact (or blob) detected by the camera. Changes in finger pressure are readily apparent to the tracking systems, but are very difficult for observers to discern. This is further improved by the fact that increasing pressure on less dexterous fingers (i.e., little finger), causes involuntary movements on other fingers that is likely to add to the confusion of the observer. The authors compared their method using the four ways of countering shoulder surfing: reducing visibility, subdividing actions, dissipating attention and transforming knowledge. The authors tested the Pressure-



(a) Numeric

(b) Numeric and alphabetic

(c) Matrix

(d) Clicking type

Figure 2.1: Pass-image selection types (Jebriel & Poet, 2011)

Grid system using finger positions as shown in Figure 2.2 on Page 17 in the study context shown in Figure 2.3 on Page 18. The authors used four authentication methods for their tests: basic (unshielded) PIN, basic (unshielded) faces, Pressure PIN and Pressure faces. The results are as shown in Table 2.1 on Page 18. It was observed that Pressure-Grid was the best, with no shoulder surfer succeeding against this method.

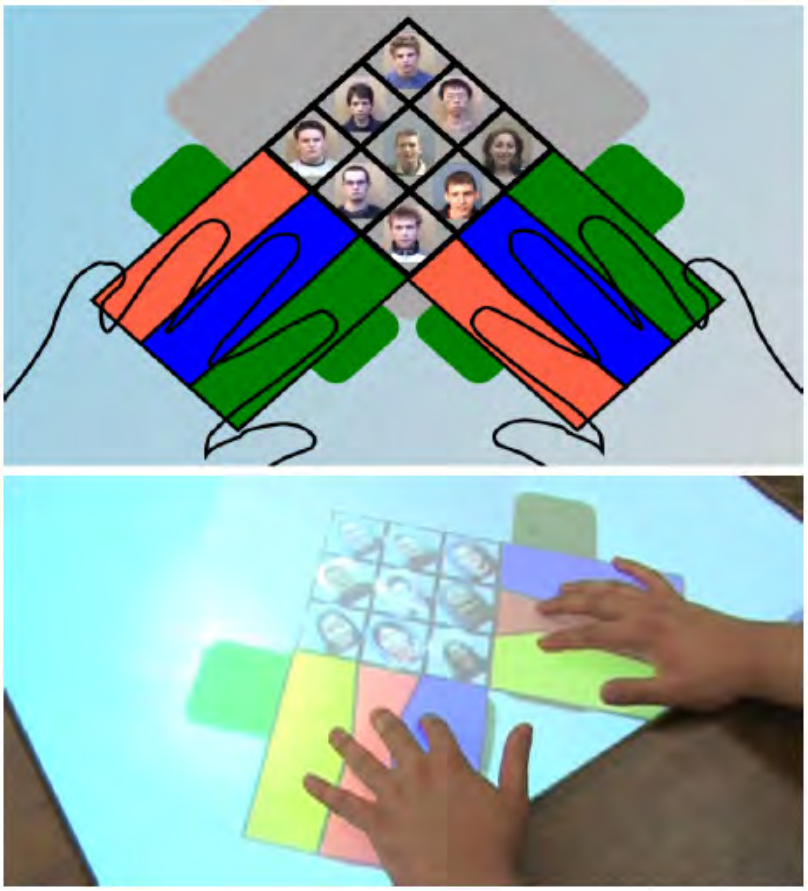


Figure 2.2: Pressure-Grid finger positions (Kim *et al.*, 2010)

Li *et al.* (2005) also proposed an authentication method that involves adding a passpoints type graphical password and grouped colour choice login to a Passfaces type password scheme, as a way of improving shoulder surfing resistance.

Miyachi *et al.* (2010) used Discreet Wavelet Transform (DWT) to blend low



Figure 2.3: Pressure-Grid study context (Kim *et al.*, 2010)

Table 2.1: Results of Pressure-Grid experiments (Kim *et al.*, 2010)

		Components guessed				
System	**Logins**	**0**	**1**	**2**	**3**	**All**
PIN	**22**	14%	18%	14%	9%	45%
Faces	**36**	25%	19%	36%	11%	8%
Pressure PIN	**38**	42%	32%	18%	3%	5%
Pressure FA.	**42**	57%	40%	2%	0%	0%

frequency components of decoy images with high frequency components of pass-images to produce a more useable system that is secure against shoulder surfing. In other words, a pass-image is faintly printed on a decoy image, which the legitimate user close to the screen can easily see, but which someone not close to the screen cannot recognize as the pass-image.

2. Draw-a-secret.

   In this method, the user draws a symbol on a screen that is divided into grids. The symbol is not actually stored, but instead the strokes (e.g. up, down movements) and the intersections with the various parts of the grid

are stored. There is some degree of tolerance as the user does not have to reproduce the exact strokes he registered as his password. Although Van Oorschot & Thorpe (2008) built a number of predictive models that succeeded in breaking this scheme, they also made a number of suggestions on how to improve it (Thorpe & Van Oorschot, 2004).

In their improvement to Draw-a-secret, Thorpe & Van Oorschot (2004) suggested the use of grid selection, where users first select a pass-image grid from a larger grid, before using it to draw their secret passwords. The authors also observed that the stroke count of the password was more important than the size of the password space, and recommended a stroke count of not less than half the maximum possible stroke count of the system, for shoulder surfing resistance to be successfully achieved.

3. Passpoints.

   In this scheme, a user clicks on a number of points on a preselected image. The image must not have too few points or the password can be easily guessed. Secondly, the image must not have too many points, otherwise the user will get easily confused. The image used must also not have hotspots. These are areas that stand out in the picture and are obvious candidates to be used as Passpoints. A number of automated brute force attacks have been known to succeed with cracking this method (Kumar, 2011; Salehi-Abari *et al.*, 2008; Thorpe & van Oorschot, 2007; van Oorschot & Thorpe, 2011; Van Oorschot *et al.*, 2010). Methods have also been designed to counter shoulder surfing when using this method (Gao *et al.*, 2010; Joshuva *et al.*, 2011).

   For example, Gao *et al.* (2010) suggested that users draw a curve across their pass-images, rather than click on them. This drawing input trick is combined with other complementary measures, like erasing the drawing trace, displaying degraded images, and starting and ending with random images to provide good resistance to shoulder surfing.

   Joshuva *et al.* (2011) suggested using a convex hull click technique to pass-points. The passpoints are not clicked on directly, instead three objects

position themselves on the screen and the user has to mentally draw a triangle around these points and determine whether the passpoint is within this area. While users were allowed to shuffle the viewport as often as they wanted, this significantly slowed the password creation process and made it difficult for the user to select easy or known hotspots.

4. Cued click points (Chiasson *et al.*, 2007)

   This method differs from the previous one in that one point per image is used and the user is authenticated by being presented with many images or rounds. This is supposed to increase entropy or password space over the Passpoints scheme. The authors also came up with another variant, called persuasive cued click points (Chiasson *et al.*, 2008).

   In persuasive cued click points, Chiasson *et al.* (2008) used persuasion to influence user choice by encouraging users to select more random, and hence more secure, click points. This is achieved by randomly generating viewpoints within which the user must select a click point (see Figure 2.4 on Page 21). The size of the viewport was intended to offer a variety of distinct points, but still cover only an acceptably small fraction of all possible points.

5. Convex Hull Click

   In this method a number of icons are preselected as the password. During authentication, the user is presented with at least three of his pass-icons along with tens of other decoy icons. Users are supposed to draw an imaginary grid around their password icons. The innovation in this method is that users do not have to actually click on their password icons, thus making it more resilient to shoulder surfing (Wiedenbeck *et al.*, 2006).

   Zhao & Li (2007) proposed the Scalable Shoulder-Surfing Resistant Textual-Graphical Password Authentication System (S3PAS) which relies on users clicking within the area encompassed by three pass-images or pass-icons for authentication, thereby making it difficult for a shoulder surfer to determine a user's pass-image or password.



Figure 2.4: Viewport of persuasive cued click points (Chiasson *et al.*, 2008)

A summary of authentication methods discussed in this section can be found in Table 2.2 on the next page

2.3 Recent developments

This section does an analysis of recent research done in related visual password schemes.

Sangore *et al.* (2014) created a system that combines alphanumeric, recall (Drawmetric) and recognition (Cognometric) based password schemes. The recall based password scheme involves drawing a secret (signature like) password in a 4×4 grid. The recognition based part of the system involves choosing from three categories of images during the registration session. The three categories of images are ideal persons, car logos and musical instruments. The alphanumeric password is meant to protect the system against shoulder surfing. The user is allowed to use either one of the graphical password systems (Drawmetric or

Table 2.2: Summary of authentication methods

Authentication method	Examples	Description
"What you have" or token based authentication	ATM or ICAM cards	Used on that can,read authentication data from cards i.e. ATM machines
"Who you are" or biometric authentication	Fingerprint or Iris scans	Physical placement,of relevant body part close to scanning machine
"What you know" or knowledge based systems	Recall based system	Alpanumeric passwords - where users type in alphanumeric characters to authenticate
		Drawmetric i.e.,draw a secret and derivatives - where users draw a secret symbol
	Recognition based systems	Cognometric i.e.,Passfaces and derivatives - where users select pass-images from a group of similar images
	Cued recall-based systems	Locimetric i.e.,Passpoints and derivatives - where users select either multiple points from a single image or a single point from each of multiple images

Cognometric) or both. The user uses the mouse for selecting the images in the Cognometric part of the system, which makes it susceptible to shoulder-surfing. The system also uses a secret password that is drawn with the mouse, but the drawn lines are not shown on the screen during login, supposedly for enhanced security but may end up confusing the user of the system and affect usability. Making the drawn line invisible may make it difficult for the user to know whether or not the correct image is being drawn. There is no provision for restarting the drawing if accidental mouse movements are made. The user has to restart the authentication process if any mistakes are made in the drawing of the secret password. The system also suffers from the usability problem of Drawmetric password schemes, especially with regards to drawn image tolerance level during login.

Srinivasa Ravi Kiran *et al.* (2013) designed a system that is virtually a graphical method of selecting or entering alphanumeric passwords using images containing four characters each in a 5×5 grid using four selection rules. The user can drag across adjacent grids or blocks, diagonally between non-adjacent blocks, across characters within a block and across a single character in a block. In all the aforementioned drag methods, called rules by the authors, the blocks or characters touched by the drawn line must contain at least one character from each set of four characters depicted on the symbol of each block. Since the system includes spaces in the four characters depicted in each shape, this gives a password space of 95^N, where $N = 3$ in their three colour character sets. Although the system may actually be resistant to shoulder surfing attacks, the system may be susceptible to intersection attack (identifying a pass-image by looking out for recurring images since the pass-image is always amongst the displayed images during authentication), if rules three and four are used frequently, and also if the user authenticates successfully with few and short mouse drags. The registration and training phase may appear complicated at first which may discourage some users. Although a login time of 42 seconds was recorded, the average registration time was not. The system will also suffer from all the problems of recall based authentication systems, along with the ability to share the passwords.

Nicholson *et al.* (2012) designed a system where a single image is chosen and the image is divided into 16 portions or tiles, with only four tiles being used in any one authentication session of four rounds. A single tile or portion of the image is displayed to the user, along with eight decoy images in each round of authentication in a 3×3 on-screen grid. There are four rounds of authentication. This is supposed to aid memorability because of the use of only one image that has to be recognised by the user for the entire authentication process. This system was tested against description and observation attacks. The system used the earth mover's distance CBIR measure to arrange the decoy images in order of similarity and to group the tiles into three groups: similar, medium and dissimilar image tiles. The overall success rate for description attack on the system when using similar tiles for decoy images was 78%, while it was 29% for observation attack. The success rate when using medium similar images was 94% and 89% for description and observation attack respectively. Similarly,

when using dissimilar images, description attack had a success rate of 97%, while it was 91% for observation attack. This is another study that shows that the more similar the decoy images are for a Passfaces type of visual password scheme, the more protected it is from description attack. However, the vulnerability of this system to both shoulder-surfing and description attacks, amongst others, makes it unusable at this time.

Nair *et al.* (2013) came up with a Passpoints style graphical password scheme that includes the use of cell phones in a multi-factor authentication system. This may be unacceptable in some use case scenarios due to additional costs incurred from having to send SMS texts to thousands or possibly hundreds of thousands of users on a regular basis. Also, there is a software, called i-rem, that can allow you to authenticate once you know the password to access it. This is because the pass-image with its associated click points are sent to any user that fails authentication. Therefore, the single point of failure becomes the password to the i-rem software, just like any other alphanumeric based authentication system.

Sarohi & Khan (2013) conceptualised a system that combines images and text in a Passfaces style of visual password scheme. The user uses the mouse to choose four pass-images out of 16 in a 4×4 on-screen grid, which gives a password space of 1820. In addition, the user also uses the keyboard to choose an additional image containing his secret number by typing the row by column positions in a provided text-box. This secret number modulo 25 is used to pick the set of 16 images that will be used for selection of pass-images during registration. Thus this system has an inbuilt mutual or two-way authentication. This system also makes the server do the same calculations on the user's machine, with a match indicating successful two-way authentication. This system is particularly strong against man-in-the-middle (an attacker successfully gaining control of communication between server and client) and relay attacks. The system however uses the mouse for pass-image selection, and is thus susceptible to shoulder surfing. Since the secret number has to be indicated by the use of the keyboard to identify row and column positions, it is susceptible to both shoulder-surfing and intersection attack, if the user is able to identify the row and column numbers keys pressed. Also, the system did not directly address the issue of description attack.

Towhidi *et al.* (2013) developed a Cognometric system that is more secure than Passfaces, called Secure-Passfaces or S-Passfaces for short. This was based on three modifications to Passfaces:

- The method of selecting password, making it a keyboard only input.
- Creating four pass-images, out of which two are used for login at any one time i.e. concurrent passwords.
- Allowing users to pick their own pass-images for improved memorability.

The use of four alternate pass-images, out of which only two are used at login at any one time means the system provides six different sets of pass-images for authentication, which makes the system moderately resistant to shoulder surfing and recording attack. In addition, the use of the mouse for selection of pass-images is completely eliminated, making the scheme a keyboard only system. The user types a randomly generated two-letter text assigned to each image, out of the 18 in the 6×3 on-screen grid, in order to indicate his pass-image. The system uses 18 faces and two rounds of authentication, which makes the password space 18^2.

Towhidi *et al.* (2013) also conducted security and usability tests on their system while comparing it to the reference Passfaces implementation. The overall conclusion of the study was that his system was slightly less usable because of the additional security measures, but the users were more confident and willing to use the system for their authentication needs because of the added security.

Rasekgala *et al.* (2014a,b) used shape grammars in the design of their visual password scheme. This is the closest to this study because of the use of grammars, since the authors used shape grammars in the generation of basic shapes under the control of the user. The authors used a draw-a-secret (drawmetric) type of visual password scheme for the registration phase of the system, for which they computed the password space. However, the computed password space is not applicable to the login process, which has a low password space of nine, since the user has to recognize one of nine images on a 3×3 on-screen grid, in their variant of the Passfaces style of visual password scheme. Suggestions for improving the password space of this model, include:

- Implementing the system as a Drawmetric visual password scheme.

- Increasing the number of rounds. This can be accomplished by:

  - Generating variations of the user-drawn images in two more rounds of authentication, and asking the user to choose the grid that most closely matches his drawn image. This is to avoid requiring the user to draw multiple sets of images in the registration phase, so as not to prolong the registration process and affect usability. This will increase the password space to 9^3 for three login rounds.

  - Requiring the user to draw multiple sets of images, but to reduce the number of images draw to at most two or three to reduce registration time. The number of sets of images should be equal to at least three and should correspond to the number of login rounds. The password space is equal to N^k where N is the number of images on the screen and k is the number of rounds.

- Increasing the number of login grids to 16 (4×4), and requiring the user to select four grids containing his pass-images out of the available 16 grids of images. This will increase the password space to 1820.

Also the use of keyboard for input will further help defeat shoulder surfing.

The use of a few basic shapes may not justify the use of shape grammars, as it might be easier and more compact in terms of memory storage to use a simple vector graphics format instead. It is suggested that slightly more complex shapes, including three dimensional shapes, should be used in order to exploit the power of shape grammars, make the shapes more interesting and allow users who are familiar with the system to create shapes that are difficult to describe and difficult for the shoulder-surfer to memorise.

The images generated have to be similar for cognometric visual password schemes, in order to help defeat shoulder surfing. In the next section, we take a look at computer based notions of similarity.

2.4 Mathematical models of similarity

Computer based similarity measures have been developed in the field of CBIR. The absence of a standard for CBIR measures led to the author conducting a survey of the most popular and effective measures found in the CBIR literature and presenting them in this section.

2.4.1 Feature extraction methods

Most of the image similarity measures today are used for CBIR from graphic or image databases. A feature is a specific visual property of an image, either globally for the entire image or locally for a small group of pixels. The following image features or metrics, as they are called, are used to compare images, to determine whether they are similar or not.

1. Colour
2. Texture
3. Shape
4. Salient points (in an image)

We will now discuss each of the above features and their extraction methods in more detail in the following sections.

2.4.1.1 Colour features

These determine the number of colour components and/or the colour distribution in the picture. The following methods have been used to determine this metric.

1. Colour moments (Stricker & Orengo, 1995)
2. Colour histograms (Swain & Ballard, 1991)
3. Colour signatures (Rubner *et al.*, 2000)

Spatial colour features

The aforementioned methods (Section 2.4.1.1 on the preceding page) are invariant to rotation and scaling and are excellent when it comes to scaling and occlusion of images. However, they do not show the distribution of colours in the image. The following method was introduced to tackle this problem, along with Colour correlograms (Auto-CorRelograms (ACR)) (Huang *et al.*, 1997) and Spatial-Chromatic histograms (Cinque *et al.*, 1999).

Colour coherence vectors (Pass *et al.*, 1997) is a method of including spatial information into colour features. This usually involves two histograms: a coherent and a non-coherent one. Coherent pixels must be part of a uniformly coloured area, which must be more than a minimum value (say 1%) of the total image area. The distance metric between two colour coherent vectors $d_{CCV}(I_1, I_2)$ is shown in Equation (2.1) below.

$$d_{CCV}(I_1, I_2) = \sum_{k=1}^{N} [|H_1^c(k) - H_2^c(k)| + |H_1^s(k) - H_2^s(k)|] \tag{2.1}$$

where H_i^c and H_i^s are the histograms of coherent and scattered pixels respectively.

2.4.1.2 Texture features

Texture features describe a neighbourhood of pixels rather than a single pixel. Texture features capture the granularity and repetitive patterns of surfaces within a picture. The attributes of texture are:

- Coarseness which measures the scale of the textures (i.e. pebbles versus boulders).
- Contrast which describes the vividness of the pattern.
- Directionality which describes whether or not the image has a favoured direction like grass or whether it is isotropic (like a smooth object).

Methods used for texture similarity analysis are:

1. Local binary patterns (LBP) (Ojala *et al.*, 1996).

2. Co-occurrence matrix (Haralick, 1979).

2.4.1.3 Shape features

This has to do with objects within a picture and their (objects) attributes are:

- Circularity.
- Eccentricity.
- Major axis orientation.
- Algebraic moment invariants (curvature and orientation).

Methods used for shape similarity techniques are:

1. Edge histograms: edge direction information (Jain & Vailaya, 1996) using an edge detection technique like that developed by Canny (1986).
2. Image moments: example set of moment invariants are given by Hu (1962).
3. Shape context (Acosta *et al.*, 2011; Belongie & Malik, 2000; Belongie *et al.*, 1998, 2001, 2002).

2.4.1.4 Salient points

These approaches model spatial relations among local image entities for retrieval. They utilise corner points or interest points. Sebe *et al.* (2000) developed a wavelet-based salient point extraction algorithm for image retrieval.

2.4.1.5 Compound or composite descriptors

The recent trend in CBIR is to use more than one image feature for image signatures. Using multiple image features for visual signatures results in compound or composite descriptors. Most recent publications try to formulate short composite descriptors for image retrieval (Chatzichristofis & Boutalis, 2008b; Chatzichristofis *et al.*, 2010, 2011b; Lux & Chatzichristofis, 2008).

One of the composite descriptors that was used in this study is called Spatial Colour Distribution (SpCD). It uses a fuzzy system that maps the number of

colours in the image into a custom palette of eight colours, while requiring only 48 bytes per image. This descriptor is considered to be suitable for generated graphics because of the relatively small number of colours and texture regions in these types of images compared to natural colour images.

2.4.2 Core techniques used in CBIR

There are four basic core techniques used in real world implementations of similarity measures in content based image retrieval. They are:

1. Visual signatures.
2. Similarity measures.
3. Classification and clustering.
4. Search paradigms.

We shall discuss the first two in detail since they directly relate to the two main problems of CBIR technology: how to mathematically model an image, and how to determine the similarity between a pair of images based on their abstracted description (mathematical model). The first problem exists because the original representation of an image, which is an array of pixel values, does not correspond to the way we see images, let alone our recognition of the objects in the image (semantic understanding). A signature can be defined as the mathematical model or description of an image for retrieval purposes. These signatures are often referred to in the literature as descriptors. The signatures formulated determine to a large extent the similarity measures that can be used. On the other hand, the design of a similarity measure places certain requirements on the construction of signatures. We can thus see the close relationship between signatures and similarity measures.

2.4.2.1 Visual signatures

Feature extraction can either be done on sections of an image (region based) or on the entire image (global). Region based feature extraction has overshadowed global feature extraction in recent years. The features mostly extracted are

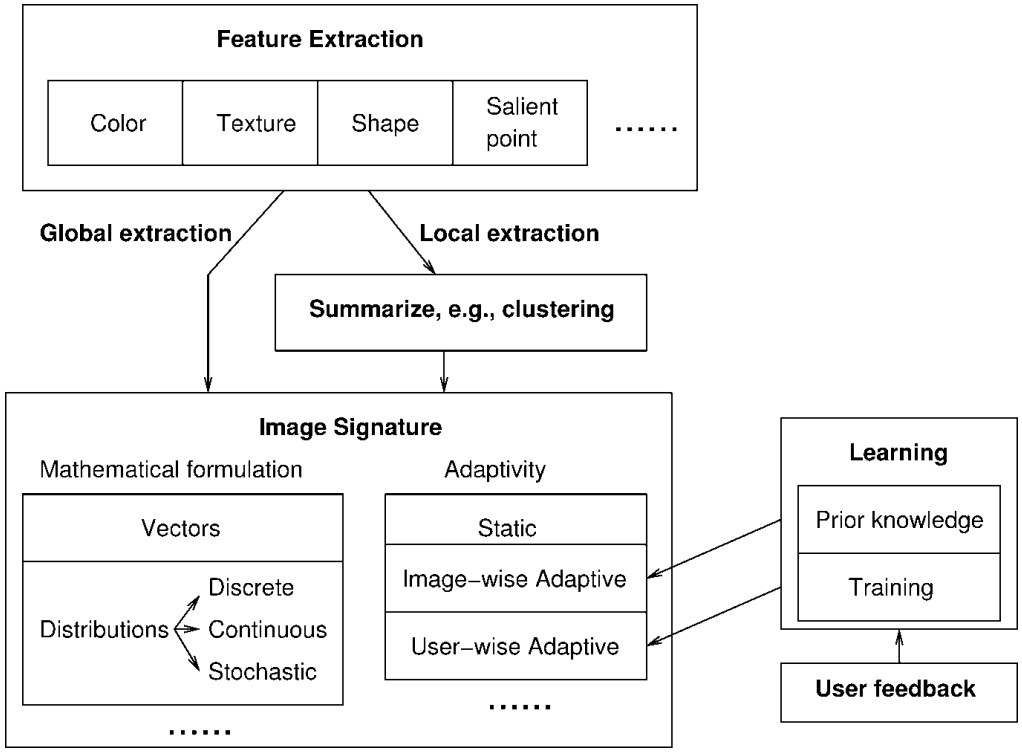


Figure 2.5: An overview of image signature formulation (Datta *et al.*, 2008)

colour, texture, shape, salient points, interest points and signs. The figure below (Figure 2.5) is an overview of image signature formulation used in CBIR systems. There are two broad types of image signature formulation, namely, mathematical formulation and adaptivity. Mathematical formulation refers to the use of mathematical models to completely describe images irrespective of type or appearance. Again there are two types: those that treat features as vectors and those that treat them as distributions. For example, colour histograms can be treated as vectors of frequencies. This approach was criticised by Rubner & Tomasi (2000), who suggested the earth mover's distance (EMD) as a histogram of feature vectors and frequency. The EMD can thus be regarded as an example of using the histogram as a discrete distribution.

Adaptivity refers to the use of learning methods to tune signatures. Static signatures are formulated for all the images in the same way, while image-wise adaptive signatures differ based on image classification. SIMPLICITY (Wang

et al., 2001) is an example of the latter approach. User-wise adaptive approach is learning in real time from user feedback.

2.4.2.2 Similarity measures

The region based signature has an image signature in the form of a set of weighted sums of feature vectors $(z_1, p_1), (z_2, p_2), .., (z_n, p_n)$, where the z_i's are the feature vectors and the p_i's are the corresponding weights assigned to them. We can denote two signatures by $I_m = [(z_1^{(m)}, p_1^{(m)}), (z_2^{(m)}, p_2^{(m)}), \ldots, (z_n^{(m)}, p_n^{(m)})], m = 1, 2$. A natural approach is to match the $z_i^{(1)}$'s with the $z_i^{(2)}$'s and then combine these vector distances as a distance between sets of vectors.

One approach is to assign a weight to every pair $z_i^{(1)}$ and $z_j^{(2)}, 1 \leq i \leq n_1, 1 \leq j \leq n_2$, and the weight $s_{i,j}$ is the significance of associating $z_i^{(1)}$ and $z_j^{(2)}$ (Wang *et al.*, 2001). The distance between I_1 and I_2 is summed up from the pair-wise distances between individual vectors as depicted below.

$$D(I_1, I_2) = \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} s_{i,j} d(z_i^{(1)}, z_j^{(2)}), \tag{2.2}$$

where I_1 and I_2 are image signatures. $s_{i,j}$ indicates the significance of associating $z_i^{(1)}$ and $z_j^{(2)}$ where d is a distance function.

In the Hausdorff distance, every $z_i^{(1)}$ is matched to its closest vector in I_2, say $z_j^{(2)}$. The Hausdorff distance between I_1 and I_2 is computed as the maximium among all $d(z_j^{(1)}, z_j^{(2)})$ and the computation is done again with the roles of I_1 and I_2 reversed before choosing the larger of the two distances.

$$D_H(I_1, I_2) = \max(\max_i \min_j d(z_i^{(1)}, z_j^{(2)}), \max_j \min_i d(z_j^{(2)}, z_i^{(1)})) \tag{2.3}$$

The Hausdorff method has been used by many researchers as a similarity measure (Gao, 2003; Hangouet, 1995; Huttenlocher *et al.*, 1992, 1993; Ko & Byun, 2002; Mémoli, 2008; Vivek & Sudha, 2006).

If $\sum_j s_{i,j} = p_i^{(1)}$, for all i, $\sum_i s_{i,j} = p_j^{(2)}$, for all j, and $s_{i,j} \geq 0$ for all i, j, then this distance becomes the Mallows distance for discrete distributions shown below (Mallows, 1972)

$$D(I_1, I_2) = \min_{s_{i,j}} \sum_{i=1}^{n_1} \sum_{j=1}^{n_2} s_{i,j} d(z_i^{(1)}, z_j^{(2)}) \tag{2.4}$$

The images used in this study are automatically generated using formal grammars. In the next section, a discussion of syntactic image generation is undertaken, specifically from the point of view of some of the tree grammars used in this study. These grammars were used for the automatic generation of pass-images and their highly similar distractors in our prototype visual password scheme.

2.5 Tree grammars and syntactic picture generation

All definitions in this section, except where otherwise specified, are taken from Drewes (2006).

2.5.1 Signatures, trees and notations

A signature is a set Σ of ranked symbols. A ranked symbol (symbol, for short) is denoted by $f : n$, with f being its name and $n \in \mathbb{N}$ its rank. The rank n of f is the number of subtrees that are allowed under f. The same name f is allowed in more than one symbol of a signature, with different ranks n. When there is no risk of confusion, for notational simplicity, $f : n$ is denoted by f. $\Sigma^{(n)}$ denotes the subset of Σ consisting of all symbols with rank n.

Given a signature Σ, we can build trees over it. The set T_Σ (Trees over [1] Σ), is defined inductively as the set of all trees that can be built using the signature Σ. Formally, T_Σ is defined as the smallest set such that $f \in \Sigma^{(n)}$ and $t_1, \ldots, t_n \in T_\Sigma$ implies $f[t_1, \ldots, t_n] \in T_\Sigma$.

The root of the tree $f[t_1, \ldots, t_n]$ is labelled by f and the direct subtrees, ordered from left to right, are $t_1, \ldots, t_n$. The symbols '[', ']', and ',' are reserved for this purpose, and do not occur in Σ. We shall refer to trees, without referring to any specific signature, if a tree over an arbitrary signature is meant. The tree

[1] "over" means "consisting of symbols from"

$f[]$ is also denoted by f. In this sense, $\Sigma^{(0)} \subseteq T_\Sigma$.

Given a signature Σ and a set T of trees, $\Sigma(T)$ denotes the set of all trees $f[t_1, \ldots, t_n]$ such that $f : n \in \Sigma$ and $t_1, \ldots, t_n \in T$.

The set $T_\Sigma(T)$ is intuitively the set of trees over Σ that have had some of their subtrees replaced by trees in T. The set $T_\Sigma(T)$ of trees over Σ with subtrees in T is the smallest set of trees such that $T \subseteq T_\Sigma(T)$ and, for every symbol $f : n \in \Sigma$ and all trees $t_1, \ldots, t_n \in T_\Sigma(T)$, the tree $f[t_1, \ldots, t_n]$ is in $T_\Sigma(T)$ as well. In particular $T_\Sigma(\emptyset)$ is equal to T_Σ.

A set $L \subseteq T_\Sigma$ is called a tree language. A tree generator is any device g defining a tree language $L(g)$. $L(g)$ is said to be the language generated by g. Examples of tree generators are Regular tree grammars, ET0L tree grammars, Grid picture grammars and Branching tree grammars, which are defined in Section 2.5.2 on page 34.

A notation useful for defining how tree grammars work is substitution, which is defined as follows. Let $X_n = \{x_1 : 0, \ldots, x_n : 0\}$ be a signature whose symbols we call variables. The names $x_1, \ldots, x_n$ will from now on be reserved for this purpose only, and will not appear in ordinary signatures.

Let $t, s_1, \ldots, s_n$ be trees. Then $t[\![s_1, \ldots, s_n]\!]$ denotes the tree t' obtained by the simultaneous substitution of s_i for every occurrence of x_i in t. As an inductive definition,

$$t' = \begin{cases} s_i & \textbf{if } t = x_i \in X_n. \\ f[t_1[\![s_1, \ldots, s_n]\!], \ldots, t_k[\![s_1, \ldots, s_n]\!]] & \textbf{if } t = f[t_1, \ldots, t_k] \notin X_n. \end{cases}$$

2.5.2 Tree grammar definitions

Chomsky (1956) classified grammars into four classes, from type 0 to 3 of increasingly restrictive or less powerful grammars (Table 2.3 on the facing page).

The tree based approach to formal languages was adopted for this study (Drewes, 2006), where syntactic generation of pictures is regarded as a domain consisting of picture elements or ranked symbols, Σ called signature, and the various operations on them. Tree generators consist of two components: a gen-

Table 2.3: Chomsky grammars

Class	Grammars	Languages	Automaton
Type 0	Unrestricted	Recursively enumerable (Turing recognizable)	Turing machine
Type 1	Context-sensitive	Context-sensitive	Linear bounded automaton
Type 2	Context-free	Context-free	Pushdown automaton
Type 3	Regular	Regular	Finite automaton

erator that outputs trees and an algebra that converts the trees into pictures by interpreting the symbols in the trees as picture operations. This approach was used to produce all the images used in this study.

Definition 2.5.1 (Regular Tree Grammars) *A regular tree grammar is a system* $g = (N, \Sigma, R, S)$ *consisting of*

- *a finite signature* N *of symbols of rank 0, called nonterminals,*
- *a finite output signature* Σ*, disjoint with* N*, of terminals,*
- *a finite set* $R \subseteq N \times T_{\Sigma \cup N}$ *of productions, and*
- *an initial nonterminal* $S \in N$*.*

A term $t \in T_{\Sigma \cup N}$ *directly derives a term* $t' \in T_{\Sigma \cup N}$ [1]*, denoted by* $t \rightarrow_P t'$*, if there is a production* $A ::= s$ *in* P *such that* t' *is obtained from* t *by replacing an occurence of* A *in* t *with* s*. The language generated by* g *is* $L(g) = \{t \in T_\Sigma | S \rightarrow_P^* t\}$*, where* $\rightarrow_P^*$ *denotes the reflexive and transitive closure of* $\rightarrow_P$*; such a language is called a regular tree language.*

Definition 2.5.2 (Grid Picture Grammars) *A grid picture grammar is a regular tree grammar of the form* $g = (N, \Sigma_k, P, S)$ *for some* $k \geq 2$*. The gallery generated by* g *is* $\Gamma(g) = \{val(t) | t \in L(g)\}$*.*

[1] Definition found on the nomenclature page.

Grid picture grammars (Drewes *et al.* (2003)), were used to produce the images in Figures 4.3 on page 86 and 5.2 on page 102.

Regular tree grammars have only one set of rules, whereas ET0L tree grammars have several sets of rules with fully parallel derivations and nonterminals may be output symbols. Also, derivations do not necessarily have to start with a single terminal, but may start with a tree called the axiom. The L in the nomenclature of ET0L grammars stands for L-systems (Lindenmayer systems). ET0L grammars are thus the tree grammar variant of the string grammars proposed by Lindenmayer (1968).

Definition 2.5.3 (ET0L Tree Grammars) *An ET0L tree grammar is a system* $g = (N, \Sigma, R, t_0)$*, with*

- *a finite signature* N *of symbols of rank 0, called nonterminals,*
- *a finite signature* Σ *of output symbols, which may or may not be disjoint with* N*, of terminals*
- R *is a finite set of tables* $R_1, \ldots, R_k$ *for some integer* k*, each table being a finite set of rules as in regular tree grammars, and*
- *an initial tree* $t_0 \in T_\Sigma(N)$*, referred to as the axiom.*

An additional requirement is that every table must contain at least one rule whose left-hand side is A*, for every* $A \in N$*.*

Let $s = s_0[\![A_1, \ldots, A_m]\!]$*, where* $s_0 \in T_{\Sigma \setminus N}(X_m)$ *(*X *being variables) and* $A_1, \ldots, A_m \in N$ *for some* $m \in \mathbb{N}$*. For* $i \in \{1, \ldots, k\}$*, there is a derivation step* $s \Rightarrow_{R_i} t$*, if there are rules* $A_1 \to s_1, \ldots, A_m \to s_m$ *in* R_i*, such that* $t = s_0[\![s_1, \ldots, s_m]\!]$*.* $L(g) = \{t \in T_\Sigma \mid t_0 \to_R^* t\}$ *is the ET0L language generated by* g*.*

ET0L grammars were used to generate Figures 5.4 on page 103 to 5.7 on page 103. An example of a complete generation sequence, from the initial ET0L grammar to the generated picture can be seen in Appendices A.2.1 to A.2.6. Branching tree grammars are a further generalisation of ET0L grammars and are defined next.

Definition 2.5.4 (Branching Tree Grammars) *Let $n \in \mathbb{N}$. A tree grammar with branching synchronization and nested tables (branching tree grammar, for short) is a tuple $g = (N, \Sigma, I, J, R, S)$, where*

- *N is a finite signature of nonterminals of rank 0;*
- *Σ is the finite output signature of terminals, where $\Sigma \cap N = \emptyset$;*
- *I and J are finite sets of synchronization and table symbols, respectively;*
- *R is a mapping, called the table specification, which assigns to every $\tau \in J^n$ a finite set $R(\tau)$ of rules $A \to t$ with $A \in N$ and $t \in T_\Sigma(N \times I^n)$; and*
- *$S \in N$ is the initial nonterminal*

The number n is the nesting depth (or just depth) of g, the sets $R(\tau)$ with $\tau \in J^n$ are the tables of g, and an element of the set $SN_g = N \times (I^n)^*$ is a synchronized nonterminal (viewed as a symbol of rank 0).

In the next section, we discuss how humans determine similarity, referred to as perceptual similarity in the literature. Perceptual similarity was used as a benchmark against which all our similarity methods had to measure up to in our experiments, in order to determine the most suitable similarity measure to use for our prototype. As mentioned earlier in the similarity section, similar distractors were generated in order to defeat shoulder surfing, and these images were made to be similar from a human point of view through perceptual similarity studies, which is the subject of the next section.

2.6 Perceptual similarity

The subject of similarity has been extensively studied in psychology and is the basis of object recognition and categorization, which are fundamental to modern cognitive research and survival in the real world. It underscores transfer of learning, errors of memory, perceptual organization, social bonding and many other problems from the psychological literature. In this study, however, we shall only be concerned with perceptual similarity in relation to images. Similarity

is a relationship that exists between two perceptual or conceptual objects. The discussion in this study will be restricted to perceptual resemblance of images to one another.

Images perceived are given internal representations in the human mind. These internal representations are used to determine perceptual similarity. There is therefore a difference between perceptual similarity and judged similarity. The latter is the only one that can be experimentally studied or manipulated. It has also been observed that it is faster to determine dissimilarity (D) than similarity (S) (Krueger, 1978), even though it is more likely for one to make false D responses to S pairs than false S responses to D pairs. Over the years, various theoretical models have been proposed to account for how humans perceive similarity. These theories have been further complicated by the fact that similarity between objects is affected by other present and immediately past stimuli and a long term relationship with a related object or image. An example of interference by other present objects is easily given in Figure 2.6, where the straight line in image *a* appears longer than the line in image *b* even though both are of the same length (Santini & Jain, 1999).

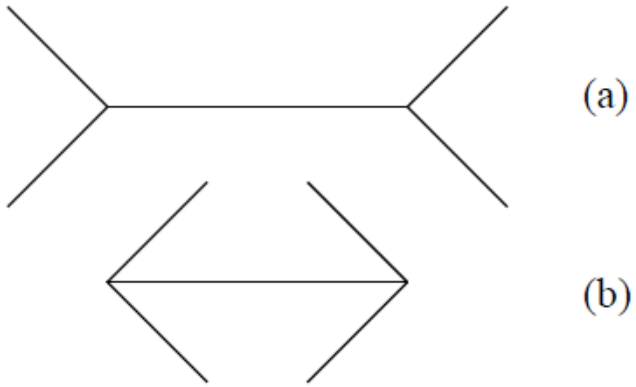


Figure 2.6: Two lines of the same length appear different

Another example of perception being affected by other present stimuli is the well known phenomenon in physics where red objects appear black under coloured light (i.e., green or blue, but not red) due to absorption. An example of similarity being affected by a long term relationship with related images is the experimen-

tal observation that similarity can be learnt (Rorissa, 2005). Another well known example is that native English speakers find the spoken L and R to be quite distinct, while to native speakers of Japanese, they sound extremely similar (Blough, 2001). Another example is that a trained musician hears things inaudible to an untrained person. People also differ in their sensory capacities (i.e. colour blind versus normal) and confusability sets in when objects are too similar, which is a central theme of experimentation in recognition studies.

Fortunately, similarities can be specified independently of the physical stimulus involved. In other words, it is possible to determine the similarities of two or more objects from behavioural responses alone, without specifying anything more about them other than which behavioural measures go with which objects. These measures need to be scaled so that they can simplify attribute relationships (e.g., multidimensional scaling) and provide generality. Generalization can help organize and predict results in other tasks that share the same stimulus. An outstanding example of a generalization of this sort is Shepard's Universal Law of Generalization, which states that the probability of a stimulus learned to another stimulus S decays exponentially with dissimilarity between the learned stimulus and stimulus S (Shepard, 1987).

Similarity has been the focus of several studies in cognitive psychology and behavioural sciences and related fields for over a century (Melara, 1992b), resulting in several theories and models of perceptual similarity (Attneave, 1950; Shepard, 1962, 1980, 1987; Thurstone, 1927; Tversky, 1977).

Fechner, from around 1850, studied perceptual similarity and came up with the just noticeable difference or jnd, which is the amount something must be changed in order for a difference to be noticeable, detectable at least half the time, and is therefore a scale to measure psychological change (Masin *et al.*, 2009; Melara, 1992a). The assumption that jnd was a fixed entity was challenged by Thurstone (1927), who argued that humans give different comparative judgements on successive occasions about the same pair of stimuli. This led to the formulation of his law of comparative judgement (Thurstone, 1927). These two classic models of similarity influenced the scaling of similarity measures as well as the development of multidimensional scaling (Melara, 1992b).

2.6.1 Theories of perceptual similarity

The five major theories of perceptual similarity found in psychological literature are discussed below.

2.6.1.1 Common element approach or alignment based models

This is one of the best known attempts to suggest that processes underlying similarity follow from the representation of stimuli or perception as a collection of elements (Estes, 1955). The higher the proportion of elements common to the stimuli or images, the more similar they are perceived to be. Similarity can be calculated by summing the number of common elements or their values. This scheme was meant to model associative processes, and cannot handle perceptual similarity of any complexity.

In this model, similarity is not just matching features, but determining how objects correspond to, or align with, one another (Goldstone & Son, 2005).

2.6.1.2 Template or transformational models

Template model representations contain much more detailed information about stimulus structure than the common element approach. Template models apply to spatially extended visual objects, and their representations can be considered to be spatially organized. Similarity is determined by the proportion of equivalent points in a point to point check between the various parts of any two image representations. This theory and its models have been criticized for their inability to detect similarities among images that have been displaced, rotated or enlarged. However, these perceived deficiencies can be eliminated by preprocessing operations that transform images to comparable orientation or size. Evidence exists to show that when training of subjects with respect to sample images is controlled, subjects may not very easily perceive similarity between images that are expanded, contracted or rotated (Tarr & Bülthoff, 1998).

This model is referred to as the transformational model in literature, and is based on the idea that two stimuli are more similar if fewer numbers of operations are required to make them identical (Goldstone & Son, 2005; Rorissa, 2005), and is the basis of the earth mover's distance metric (Rubner *et al.*, 2000).

2.6.1.3 Geometric models

The geometric model tries to place objects in a geometric or similarity space (usually Euclidean) and uses the distance or dissimilarity between any two objects to judge similarity. In other words, the closer two objects or images are in the geometric space, the more similar they are. This class of models is subsumed within the larger class of multidimensional scaling (MDS) models of perceptual similarity. There are two basic assumptions of the geometric models:

- That images or objects can be represented by values in a few continuous dimensions, and
- That similarity can be represented by distance in a coordinate space.

MDS is both a perceptual similarity theory and a data reduction technique, and these concepts should not be confused. As a method of analysing data, MDS has been very useful even if the theory from which it emanates is controversial as discussed in the next few paragraphs.

Let d be a metric distance function, which is a scale that ascribes a positive number to every pair of points (a, b, c) in accordance with the following distance axioms:

1. Self similarity: $d(a, a) = 0$.
2. Minimality: $d(a, b) \geq d(a, a)$.
3. Symmetry: $d(a, b) = d(b, a)$.
4. The triangle inequality: $d(a, b) + d(b, c) \geq d(a, c)$.

Geometric models rely on distance to predict perceived dissimilarities and as such must satisfy the four distance axioms, the empirical validity of which has been questioned.

The first axiom states that self-dissimilarity of all images are equal. Krumhansl (1978) discussed empirical evidence against this axiom. For example, she argued that distinctive or unique images amongst a set of perceived images, have a greater perceived self-similarity and so a smaller perceived self-dissimilarity.

The second axiom is minimality and states that two images are always at least as dissimilar as either image is to itself. In other words, any image is least as similar to itself as it to any other image. Tversky (1977) criticized this axiom as being sometimes inappropriate.

The third axiom states that similarity is symmetric. Tversky (1977) attacked this axiom by giving the example that similarity of North Korea to Red China is judged to be greater than the similarity of Red China to North Korea.

The fourth and final important axiom is the triangle inequality. This states that for any three images a, b and c, the sum of the distance or dissimilarity between a and b and b and c should be equal to or greater than a and c. Tversky (1977) has argued that this axiom may sometimes be violated. James (1890), over a century ago, gave an example of what seems to be a clear violation. A flame and the moon are similar because they are both luminous, and the moon is similar to a ball because they are both round. However, a flame and a ball are very dissimilar, in contradiction to the triangle inequality.

2.6.1.4 Feature-contrast models

In a pioneering article, Tversky (1977), formulated and tested a set-theoretical model of similarity called the contrast model. He defined stimuli as sets of features and similarity of any two stimuli as a linear function of a measure of both their common and unique/distinctive features. Tversky & Gati (1982) used the term features to describe attributes or properties of objects that are relevant to the task under study. This catch-all definition was criticised by Ashby & Perrin (1988), as being the reason why the task of empirically disproving this model is very difficult.

A major difference between this model and geometric models is that while dimensions of geometric models are mutually exclusive, features are an all or nothing affair (dichotomous: either possessed by a stimulus or not) (Lloyds & Rosch, 1978). The feature-contrast model posits an increase in similarity with the number of features two perceived images have in common and a decrease with the number of distinct features. The converse is also true. Two perceived images are less similar if they have more unique features and fewer common features.

Given any two images or stimuli a and b with feature sets A and B respectively, the perceived similarity of a and b, denoted by $S(a, b)$, is a linear function of the measures of their common and unique features (Tversky, 1977; Tversky & Gati, 1982), and is given by the following equation.

$$S(a, b) = \theta f(A \cap B) - \alpha f(A - B) - \beta f(B - A) \tag{5}$$

Where :

- $A \cap B$ represents the common features of a and b
- $A - B$ represents features of a that are not in b (i.e., unique features of a)
- $B - A$ represents features of b that are not in a (i.e., unique features of b)
- θ, α, and β reflect weights given to the common and unique features and are nonnegative $(\theta, \alpha, \beta \geq 0)$
- S is an interval scale
- f is an additive function [i.e., $f(A \cup B) = f(A) + f(B)$], whenever A and B are disjoint $(A \cap B = \emptyset)$

Another form of the contrast model, called the ratio model, is given by the following equation:

$$S(a, b) = \frac{f(A \cap B)}{f(A \cap B) + \alpha f(A - B) + \beta f(B - A)} \tag{6}$$

where $\alpha, \beta \geq 0$. This model defines a normalized value of similarity, such that $0 \leq S \leq 1$. This ratio model equation is a generalized form of the contrast model (Tversky, 1977). In the two previous equations ((5) and (6)), the function f, has a dual purpose. The first is to measure the common and unique features of the stimuli. The second purpose is to act as an indicator of the salience or prominence of the stimuli. The factors which contribute to the salience of a stimulus are information content, intensity, frequency, familiarity and good form (Tversky & Gati, 1978).

Several tests were conducted on the contrast model (Gati & Tversky, 1984; Johnson, 1981; Rorissa, 2005; Tversky, 1977; Tversky & Gati, 1978), and results show that humans focus their attention more on common features when judging similarity than when judging differences of stimuli.

The main reason for the formulation of this model was the violation of the metric axioms by the geometric models, as highlighted earlier. Tversky (1977) noted that the triangle inequality axiom is violated by the similarity relations of three stimuli by the fact that even though Jamaica is similar to Cuba (in geographic location) and Cuba is similar to Russia (in political ideology), that does not make Jamaica and Russia similar.

Many researchers have used the feature-contrast model as a framework for their studies in marketing and advertising (Johnson, 1981, 1986; Johnson & Horne, 1988), psychology (Ben-Shakhar & Gati, 1992; Dopkins & Ngo, 2001; Gati & Tversky, 1984), and consumer research (Ulhaque & Bahn, 1992).

2.6.1.5 Geon model

Biederman's geon theory (Biederman, 1987) is a recognition by components (RBC) theory and centers on the representation of visual forms. This theory postulates that stimulus objects are represented by primitive shapes or elementary parts called geons, like cylinders, bricks, or cones, and their relationships to one another. According to the theory, two images or objects will be perceived to be similar if the same parts and relations are visible in both, even if details of the images of various parts change considerably. As an example, a rotated object with none of the parts or relations obscured is still recognized as being similar to the unrotated object. The four assumptions of this theory are:

- Objects are represented as an arrangement of geons, which are simple convex or concave parts.

- Geons have viewpoint invariant properties which can distinguish two way contrasts or differences, like straight versus curved, instead of metric properties like degree of curvature.

- Geon relations are explicit (i.e., PERPENDICULAR-TO or TOP-OF) structural descriptions, rather than implicit in a coordinate space, and
- The number of geons required are relatively few.

The figure below (Figure 2.7) from Ashby & Perrin (1988) shows five geons and five objects.

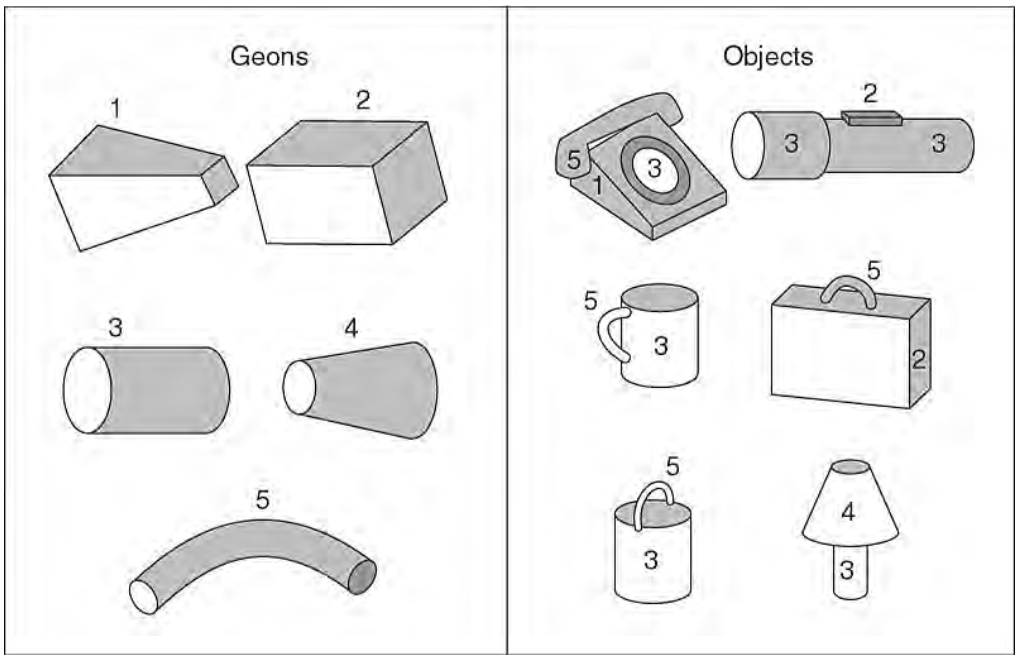


Figure 2.7: Geons and related objects (Biederman, 1987)

Note that the cup and the bucket objects are composed of the same geons, but in different relations. TOP-OF is a relation. Also notice that if the page is rotated 180 degrees, the cup will resemble a cap and the lamp a trowel or shovel.

2.6.2 Experimental measurement and computation of similarity

The process of assigning numeric values to objects using a set of rules is called measurement (Blough, 2001). This process serves both to understand and test

theoretical models of the measured objects. This section describes how various similarity measures are obtained and used.

2.6.2.1 Stimulus generalization

This is the transfer of a response learned from one stimulus to a similar stimulus. In other words, it is the tendency of a subject to respond to similar, but not identical, stimuli to the original conditioning stimulus. For example, if a dog has been conditioned to salivate at the sound of a bell ringing, it will still salivate if the bell rings at a higher or lower tone. Another example is that of a child taught to fear a stuffed white rabbit. It will exhibit fear of objects similar to the conditioned stimulus, such as a white toy rat. Guttman & Kalish (1956) used this method in similarity experiments with pigeons being rewarded for pecking a key illuminated at 550 nm. Eleven wavelengths from 490 to 610 nm were used in random order. The response of the birds to the different wavelengths was related to how similar they were to the conditioning stimulus (550 nm). This method of experimental measurement is mostly applicable to non-verbal subjects (i.e., animals) and has not been used much in human perceptual similarity studies.

2.6.2.2 Multidimensional scaling

Multidimensional scaling (MDS), as a data reduction method, is the measurement procedure that is most closely linked to the geometric approach to similarity discussed earlier. In this experimental procedure, human subjects are made to rate image or object pairs for similarity. A matrix is created with all possible image pair values represented. These data are then used to create a spatial map, in which image distances correspond to dissimilarity between images. This map efficiently describes patterns within such data and is directly related to models for the mental representation of similarity (Nosofsky, 1992; Shepard, 1987; Tversky, 1977).

Multidimensional scaling has the following features:

- It can reduce a large amount of data to a relatively simple structure that is easy to visualize and can present important relationships in an efficient manner.

- The effectiveness of this method increases with the number of images (n) employed. Each pair of images provides a data value and the number of binary relations (R) between pairs is

$$R = \frac{n(n-1)}{2} = \frac{(n^2 - n)}{2},$$

  if self-similarity is excluded.

- A multidimensional map of similarity is like a geographic map, as it compactly captures multiple relations among perceptual images or objects. Each point in the similarity space represents an image and the distances between points represent dissimilarities between objects. In other words, the smaller the distance between image points the greater their similarity. Conversely, the greater the distance between images the less similar they are.

- The correspondence between physical and psychological measures is usually non-linear, even when the dimensions of an MDS map correspond to physical attributes, like size or intensity.

Non-metric multidimensional scaling algorithms assume only ordinal relations in the data measurement scale. The rank order of data values is closely matched to the rank order of the distances in a multidimensional space. Shepard (1980) has shown that with a sufficient number of objects, the metric structure of the space can be accurately constructed from rank order alone.

Rogowitz *et al.* (1998) used multidimensional scaling in his perceptual similarity experiments. He developed and validated a new technique for collecting similarity judgements which can provide meaningful results with a factor of four fewer judgements as compared with the paired comparison method. For these experiments, they selected a set of 97 digitized photographic images which represent a range of semantic categories, viewing distances and colours. In the first experiment, which they referred to as the "Table Scaling" experiment, nine test subjects organized printed thumbnail images (3 cm by 2 cm) on a tabletop so that similar images would be near each other and dissimilar images would be far

apart. For each test subject's solution, they measured the physical distance between each pair of images and created a similarity matrix. A pooled matrix was created by accumulating these distances across subjects. In the second experiment, which they referred to as "Computer Scaling", 15 observers compared each image to every other, taken eight images at a time, thereby reducing the number of pairwise comparisons by a factor of four and removing the bias of subjective magnitude judgements. The results in two dimensions showed a smooth progression from less human-like to more human-like in one dimension, and natural to man-made in the second dimension.

Other examples of pairwise comparison and perceptual similarity experiments without using MDS are discussed in Hastings (2005), Rosman (2007), Santini & Jain (1999) and Russell & Sinha (2001). MDS has also been used in other perceptual similarity experiments involving musical notes (Berenzweig *et al.*, 2003; Grey, 1977; McAdams *et al.*, 1995).

2.6.2.3 Cluster analysis

Cluster analysis is a procedure for identifying homogeneous subgroups of objects in a population. Clustering is an alternative to spatial representation, as it can also show how subjects perceive images and the image features used for this procedure.

A cluster analysis procedure consists of:

- Choose images or objects
- Choose variables
- Determine similarity between images
- Group objects into clusters
- Evaluate the created clusters.

The goal of the above procedure is the division of images into clusters with:

- Small within cluster variation, and

- Large between cluster variation.

Squire & Pun (1997) used clustering for their perceptual similarity experiments. Human subjects were asked to partition a set of N images into at most M unlabelled subsets. One hundred colour images were used and images could be dragged from any image set and dropped into another image set using the mouse. The partitioning could be saved when all the source images have been assigned to subsets. Test subjects were told that the notion of image similarity was entirely their choice, after being given a demonstration. The test subjects were 18 in all: 10 members of the article research team (who may be considered to have expert knowledge), and 8 undergraduate students and lay people.

The authors' findings showed that the agreement between experts is significantly higher than that between lay people. They also observed that the variation between experts was less than that between lay people, which confirmed that image similarity measures may be partially learnt. They also found positive agreement between human test subjects and some computer models of similarity.

Squire & Pun (1997) derived a variety of distance measures by applying Correspondent Analysis (CA), Principal Component Analysis (PCA) and Normalised Principal Component Analysis (NPCA) to a range of colour, segment, and arc statistics. The authors observed that the average K_b (Kappa Agreement measure) between humans was 0.3468, whereas that between humans and machine partitionings was 0.1067. They concluded that the machine technologies provided better than chance agreement with human subjects, but that they were not as good as the "average" human.

Scassellati *et al.* (1994) used clustering for their study of how computational shape similarity corresponds to the human psychological idea of shape similarity. Shape similarity using algebraic moments, spline curve distances, cumulative turning angle, sign of curvature and Hausdorff-distance were compared to 40 human similarity judgements on twenty test shapes against a 1415 image database. A graphical interface allowed subjects to see both the query image and the database while selecting images. Test subjects were instructed that:

- They would be shown one image and then asked to look at a photo album to find similar pictures (no definition of similarity was offered).

- The relative size and orientation of the images in the photo album were unimportant.
- They (test subjects) could choose up to ten matching images for each query image and that the selection order was unimportant.

Both the query image and the database images were presented in random order. Test subjects could not select the same object twice for the same query image. The 40 test subjects took approximately 100 minutes to complete the 20 query images, with an about 70% completion rate. Partial results from incomplete tests were accepted for completed query images. Their results showed that some computational methods were better at identifying certain shapes than others, but the turning angle seemed to provide the overall best results.

Rosman (2007) also used pairwise comparisons of abstract images in his perceptual similarity research. His study is of interest, because he not only used abstract graphics in his experiments, but also compared perceptual similarity with a computer similarity model. He used three classifications in his pairwise comparison experiments: yes, no and maybe. He further went ahead to compare his perceptual similarity measures with that of the earth mover's distance computer model, using colour and texture features. The application of colour and texture features similarity matching algorithm to his abstract images showed a 70% and 63% successful matching rate respectively, when used alone. He then experimented with combining computer models of similarity by plotting a graph, using various weights of both colour and texture methods. He concluded that an optimal result of 83% successful matching rate was achievable using weights (i.e., 0.5 or 50%) for both the colour and texture algorithms. This idea of combining various computer models of similarity, although not used in the current work, may be relevant in further studies where variations of our research that involves working with the raw generated images may be used, if one computer similarity model proves to be unsatisfactory.

2.6.2.4 Summary of perceptual similarity

This section has covered the development of perceptual similarity theories and the experimental procedures used in similarity experiments by various researchers

Table 2.4: Summary of perceptual similarity theories

Perceptual similarity theory or model	**Description**
Common element approach or alignment based model	Uses common features of both images to determine similarity
Template or transformational model	Uses amount of work needed to convert one image into the other to determine similarity
Geometric model	Uses shortness of distance between images placed in a multidemensional feature space to determine similarity
Feature-contrast model	Uses both common and uncommon features of both images to determine similarity
Geon model	Uses the number of common primitive shapes to determine similarity

over the years. This helped in the design and implementation of perceptual similarity experiments for the current study. The results of these experiments guided the development of highly similar pass-images used in our visual password scheme.

A summary of all the perceptual similarity measures discussed in this section is in Table 2.4.

2.7 Summary of literature review

This chapter looked at the various authentication methods currently in use, and discussed why visual password schemes have been considered as a viable alternative to alphanumeric passwords: namely, they are generally more secure and easier to remember. It then summarised the core techniques of CBIR, for both querying and retrieving similar images from an image database, and their related feature extraction methods. Theoretical and experimental approaches to perceptual similarity studies were then discussed, and the various techniques used for obtaining similarity measures were then studied along with some case studies. These perceptual similarity studies assisted in the design and implementation of our perceptual similarity research design, which is further discussed in the next chapter, and it reflects the importance of perceptual similarity studies in

our research. The results of the perceptual similarity experiments allowed for the generation of highly similar, but not identical images, in our visual password scheme.

Perceptual similarity study was used in this study to:

- Determine whether the human idea of similarity is reasonably consistent across humans.
- Determine how closely the computer model of similarity matches that of humans. This pre-supposes that perceptual similarity judgements are consistent enough to be compared with computer models.
- Determine what modifications will be made to the process of automated image generation and / or what images to display out of all the possible images that could be syntactically generated.
- Determine what images not to display by reason of being too similar, to avoid confusing human users of the authentication system, in order to satisfy the requirement of ease of use of any viable authentication system.

The next chapter discusses how this research study was carried out.

Chapter 3

Research Method

3.1 Introduction

The purpose of this research as discussed in the last two chapters, is to use tree grammars to generate images, out of which are selected the most similar distractor images for display along with the user's pass-image. The various similarity measures used in this study are discussed. The purpose of using similarity measures in this study is to discover the most suitable similarity measure to use for tree grammar-based generated images. These distractor images are selected using the best possible similarity measure, vis-á-vis the human idea of similarity, for these types of abstract computer generated images. This chapter shows how the various disciplines discussed in the previous chapter (Chapter 2), are brought together chronologically to achieve the overall aim of this research, which is the developement of a Passfaces type of visual password scheme that overcomes most of the limitations of existing systems. This chapter, therefore, presents the five phases that were followed in this research:

- Tree grammar-based picture generation: this was the approach used to generate the images used in this study.
- Pilot and complete online perceptual similarity study using generated images to understand the human idea of similarity: this was to ensure that the images generated by the system are generally considered to be similar by humans.

- Experiments with ranking of generated images using mathematical models of similarity: this was to correlate the computer models of similarity with the human idea of similarity, to ensure that the images automatically generated by the best similarity measure for the prototype system correspond, as much as possible, to what humans consider to be similar.
- Using knowledge gained from the aforementioned stages to develop a visual password scheme that displays nine similar images in a three-by-three on-screen grid. Java was used in this study because it is a cross-platform object oriented programming language with a rich set of available libraries and add-on components. Java programs can be executed without being installed locally, as Java applets or by using Java Web Start.
- Characterisation of the tree grammars used in this study with a view to further understanding the factors affecting the similarity of the generated images.

The next few sections will go into slightly more detail about the research methods that were used in each of the five phases of this research.

3.2 Tree Grammar-based picture generation

One way of looking at formal languages is to consider them as equivalent to tree grammars and one or more tree transducers. This is the approach adopted by Drewes (2006) and implemented in TREEBAG, a software for experimenting with formal languages. In this study syntactic picture generation is considered as consisting of four separate components, namely: trees, transducers, algebras and displays. These four separate phases represent the system flow that was used to generate images for this research.

3.2.1 Trees

An abstract syntax tree (AST), or just syntax tree, is a tree representation of the abstract syntactic structure of source code written in a programming language. Each node of the tree denotes a construct occurring in the source code. An

expression can be represented by either its tree or term forms. For example, the expression 11+2*3 can be represented by the term [+,11,[*,2,3]] or the tree form below (Figure 3.1).

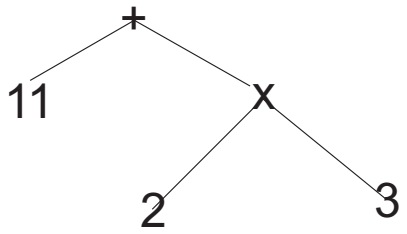


Figure 3.1: Abstract syntax tree form

The tree can be read from left to right or right to left. The leaves represent terminals or alphabet letters and the nodes represent nonterminals or algebraic operations. In picture generation, the leaves are usually graphic symbols while the nodes are graphic operations. In this study, tree grammars were used in the first stage of picture generation.

3.2.2 Tree transducers

Tree transducers transform trees by taking a tree as input and outputting a tree. A transducer can also transform an output tree into another output tree. For example, 11+2*3 can be converted into a different tree of only addition operators 11+2+2+2 by a tree transducer, if both expressions are in tree form (Figure 3.2 on the next page).

Transducers allow most formal languages to be interpreted in terms of trees and transducers. This study used one or more tree transducers for this stage of picture generation.

3.2.3 Algebras

Algebras are the various ways nodes or nonterminals are interpreted. The symbols or leaves and nodes or operations of trees can be handled differently by different algebras. It is the algebras that actually transform trees to pictures. The process is depicted in Figure 3.3 on the following page.

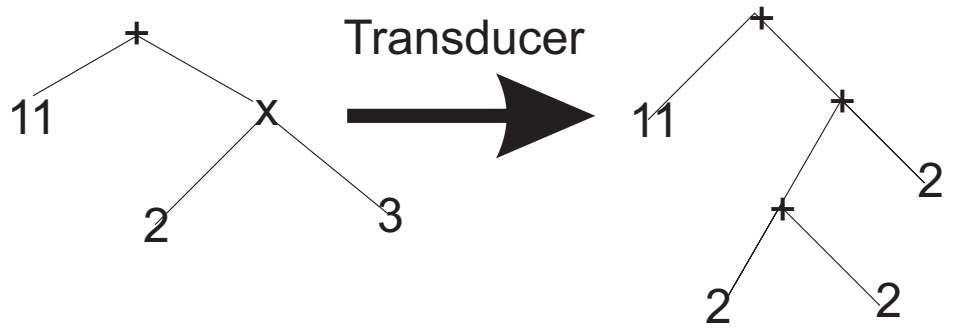


Figure 3.2: Tree transformation by transducer

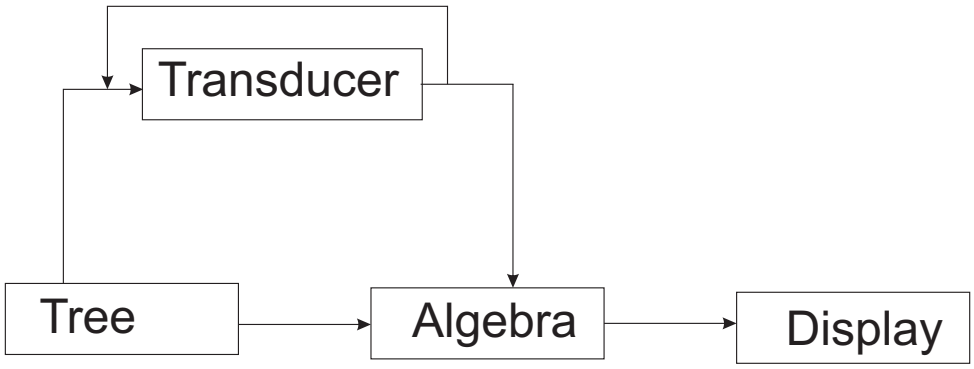


Figure 3.3: TREEBAG system flow

It is possible to use more than one transducer to generate the equivalent of a target grammar.

3.2.4 Display

The fourth component of Treebag is the display, which is responsible for displaying the target image and may be different for different types of grammars. The displayed images were saved to disk in any of the popular graphics formats. These images are used in the perceptual similarity online survey, which is the subject of the next section (Section 3.3 on page 62).

3.2.5 Tree picture grammars implemented in this study

The following grammar components in Treebag are also implemented in the Visual Password Scheme (VPS) prototype. The following grammars are as defined in Chapter 2 and / or in Drewes (2006).

3.2.5.1 Regular tree grammars

These grammars are implemented as defined in Chapter 2. A regular tree grammar is a 4-tuple (N, Σ, R, S), with the syntactically correct input shown below.

```
generators.regularTreeGrammar:
( { S,A1,A2,A3,B1,B2,B3,C1,C2,C3,ha,hb,hc,hd,Q1,Q2,Q3,Q4 },
  { S:9, C1:4, C2:2, C3:4,B1:2,B2:9,B3:2,A1:4,A2:2,A3:4,ha:9,
  hb:9,hc:9,hd:9,Q:9, Q1:9,Q2:9,Q3:9,Q4:9,sq:0, -:0 },
  {
    S -> S[A1,A2,A3,B1,B2,B3,C1,C2,C3],
    A1 -> A1[hd,hc,hb,sq],

    A2 -> A2[hc,hd],

    A3 -> A3[hd,hc,sq,ha],

    B1 -> B1[hb,hd],

    B2 -> B2[sq,Q3,sq,Q2,sq,Q4,sq,Q1,sq],

    B3 -> B3[ha,hc],

    C1 -> C1[hd,sq,hb,ha],

    C2 -> C2[ha,hb],

    C3 -> C3[sq,hc,hb,ha],
```

```
    ha  -> ha[sq,sq,ha,sq,ha,-,ha,-,-],
    ha  -> ha[sq,sq,sq,sq,sq,-,sq,-,-],
    hb  -> hb[hb,sq,sq,-,hb,sq,-,-,hb],
    hb  -> hb[sq,sq,sq,-,sq,sq,-,-,sq],
    hc  -> hc[hc,-,-,sq,hc,-,sq,sq,hc],
    hc  -> hc[sq,-,-,sq,sq,-,sq,sq,sq],
    hd  -> hd[-,-,hd,-,hd,sq,hd,sq,sq],
    hd  -> hd[-,-,sq,-,sq,sq,sq,sq,sq],
    Q1  -> Q1[sq,sq,sq,sq,Q1,sq,ha,-,hb],
    Q1  -> Q1[sq,sq,sq,sq,sq,sq,sq,-,sq],
    Q2  -> Q2[hb,sq,sq,-,Q2,sq,hd,sq,sq],
    Q2  -> Q2[sq,sq,sq,-,sq,sq,sq,sq,sq],
    Q3  -> Q3[hc,-,hd,sq,Q3,sq,sq,sq,sq],
    Q3  -> Q3[sq,-,sq,sq,sq,sq,sq,sq,sq],
    Q4  -> Q4[sq,sq,ha,sq,Q4,-,sq,sq,hc],
    Q4  -> Q4[sq,sq,sq,sq,sq,-,sq,sq,sq]

    },
  S
)
```

Each pair of braces represents an element as defined in the 4-tuple, with each element separated by a comma, and the complete grammar definition enclosed in normal brackets. In other words, the first set of braces represent N, the second Σ, the third represents the set of production rules R and the last is the start symbol S. The symbols in Σ are defined within the second pair of braces, where $S : 9$ represents a symbol S of rank 9. This is actually the grid picture grammar used in the prototype, as grammar 21.

3.2.5.2 Parallel deterministic total tree grammars (pdt)

These grammars are like regular tree grammars, except that, for every nonterminal, there are only two productions, the second of which is terminating. In other words, it does not contain nonterminals. An example input file is depicted below.

```
({begin, lside, rside},
{lside:2, rside:2, .:2, 0:0},
{begin -> (lside, rside) | (0 . ),
lside -> lside[lside, lside] | lside[0, 0],
rside -> rside[rside, rside] | rside[0,  0]},
begin)
```

The above grammar generates all trees $(t.t^{'})$ such that t and $t^{'}$ are fully balanced trees of equal height, where t consists of the symbols **lside** and 0, and $t^{'}$ consists of the symbols **rside** and 0.

3.2.5.3 ET0L tree grammars

These grammarr are implemented as defined in Chapter 2. They are parallel grammars which generalize the previous pdt grammars. Below is an example input file.

```
generators.ET0LTreeGrammar:
( {S,A2,B2,C2,Ht,Hb,Vl,Vr},
  { H:1,S:3,A2:7,B2:9,C2:7,Hb:3,Ht:3,Vr:3,Vl:3,X1:5,X2:5, sq:0, -:0 },
  {
    {
    S -> S[A2,B2,C2]},
    {A2 -> A2[Ht,Vr,sq,Vl,Vr,sq,Vl],
    B2 -> B2[Vr,sq,Vl,Vr,sq,Vl,Vr,sq,Vl],
    C2 -> C2[Vr,sq,Vl,Vr,sq,Vl,Hb]

   },
   {
    Hb -> Hb[Hb,Hb,Hb]},
    {Ht -> Ht[Ht,Ht,Ht]},
    {Vr -> Vr[Vr,Vr,Vr]},
    {Vl -> Vl[Vl,Vl,Vl]
    },
    {
```

```
    Hb -> Hb[sq,-,sq],
    Ht -> Ht[sq,-,sq],
    Vr -> Vr[sq,-,sq],
    Vl -> Vl[sq,-,sq]
    }
   },
  H[S], 1 2 3+ 4+ 5+ 6+ 7
)
```

The above example also shows the use of regular expressions to control the application of production rules. The symbols $*$ and $+$ have their usual meanings of "zero or more" and "one or more" respectively. Commas are used to separate alternatives, square brackets enclose optional parts, and parentheses $(\dots)$ are used to override the precedence rule. For example, in a grammar with three tables, the expression (1, 2 3)*2+ will start with a sequence of table 1 or tables 2 and 3, where tables 2 and 3 occur only in direct succession and in the stipulated order.

3.2.5.4 Branching tree grammars

This is implemented as defined in Chapter 2, Section 2.5.2 on page 34. The input file for this grammar is similar to that of an ET0L grammar, with the following differences.

- The third and fourth components specified must be the nesting depth n and the set of synchronization symbols respectively. This is specified after the set of nonterminals and the output signatures.
- The structure of the table is denoted as a nested hierarchy of sets of rules. In other words, $n = 0$ equates to a regular tree grammar and the table is just a set of rules, $n = 1$ equates to an ET0L grammar with a set of sets of rules, etc.
- Every nonterminal in the axiom and in the right-hand side of a rule is followed by n synchronisation symbols between angular brackets and separated by white space.

3.2.5 Tree picture grammars implemented in this study

```
generators.BSTGrammar:
  ( { tri1, tri2 },
    { F:8, G:4, p1:0, p2:0 },
    { 1, 2 },
    1,
    {
      {
        tri1 -> G[tri1<1>,tri1<2>,tri2<2>,tri1<2>],
        tri2 -> G[tri2<2>,tri2<1>,tri1<1>,tri2<1>]
      },
      {
        tri1 -> p1,
        tri2 -> p2
      }
    },
    F[tri1<1>,tri1<1>,tri1<1>,tri1<1>,tri1<1>,tri1<1>,tri1<1>,tri1<1>] )
```

The above grammar was used as grammar 16 in the prototype. The following components are also implemented in the prototype, and further details can be found in (Drewes, 2006):

- Tree transducers
  - Deterministic bottom-up tree transducers
  - Top-down tree transducers
  - Tolerant top down tree transducers
  - YIELD mappings
  - Macro tree transducers
  - Tolerant macro tree transducers
  - Tree transducer iterator
- Algebras

  - Chain code algebra
  - Turtle algebras
  - Link algebras
  - Collage algebras
  - Grid collage algebras
  - YIELD algebra
  - Free term algebra
  - String algebra

- Displays
  - Line drawing display
  - Collage display
  - Textual display
  - Tree display

3.3 Perceptual similarity

An online perceptual similarity survey was conducted to determine how humans determine similarity of grammatically generated abstract images. Human subjects were contacted via emails and given the url (uniform resource locator) of the survey site. After the test subjects have been provided assurances of anonymity and confidentiality, they then gave their consent. Ethics clearance was obtained for this research.

Background information was requested on Gender, Age range, Ethnicity, Computer literacy, Educational qualification and Country of respondent. After the preliminary series of questions mentioned above, Figure sets 5.2 on page 102 to 5.7 on page 103 were presented to the respondents to arrange in order of similarity to the first image in each figure set. The pictures were arranged in random order, different for each respondent. Initially, all the images were placed in the leftmost of two columns, and the respondent were required to drag the

images to the right column in order of similarity to the question image. As a control measure to determine outliers, the first image in every figure set was used as both the question image and an answer image.

The online survey can be found in Appendix B.

3.4 Similarity methods used in this study

The online perceptual similarity survey images were also compared with eight, state of the art, computer based similarity measures to determine how closely they modelled perceptual similarity. Since all the images were generated with tree grammars, the most effective and successful measure of tree similarity, the tree edit distance, was also used to compare the images. Eight different types of tree edit distance measures were used in order to cover the broad range of tree edit distance and tree edit distance approximation methods. Tree edit distance approximation methods are known to run in linear time compared to their full tree edit distance in quadratic time. All the computer based similarity methods were then correlated with the online similarity survey results to determine which ones more closely modelled perceptual similarity. The results were then analysed in the light of some modern psychological theories of perceptual similarity discussed in the literature review chapter.

The section discusses the two classes of similarity measures used in this study. The first class of similarity measures are drawn from variations of the most popular measure of tree similarity: the tree edit distance. These measures can be considered to be internal, in that they work on the abstract syntax trees of generated images while the CBIR measures are external because they operate on the bitwise graphical representation of images.

3.4.1 Tree edit distances

A Tree is a graph with nodes $N(T)$ and edges $E(T)$, that is directed, acyclic, connected and non-empty. An ordered pair (n, m) is an edge if $n, m \in N(T)$ and n is the parent of m. Graphs with trees as connected components are forests. Thus, a tree is also a forest. Each node, n, has at most one parent. Nodes with

the same parents are siblings. The number of n's children is its fanout f_n, and is equal to the number of outgoing edges. The root node, $root(T)$, is the node with no parent and a leaf is a node without children. Each node n has a label, $l(n) \in \Sigma$, where Σ is a finite alphabet. If a total order is defined for each group of labelled sibling nodes, such a tree is referred to as an ordered labelled tree, which is the focus of this thesis. The depth of a tree is the length of the longest path from the root to any one of the leaves. We use the standard edit operation costs: $c_d(v)$ for deleting node v and connecting its children to its parents and maintaining the order; $c_i(v)$ for inserting a new node, v, between an existing node, say w, and the consecutive subsequence of w's children; $c_r(v, w)$ for renaming the label of node v to w (Figure 3.4) on page 65 (Pawlik & Augsten, 2011). The recursive solution to the tree edit distance computation problem is shown in Figure 3.5 on page 66 (Zhang & Shasha, 1989).

Approximate tree edit distance measures were developed as a result of the long running times of "normal" tree edit distance measures. A general feature of these approximation tree edit distance measures is near linear running times. What follows is a brief discussion of the eight tree edit distance measures, both normal and approximation, used in this study.

3.4.1.1 General tree edit distance (ED)

The tree edit distance is the minimum sequence of node edit operations needed to transform one tree into another. The edit operations are usually node insertion, node deletion and node relabelling (Figure 3.4 on the facing page). Zhang & Shasha (1989) presented an algorithm for the recursive solution to the tree edit distance problem. This is given in Figure 3.5 on page 66. The nodes v and w are either both the leftmost or rightmost root nodes of the respective forests. Zhang & Shasha (1989)'s leftmost tree path decomposition algorithm used in this study is as proposed in Pawlik & Augsten (2011). This algorithm uses $O(n^2 \min^2(l, d))$ time and $O(n^2)$ space for trees with n nodes, l leaves and depth d.

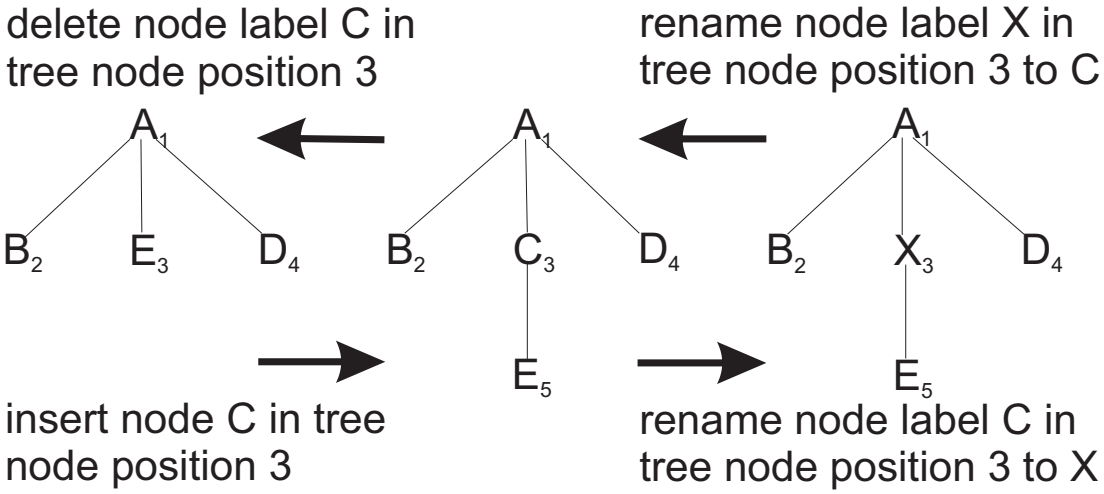


Figure 3.4: Tree edit operations

3.4.1.2 Fanout weighted (FW), pq-gram (PQ) and Windowed pq-gram (WPG) distance measures

The cost c of changing a non-leaf node is proportional to the fanout. The cost of changing a leaf node is a constant, $c = 1$, in this study and in Figure 3.6 on the following page.

Augsten *et al.* (2005, 2010), proposed both the Fanout and pq-gram distance measures. The Fanout weighted edit distance was developed to correct the non intuitive behaviour of the general tree edit distance in specific scenarios. This is illustrated in Figure 3.6 on the next page (Augsten *et al.*, 2010). The Fanout edit distance takes into consideration the number of children possessed by nodes involved in tree edit operations.

Definition 3.4.1 (Fanout Weighted Edit Distance) *Let T_1 and T_2 be two trees and let $\Sigma N(T_1)$ be the set of all the nodes in T_1. Then $w_1 \in \Sigma N(T_1)$ is a node with fanout f_1, $w_2 \in \Sigma N(T_2)$ is a second node with fanout f_2 and $c \geq 0$ is a constant. The Fanout Weighted Edit Distance, $\delta_f(T_1, T_2)$, between T_1 and T_2 is as defined in the general tree edit distance with the following costs for the edit operations.*

- *Delete:* $\alpha(w_1 \to \epsilon) = f_1 + c$,
- *Insert:* $\alpha(\epsilon \to w_2) = f_2 + c$, *and*

$\delta(\emptyset, \emptyset) = 0,$
$\delta(F, \emptyset) = \delta(F - v, \emptyset) + c_d(v),$
$\delta(\emptyset, G) = \delta(\emptyset, G - w) + c_i(w),$
if F is not a tree or G is not a tree:

$$\delta(F, G) = \min \begin{cases} \delta(F - v, G) + c_d(v) \\ \delta(F, G - w) + c_i(w) \\ \delta(F_v, G_w) + \delta(F - F_v, G - G_w) \end{cases}$$

if F is a tree or G is a tree:

$$\delta(F, G) = \min \begin{cases} \delta(F - v, G) + c_d(v) \\ \delta(F, G - w) + c_i(w) \\ \delta(F - v, G - w) + c_r(v, w) \end{cases}$$

Figure 3.5: Recursive formula for tree edit distance

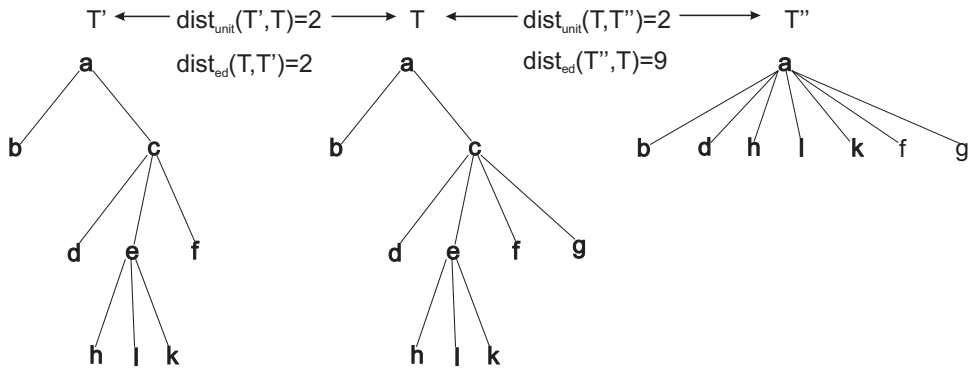


Figure 3.6: Tree edit distance with unit cost ($dist_{unit}$) and Fanout Weighted ($dist_{ed}$) cost model (Pawlik & Augsten, 2011).

- *Rename:* $\alpha(w_1 \to w_2) = (f_1 + f_2)/2 + c.$

The cost c of changing a non-leaf node is proportional to the fanout. The cost of changing a leaf node is a constant, $c = 1$, in this study and in Figure 3.6.

pq-gram distance is an approximation of fanout weighted tree edit distance for ordered labelled trees. The pq-gram pattern of a tree consists of an anchor node with $p - 1$ ancestors and q children, which translate to subtrees of a particular shape. The number of pq-grams in common determines the similarity of the two trees. This study uses a pq-gram of $p = 2$ and $q = 3$. Other values of p and q

were tried by us with no noticeable difference in the results obtained. The pq-gram distance can be computed in $O(n \log n)$ time and $O(n)$ space. Pawlik & Augsten (2011) were able to show that this distance is an effective and efficient approximation of the Fanout Weighted edit distance, and that it provides its lower bound. Windowed pq-gram (Augsten, 2008; Augsten *et al.*, 2012) extends pq-grams to unordered trees where sibling order is not important. It can also be applied to ordered labelled trees and the values of $w = 2$, $p = 2$ and $q = 3$ were used for this study, since changes in these values had no noticeable effect on the results obtained.

3.4.1.3 Tree embedding distance (TE)

Garofalakis & Kumar (2005) introduced an additional edit operation, with subtree move, in their variation of tree edit distance called tree embedding distance. This approach embeds the tree edit distance with subtree move into a numeric vector space. It runs in $O(n \times \log^* n)$ time and $O(n)$ space, computing an approximate tree edit distance with subtree move to within $O(\log^2 \times \log^* n)$ factor. $log^* n$ represents the number of log applications required to reduce n to a quantity that is less than or equal to 1. A possible weakness in this method is that it uses a unit cost model for edit operations and weighs structural changes less than the Fanout Weighted Edit Distance.

3.4.1.4 Binary branch distance (BB)

This approach (Yang *et al.*, 2005) splits trees into smaller subtrees which are their binary branches. Each tree node is transformed into the node itself, its right sibling and its first child, while discarding the edges to other children. Similarity of two trees depends on the number of binary branches in common and is a lower bound for unit cost model tree edit distance, which is the general tree edit distance with a unit cost for all tree edit node operations. A possible weakness of this approach is that the discarded edges are usually very important in distinguishing between trees with different structures with similar sibling sequences.

3.4.1.5 Bottom-up tree edit distance (BU)

The bottom-up tree edit distance (Valiente, 2001), for two trees T_1 and T_2, is equal to $1 - f/\max(n_1, n_2)$. The algorithm starts from the leaf nodes and tries to find the largest possible common subtree of the two trees, f. n_1 and n_2 are the number of nodes of T_1 and T_2 respectively. Advantages of this method are:

- Edit operations and their costs need not be defined,
- low complexity: $O(n_1 + n_2)$ time, and
- has the same low complexity for unordered trees.

A possible weakness of this approach is its sensitivity to differences in leaf nodes, such that inner nodes are not compared once the leaf nodes are different.

3.4.1.6 Fullpath tree edit distance (FP)

Fullpath (Buttler, 2004) uses tree paths from the root node to the leaf nodes in its computation of approximate tree edit distance. The similarity between two trees or documents is measured by path similarity. The definition of a path is a list of connected nodes, starting from the root and terminating at a leaf node. Path similarity can be measured in several ways. The cheapest option is binary similarity, where a path is either equivalent or not to another path. Unique paths are matched using joins or connections. Similarity is then computed as the ratio of matched paths to the total number of paths in the two trees. This approach requires $O(n)$ time to create the representation of the tree or document, with a constant time comparison thereafter, where n is the number of tree nodes.

3.4.2 CBIR measures used in this study

The following are the eight visual descriptors used in this study. These eight image descriptors represent the broad spectrum of recent and effective image similarity matching techniques in current use.

1. State of the art descriptors:

- Colour Histogram (CH).
- Auto Correlogram (AC).
- Tamura Texture Features (TTF).

2. MPEG-7 standard descriptors:

- Edge Histogram Descriptor (EHD).
- Colour Layout Descriptor (CLD).

3. Compact composite descriptors:

- Colour and Edge Directivity Descriptor (CEDD).
- Fuzzy Colour and Texture Histogram (FCTH).
- Spatial Colour Distribution Descriptor (SpCD).

The first three (CH, AC and TTF) fall under the category of state of the art descriptors (Chatzichristofis *et al.*, 2009; Lux & Marques, 2013). EHD and CLD are from the MPEG-7 standard, while the last three (CEDD, FCTH and SpCD) are compact composite descriptors. All of them are meant to work with all types of images (coloured, black and white, hand drawn, etc), but some may be more suited to certain types of images than others, which is the reason for these experiments.

In the category of state of the art descriptors, Colour Histogram was selected as it is the colour feature that started the field of CBIR (Swain & Ballard, 1991) and is still the most widely used colour descriptor (Deselaers, 2008). Its major disadvantage is the lack of spatial colour distribution information. Auto correlogram (Long *et al.*, 2003) was selected to compensate for this lack of spatial colour distribution information in Colour Histograms. Auto correlogram is a popular and highly effective colour descriptor with spatial colour distribution information.

Texture descriptors can be broadly categorised into three (Gonzalez *et al.*, 2009).

1. Statistical models: This approach can be further classified into

- the use of statistical moments of the gray-level histogram of an image or region to determine texture properties.
- the use of a descriptor, like energy, contrast and homogeneity derived from the image's gray-level co-occurence matrix, which was originally proposed by Haralick *et al.* (1973).

2. Spectral models: These models depend on the analysis of the power spectral density function in the frequency domain, like wavelet transform (Chang & Kuo, 1993; Gross *et al.*, 1994; Laine & Fan, 1993).

3. Structural models: These models include methods that describe textures in terms of primitive texels (Shapiro & Stockman, 2001) in some regular or repeated relationship, which is especially suited to artificial regular patterns.

Tamura Texture Features (Tamura *et al.*, 1978), which is a classical set of six different characteristic texture features was selected for this study. Motion Photographers Expert Group 7 (MPEG-7) also recommends a set of descriptors for determining similarity of images. One of the MPEG-7 colour descriptors (Colour Layout Descriptor) and one texture descriptor (Edge Histogram Descriptor) were selected for this study.

A number of descriptors that combine two or more descriptors into one composite descriptor are included in this study. Colour and Edge Directivity Descriptor, Fuzzy Colour and Texture Histogram and Spatial Colour Distribution Descriptor were selected because they are currently being used in some CBIR systems (Chatzichristofis *et al.*, 2009; Lux & Marques, 2013) and have been shown to have both high precision and recall rates.

3.4.2.1 Colour Histogram

Colour Histograms (Huang *et al.*, 1997) record the frequencies of colours in an image and are used to describe images in order to perform image retrieval. Swain & Ballard (1991) introduced the complement of the histogram intersection method, which is defined as

$$d_{\text{HIS}}(I_1, I_2) = 1 - \sum_{k=1}^{N} \min(H_1(k), H_2(k)) \ , \tag{3.1}$$

where H_1 and H_2 are the colour histograms of images I_1 and I_2, and N is the number of bins used for representing the histogram. It can be shown that the histogram intersection is equivalent to the l_1 norm and hence a metric.

3.4.2.2 Auto-Correlograms

Colour correlograms is a method that incorporates information on the spatial correlation between colours present in an image and is defined as

$$\gamma_{c_i,c_j}^{(k)}(I) = \mathrm{PR}_{p_1 \epsilon I_{c_i}, p_2 \epsilon I}[p_2 \epsilon I_{c_j}, |p_1 - p_2| = k] \ , \tag{3.2}$$

with

$$|p_1 - p_2| = \max(|X_1 - X_2|, |Y_1 - Y_2|) \ , \tag{3.3}$$

where c_i and c_j denote two colours and (X_k, Y_k) denote pixel locations. In other words, given any colour c_i in the image, γ gives the probability that a pixel at distance k away is of colour c_j.

Full colour correlograms are both computationally and memory intensive operations, therefore a simpler form called auto-correlogram (Long *et al.*, 2003) is often used in similarity matching. Auto-correlogram is defined as

$$\gamma_c^{(k)}(I) = \gamma_{c,c}^{(k)}(I) \ , \tag{3.4}$$

and is the spatial correlation of each colour to itself.

3.4.2.3 Tamura Texture Features

The Tamura texture includes six features obtained from psychological experiments (Tamura *et al.*, 1978): (i) coarseness (coarse vs.fine); (ii) contrast (high contrast vs. low contrast); (iii) directionality (directional vs. nondirectional); (iv) line-likeness (line-like vs. blob-like); (v) regularity (regular vs. irregular); and (vi) roughness (rough vs. smooth). The first three features were found to correlate well with human perception.

Coarseness provides information about the size of the texture elements of the image. An image will contain textures at several scales and the aim of coarseness

is to identify the largest size at which a texture exists, even in the presence of smaller microtextures. The coarseness of an image is calculated by first taking averages at every point over neighbourhoods, the linear size of which is determined by using powers of two i.e. $1 \times 1, 2 \times 2, 4 \times 4, \ldots, 32 \times 32$. The average over the neighbourhood of size $2^k \times 2^k$ at the point (x, y) is

$$A_k(x, y) = \sum_{i=x-2^{k-1}}^{i=x+2^{k-1}} \sum_{i=y-2^{k-1}}^{i=y+2^{k-1}-1} f(i, j)/2^{2k} \tag{3.5}$$

For each point, (x, y), the differences between pairs of averages corresponding to non-overlapping neighbourhoods on opposite sides of the point are calculated in both the horizontal and vertical directions. In the horizontal case this is

$$E_{k,h}(x, y) = |A_k(x + 2^{k-1}, y) - A_k(x - 2^{k-1}, y)| \ , \tag{3.6}$$

and the vertical is

$$E_{k,v}(x, y) = |A_k(x, y + 2^{k-1}, y) - A_k(x, y - 2^{k-1})| \tag{3.7}$$

For each point (x, y), the value which leads to the maximum difference is selected. Thereafter, the average value is taken and used as the standard for measuring the coarseness of the image.

Contrast is obtained from the following four factors:

- Dynamic range of levels of grey.
- Polarisation of the white-black distribution in the grey level histogram.
- Edge sharpness.
- Frequency of repeated patterns.

The first item in the list above is measured using the standard deviation of grey levels and the second by the kurtosis α_4. The contrast of an image is therefore defined as

$$F_{\text{con}} = \sigma/(\alpha_4)^n \ \text{ where } \ \alpha_4 = \mu_4/\sigma^4 \ , \tag{3.8}$$

where μ_4 is the fourth moment about the mean and σ^2 is the variance. The value $n = 14$ was experimentally found to be closest to human measurements.

Tamura directionality is a global property over a region and its histogram is a graph of local edge probabilities against their directional angle. The fact that gradient is a vector (i.e. has both magnitude and direction) is used in extracting directionality information from an image. In the discrete case, the magnitude $|\Delta G|$ and the local edge direction θ are approximated as follows

$$|\Delta G| = (|\Delta_H| + |\Delta_V|)/2 \ , \tag{3.9}$$

$$\theta = \tan^{-1}(|\Delta_H|/|\Delta_V|) + \frac{\pi}{2} \ , \tag{3.10}$$

where Δ_H and Δ_V are the horizontal and vertical differences measured by the Sobel edge detector in a 3×3 moving window. A real number θ $(0 \leq \theta < \pi)$ is obtained which is measured counter clockwise so that the horizontal direction is zero.

H_D, the desired histogram can be obtained by quantizing θ in n values and counting the points with magnitude $|\Delta G|$ over the threshold t.

$$H_D(k) = N_\theta(k) / \sum_{i=0}^{n-1} N_\theta(i), \qquad k = 0, 1, \ldots, n-1 \ , \tag{3.11}$$

where $N_\theta(k)$ is the number of points at which $|\Delta G| \geq t$. Thresholding $|\Delta G|$ by t is aimed at preventing counting of unreliable directions which cannot be regarded as edge points. In their experiments, they used $n = 16$ and $t = 12$. Observe that the shape of each histogram is not sensitive to the value of t.

3.4.2.4 Edge Histogram Descriptor

The spatial distribution of edges in an image is a useful similarity measure exemplified by EHD (Cieplinski, 2001). EHD represents local edge distribution in an image, by dividing the image into 4×4 subimages and representing the local edge distribution in a histogram. To produce the histogram, edges in the subimages are categorised into five types: vertical, horizontal, 45° diagonal, 135° diagonal and nondirectional edges (i.e., edges with no particular direction). As there are

16 subimages, a total of $5 \times 16 = 80$ histogram bins are required. A block based edge extraction scheme is used to extract the five types of edges. To achieve that, each subimage is further divided into nonoverlapping square image blocks whose size is dependent on the image resolution. However, a predetermined number of image blocks of around 1100 is used to capture good directional edge features. Each image block is then classified into one of the five edge categories mentioned previously or as a non edge block.

After classification, the five-bin-histogram for each subimage is obtained by counting each edge type for all image blocks in the image. Each bin value is then normalised by the total number of image blocks in the image. The normalised bin values are then nonlinearly quantized for binary representation. The l_1 distance measure $D(A, B)$, can be used for similarity matching for two histograms A and B as shown in Equation 3.12.

$$D(A,B) = \sum_{i=0}^{79} |h_A(i) - h_B(i)| + \sum_{i=0}^{4} |h_A^g(i) - h_B^g(i)| + \sum_{i=0}^{64} |h_A^S(i) - h_B^S(i)| \ , \tag{3.12}$$

where $h_A(i)$ and $h_B(i)$ are the normalised histogram bin values, $h_A^g(i)$ and $h_B^g(i)$ are the normalised bin values for the global edge histograms obtained from the corresponding local histograms $h_A(i)$ and $h_B(i)$, while $h_A^S(i)$ and $h_B^S(i)$ represent the histogram bin values for the semiglobal-edge histograms of image A and B repectively.

3.4.2.5 Colour Layout Descriptor

The Colour Layout Descriptor (Sikora, 2001) describes the spatial distribution of colour in a very compact form. It can be used for a wide variety of similarity-based retrieval types, like sketch based image retrieval, content filtering using image indexing and visualisation. The functionality of this descriptor can be achieved using a combination of the Grid-layout datatype of MPEG-7 and the Dominant Colour Descriptor. However, a combination like this would require

a relatively large number of bits, and matching would be more complex and expensive.

CLD is obtained by dividing the image into an 8×8 grid, and then getting the representative colour of each tile using the YC_bC_r colour space. CB and CR are the blue-difference and red-difference chroma components, while Y is the luminance. A Discrete Cosine Transform (DCT) is performed on the 8×8 block and the DCT coefficients are used as the descriptor.

For matching two CLDs, (DY, DC_r, DC_b) and (DY', DC'_r, DC'_b), the following distance measure is used

$$D = \sqrt{\sum_i w_{yi}(DY_i - DY'_i)^2} + \sqrt{\sum_i w_{bi}(DCb_i - DCb'_i)^2} + \sqrt{\sum_i w_{ri}(DCr_i - DCr'_i)^2} \tag{3.13}$$

3.4.2.6 Colour and Edge Directivity Descriptor and Fuzzy Colour and Texture Histogram

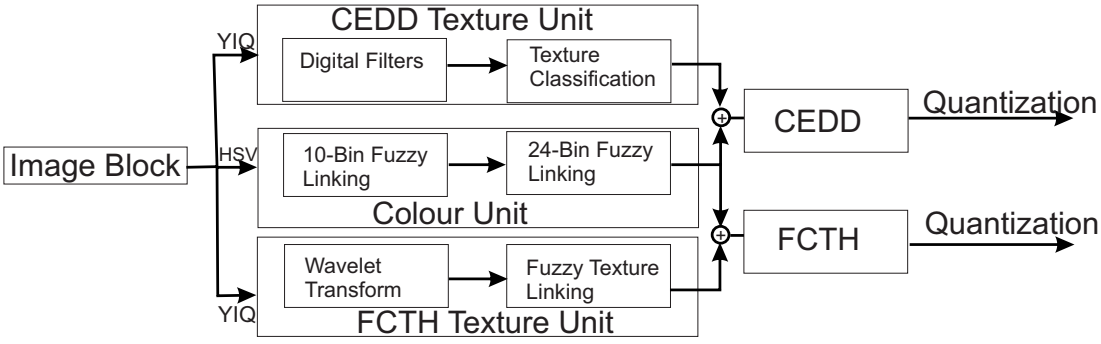


Figure 3.7: CEDD and FCTH flowchart

Figure 3.7 shows the implementation flowchart for CEDD (Chatzichristofis & Boutalis, 2008a) and FCTH (Chatzichristofis & Boutalis, 2008b). The colour and texture units are reponsible for the extraction of colour and texture information respectively. Observe that both CEDD and FCTH share the same colour unit.

The CEDD texture unit is used to determine the six regions of a CEDD histogram. Each region is further subdivided into 24 regions by the colour unit. The resulting histogram now has 144 (6×24) regions. As a compromise between the image detail and computational power, the image is separated into 1600 image blocks. Each image block is further divided into four regions called sub blocks in the texture unit. The sub block value is determined from the mean value of the luminosity of its pixels.

FCTH texture unit transforms the YIQ colour space it uses with the Haar Wavelet transform such that each block is classified into one of eight output bins. In YIQ colour space, Y represents the luma information, I stands for in-phase which is the orange-blue range while Q stands for quadrature and the purple-green range.

The colour unit of both CEDD and FCTH converts every image block into HSV (Hue Saturation Value) colour space. The inputs to the fuzzy system are the mean values of H, S and V which shape the fuzzy 10-bin histogram. The histogram is then normalised in the interval zero to one and quantized in three bits per bin. The result of this quantization is a very compact 54 byte CEDD descriptor and a 72 byte FCTH descriptor.

3.4.2.7 Spatial Colour Distribution Descriptor

This descriptor was created especially for hand drawn sketches and colour graphics, since such images contain relatively small numbers of colours and less texture than natural coloured images.

SpCD (Chatzichristofis *et al.*, 2011a, 2010) combines colour and spatial colour information. The image is first decomposed into 16 sub-images (4×4), with each pixel being transformed into the HSV colour space. The values of H, S and V then go into the fuzzy-linking system which outputs either a zero or a one for each of the eight preset colours. The eight preset colours are: (0) Black, (1) Red, (2) Orange/Yellow, (3) Green, (4) Cyan, (5) Blue, (6) Magenta and (7) White. A set of 28 rules is used for the defuzzification process, with fuzzy antecedents and crisp consequents. At the end of this process, eight images called tiles of three bits each are produced, with each of them describing the quantitative and spatial

distribution of each colour.

Scanning row-by-row, each tile is used to form a 16-dimensional vector, with each vector requiring three bits for its representation. This process is repeated for all the tiles to produce eight vectors, which when combined by successive placement, forms a 48 byte compact composite descriptor.

The prototype developed is for use by humans, therefore the images presented must be similar from the human point of view. Perceptual similarity is therefore the subject of the next chapter.

3.5 Visual password scheme

In this phase the best mathematical models were incorporated into the syntactic image generation process.

The approach considered was to generate the images and then use the most suitable distance measure to determine whether they are similar enough to our password image, before displaying it to the user. The generated images will then be displayed in a three-by-three on-screen grid or matrix for authentication in a prototype visual password scheme.

The prototype was then developed based on our cumulative findings.

The next section discusses two issues addressed in the course of collecting the data used for our research.

3.6 Data collection method

An online perceptual similarity survey was conducted in this study. The next two sections covers the sample size and research instruments used.

3.6.1 Sample size

The population size is represented by all computer users. A convenience sampling method was used, because it is economical in terms of time, effort and finance. This non probability method is often used during preliminary research efforts to get a gross estimate of the results, without incurring the cost or time required

to select a random sample. The sample size needed for our study for the earth's population at 95 % Confidence level and 5% Confidence Interval is 384 (Creative Research Systems, 2012; Hsieh *et al.*, 1998; Kadam & Bhalerao, 2010; The Research Advisors, 2006). The sample size used in this research study is 661, which is well above the required 384.

3.6.2 Research instruments

A self administered on-line survey was used. Since the survey participants needed to compare sets of graphics, it was impractical to conduct:

- On the streets
- By post
- Without orientation

The self administered survey process has the following obvious advantages (Eiselen, 2012).

- Self administered surveys are usually more cost-effective to administer than personal (face-to-face) interviews.
- They are easy to administer and analyse when compared to other methods.
- People are familiar with the concept of a survey.
- They are more likely to reduce the possibility of interviewer bias.
- Most people perceive surveys to be less intrusive than telephone or face-to-face surveys and hence, respondents will more readily respond truthfully to sensitive questions.

The fact that the survey was conducted on-line meant that test subjects could participate from the comfort of their homes or offices.

A sample set of images and associated questionnaire can be found in Appendix A and B respectively.

3.7 Summary of research methods

This chapter looked at the four different areas or disciplines relevant to this study. After a brief introduction in the first section, the next section indicated how the syntactically generated images were obtained: specifically, through the use of Treebag software and its associated grammars.

The third section (Section 3.3 on page 62) reviewed the conduct of the on-line perceptual similarity experiments, the aim of which was to find a similarity measure that matches as much as possible the human one. This was to ensure that the automatically generated images were similar from the human point of view (Section 3.4 on page 63).

The next section (Section 3.5 on page 77), showed how the various experiments were integrated into a visual password scheme that is resilient to shoulder surfing. This was done by using the best tree edit distance measure from our experiments to select, from a pool of generated images from the various grammars, a set of highly similar distractors for on-screen display to the users of our prototype system.

The next section, Section 3.6 on page 77 highlighted the on-line survey system that was used to collect our similarity data, while this section provides an overview of the entire chapter. The next chapter takes a look at the initial set of experiments conducted to determine accuracy and precision of similarity measures for tree grammar-based generated images. This was the first proof-of-concept search for appropriate similarity measures that were able to select highly similar distractor images for our visual password scheme.

Chapter 4

Similarity Measures

This chapter details the search for appropriate similarity measures to use for the types of images generated by tree picture grammars. This is an expanded version of the paper presented at PRASA 2013 without the literature review (Okundaye *et al.*, 2013).

4.1 Experimental Determination of Appropriate Similarity Measures

The explosion of images on the Internet has led to the development of various similarity measures to help in the search and retrieval of similar images from large image databases or image repositories, like flickr, WANG's and facebook, collectively referred to as CBIR. Not all computer based similarity measures are suitable for all types of images, since images range from natural images taken with digital cameras, through the gamut of computer generated graphics, to hand drawn images. Image similarity measures have been useful in other areas as well, such as computer robotic vision (Cao *et al.*, 2010) and similarity of video sequences (Adjeroh *et al.*, 1998). Determining similarity of images could be very useful in certain application areas, like visual password schemes, which are of special interest in this study, where password images and distractor images need to be very similar to defeat shoulder surfing (Lashkari *et al.*, 2009). We have created a corpus of thousands of syntactically generated images as an

experimental test-bed to use to experimentally determine the best image similarity measure for use in our study. Eight computer based similarity measures were selected for their diverse methodology and effectiveness. The eight methods are Colour Histogram, Auto-Correlograms, Tamura Texture Features, Edge Histogram Descriptor, Colour Layout Descriptor, Colour and Edge Directivity Descriptor, Fuzzy Colour and Texture Histogram and Spatial Colour Distribution Descriptor.

The purpose of the sets of experiments discussed in this chapter was to determine the most suitable computer based similarity measure for syntactically generated images. It also appeared logical to test for similarity of our images using tree pattern matching, since all of our images are tree based. Tree pattern matching has been used on tree structured data, like computing the difference between XML documents represented as ordered labelled trees. The standard tree pattern matching method for finding the difference between two trees is the tree edit distance. The tree edit distance is the minimum-cost sequence of node edit operations required to transform one tree into another. The tree edit distance has been applied successfully in different application areas, such as bio-informatics (Akutsu, 2010; Heumann & Wittum, 2009; Ma *et al.*, 2002), image analysis (Bellando & Kothari, 1999), pattern recognition (Klein *et al.*, 2000), melody recognition (Habrard *et al.*, 2008), natural language processing (Lin *et al.*, 2010), or information extraction (Kim *et al.*, 2007; Reis *et al.*, 2004), and databases (Augsten *et al.*, 2010; Chawathe, 1999; Cobena *et al.*, 2002; Dalamagas *et al.*, 2006; Garofalakis & Kumar, 2005; Guha *et al.*, 2002; Lee *et al.*, 2004). In the third set of experiments, the most successful computer based similarity measure (SpCD) is compared with tree edit distance for ten sets of syntactically generated images, representative of our corpus of images, to determine the better method. In brief, the contribution of this chapter to the literature is as follows:

- The use of tree edit distance for pattern matching of abstract syntax trees of syntactically generated images with a view to determining their similarity;
- This is the first time the effectiveness of a tree pattern matching method, specifically tree edit distance, as an image similarity measure is being compared to computer based image similarity measures;

- This study has produced a corpus of thousands of syntactically generated images, with the aim of determining the most suitable similarity measures to use for these kinds of images.

This chapter is structured as follows. Section 4.2 discusses picture grammars, tree edit distance and the visual descriptors used in this study. Section 4.3 on page 85 covers both the experimental procedures used in this research and the discussion of results, while Section 4.4 on page 96 summarises the chapter and highlights further research done in the next chapter.

4.2 Background to the study

This section covers the three disciplines brought together for this study. The formal definitions for these concepts can be found in the literature review chapter.

4.2.1 Picture grammars

The tree based approach to formal languages adopted for this study (Drewes, 2006), considers syntactic generation of pictures as a domain consisting of some picture elements or symbols, and the various operations on them. Thus tree based generators are made up of two components: a tree generator that produces the trees and a picture algebra that evaluates the trees and turns them into pictures by interpreting the symbols in those trees as picture operations. All the images in this study were produced using this approach.

Grid picture grammars (Drewes *et al.*, 2003), were used to produce the images in Figures 4.2 on page 86 and Figures 4.3 on page 86. The images were generated in the tree based approach using a regular tree grammar, a top down tree transducer and an algebra which was used to interpret the symbols of the transducer output into pictures.

4.2.2 Tree edit distance

A directed, acyclic, connected graph can be referred to as a tree T. T has nodes $N(T)$ and edges $E(T) \subseteq N(T) \times N(T)$, where each node has at most one incoming

edge. The trees are ordered labelled trees. A graph in which each connected component is a tree is called a forest F, which implies that a tree is also a forest. The tree edit distance between two trees F and G, $\delta(F, G)$, is defined as the minimum cost sequence of node edit operations (node deletion, node insertion and node renaming) needed to transform one tree into the other.

Pawlik & Augsten (2011) discussed the various state of the art approaches to providing a fast and efficient dynamic programming solution to the recursive formula, about using LRH (Leftmost tree node path, Rightmost tree node path and Heavy (longest) tree node path) paths tree decomposition algorithms. This study uses Zhang & Shasha (1989) leftmost tree path decomposition algorithm as proposed in Pawlik & Augsten (2011).

4.2.3 Visual descriptors

The following are the visual descriptors used in this study.

1. State of the art descriptors:
    - Colour Histogram (Swain & Ballard, 1991)
    - Auto Correlogram (Long *et al.*, 2003)
    - Tamura Texture Features (Tamura *et al.*, 1978)
2. Motion Photographers Expert Group 7 (MPEG-7) standard descriptors:
    - Edge Histogram Descriptor (Cieplinski, 2001)
    - Colour Layout Descriptor (Sikora, 2001)
3. Compact composite descriptors:
    - Colour and Edge Directivity Descriptor (Chatzichristofis & Boutalis, 2008a)
    - Fuzzy Colour and Texture Histogram (Chatzichristofis & Boutalis, 2008b)
    - Spatial Colour Distribution Descriptor (Chatzichristofis *et al.*, 2010)

These eight image descriptors represent the broad spectrum of recent and effective image similarity matching techniques in current use. See the literature review chapter, as well as the cited references, for a more detailed description of these descriptors.

These descriptors work with all types of images (coloured, black and white, hand drawn, etc), but these experiments were meant to find the descriptor most suited to the type of tree grammar-based generated images used in this research.

4.3 Experiments and Results

The visual descriptors used in this study are as implemented in Chatzichristofis *et al.* (2009) using C#, which is a re-implementation of Lux & Marques (2013) in Java.

4.3.1 Experimental Studies

This section discusses the experimental procedures used in this study. The first set of images (Figure 4.2 on page 86) was generated from the same grammar and they are all sub-pictures of the first image, I_1. A sub-picture is a clearly identifiable part of another picture. The second set of images (Figure 4.3 on page 86) was generated from different grammars and they are all not sub-pictures of the first image, E. This division of our data into two sets was to allow us to determine how well the various similarity methods handle both types of sets, e.g. in the second set, I is a subpicture of the rest.

4.3.1.1 Experiment on first set of data

The production rules for the first set of images are shown in Figure 4.1 on the following page. These production rules were entered into TREEBAG (Drewes, 2006) and the following seven images were generated (Figure 4.2 on the next page) at the same tree depth level i.e. same tree height.

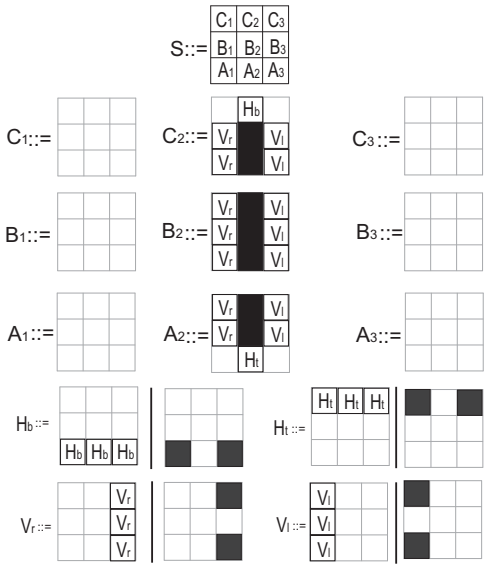


Figure 4.1: Production rules for letter I

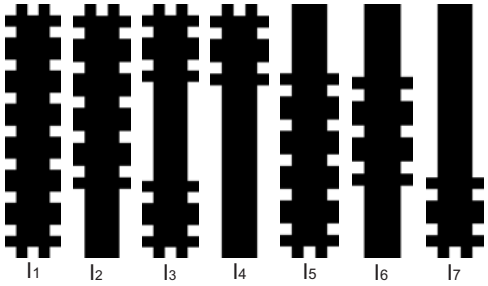


Figure 4.2: Various versions of I used as set 1 in this study



Figure 4.3: Various letters used as set 2 in this study

4.3.1.2 Experiment on second set of data

A similar procedure was followed for the letters E, F, L, I, H, T (Figure 4.3). The following assumptions were made in the experiments for these grammars.

- The non-terminals have the same meaning, i.e., refer to the same section of the grid ($(A_1 \dots C_3)$ in Figure 4.1) on page 86.



(a) I (b) II (c) III (d) IV (e) V (f) VI (g) VII (h) VIII (i) IX

Figure 4.4: Series 1: Barnsley fern



(a) I (b) II (c) III (d) IV (e) V (f) VI (g) VII (h) VIII (i) IX

Figure 4.5: Series 2: Spiral

- The images are all at the same tree depth level, which is the maximum number of nodes from the root node to the leaf of the tree.

4.3.1.3 Experiment on third set of data

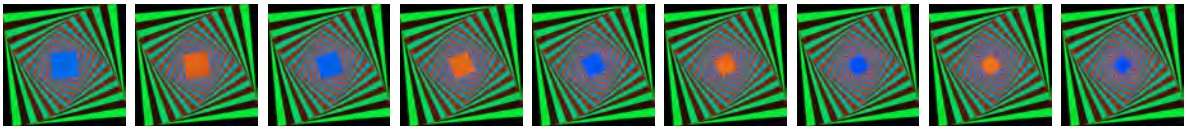
The ten sets of images generated in Figures 4.4 on page 87 to 4.13 on page 89 were the results of nine consecutive iterations of the production rules of the different grammars. Both SpCD and the general tree edit distance were used to compare all nine images to the first image in each set. Nine images were generated because nine images are usually displayed in a 3-by-3 on-screen matrix for some image based visual password schemes.

4.3.2 Results

This section contains the results of the study for all the sets of images.

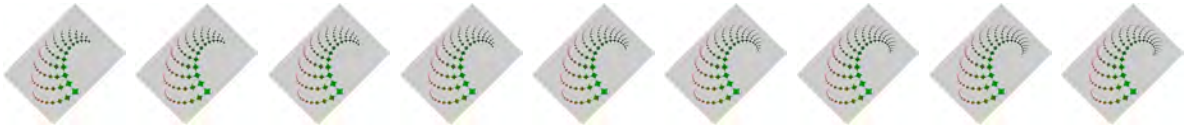
4.3.2.1 Results for data set 1

The results for the first set of experiments are presented in Table 4.1 on page 89 and graphed in Figure 4.14 on page 91. The normalised values are given in



(a) I (b) II (c) III (d) IV (e) V (f) VI (g) VII (h) VIII (i) IX

Figure 4.6: Series 3: Rotating squares



(a) I (b) II (c) III (d) IV (e) V (f) VI (g) VII (h) VIII (i) IX

Figure 4.7: Series 4: Colour operations



(a) I (b) II (c) III (d) IV (e) V (f) VI (g) VII (h) VIII (i) IX

Figure 4.8: Series 5: Spirals2



(a) I (b) II (c) III (d) IV (e) V (f) VI (g) VII (h) VIII (i) IX

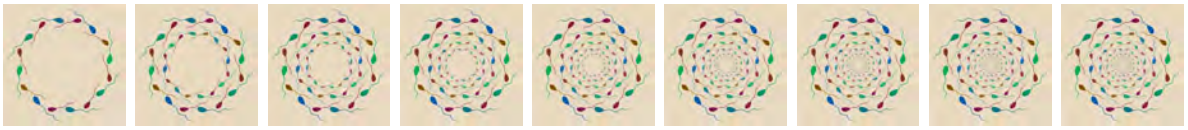
Figure 4.9: Series 6: Circles



(a) I (b) II (c) III (d) IV (e) V (f) VI (g) VII (h) VIII (i) IX

Figure 4.10: Series 7: Mosaic1

Table 4.2 on the next page on page 89 and the similarity rankings appear in Table 4.3 on page 90. Four of the methods (TTF, EHD, CLD and FCTH) give



(a) I (b) II (c) III (d) IV (e) V (f) VI (g) VII (h) VIII (i) IX

Figure 4.11: Series 8: Mosaic2



(a) I (b) II (c) III (d) IV (e) V (f) VI (g) VII (h) VIII (i) IX

Figure 4.12: Series 9: Parametric Colour Operation



(a) I (b) II (c) III (d) IV (e) V (f) VI (g) VII (h) VIII (i) IX

Figure 4.13: Series 10: Triangles

Table 4.1: Comparison of I_1 to other I's

Image	CH	AC	TTF	EHD	CLD	CEDD	FCTH	SpCD	T
I_1	0	0	0	0	0	0	0	0	0
I_2	22.0227	0.3290	0.0033	0.4103	10.5357	4.2530	6.6490	7.8292	15
I_3	45.0111	0.6840	0.0020	0.4011	14.2478	8.1301	7.8109	13.0435	18
I_4	77.0325	1.1637	0.0119	0.7081	18.4932	14.7674	9.5310	22.9508	33
I_5	22.0000	0.3346	0.0037	0.8063	11.9164	4.5127	5.0388	7.8292	15
I_6	50.0200	0.7514	0.0123	0.6347	8.6603	10.5551	13.0372	14.7651	30
I_7	78.0128	1.1751	0.0122	1.1457	19.3132	14.7674	14.8397	22.9508	33

Table 4.2: Comparison of I_1 to other I's (Normalised)

Image	CH	AC	TTF	EHD	CLD	CEDD	FCTH	SpCD	T
I_1	0	0	0	0	0	0	0	0	0
I_2	0.282	0.280	0.268	0.358	0.545	0.288	0.448	0.341	0.4545
I_3	0.577	0.582	0.164	0.350	0.737	0.550	0.526	0.568	0.5455
I_4	0.987	0.990	0.967	0.618	0.957	1.000	0.642	1.000	1.0000
I_5	0.282	0.284	0.302	0.703	0.617	0.305	0.339	0.341	0.4545
I_6	0.641	0.639	1.000	0.554	0.448	0.714	0.878	0.643	0.9091
I_7	1.000	1.000	0.999	1.000	1.000	1.000	1.000	1.000	1.0000

Table 4.3: Comparison of I_1 to other I's (Similarity ordering)

Image	CH	AC	TTF	EHD	CLD	CEDD	FCTH	SpCD	T
1	I_1	I_1	I_1	I_1	I_1	I_1	I_1	I_1	I_1
2	I_5	I_2	I_3	I_3	I_6	I_2	I_5	I_2	I_2
3	I_2	I_5	I_2	I_2	I_2	I_5	I_2	I_5	I_5
4	I_3	I_3	I_5	I_6	I_5	I_3	I_3	I_3	I_3
5	I_6	I_6	I_4	I_4	I_3	I_6	I_4	I_6	I_6
6	I_4	I_4	I_7	I_6	I_4	I_4	I_6	I_4	I_4
7	I_7	I_7	I_6	I_7	I_7	I_7	I_7	I_7	I_7

Table 4.4: Comparison of E to other letters

Image	CH	AC	TTF	EHD	CLD	CEDD	FCTH	SpCD	T
E	0	0	0	0	0	0	0	0	0
F	76.0000	1.1814	0.0004	0.2599	26.1725	1.5038	0.9375	11.7922	2.0000
L	118.1524	1.8526	0.0037	0.6966	34.9714	18.0541	5.7252	718.6700	4.0000
I	157.0127	2.4572	0.0119	0.9634	32.2800	68.7732	62.3656	41.4210	6.0000
H	55.0000	0.8655	0.0038	1.1382	27.4955	22.5806	7.9045	27.2450	2.0000
T	169.2720	2.6656	0.0035	0.9313	57.4804	16.3968	7.7061	59.6136	5.0000

Table 4.5: Comparison of E to other letters (Normalised)

Image	CH	AC	TTF	EHD	CLD	CEDD	FCTH	SpCD	T
E	0	0	0	0	0	0	0	0	0
F	0.4490	0.4432	0.0333	0.2284	0.4553	0.0219	0.0150	0.1978	0.3333
L	0.6980	0.6950	0.3151	0.6120	0.6084	0.2625	0.0918	0.3132	0.6667
I	0.9276	0.9218	1.0000	0.8465	0.5616	1.0000	1.0000	0.6948	1.0000
H	0.3249	0.3247	0.3214	1.0000	0.4783	0.3283	0.1267	0.4570	0.3333
T	1.0000	1.0000	0.2926	0.8183	1.0000	0.2384	0.1236	1.0000	0.8333

Table 4.6: Comparison of E to other letters (Similarity ordering)

Image	CH	AC	TTF	EHD	CLD	CEDD	FCTH	SpCD	T
1	E	E	E	E	E	E	E	E	E
2	H	H	F	F	F	F	F	F	F
3	F	F	T	L	H	T	L	L	H
4	L	L	L	T	I	L	T	H	L
5	I	I	H	I	L	H	H	I	T
6	T	T	I	H	T	I	I	T	I

random rankings (when using the raw scores in Table 4.1 on page 89), which may make them unsuitable for these kinds of images. The other four methods (CH, AC, CEDD and SpCD) and tree edit distance (T) give similar rankings.

SpCD and T correctly determine that image I_2 and I_5 are equally similar to I_1. CH, AC and CEDD come close (within 0.02 range) to agreeing with

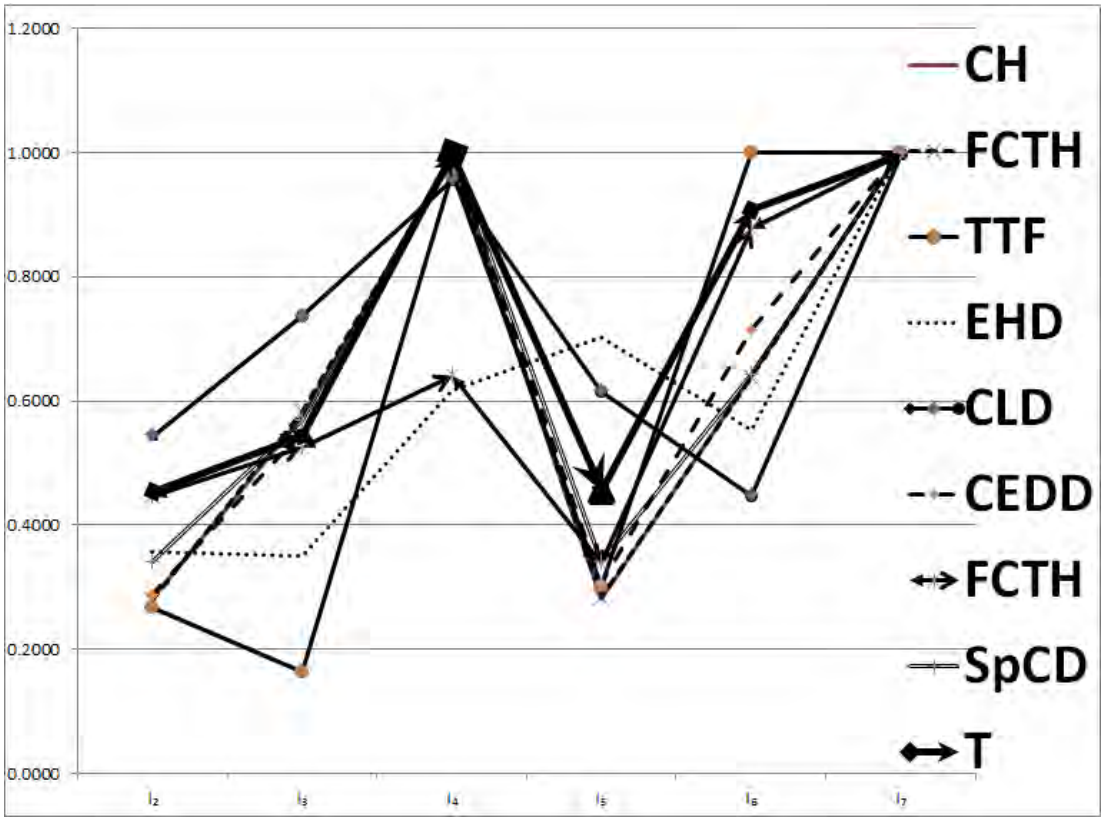


Figure 4.14: Graph for data set 1

SpCD. The other methods give varying results with the worst being CLD. This is understandable, since CLD is a colour layout or distribution descriptor, while the images being considered are grayscale images.

T, SpCD and CEDD were able to determine correctly that I_4 and I_7 are rotated images of each other and can therefore be considered as equally similar to I_1. Three other similarity methods (CH, AC and TTF) are also in agreement (within the 0.02 range). Although FCTH gives a similarity ordering close to AC and CEDD, its normalised similarity values for (I_2, I_5) and (I_4, I_7) do not reflect how closely matched these images are.

A summary of the results of experiments on data set 1 in Table 4.3 on the facing page, shows that four methods (CH, AC, CEDD and SpCD) rank I_2 and I_5 to be equally similar to I_1. This is followed by I_3 and I_6. I_4 and I_7 are also ranked to be equal and are judged to be the least similar to I_1 by the above four

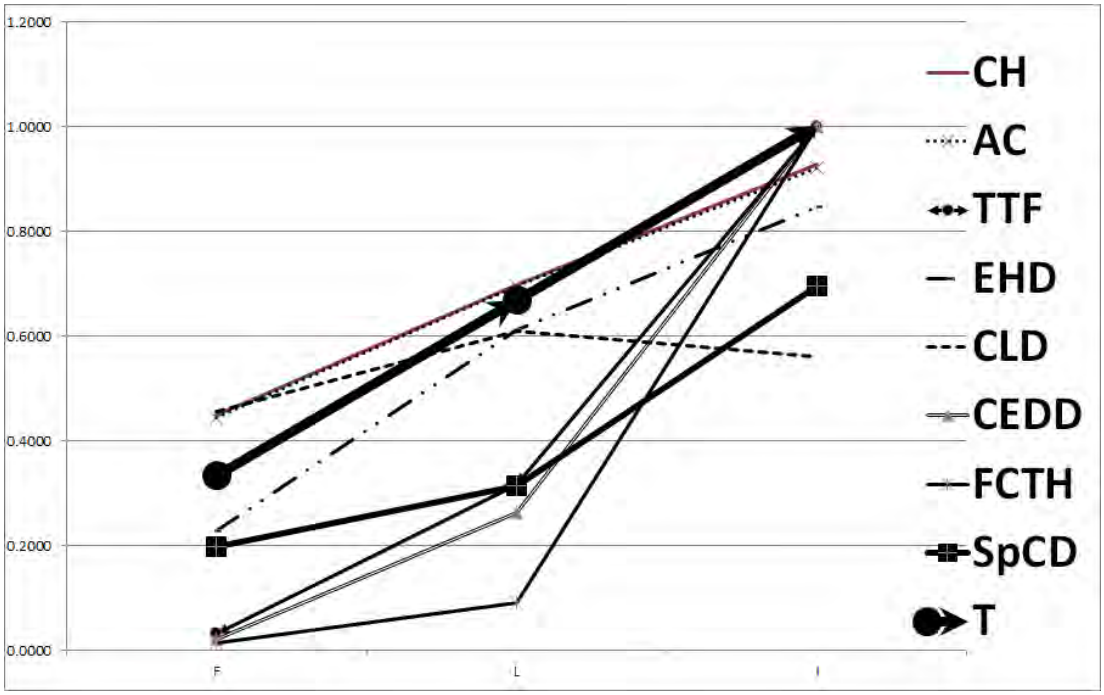


Figure 4.15: Graph for data set 2 (F, L, I)

methods. This is consistent with the researcher's observation, since these images (which are variations of the letter I) are all sub-pictures of the letter I_1.

It is interesting to note that the four similarity measures mentioned above and T in this study give very close rankings, even though they all have different normalised similarity values due to the different methodologies employed in computing similarity for this data set.

This shows that all four methods and T can produce similar results when the images are sub-pictures of the image that they are being compared to.

4.3.2.2 Results for data set 2

The results of the experiment that measures the similarity between the images F, L, I, T and H and image E (Figure 4.3 on page 86), are depicted in Table 4.4 on page 90, while the normalised values can be seen in Table 4.5 on page 90. The similarity rankings for the various methods are shown in Table 4.6 on page 90.

When comparing similarity across different grid pictures, the images F, L, and I were ranked in the same order of similarity to image E (see Figure 4.15 on

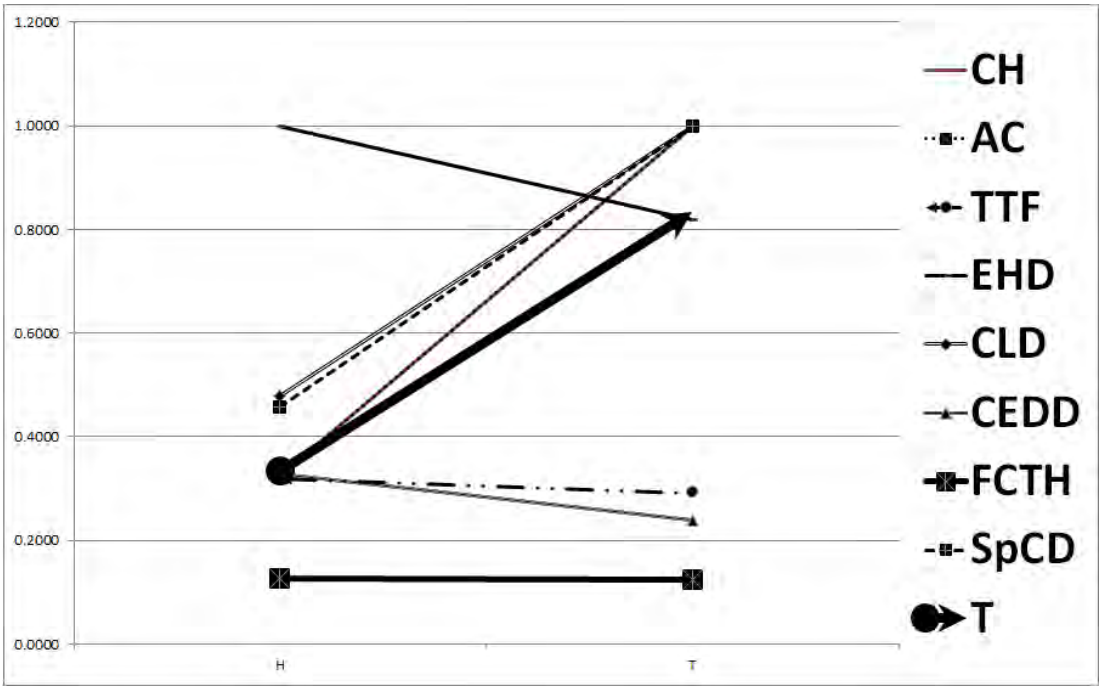


Figure 4.16: Graph for data set 2 (T and H)

page 92) by all the methods (except CLD), when isolated from the rest of the images in this set. This is consistent with the researcher's observation, since these images are all sub-pictures of the letter E.

The images T and H were ranked differently. Four of the similarity methods (CH, AC, CLD and SpCD) rank H as being more similar to E than T while the other four (TTF, EHD, CEDD and FCTH) rank T as being more similar to E than H (Figure 4.16).

Once again, CLD is the worst performer in determining that I is more similar to E than L, which makes it the worst similarity matching technique for this set of images, because of the reason given earlier (i.e., meant for colour images). FCTH also finds (E, F) and (T, H) to be very close matches (< 0.2 range), which may not be acceptable from a human point of view as these letters are clearly distinguishable.

Table 4.7: Comparison of Treediff (T) and SpCD (S)

No.	SM	I	II	III	IV	V	Vi	VII	VIII	IX
1	T	0	36	84	144	216	306	432	618	888
	S	0	0.8922	1.4015	2.0679	2.7901	3.2441	3.5298	3.3547	3.7573
2	T	0	20	42	66	92	120	150	182	216
	S	0	8.9554	6.5821	13.4284	9.5097	15.1449	16.2514	17.4556	19.7557
3	T	0	2	3	5	6	8	9	11	12
	S	0	4.5108	1.3960	4.5108	1.3960	4.5108	1.3960	4.5108	2.7656
4	T	0	26	54	84	116	150	186	224	264
	S	0	0.6443	1.0299	1.1707	1.3072	1.5223	1.7462	1.7462	1.7462
5	T	0	122	239	357	470	588	702	812	931
	S	0	3.1573	6.0691	**7.2281**	**7.2281**	7.722	7.722	7.722	7.722
6	T	0	142	284	426	568	710	852	994	1136
	S	0	5.131	8.9522	12.945	**18.6039**	**18.6039**	26.3577	26.3577	26.3577
7	T	0	8	16	24	32	40	48	56	64
	S	0	10.5063	13.0676	12.4026	12.4026	12.4026	12.4026	12.4026	12.4026
8	T	0	8	16	24	32	40	48	56	64
	S	0	6.2364	9.0634	9.8941	9.8941	9.8941	9.8941	9.8941	9.8941
9	T	0	720	1440	2160	2880	3600	4320	4866	5760
	S	0	0.7192	**0.9488**	**0.9488**	**0.9488**	1.1996	1.1996	1.1996	1.1996
10	T	0	78	54	80	7	99	153	135	155
	S	0	0.5382	**0.5821**	**0.5821**	0.0000	2.0432	2.0432	2.0432	2.0432

4.3.2.3 Results for data set 3

The results for data set 3 are given in Table 4.7 and the rankings can be seen in Table 4.8 on the next page. The results show that for row numbers 4, 5, 6, 8 and 9 in above mentioned tables, there is apparently close agreement between SpCD and T. However, on closer inspection of the results, it is observed that SpCD incorrectly classifies some of the images (bold and shaded regions of Table VII), as identical instead of being very similar. This shows that T is actually better at detecting minute changes in these generated images than SpCD. This is because the fuzzy colour quantization scheme used for SpCD is not large enough to accommodate enough visual information to distinguish between these images. SpCD is designed for speed of similarity retrieval, hence it is a very compact composite visual descriptor, rather than for extracting detailed visual information from images. For all other figures where there are differences in rankings of both methods, the ranking of T reflects the ordering of iteration steps used to derive the images by the various grammars. The orderings given by tree edit distance are consistent with the author's observations.

Table 4.8: Similarity ordering of Treediff (T) and SpCD (S)

Fig.	SM	I	II	III	IV	V	Vi	VII	VIII	IX
1	T	1	2	3	4	5	6	7	8	9
	S	1	2	3	4	5	6	8	7	9
2	T	1	2	3	4	5	6	7	8	9
	S	1	3	2	5	4	6	7	8	9
3	T	1	2	3	4	5	6	7	8	9
	S	1	**6**	**2**	**6**	**2**	**6**	**2**	**6**	**5**
4	T	1	2	3	4	5	6	7	8	9
	S	1	2	3	4	5	6	7	**7**	**7**
5	T	1	2	3	4	5	6	7	8	9
	S	1	2	3	4	4	6	**6**	**6**	**6**
6	T	1	2	3	4	5	6	7	8	9
	S	1	2	3	4	5	5	7	**7**	**7**
7	T	1	2	3	4	5	6	7	8	9
	S	1	2	**9**	**3**	**3**	**3**	**3**	**3**	**3**
8	T	1	2	3	4	5	6	7	8	9
	S	1	2	3	4	**4**	**4**	**4**	**4**	**4**
9	T	1	2	3	4	5	6	7	8	9
	S	1	2	3	**3**	**3**	6	**6**	**6**	**6**
10	T	1	4	3	5	2	6	8	7	9
	S	1	**3**	**4**	4	**1**	6	**6**	**6**	**6**

4.3.2.4 Correlation studies

Correlation coefficient is a unitless index of strength of association between two variables X and Y ($+$ = positive association, - = negative, 0 = no association). It measures the linear relationship between X and Y and has a range of $-1 \leq r \leq 1$. Pearson (Equation 4.1) (Nikolić *et al.*, 2012; Pearson, 1895) and Kendall Tau B (Equation 4.2) (Agresti, 2010; Kendall, 1938) are two commonly used correlation coefficients, which were computed using Tables 4.7 on the facing page and 4.8 respectively.
The results can be seen in Table 4.9 on the following page.

$$r = \frac{\Sigma_{x=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\Sigma_{x=1}^{n}(x_i - \bar{x})^2 \Sigma_{x=1}^{n}(y_i - \bar{y})^2}} \tag{4.1}$$

Table 4.9: Correlation studies using Pearson and Kendall Tau

Figure Series		1	2	3	4	5	6	7	8	9	10
Method	Pearson	0.83	0.97	0.34	0.91	0.82	0.98	0.33	0.76	0.87	0.82
	Kendall	0.94	0.89	0.22	0.96	0.90	0.94	0.59	0.71	0.83	0.92

$$\tau = \frac{n_c - n_d}{\sqrt{(n_0 - n_1)(n_0 - n_2)}} \tag{4.2}$$

where

- $n_0 = n(n-1)/2$
- $n_1 = \sum_i t_i(t_i - 1)/2$
- $n_2 = \sum_j u_j(u_j - 1)/2$
- n_c = Number of concordant pairs
- n_d = Number of discordant pairs
- t_i = Number of tied values in the ith group of ties for the first quantity
- u_j = Number of tied values in the jth group of ties for the second quantity

The results show high significantly positive correlation (at p=95%) between tree edit distance and SpCD figure values. The non-significant correlations are shaded in Table 4.9 (column numbers 3 and 7 data in the table).

4.4 Summary

The results show that four methods (CH, AC, CEDD and SpCD) produce ranking results that correlate with each other, when dealing with sub-pictures of images. The normalised values of the similarity ratings are different, as is to be expected. This is as a result of the different computational procedures used in each method. The variations in similarity ratings may or may not actually coincide with human differences of judgement when asked to rate these letters according to their degree of similarity.

The overall results show that SpCD is the best choice for determining similarity of grayscale images, followed closely by CEDD. CH and AC also give good or acceptable results with these kinds of images.

Both tree edit distance and SpCD can be used to detect similarity of syntactically generated images with a high degree of success or accuracy in our study. However, the tree edit distance more accurately reflects the various iterations used for generating the pictures by the grammars. The tree edit distance is therefore, more accurate in detecting similarity of generated images by tree grammars than SpCD or most other computer based similarity measures. SpCD cannot detect subtle differences between the pictures used in this study that tree edit distance and the human eye can detect. The reason for this may be attributed to the fact that sampling of the colour distribution in those areas by SpCD may not be sufficiently detailed to distinguish between these images. Hence, SpCD consistently rates these images as identical, instead of highly similar, but different. This implies that it may rank images in our visual password scheme as identical instead of similar, which may make it unsuitable for our visual password scheme.

The next chapter discusses further research done with similarity measures. This was to obtain similarity ratings from human perceptual similarity experiments, and then compare results of their normalised values with what was obtained here. The success of the tree edit distance measure led to the search for additional tree edit distance measures that could be more suitable for our study. For speed of computation considerations, approximate tree edit distance measures were also included in our next set of experiments. This allowed us to determine conclusively which of the methods came closest to the human idea of similarity. The most successful similarity measure made it possible for us to generate images that are highly similar from a human point of view, ensuring that the distractor images are similar enough to the pass-image to defeat shoulder-surfing.

Chapter 5

Perceptual Similarity and Computer-based Similarity Measures

5.1 Introduction

This chapter looks at how humans determine similarity of grammatically generated abstract images, called perceptual similarity in the literature, and how to best model it using algorithmic methods. This is an expanded version of the paper presented at SAICSIT 2014 without the literature review and with only tree edit distance measures (Okundaye *et al.*, 2014b).

There are a number of perceptual similarity theories in the psychology literature that try to model this aspect of human behaviour. The best known ones are Common element approach or alignment based models, Geometric models, Template or transformational models, Tversky's feature-contrast model and the Geon model. Perceptual similarity is important in such areas as Robotics, CBIR, Psychology and Visual password schemes where users have to distinguish between their pass-images and similar but not identical distractor images to defeat shoulder surfing, which is of primary interest in this study (Lashkari *et al.*, 2009).

Our generated abstract images were used to determine perceptual similarity. Eight CBIR and eight tree edit distance measures were used to compare the

images.

In summary, the contributions of this chapter to the literature are as follows:

- The use of an online perceptual similarity survey to ascertain the way humans determine similarity of tree based generated abstract images;
- The use of eight tree edit distance measures for pattern matching of abstract syntax trees of syntactically generated images with a view to determining the most suitable tree edit distance measure for these kinds of images;
- To the knowledge of the researcher, this is the first time the effectiveness of tree edit distance measures and CBIR similarity measures are being compared to perceptual similarity measures;
- An attempt is being made to compare eight tree edit distance measures to eight CBIR measures for tree based grammatically generated images to determine the most effective class of similarity measures;
- This is the first time, to the best of our knowledge, that tree edit distance is being proposed as a special case of the Tversky's feature-contrast model.

This chapter is structured as follows. Section 5.2 discusses perceptual similarity models, picture grammars, tree edit distances and the visual descriptors used in this study. Section 5.3 on the next page covers the experimental procedures used in this research and Section 5.4 on page 107 the results. Section 5.5 on page 115 is a correlation study of all the distance measures used in this study with perceptual similarity, while Section 5.6 on page 119 concludes the chapter and highlights further research done in the next chapter.

5.2 Background to the study

This section covers the four disciplines brought together for this study. These have been covered in depth in the literature review chapter. The aforementioned disciplines are:

- Perceptual similarity. This was used to interpret the results of the online study.

- Tree picture grammars. This was to enable us generate the images used for the experiments.
- CBIR measures and
- Tree edit distance measures were used to model perceptual similarity in the experiments conducted in order to find the one that most closely models perceptual similarity.

A general feature of approximation tree edit distance measures are near linear running times compared to their normal counterparts. The following tree edit distance measures were used in this study: general tree edit distance (ED) (Zhang & Shasha, 1989), fanout weighted (FW), pq-gram (PQ), windowed pq-gram (WPG) (Augsten *et al.*, 2005, 2010), tree embedding distance (TE) (Garofalakis & Kumar, 2005), binary branch distance (BB) (Yang *et al.*, 2005), bottom-up tree edit distance (BU) (Valiente, 2001) and the fullpath (FP) (Buttler, 2004) tree edit distance measures. PQ, WPG, TE, BB and FP are approximate tree edit distance measures while ED, FW and BU are not.

The next section discusses the series of experiments conducted, in order to determine the best similarity measure.

5.3 Experiments

The production rules for the first set of images are shown in Figure 5.1 on the next page. These production rules were entered into TREEBAG (Drewes, 2006) and eight images were generated (Figure 5.2 on the following page).

A similar procedure was followed for the letters E, F, L, I, H, T (Figure 5.3 on the next page).

Figure 5.3 on the following page presented some challenges because it is the only figure whose images were generated using different grammars. Alternative grammar design approaches were used in order to study the effect of different grammar designs on tree edit distance measures. In addition, the letters H and T were designed to have a slightly different width (H2) and to be centred in the 3×3 grid (T2) respectively. H2 and T2 were not used in our online survey as

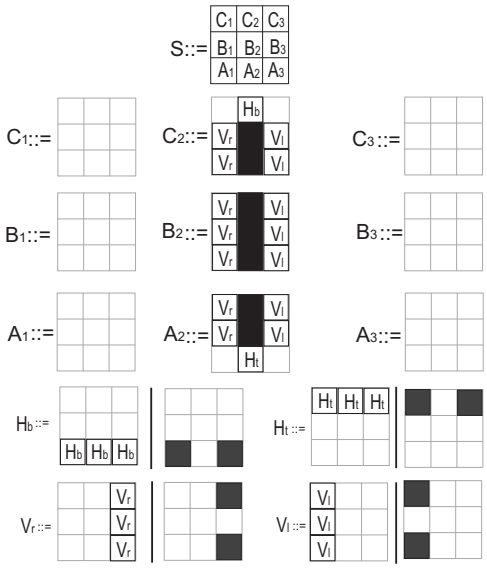


Figure 5.1: Production rules for letter I

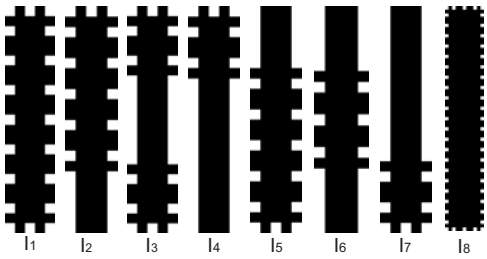


Figure 5.2: Various versions of I used as set 1 in this study



Figure 5.3: Various letters used as set 2 in this study

they were designed after the survey to determine the effect of design changes on the tree edit distance values. In addition, four other sets of images were used in both our online survey and experiments (Figures 5.4 on the next page to 5.7 on the facing page).



(a) I (b) II (c) III (d) IV (e) V (f) VI (g) VII (h) VIII (i) IX

Figure 5.4: Barnsley fern



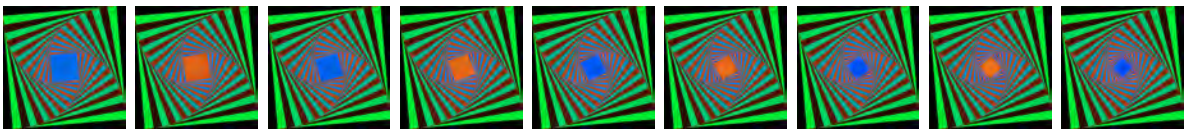
(a) I (b) II (c) III (d) IV (e) V (f) VI (g) VII (h) VIII (i) IX

Figure 5.5: Spiral



(a) I (b) II (c) III (d) IV (e) V (f) VI (g) VII (h) VIII (i) IX

Figure 5.6: Mosaic



(a) I (b) II (c) III (d) IV (e) V (f) VI (g) VII (h) VIII (i) IX

Figure 5.7: Rotating squares

5.3.1 Perceptual similarity study

An online perceptual similarity survey was conducted to determine how humans determine similarity of grammatically generated abstract images. Background information was requested on Gender, Age range, Ethnicity, Computer literacy,

Table 5.1: Background information

Gender			Age			Ethnicity			Computer Literacy			HEQ			Country		
Groups	V	%	Groups	V	%	Groups	V	%	Groups	V	%	Groups	V	%	Groups	V	%
Male	308	46.6	Under 18	17	2.6	Asian/Indian	81	12.3	Excellent	228	34.55	SE	4	0.6	Australia	2	0.30
Female	353	53.4	18-24	472	71.5	Black	292	44.2	Good	329	49.85	HS	415	62.9	Austria	1	0.15
			25-34	83	12.6	Caucasian/White	249	37.7	Fair	94	14.24	TE	241	36.5	Belgium	1	0.15
			35-54	74	11.2	Coloured	18	2.7	Poor	9	1.36				Botswana	3	0.45
			55+	14	2.1	Others	20	3.1							Kazaskhstan	1	0.15
															Nigeria	51	7.73
															Portugal	1	0.15
															Somalia	1	0.15
															South Africa	594	90.00
															Swaziland	1	0.15
															Sweden	2	0.30
															United Kingdom	1	0.15
															United States	1	0.15
Total	661	100		660	100		660	100		660	100		660	100		660	100

Table 5.2: Online similarity survey data

Figure 7: Various I's			Figure 8: E to T			Figure 9: Barnsley fern			Figure 10: Spiral			Figure 11: Mosaic			Figure 12:Rotating squares		
V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O
5225	0.00	I_1	3941	0.00	E	5816	0.00	I	5867	0.00	I	5896	0.00	I	5830	0.00	I
4307	0.18	I_2	3271	0.17	F	5233	0.10	II	5172	0.12	II	5228	0.11	II	5087	0.13	III
3747	0.28	I_5	2385	0.39	L	4569	0.21	III	4562	0.22	III	4569	0.23	III	4151	0.29	V
2887	0.45	I_3	1515	0.62	I	3808	0.35	IV	3914	0.33	IV	3872	0.34	IV	3342	0.43	II
2660	0.49	I_6	1393	0.65	T	3142	0.46	V	3262	0.44	V	3212	0.46	V	3269	0.44	VII
1991	0.62	I_4	1376	0.65	H	2545	0.56	VI	2592	0.56	VI	2570	0.56	VI	2573	0.56	IV
1507	0.71	I_7				1997	0.66	VII	2020	0.66	VII	1841	0.69	VII	2495	0.57	IX
1472	0.72	I_8				1434	0.75	VIII	1385	0.76	VIII	1390	0.76	VIII	1775	0.70	VI
						1005	0.83	IX	867	0.85	IX	1055	0.82	IX	1027	0.82	VIII

Educational qualification[1] and Country of respondent. After the preliminary series of questions mentioned above, Figures 5.2 on page 102 to 5.7 on the previous page were presented to the respondents to arrange in order of similarity to the first image in each figure set. The pictures were arranged in random order, different for each respondent. Initially, all the images were placed in the leftmost of two columns, and the respondents were required to drag the images to the right column in order of similarity to the question image. As a control measure to determine outliers, the first image in every figure set was used as both the question image and an answer image.

[1]in Table 5.1 HEQ=Highest Educational Qualification, SE=did not complete Secondary Education, HS=completed High School or equivalent and TE= completed Tertiary Education

5.3.2 Cluster analysis

A comprehensive cluster analysis of the online survey can be found in Appendix E. This was done using custom excel formulas to group similar responses together. Traditional clustering methods could not be used as all the responses had the same mean values. A table containing the largest clusters, total number of clusters, total respondents and percentage of the largest cluster for each question is shown in Table 5.3 on the following page.

In Table 5.3 on the next page, the most successful clusters are displayed in the order in which they were ranked by the respondents. There is a column for each answer option. They are in the same order as they were in the survey question. So the first column is the first answer option and the second is the second. The number in the column is the rank that was selected for that answer. Thus, 1, 6, 2, 7, 3, 8, 4, 9, 5 means that the first option was ranked first, the second option was ranked sixth, the third option was ranked second, and so on.

For all the questions, except the last (Figure number 5.8 in the table), the largest cluster for each question clarifies and tallies with the overall trend in the survey. Figure 5.7 on page 103 in the online survey produced a mixed ordering, however in Table 5.3 on the next page, the dominant ordering can be clearly seen. Thus, those who arranged the squares first using the colour of the centre square and then ordering the other images in order of the size of the centre square, from the largest to the smallest, were clearly the majority. The mixed result for Figure 5.8 in the online survey was one of the reasons for doing this cluster analysis. The other reason was to seek a clearer insight into all the group ordering by respondents, especially for Figures 5.2 on page 102 and 5.3 on page 102.

5.3.3 CBIR similarity experiments

The eight CBIR similarity measures discussed were tested on the sets of images used in the online study to see how closely they model perceptual similarity.

Table 5.3: Online survey respondents clusters

Fig. ref.	Image position in survey question									Largest cluster count	Total no. of clusters	Total respondents	Cluster percentage
	I	II	III	IV	V	VI	VII	VIII	IX				
5.3	1	2	4	6	3	5	7	8		132	174	661	19.96974281
5.4	1	2	3	4	6	5				194	44	661	29.3494705
5.5	1	2	3	4	5	6	7	8	9	287	159	648	44.29012346
5.6	1	2	3	4	5	6	7	8	9	443	99	656	67.5304878
5.7	1	2	3	4	5	6	7	8	9	358	95	655	54.65648855
5.8	1	6	2	7	3	8	4	9	5	273	116	655	41.67938931

5.3.4 Tree edit distances experiments

As all the images used for the online perceptual similarity study were generated by tree based grammars, the most popular, effective and state of the art tree edit distance measures found in the literature were also tested on the sets of images used in the online study. This was done to:

- Determine how closely tree edit distance measures can model perceptual similarity, and
- Determine whether tree edit distance was better at modelling perceptual similarity than CBIR methods.

5.4 Results

The results of the online survey, and the CBIR and tree edit distance experiments are presented in Tables 5.2 on page 104 to 5.15 on page 116. The tables are structured as follows. Table 5.2 on page 104 presents the results for the online survey (perceptual similarity), while Tables 5.4 on page 111 to 5.9 on page 112, present the results for the CBIR experiments for Figures 5.2 on page 102 to 5.7 on page 103 respectively. Tables 5.10 on page 115 to 5.15 on page 116 are for the tree edit distance experiments. Each table corresponds to the results of the experiments on a single figure series. Each table is grouped into eight sets of three columns each, corresponding to the eight distance measures used. The three columns are labelled V for values, NV for normalised values and O for ordering of the images by similarity by the distance measure involved. Sometimes, instead of V for the first column, Vx100 is used to indicate that the values obtained were multiplied by 100 to avoid loss of significant digits in the cells when displayed to two decimal places.

5.4.1 Results of perceptual similarity study

Table 5.1 on page 104 presents the results of the online survey for the 661 respondents on background information: Gender, Age range, Ethnicity, Computer literacy, Educational qualification and Country respectively. Since drag and drop

table ranking was used for this survey, the similarity measures were obtained in the form of weighted scores and are shown in Table 5.2 on page 104. The first image question, Figure 5.2 on page 102 and survey question 8, shall be used to illustrate how the values in Table 5.2 on page 104 for Figure 5.2 on page 102 were obtained. There are eight images, so the images are weighted from 1 to 8. Any image selected as the first image position (most similar) is given a weight of 8. The least similar is given a weight of 1 (i.e. the respondent placed it at the bottom of the image selections for that question). In this way, the image that is always selected first will have the highest total score. Analogously, the images selected in lower positions will have lower scores next to them. This method is called weighted ranking, with each image being given a ranking weight according to whether the respondents selected it as being more or less similar to the question image. The higher the weight of an image as determined by the respondents, the more similar it is to the question image and the higher its total weighted score.

The values obtained from the online survey are similarity measures which have to be converted to distance measures in order to be compared with both the CBIR and tree edit distance measures. This is obtained by subtracting their normalised similarity values from one. The NV column contains the converted normalised values for the various images. The O column contains the overall ranking for each image in an image series by all the respondents in the survey.

When looking at the results of the online survey, it is apparent that the last two images of Figures 5.2 on page 102 and 5.3 on page 102 have very close weighted scores, which signifies that the respondents were almost equally divided between the ordering of the two images. The results for Figures 5.4 on page 103 to 5.6 on page 103 follow from a logical ordering of the images: the number and density of the branches in Barnsley ferns (Figure 5.4 on page 103), the number and placement of the circles in Spirals (Figure 5.5 on page 103) and the size of the space at the centre of the Mosaic (Figure 5.6 on page 103).

The results for Figure 5.7 on page 103 produced the most interesting results. There are at least three possible ways to logically order the images for this series. The ordering in the figure corresponds to the order of sequential generation by the grammar and is also the expected ordering when the size of the centre square is used. Two other possible orderings can be obtained, if the colour of the centre

square is used or when both the size and colour of the centre square are used. The survey respondents used all three methods plus other miscellaneous orderings. The overall result is the effect of all these ranking methods used.

A possible explanation for the mixed results obtained in Table 5.2 on page 104 can be found in Tversky's feature contrast model. This model predicts that respondents will order images according to the most salient (prominent) features possessed by both the source and target images, while perceived feature differences will lower the similarity ranking. This also explains the difficulty in respondents' determinations of the similarity of H and T to E. The same is also true for the last two images of Figure 5.2 on page 102.

5.4.2 Results of CBIR similarity experiments for perceptual similarity images

Table 5.4 on page 111 shows the ranking for Figure 5.2 on page 102. In two pairs (I_2,I_5 and I_4,I_7), the images are rotated replicas of each other. CH, AC, CEDD and SpCD were able to detect this, although SpCD was the only one to give them identical values. Thus the aforementioned methods came closest to the online survey results.

Table 5.5 on page 111 shows the results of CBIR distance measures for Figure 5.3 on page 102. All the measures, with the exception of CLD and SpCD, which ranked I as being more similar to E than L, ranked E, F, L and I, taken in isolation, in the same order as humans. However, none of them came close to the results of the online survey.

In Table 5.6 on page 112, only CLD and SpCD came close to the results of the online survey, with CLD wrongly ranking images VI to IX as equal in similarity. The graph for this table (Table 5.6 on page 112) is in Figure 5.8 on the following page. One can see the zig-zag nature of the various measures compared to that of the online survey.

Table 5.7 on page 112 shows CH, AC, TTF, EHD and CEDD correctly modelling perceptual similarity from the survey. In Table 5.8 on page 112, the measures that came closest to perceptual similarity incorrectly determined that at least 3 of the images were equally similar i.e. same similarity values. For exam-

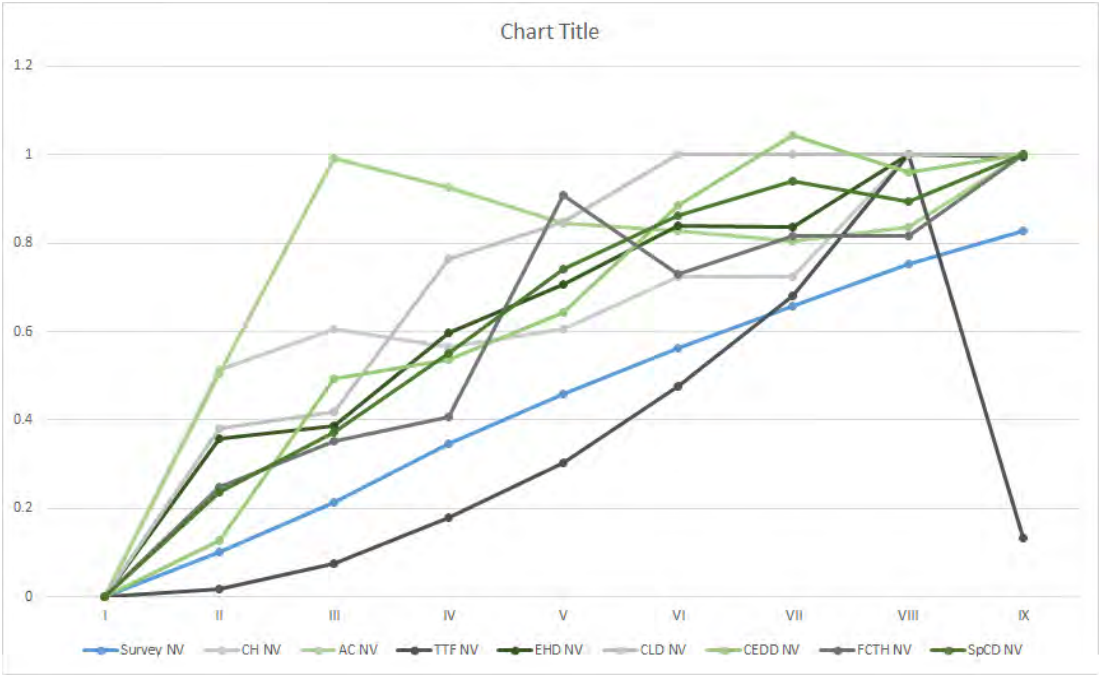


Figure 5.8: Graph of survey results versus CBIR distance measures for Table 5.5 (Barnsley fern)

ple, CH, AC, TTF, CLD and CEDD incorrectly determine as equal 3, 4, 6, 4 and 4 images, respectively, of the nine images provided.

Table 5.9 on page 112 for Figure 5.7 on page 103 produced another set of interesting results. TTF orders the images according to the size of the centre square, while AC orders the images according to both the colour and size of the centre square. However, all the results differ in varying degrees from the results of the online survey.

The varying results obtained by the various CBIR measures reflect the image features used to obtain the results, as discussed in the background to this study. For example, SpCD that emphasizes the spatial distribution of colours in an image is expected to give a different result, from say, CH that uses colour frequencies in an image.

5.4.2 Results of CBIR similarity experiments for perceptual similarity images

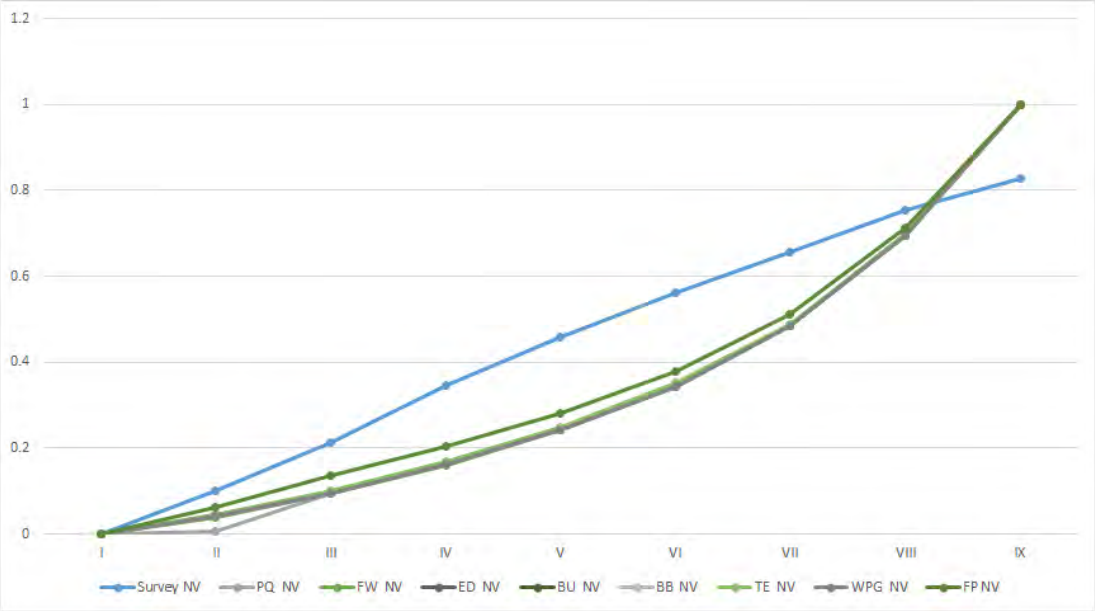


Figure 5.9: Graph of survey results and tree edit distance measures for Table 5.11 (Barnsley fern)

Table 5.4: CBIR comparison of I_1 to other I's

CH			AC			TTF			EHD			CLD			CEDD			FCTH			SpCD		
V	NV	O	V	NV	O	Vx100	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O
0.00	0.00	I_1	0.00	0.00	I_1	0.00	0.00	I_1	0.00	0.00	I_1	0.00	0.00	I_1	0.00	0.00	I_1	0.00	0.00	I_1	0.00	0.00	I_1
22.00	0.28	I_5	0.33	0.28	I_2	0.20	0.16	I_3	0.40	0.16	I_3	7.94	0.41	I_8	4.25	0.26	I_2	5.04	0.23	I_5	1.81	0.08	I_8
22.02	0.28	I_2	0.33	0.28	I_5	0.33	0.26	I_2	0.41	0.16	I_2	8.66	0.45	I_6	4.51	0.28	I_5	6.65	0.31	I_2	7.83	0.34	I_2
36.01	0.46	I_8	0.55	0.47	I_8	0.37	0.30	I_5	0.63	0.25	I_6	10.54	0.55	I_2	8.13	0.50	I_3	7.81	0.36	I_3	7.83	0.34	I_5
45.01	0.58	I_3	0.68	0.58	I_3	1.19	0.95	I_4	0.71	0.28	I_4	11.92	0.62	I_5	10.56	0.65	I_6	9.53	0.44	I_4	13.04	0.57	I_3
50.02	0.64	I_6	0.75	0.64	I_6	1.22	0.97	I_7	0.81	0.32	I_5	14.25	0.74	I_3	14.77	0.91	I_4	13.04	0.60	I_6	14.77	0.64	I_6
77.03	0.99	I_4	1.16	0.99	I_4	1.23	0.98	I_6	1.15	0.45	I_7	18.49	0.96	I_4	14.77	0.91	I_7	14.84	0.68	I_7	22.95	1.00	I_4
78.01	1.00	I_7	1.18	1.00	I_7	1.25	1.00	I_8	2.53	1.00	I_8	19.31	1.00	I_7	16.20	1.00	I_8	21.79	1.00	I_8	22.95	1.00	I_7

Table 5.5: CBIR comparison of E to other letters

CH			AC			TTF			EHD			CLD			CEDD			FCTH			SpCD		
V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O
0.00	0.00	E	0.00	0.00	E	0.00	0.00	E	0.00	0.00	E	0.00	0.00	E	0.00	0.00	E	0.00	0.00	E	0.00	0.00	E
55.00	0.32	H	0.87	0.32	H	0.04	0.03	F	0.26	0.23	F	26.17	0.46	F	1.50	0.02	F	0.94	0.02	F	11.79	0.02	F
76.00	0.45	F	1.18	0.44	F	0.35	0.29	T	0.70	0.61	L	27.50	0.48	H	16.40	0.24	T	5.73	0.09	L	27.25	0.04	H
118.15	0.70	L	1.85	0.70	L	0.37	0.31	L	0.93	0.82	T	32.28	0.56	I	18.05	0.26	L	7.71	0.12	T	41.42	0.06	I
157.01	0.93	I	2.46	0.92	I	0.38	0.32	H	0.96	0.85	I	34.97	0.61	L	22.58	0.33	H	7.90	0.13	H	59.61	0.08	T
169.27	1.00	T	2.67	1.00	T	1.19	1.00	I	1.14	1.00	H	57.48	1.00	T	68.77	1.00	I	62.37	1.00	I	718.67	1.00	L

Table 5.6: CBIR distance measure for Figure 5.4 on page 103 (Barnsley fern)

CH			AC			TTF			EHD			CLD			CEDD			FCTH			SpCD		
V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O
0.00	0.00	I	0.00	0.00	I	0.00	0.00	I	0.00	0.00	I	0.00	0.00	I	0.00	0.00	I	0.00	0.00	I	0.00	0.00	I
2.24	0.51	II	0.06	0.50	II	0.00	0.02	II	0.98	0.36	II	2.73	0.38	II	3.45	0.12	II	5.86	0.25	II	0.89	0.24	II
2.46	0.57	IV	0.10	0.80	VII	0.00	0.08	III	1.06	0.39	III	3.00	0.42	III	13.29	0.47	III	8.27	0.35	III	1.40	0.37	III
2.65	0.61	III	0.10	0.83	VI	0.00	0.13	IX	1.63	0.60	IV	5.50	0.77	IV	14.51	0.52	IV	9.55	0.41	IV	2.07	0.55	IV
2.65	0.61	V	0.10	0.84	VIII	0.00	0.18	IV	1.94	0.71	V	6.10	0.85	V	17.35	0.62	V	17.09	0.73	VI	2.79	0.74	V
3.16	0.73	VI	0.10	0.85	V	0.01	0.30	V	2.29	0.84	VII	7.18	1.00	VI	23.91	0.85	VI	19.19	0.82	VII	3.24	0.86	VI
3.16	0.73	VII	0.11	0.93	IV	0.01	0.48	VI	2.30	0.84	VI	7.18	1.00	VII	25.94	0.92	VIII	19.19	0.82	VIII	3.36	0.89	VIII
4.36	1.00	VIII	0.12	0.99	III	0.01	0.68	VII	2.73	1.00	IX	7.18	1.00	VIII	27.00	0.96	IX	21.33	0.91	V	3.53	0.94	VII
4.36	1.00	IX	0.12	1.00	IX	0.02	1.00	VIII	2.74	1.00	VIII	7.18	1.00	IX	28.16	1.00	VII	23.47	1.00	IX	3.76	1.00	IX

Table 5.7: CBIR distance measure for Figure 5.5 on page 103 (Spiral)

CH			AC			TTF			EHD			CLD			CEDD			FCTH			SpCD		
V	NV	O	V	NV	O	Vx100	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O
0.00	0.00	I	0.00	0.00	I	0.00	0.00	I	0.00	0.00	I	0.00	0.00	I	0.00	0.00	I	0.00	0.00	I	0.00	0.00	I
8.82	0.47	II	0.16	0.44	II	0.12	0.04	II	0.64	0.23	II	2.83	0.29	II	3.55	0.34	II	0.00	0.00	II	6.58	0.33	III
11.09	0.60	III	0.21	0.56	III	0.43	0.14	III	1.02	0.36	III	6.10	0.63	III	3.55	0.34	III	0.00	0.00	III	8.96	0.45	II
13.53	0.73	IV	0.25	0.68	IV	0.84	0.28	IV	1.25	0.44	IV	7.16	0.74	IV	5.29	0.50	IV	0.65	0.26	IV	9.51	0.48	V
14.73	0.79	V	0.30	0.82	V	1.30	0.43	V	2.01	0.71	V	7.40	0.76	VI	6.55	0.62	V	1.74	0.69	VII	13.43	0.68	IV
15.88	0.85	VI	0.31	0.86	VI	1.79	0.59	VI	1.95	0.69	VI	7.86	0.81	V	6.84	0.65	VI	1.94	0.77	V	15.15	0.77	VI
17.15	0.92	VII	0.33	0.91	VII	2.30	0.75	VII	2.41	0.85	VII	8.87	0.91	VII	7.06	0.67	VII	2.29	0.90	VI	16.25	0.82	VII
18.03	0.97	VIII	0.35	0.97	VIII	2.69	0.88	VIII	2.60	0.91	VIII	8.87	0.91	VIII	9.09	0.86	VIII	2.29	0.90	VIII	17.46	0.88	VIII
18.57	1.00	IX	0.36	1.00	IX	3.06	1.00	IX	2.84	1.00	IX	9.72	1.00	IX	10.53	1.00	IX	2.53	1.00	IX	19.76	1.00	IX

Table 5.8: CBIR distance measure for Figure 5.6 on page 103 (Mosaic)

CH			AC			TTF			EHD			CLD			CEDD			FCTH			SpCD		
V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O
0.00	0.00	I	0.00	0.00	I	0.00	0.00	I	0.00	0.00	I	0.00	0.00	I	0.00	0.00	I	0.00	0.00	I	0.00	0.00	I
30.27	0.62	II	0.13	0.55	II	0.02	0.56	II	2.82	0.70	II	9.90	0.59	II	13.27	0.71	II	6.46	0.32	II	10.51	0.80	II
42.73	0.87	III	0.20	0.86	III	0.03	0.88	III	3.56	0.89	III	14.78	0.88	III	18.77	1.00	III	18.59	0.93	III	12.40	0.95	IV
46.82	0.96	IX	0.22	0.96	IV	0.03	1.00	IV	3.97	0.99	V	15.29	0.91	IV	18.77	1.00	IV	18.59	0.93	IV	12.40	0.95	V
47.01	0.96	VIII	0.23	0.99	V	0.03	1.00	V	3.97	0.99	VIII	15.86	0.95	V	18.77	1.00	V	18.59	0.93	V	12.40	0.95	VI
47.09	0.96	VII	0.23	1.00	VI	0.03	1.00	VI	3.97	0.99	IX	16.77	1.00	VI	18.77	1.00	VI	18.59	0.93	VII	12.40	0.95	VII
47.49	0.97	VI	0.23	1.00	VII	0.03	1.00	VII	3.99	0.99	VI	16.77	1.00	VII	18.77	1.00	VII	18.59	0.93	VIII	12.40	0.95	VIII
48.08	0.98	V	0.23	1.00	IX	0.03	1.00	VIII	3.99	0.99	VII	16.77	1.00	VIII	18.77	1.00	VIII	18.59	0.93	IX	12.40	0.95	IX
49.01	1.00	IV	0.23	1.00	VIII	0.03	1.00	IX	4.01	1.00	IV	16.77	1.00	IX	18.77	1.00	IX	19.94	1.00	VI	13.07	1.00	III

Table 5.9: CBIR distance measure for Figure 5.7 on page 103 (Rotating squares)

CH			AC			TTF			EHD			CLD			CEDD			FCTH			SpCD		
V	NV	O	N	NV	O	Vx100	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O
0.00	0.00	I	0.00	0.00	I	0.00	0.00	I	0.00	0.00	I	0.00	0.00	I	0.00	0.00	I	0.00	0.00	I	0.00	0.00	I
54.13	0.85	IX	0.56	0.22	III	0.01	0.20	II	0.25	0.30	II	3.41	0.24	III	1.68	0.29	III	0.00	0.00	III	1.40	0.30	III
54.41	0.85	VIII	0.98	0.38	V	0.02	0.23	III	0.46	0.55	III	3.41	0.24	V	3.82	0.67	V	0.00	0.00	V	1.40	0.30	V
54.62	0.86	VII	1.28	0.50	VII	0.03	0.41	IV	0.48	0.58	IV	4.83	0.33	VII	3.82	0.67	VII	0.57	0.16	VII	1.40	0.30	VII
55.96	0.88	VI	1.50	0.58	IX	0.04	0.59	V	0.56	0.67	V	4.83	0.33	IX	3.82	0.67	IX	0.57	0.16	IX	2.77	0.60	IX
56.92	0.89	V	2.01	0.78	VIII	0.05	0.68	VI	0.72	0.86	VII	11.07	0.76	VI	5.35	0.94	IV	2.33	0.66	VI	4.51	0.97	IV
58.00	0.91	III	2.02	0.78	VI	0.06	0.88	VII	0.73	0.88	VIII	11.07	0.76	VIII	5.41	0.95	VI	2.33	0.66	VIII	4.51	0.97	VI
59.20	0.93	IV	2.15	0.84	IV	0.06	0.93	VIII	0.79	0.94	VI	12.49	0.86	IV	5.41	0.95	VIII	3.54	1.00	II	4.51	0.97	VIII
63.80	1.00	II	2.57	1.00	II	0.07	1.00	IX	0.83	1.00	IX	14.49	1.00	II	5.70	1.00	II	3.54	1.00	IV	4.65	1.00	II

5.4.3 Results of tree edit distance measurements experiment for perceptual similarity images

Table 5.10 on page 115 shows the results of tree edit distance experiments on Figure 5.2 on page 102. All the distance measures correctly ordered the images in line with the online survey results. It is also interesting to note that there was no difference between the ordering of normal tree edit distance measures compared to their corresponding approximate measures.

Table 5.11 on page 115 produced some interesting results regarding the sensitivity of tree edit distance measures to the design of the grammar. The results show markedly different results for the two grammar designs. A rule of thumb that came out of these experiments, is that when applying tree edit distance to different grammars, one must try, as much as possible to exploit what is common to both grammars. In other words, the production rules, labels or signatures and derivation steps, should as much as possible be similar in order to have a more accurate distance measure. In line with this observation, we shall adopt the second grammar design for use in comparative analysis in this study. The ordering results for grammar design II show an ordering equal to or better than grammar design I for all the images, when compared to the results obtained from the online survey. FW and ED are the most successful tree edit distance measures for this set of images when using grammar design II. Both distance measures rank images I, H and T as being equally similar to E. Since respondents also found H and T almost similar to E, these are the closest distance measures to perceptual similarity. Also of note, is that both are normal tree edit distance measures. This shows the superiority of normal tree edit distance measures compared to their approximation counterparts for some types of images.

Tables 5.12 on page 115, 5.13 on page 116 and 5.14 on page 116 show perfect correlation between all the tree edit distance measures and the online survey results. This shows the superiority of tree edit distance measures to all the CBIR methods used in this study. All the images were detected as different and given the correct similarity ordering, which is a better result than that provided by CBIR measures. A graph of survey results and tree edit distance measures for Table 5.12 on page 115 (Barnsley fern) is in Figure 5.11 on the following page

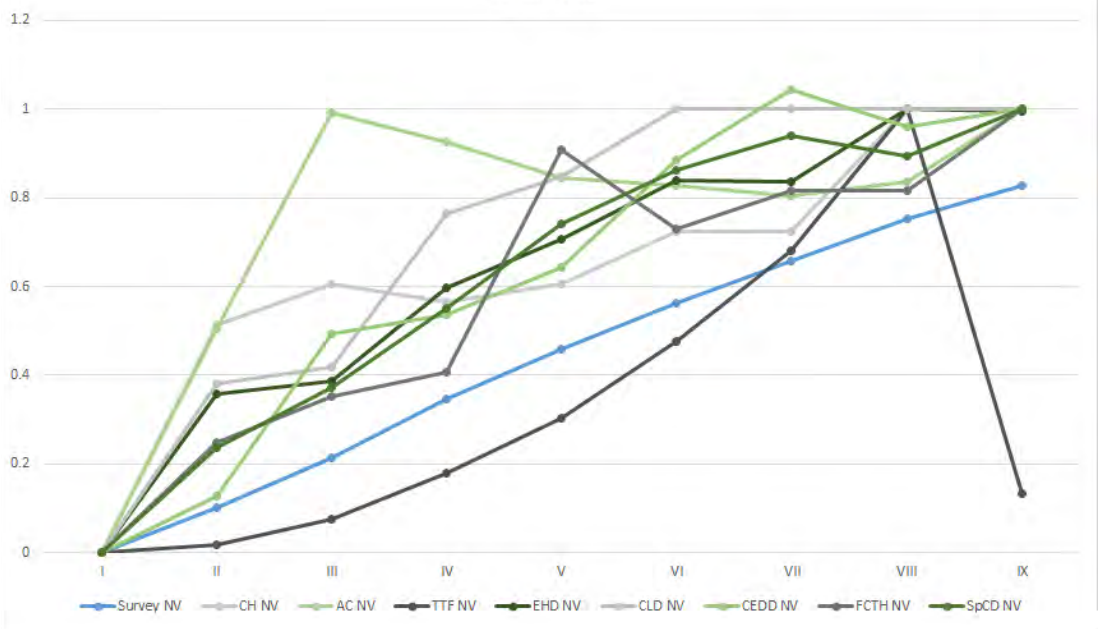


Figure 5.10: Graph of survey results and CBIR measures for Table 5.6 (Spiral)

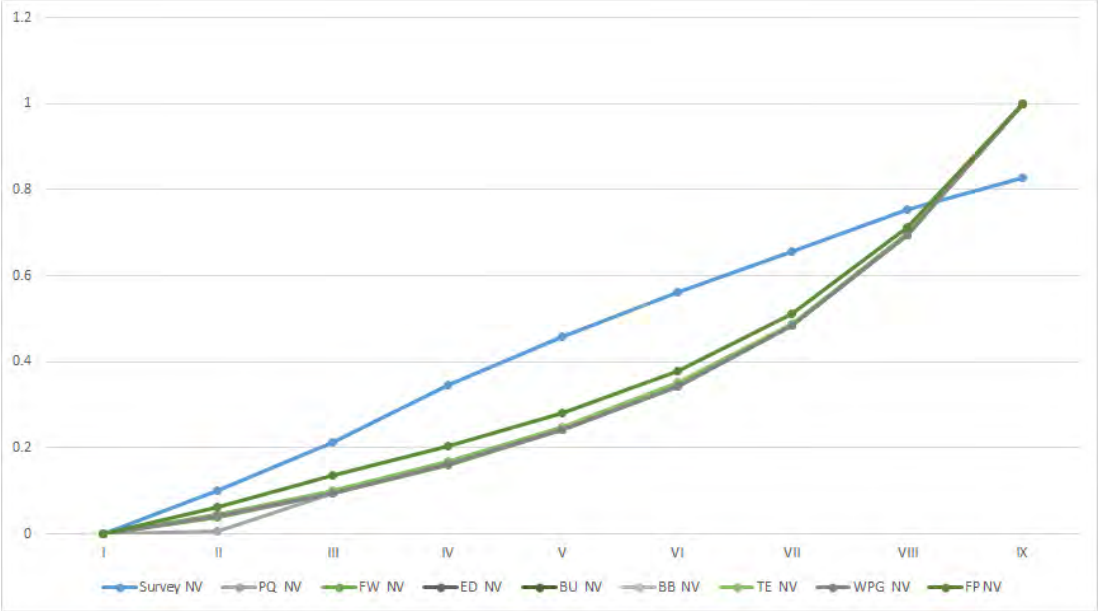


Figure 5.11: Graph of survey results and tree edit distance measures for Table 5.12 (Spiral)

Figure 5.7 on page 103 produced interesting results for tree edit distance measures as it did for CBIR measures. BU gave the same ordering as humans

Table 5.10: Tree edit distance measures (I)

PQ			FW			ED			BU			BB			TE			WPG			FP		
V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O
0.00	0.00	I_1	0.00	0.00	I_1	0.00	0.00	I_1	0.00	0.00	I_1	0.00	0.00	I_1	0.00	0.00	I_1	0.00	0.00	I_1	0.00	0.00	I_1
63.00	0.13	I_2	27.50	0.10	I_2	15.00	0.10	I_2	17.00	0.11	I_2	23.00	0.14	I_2	56.00	0.16	I_2	76.00	0.13	I_2	25.00	0.10	I_2
63.00	0.13	I_5	27.50	0.10	I_5	15.00	0.10	I_5	17.00	0.11	I_5	23.00	0.14	I_5	56.00	0.16	I_5	76.00	0.13	I_5	25.00	0.10	I_5
76.00	0.15	I_3	33.00	0.13	I_3	18.00	0.13	I_3	20.00	0.14	I_3	26.00	0.15	I_3	67.00	0.19	I_3	90.00	0.16	I_3	30.00	0.13	I_3
126.00	0.25	I_6	55.00	0.21	I_6	30.00	0.21	I_6	33.00	0.22	I_6	46.00	0.27	I_6	101.00	0.28	I_6	152.00	0.26	I_6	50.00	0.21	I_6
139.00	0.28	I_4	60.50	0.23	I_4	33.00	0.23	I_4	36.00	0.24	I_4	49.00	0.29	I_4	113.00	0.31	I_4	166.00	0.29	I_4	55.00	0.23	I_4
139.00	0.28	I_7	60.50	0.23	I_7	33.00	0.23	I_7	36.00	0.24	I_7	49.00	0.29	I_7	113.00	0.31	I_7	166.00	0.29	I_7	55.00	0.23	I_7
496.00	1.00	I_8	264.00	1.00	I_8	144.00	1.00	I_8	148.00	1.00	I_8	170.00	1.00	I_8	361.00	1.00	I_8	576.00	1.00	I_8	240.00	1.00	I_8

Table 5.11: Tree edit distance measures (E to I)

PQ				FW				ED				BU				BB				TE				WPG				FP			
Grammar type				Grammar type				Grammar type				Grammar type				Grammar type				Grammar type				Grammar type				Grammar type			
I		II		I		II		I		II		I		II		I		II		I		II		I		II		I		II	
V	O	V	O	V	O	V	O	V	O	V	O	V	O	V	O	V	O	V	O	V	O	V	O	V	O	V	O	V	O	V	O
0	E	0	E	0	E	0	E	0	E	0	E	0	E	0	E	0	E	0	E	0	E	0	E	0	E	0	E	0	E	0	E
4	F	12	F	2	F	2	F	2	F	2	F	4	F	4	F	2	F	6	F	11	H	31	F	6	H	12	F	2	F	4	F
4	H	22	L	2	H	4	L	2	H	4	L	4	H	6	H	2	H	10	L	15	F	51	L	12	F	12	H	2	H	4	H
8	L	30	H	4	L	6	I	4	L	6	I	5	T	6	T	4	L	12	T	19	L	57	T	18	L	24	L	4	L	8	L
12	I	34	I	6	I	6	H	5	T	6	H	5	T2	7	L	6	I	16	I	19	T	60	H	30	I	36	I	6	I	12	I
19	T	36	T	9	T	6	T	6	I	6	T	7	L	8	T2	7	T	16	H	26	I	72	I	32	T	42	T	7	T	12	T
24	T2	56	T2	10	T2	11	H2	8	T2	11	H2	9	H2	9	H2	12	H2	24	T2	27	H2	74	T2	42	T2	44	T2	8	T2	12	T2
39	H2	78	H2	21	H2	12	T2	10	H2	12	T2	10	I	10	I	12	T2	32	H2	28	T2	93	H2	61	H2	84	H2	14	H2	22	H2

Table 5.12: Tree edit distance measures (Barnsley fern)

PQ			FW			ED			BU			BB			TE			WPG			FP		
V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O
0	0.00	I	0	0.00	I	0	0.00	I	0	0.00	I	0	0.00	I	0	0.00	I	0	0.00	I	0	0.00	I
16	0.01	II	70	0.04	II	36	0.04	II	40	0.04	II	36	0.04	II	110	0.04	II	156	0.04	II	58	0.06	II
224	0.09	III	165	0.09	III	84	0.09	III	88	0.10	III	84	0.09	III	245	0.10	III	364	0.09	III	128	0.14	III
384	0.16	IV	284	0.16	IV	144	0.16	IV	148	0.17	IV	144	0.16	IV	413	0.17	IV	624	0.16	IV	192	0.21	IV
576	0.24	V	427	0.24	V	216	0.24	V	220	0.25	V	216	0.24	V	616	0.25	V	936	0.24	V	264	0.28	V
816	0.34	VI	606	0.34	VI	306	0.34	VI	310	0.35	VI	306	0.34	VI	871	0.35	VI	1326	0.34	VI	354	0.38	VI
1152	0.49	VII	855	0.48	VII	432	0.49	VII	433	0.49	VII	432	0.49	VII	1204	0.49	VII	1872	0.49	VII	480	0.51	VII
1648	0.70	VIII	1223	0.69	VIII	618	0.70	VIII	622	0.70	VIII	618	0.70	VIII	1723	0.70	VIII	2678	0.70	VIII	666	0.71	VIII
2368	1.00	IX	1763	1.00	IX	888	1.00	IX	892	1.00	IX	888	1.00	IX	2467	1.00	IX	3848	1.00	IX	936	1.00	IX

when the colour of the centre square is used to order the images. FP gives the similarity ordering according to the size of the centre square. None of them was able to provide the mixed method ordering of the online survey.

5.5 Correlation studies

This section discusses the relationship between perceptual similarity and the various similarity measures used in this study. It also tries to explain the results of

Table 5.13: Tree edit distance measures (Spiral)

PQ			FW			ED			BU			BB			TE			WPG			FP		
V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O
0	0.00	I	0	0.00	I	0	0.00	I	0	0.00	I	0	0.00	I	0	0.00	I	0	0.00	I	0	0.00	I
59	0.09	II	39	0.09	II	20	0.09	II	22	0.10	II	20	0.09	II	52	0.11	II	115	0.09	II	38	0.16	II
124	0.19	III	83	0.19	III	42	0.19	III	44	0.20	III	42	0.19	III	103	0.22	III	242	0.19	III	60	0.26	III
195	0.30	IV	131	0.30	IV	66	0.31	IV	68	0.31	IV	66	0.31	IV	156	0.33	IV	381	0.30	IV	84	0.36	IV
272	0.43	V	183	0.42	V	92	0.43	V	94	0.43	V	92	0.43	V	212	0.44	V	532	0.42	V	110	0.47	V
355	0.55	VI	239	0.55	VI	120	0.56	VI	122	0.56	VI	120	0.56	VI	273	0.57	VI	695	0.55	VI	138	0.59	VI
444	0.69	VII	299	0.69	VII	150	0.69	VII	152	0.70	VII	150	0.69	VII	336	0.70	VII	870	0.69	VII	168	0.72	VII
539	0.84	VIII	363	0.84	VIII	182	0.84	VIII	184	0.84	VIII	182	0.84	VIII	406	0.85	VIII	1057	0.84	VIII	200	0.85	VIII
640	1.00	IX	431	1.00	IX	216	1.00	IX	218	1.00	IX	216	1.00	IX	477	1.00	IX	1256	1.00	IX	234	1.00	IX

Table 5.14: Tree edit distance measures (Mosaic)

PQ			FW			ED			BU			BB			TE			WPG			FP		
V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O
0	0.00	I	0	0.00	I	0	0.00	I	0	0.00	I	0	0.00	I	0	0.00	I	0	0.00	I	0	0.00	I
34	0.18	II	16	0.13	II	8	0.13	II	9	0.14	II	12	0.18	II	20	0.16	II	54	0.17	II	12	0.18	II
56	0.30	III	32	0.25	III	16	0.25	III	17	0.26	III	20	0.29	III	37	0.29	III	92	0.29	III	20	0.29	III
78	0.41	IV	48	0.38	IV	24	0.38	IV	25	0.38	IV	28	0.41	IV	49	0.39	IV	130	0.41	IV	28	0.41	IV
100	0.53	V	64	0.50	V	32	0.50	V	33	0.51	V	36	0.53	V	63	0.50	V	168	0.53	V	36	0.53	V
122	0.65	VI	80	0.63	VI	40	0.63	VI	41	0.63	VI	44	0.65	VI	82	0.65	VI	206	0.64	VI	44	0.65	VI
144	0.77	VII	96	0.75	VII	48	0.75	VII	49	0.75	VII	52	0.76	VII	96	0.76	VII	244	0.76	VII	52	0.76	VII
166	0.88	VIII	122	0.95	VIII	56	0.88	VIII	57	0.88	VIII	60	0.88	VIII	110	0.87	VIII	282	0.88	VIII	60	0.88	VIII
188	1.00	IX	128	1.00	IX	64	1.00	IX	65	1.00	IX	68	1.00	IX	126	1.00	IX	320	1.00	IX	68	1.00	IX

Table 5.15: Tree edit distance measures (Rotating squares)

PQ			FW			ED			BU			BB			TE			WPG			FP		
V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O	V	NV	O
0	0.00	I	0	0.00	I	0	0.00	I	0	0.00	I	0	0.00	I	0	0.00	I	0	0.00	I	0	0.00	I
9	0.24	III	6	0.25	III	2	0.17	II	3	0.08	III	3	0.21	III	12	0.40	III	16	0.22	II	3	0.21	II
11	0.29	II	8.5	0.35	IV	3	0.25	III	6	0.15	V	5	0.36	II	15	0.50	II	18	0.25	III	5	0.36	III
18	0.47	V	9.5	0.40	II	5	0.42	IV	9	0.23	VII	6	0.43	V	15	0.50	V	34	0.47	IV	6	0.43	IV
20	0.53	IV	12	0.50	V	6	0.50	V	12	0.30	IX	8	0.57	IV	19	0.63	VII	36	0.50	V	8	0.57	V
27	0.71	VII	14.5	0.60	VI	8	0.67	VI	31	0.78	II	9	0.64	VII	21	0.70	IV	52	0.72	VI	9	0.64	VI
29	0.76	VI	18	0.75	VII	9	0.75	VII	34	0.85	IV	11	0.79	VI	25	0.83	VI	54	0.75	VII	11	0.79	VII
36	0.95	IX	20.5	0.85	VIII	11	0.92	VIII	37	0.93	VI	12	0.86	IX	27	0.90	IX	70	0.97	VIII	12	0.86	VIII
38	1.00	VIII	24	1.00	IX	12	1.00	IX	40	1.00	VIII	14	1.00	VIII	30	1.00	VIII	72	1.00	IX	14	1.00	IX

the perceptual similarity study, by pointing out similarities between the formulas of Tversky's feature-contrast model and that of the tree edit distance measure.

Table 5.16: Correlation of perceptual similarity, tree edit distances and CBIR similarity measures

	Figure Series	Correlation method	PQ	FW	ED	BU	BB	TE	WPG	FP	CH	AC	TTF	EHD	CLD	CEDD	FCTH	SpCD
Online survey	Fern (Fig. 9)	Pearson	0.930	0.925	0.927	0.930	0.927	0.930	0.927	0.941	0.894	0.664	0.689	0.972	0.932	0.964	0.934	0.974
		Kendall	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.899	0.278	0.722	0.972	0.913	0.889	0.817	0.944
	Spiral (Fig. 10)	Pearson	0.994	0.993	0.994	0.995	0.994	0.996	0.993	0.997	0.913	0.927	0.993	0.984	0.905	0.969	0.958	0.927
		Kendall	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.944	0.930	0.986	0.943	0.889
	Mosaic(Fig. 11)	Pearson	0.997	0.995	0.997	0.997	0.997	0.996	0.997	0.997	0.721*	0.784*	0.766*	0.721*	0.787*	0.687*	0.751*	0.617**
		Kendall	1.000	1.000	1.000	1.000	1.000	1.000	1.000	1.000	0.229**	0.913	0.764	0.487**	0.913	0.645*	0.546**	0.327**
	Squares (Fig. 12)	Pearson	0.885	0.764*	0.775*	0.846	0.946	0.948	0.820	0.715*	0.571**	0.810	0.742*	0.747*	0.716*	0.864	0.625**	0.832
		Kendall	0.833	0.611*	0.611*	0.833	0.833	0.930	0.611*	0.611*	-0.141**	0.535*	0.611*	0.611*	0.493**	0.589*	0.487**	0.548**

5.5.1 Correlation of perceptual similarity, tree edit distances and CBIR similarity measures

Both correlation methods give very similar correlation values, which shows that both are equally accurate coefficients for analysing the research data. However, Kendall's Tau B is able to show the perfect correlation score of 1 for the distance measures that actually had the same image ordering as the online survey. It is a non-parametric statistic that is less susceptible to outliers and unit differences. The values without asterisks are significant at 0.01 level, while those with single asterisks are significant at 0.05 level. The values with double asterisks are not significantly correlated with the results of the online survey.

Generally, the tree edit distance measures performed better, as a group, than the CBIR measures. All the tree edit distance measures had three perfect correlation scores (Kendall's Tau B) out of the four Figures compared. Only CH, AC and TTF had perfect correlation scores for Figure 5.5 on page 103 amongst the CBIR measures. BU has a perfect correlation with the largest cluster in Figure 5.8. Thus BU is the best distance measure in this study. TE appears to be the second best correlation with the online survey in this study. It is closely followed by BB and PQ. The correlation ordering for the others are WPG, ED, FW, FP, CEDD, CLD, EHD, TTF, FCTH, SpCD, AC and CH.

5.5.2 Tversky's feature-contrast model and tree edit distance

The relationship between Tversky's feature-contrast model and tree edit distance is depicted in Figure 5.12 on the next page. In the diagram, the following map-

pings can be observed:

- θ corresponds to w_s is the cost of the substitution operation in tree edit distance,
- α corresponds to w_d is the cost of the deletion operation in tree edit distance,
- β corresponds to w_i if the cost of the insert operation in tree edit distance, and
- f corresponds to n the number of nodes

In Tversky's model, $(A - B)$ represents the features of A not in B. In tree edit distance $(A - B)$ becomes deletion of nodes. Similarly $(B - A)$ becomes node insertion, while the common features of A and B, $(A \cap B)$, are represented by node substitution. Thus, tree edit distance is a special case of Tversky's feature-contrast model, where the number of features being compared are reduced to one: node edit operations. This may account for the high success rate when tree edit distance is used as a similarity measure in highly diverse application areas. The "+" operator in Tversky's model becomes a "-" operator in tree edit distance because the former is a similarity measure, while the later is a distance measure.

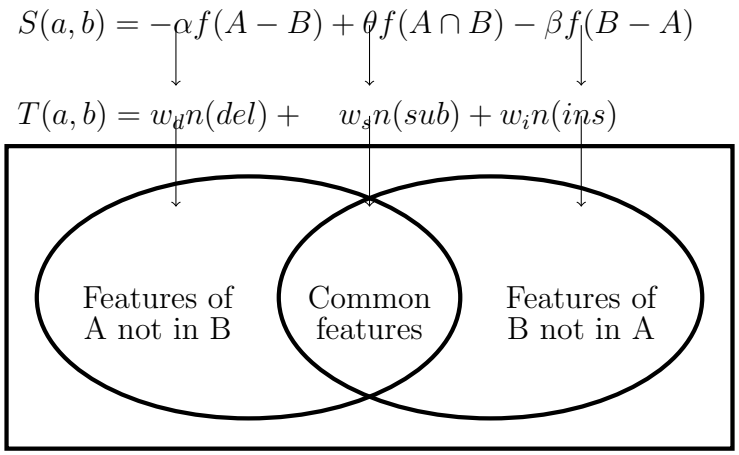


Figure 5.12: Tree edit distance as a special case of Tversky's feature-contrast model

A major issue raised by Tversky's model is asymmetry. This is an observed phenomena in similarity experiments, where comparing image or stimulus A to image or stimulus B gives a different similarity measure from when comparing B to A. Tversky suggested that when an observer focused on a particular image, the features of that image are weighted more heavily than the features of the other image. This is also simulated in tree edit distance by some asymmetric tree edit distance measures and weighting of the costs of edit operations.

5.6 Summary

The online perceptual similarity survey showed that humans can significantly agree when judging the similarity of grammatically generated images when a predominant feature can be singled out for use in determining similarity of groups of images. When more than one feature can be used, all possible combinations of features will most likely be used. The overall result will be a mixed ordering that may not be logical or intuitive. However, cluster analysis is able to identify the predominant ordering used by respondents. This was clearly the case with similarity judgements for our rotating squares series of images. A similar result can be seen when some of the features are in common and others are not. This results in different weights being assigned to both the common and distinct features, resulting in different similarity ordering for different individuals.

Tversky's feature-contrast model can be used to explain these behaviours. This model postulates that humans use not only similar features, but also distinct features when determining similarity of images. This may account for the differences in similarity of respondents noted in the online survey conducted as a part of this study. This model also accounts for the superior performance of tree edit distance measures over CBIR similarity measures used in this study. This study also put forward the notion that tree edit distance is a special case of Tversky's feature contrast model, with a single feature space. CBIR measures use only similarity of features of graphic images, like colour and texture, in determining the similarity of images. The distinct features are ignored. Tree edit distance being a special case of the feature-contrast model, uses both similar and distinct features of tree representation of images. The tree edit distance substitution op-

eration corresponds to common features of images A and B, deletions for features in A not in B and insertions for features of B not in A in the feature-contrast model. This explains the higher correlation with the online perceptual similarity survey results of all the tree edit distance measures when compared to all the CBIR methods.

This study also found that there is no significant difference between normal tree edit distance measures and their approximate counterpart when it came to accuracy and precision of determining similarity of grammatically generated images used in this study.

The results from this study show that bottom-up tree edit distance is the closest to perceptual similarity. Therefore, the normalised bottom-up tree edit distance was used in our prototype visual password scheme, which is the subject of the next chapter. The next chapter therefore covers further work done in this research, which involved using the results obtained from this study to implement a system that uses perceptual similarity judgements of images, a tree grammar-based visual password scheme.

Chapter 6

Design and Implementation of a Novel Visual Password Scheme

6.1 Introduction

This is an expanded version of the paper presented at PRASA 2014 without the literature review (Okundaye *et al.*, 2014a).

Psychological theories have long proposed that forgetting is due to decay as a result of the passage of time and to interference between new and old items in memory (Wixted, 2004, 2005). There are two types of memory interference. Retroactive interference in which new items in memory disrupt existing items in memory and proactive interference in which new items in memory are disrupted by existing ones. Retroactive interference has been advanced as the more relevant to everyday forgetting, like the inability to recall passwords (Wixted, 2004).

It has been shown that recognition based authentication systems, like the Passfaces type of visual password schemes, are easier to use or remember (Elftmann, 2006; Madigan, 1983). Alphanumeric based schemes are based on recollection and as such can be difficult to use or remember (Sobrado & Birget, 2002).

This part of the thesis aims at using formal grammars, specifically tree based grammars, in the generation of similar graphic images for visual password authentication. In doing this, the idea of similarity of graphic images will be examined,

along with attributes possessed by images that can be said to be mathematically similar.

In summary, the contributions of this chapter of my thesis to the literature are as follows:

- To the knowledge of the researcher, this is the first time dynamically generated images are being used in the design of a prototype visual password scheme.
- To the knowledge of the researcher, this is the first time a grammatical picture generation approach, or more specifically, a tree based grammatical picture generation approach, is being used in a prototype visual password scheme.
- To the knowledge of the researcher, this is the first time the tree edit distance measure is being proposed as a selection mechanism for obtaining the best set of distractor images in a visual password scheme.

This chapter is structured as follows. Section 6.2 provides a highlight of tree grammars and tree edit distance measures used in this study. Section 6.3 on the next page covers both the experimental procedures used in this research and the discussion of the design and implementation of a prototype of the visual password scheme. Section 6.4 on page 134 concludes the chapter and highlights areas for further research.

6.2 Background to this study

This section discusses the two research areas brought together to design the prototype used in this study, which are grammatical picture generation and the tree edit distance measure. This has previously been comprehensively discussed in the literature review chapter.

The grammars selected for this study were taken from as many different categories of tree picture grammars as possible. There are therefore grammars from regular tree picture grammars, grid picture grammars, ET0L picture grammars and branching synchronisation grammars. The images are generated, one after

the other, with increasing number of terminals for all the grammars used in this study. It was observed that, for some grammars, the initial images generated were not very similar, when compared to those generated later on, after successive applications of the production rules. This initial set of images was left out for some of the grammars because these images were not considered good enough for use in a visual password scheme. These grammars were also observed to generate increasingly more similar images, after successive iterations, until the images became indistinguishable to the human eye. This point where the images become indistinguishable is referred to in this study as the similarity convergence point and the region before it of progressively similar images as the similarity convergence region. This led to the decision to select eight images immediately before and immediately after the pass-image in the design of the visual password scheme, because it was considered to be the region that had the highest probability of providing us with the most similar images.

Another set of grammars was also observed to generate uniformly similar images throughout the image generation process. However, a major disadvantage of this set of grammars was the tendency to repeat some images in their generation sequence. This meant that steps must be taken to prevent appearance of these repeating images when designing or using these grammars in a visual password scheme. Also, with this type of grammars, the design decision of picking the most similar eight images to the pass-image out of the 16 generated images (eight images generated before and after the pass-image), was found to be suitable for these grammars.

6.3 Design and implementation

This section discusses the importance of the online survey in this study and the design and implementation of the prototype visual password system.

6.3.1 Determination of similarity

In order to create a Passfaces type of visual password system, we need to grammatically generate similar pictures. The most important research questions an-

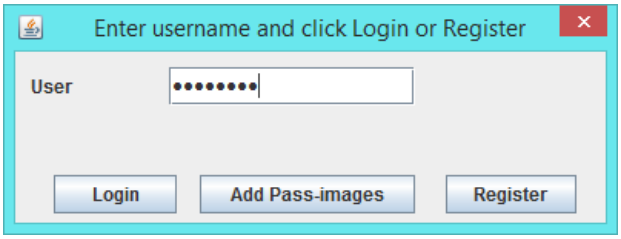


Figure 6.1: Login / Add Pass-images / Registration screen for prototype

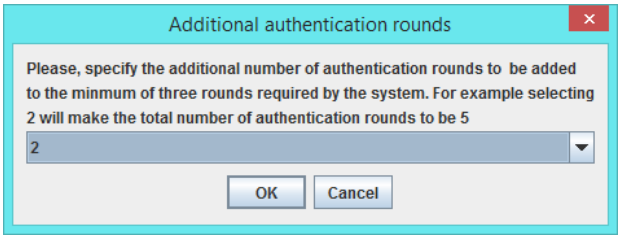


Figure 6.2: Addition of authentication rounds dialog box

swered at this stage were:

1. how to determine that the generated images are universally regarded as similar? and

2. how to model this human idea of similarity using a computer system?

To answer the first question, an online survey was conducted with 661 respondents. The survey showed that the human idea of similarity is consistent enough to be modelled. This was the subject of the last chapter

The second question was addressed by experimentally determining which of a broad range of state of the art similarity measures was most suitable for this class of images (Okundaye *et al.*, 2013). This is because a perceptual similarity survey can only be used on a small subset of the images that can be generated grammatically. The similarities of all the other images that can be generated have to be determined automatically. Also, the prototype visual password scheme should be able to dynamically determine the similarity of generated images. Bottom-up tree edit distance was shown, in the last chapter, to be the most appropriate measure for determining similarity of this class of images. Thus, our prototype uses Bottom-up tree edit distance in determining similarity of our generated images (Okundaye *et al.*, 2014b).

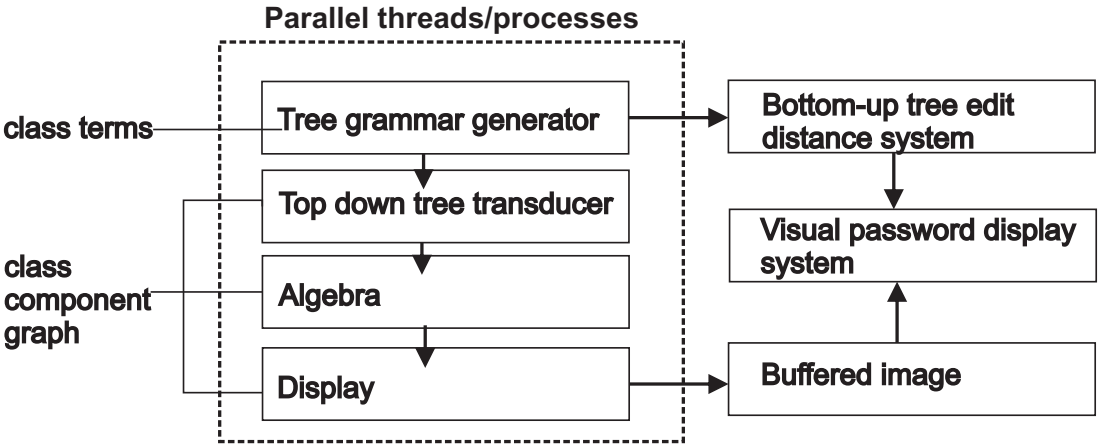


Figure 6.3: Parallel threads/processes

6.3.2 Grammar design

Twenty four tree picture grammars were used in this prototype. Regular tree picture grammars, ET0L tree picture grammars, grid picture grammars and branching tree grammars and Links grammar were used in the prototype. They are all implemented as defined in Drewes (2006, 2007) and Treebag. ET0L tree grammars are a generalisation of Regular tree grammars and Branching tree grammars are a generalisation of ET0L grammars with fully parallel derivations. The Links formalism is a generalisation of Chain-code, Turtle geometry and Collage formalisms Drewes (2007). For example, Regular tree grammars are not able to generate balanced trees (i.e. where all leaves are at the same distance from the root node). Similarly, it is difficult for an ET0L grammar to generate balanced trees that are symmetric about the vertical line through the root, but this can easily be achieved with a branching tree grammar.

The tree picture grammar generator module translates these grammars, with the exception of the regular tree grammar which is is not translated, into one or more top down tree transducers (Figure 6.3). It has been shown that ET0L and branching tree grammars can be converted into an equivalent regular tree grammar and one or more tree transducers (Drewes, 2006; Drewes & Engelfriet, 2004).

6.3.3 Design and implementation of the system

The system is made up of several subsystems: the tree picture grammar module, the tree edit distance module and the visual password module. The picture grammar generation module reads the grammar and algebra input files from the local file system, and based on the type of grammar, decides whether translation is necessary or not. If the grammar is a regular tree grammar, no translation is necessary. An ET0L grammar is translated into a regular tree grammar and a top-down tree transducer while a branching synchronisation grammar is translated into a regular tree grammar and one or more tree transducers. This is then interpreted by the algebra into an image consisting of graphic operations that are then passed to the relevant display. A collage display is used for all the tree grammars except for the grid picture grammars that use the grid collage display.

The image generated by the appropriate display is stored in a buffered image array, while the string representation of the tree is stored in a string array. After obtaining the 16 images and the string representation of their tree forms, the relevant tree edit distance measure (bottom-up tree edit distance in this instance), is used to find the tree edit distance between the pass-image and the 16 images generated using their string representation. The number of images used was to strike a balance between having enough images to pick eight highly similar distractors from and the time needed to generate them. The images and their computed tree edit distance measures are then sorted in order of similarity to the pass-image and a cut-off point established, which is the tree edit distance value of the ninth image in the array of sorted values. This cut-off value is referred to as the similarity threshold value for the pass-images in this study. The nine images, consisting of the pass-image and the most similar eight of the 16 images, are then displayed to the user.

The Unified Modelling Language (UML) class diagrams for the visual password systems are presented in Figures 6.4 on page 130 to 6.6 on page 132. The overview of the visual password system is depicted in Figure 6.4 on page 130. It consists of seven sub-systems in total. The **componentGraph** sub-system class and the **terms** sub-system are responsible for generating the images and their tree forms respectively and are part of the picture grammar generating module.

The tree edit distance module named **matchtrees** can be seen in Figure 6.6 on page 132 and consists of the eight tree edit distance measures used in this study and their normalised forms.

The design of the prototype visual password scheme is shown in Algorithms 1, 2 and 3. It basically consists of a pass-image gallery (Figure 6.7a on page 136) that has a snapshot of one image per grammar for the user to select from. The user picks a particular type of image to use for authentication. The user is then presented with images generated from the selected grammar for selection of a pass-image.

The user uses the mouse to select a given pass-image and a number representing the number of steps used to generate the image is stored. This procedure is repeated five more times to complete the registration process, with the user having six pass-images. During this phase the user indicates the number of authentication rounds desired.

The "Add Pass-images" module (Algorithm 2 on page 134) was added to allow users add three pass-images at a time to their set of pass-images. The user can also change the number of authentication rounds after each run of this module. The user is expected to run the "Add Pass-images" twice to bring the total number of pass-images to 12 for a minimal security system. The higher the number of pass-images the higher the number of alternate authentication screens available to defeat shoulder surfing. This is more comprehensively discussed in the research discussion chapter. For medium to high security environments, a total of 24 pass-images are recommended. The number of authentication rounds recommended for high security environments is six, in order to increase the password space and to further enhance security against shoulder surfing and guessing attacks. At the end of both the registration and "Add Pass-images" phases, the user is expected to indicate the number of additional rounds of authentication to be added to the minimum of three required by the system. For example, selecting two in Figure 6.2 on page 124 means that a total of of five rounds, three required by the system plus two selected, of authentication will be used for the user.

The login screen (Figure 6.1 on page 124) contains the option for registration of a new user or login of an existing user. When the user types in a username, the system checks if the username is in the list of registered users. If the username is

not among the list of registered users, the user is taken to the user registration screen. However, if the username is found in the list of registered users, the system picks the user's first image token, generates 16 images around the image token and selects the most similar eight images to use as distractors. In the next step, the pass-image and the eight distractor images are then presented to the user. The user then selects his pass-image. At this point two processing paths are possible. The first processing path is when the user selects the right pass-image by using the keyboard, the authentication process is repeated two more times to complete a successful authentication. If however, the wrong pass-image is selected, random images from subsequent grammars without the user's pass-image are generated and displayed for the rest of the authentication process and access to the resource is denied.

There are two reasons for this design decision. The first is to alert the legitimate user that he has made the wrong choice of a pass-image and that he will be required to repeat the authentication process, while the second reason is to leave the illegitimate user uninformed as to whether he has made the right or wrong pass-image choice. This prevents the illegitimate user from knowing how many pass-images he has been able to guess correctly.

Another design decision made, was to use the same eight distractor images with a given pass-image. The use of different distractor images may allow a shoulder surfer to guess the right pass-image by the simple process of elimination i.e. by observing or looking out for an image that is always appearing on the screen, since the pass-image must always be on the screen during normal authentication (intersection attack). Since the images are supposed to be similar and are randomly placed on the screen during each authentication session, it will be difficult for the shoulder surfer to guess the right pass-image. Also, the choice of keyboard for authentication is also based on the principle of dividing the attention of the shoulder surfer between the screen and the keyboard in order to reduce the chances of success. Using the mouse to select the pass-image may allow the shoulder surfer to memorise the pass-image by noting the position of the mouse pointer when the mouse was clicked. Using one hand to select the pass-image row position (Control, Alt and Shift keys for rows one, two and three respectively) and the other hand to select the pass-image column position (keys

a or 1, b or 2 and c or 3 for columns one, two and three respectively) simultaneously, will divide the attention of the shoulder surfer between the keyboard and the screen and will further reduce his chances of success in being able to identify a pass-image. This process of selecting a pass-image is repeated two more times to complete the authentication process.

The user interface aspect of the prototype of the visual password scheme has been completed. A simplified pseudocode of the registration process is shown in Algorithm 1 on page 133. Algorithms 2 on page 134 and 3 on page 135 are the pseudocode for the addition of pass-images and the user authentication process respectively.

6.3.4 Scalability and timing

The system consists of components running in parallel as threads or processes depending on the underlying processor, and is therefore highly responsive and fast (Figure 6.3 on page 125). The test-bed system is a HP pavilion g series 64 bit Intel pentium P600 2.13 dual core processor laptop, running Windows 8.1 with 4 gigabytes of RAM. The system does not have a dedicated graphics card, and as such its performance will be lower than that of a computer equipped with one. The system is equipped with an embedded Intel graphic chipset that uses about 0.2 Gb of the system RAM. The tree edit distance system has $O(n_1 + n_2)$ time, where n_1 and n_2 are the number of nodes in tree one and two respectively, and the buffered image memory takes an average of about 70ms to be completely refreshed. It takes an average of between two to five seconds to display all nine images on the screen for all the grammars. Overall, the authentication system

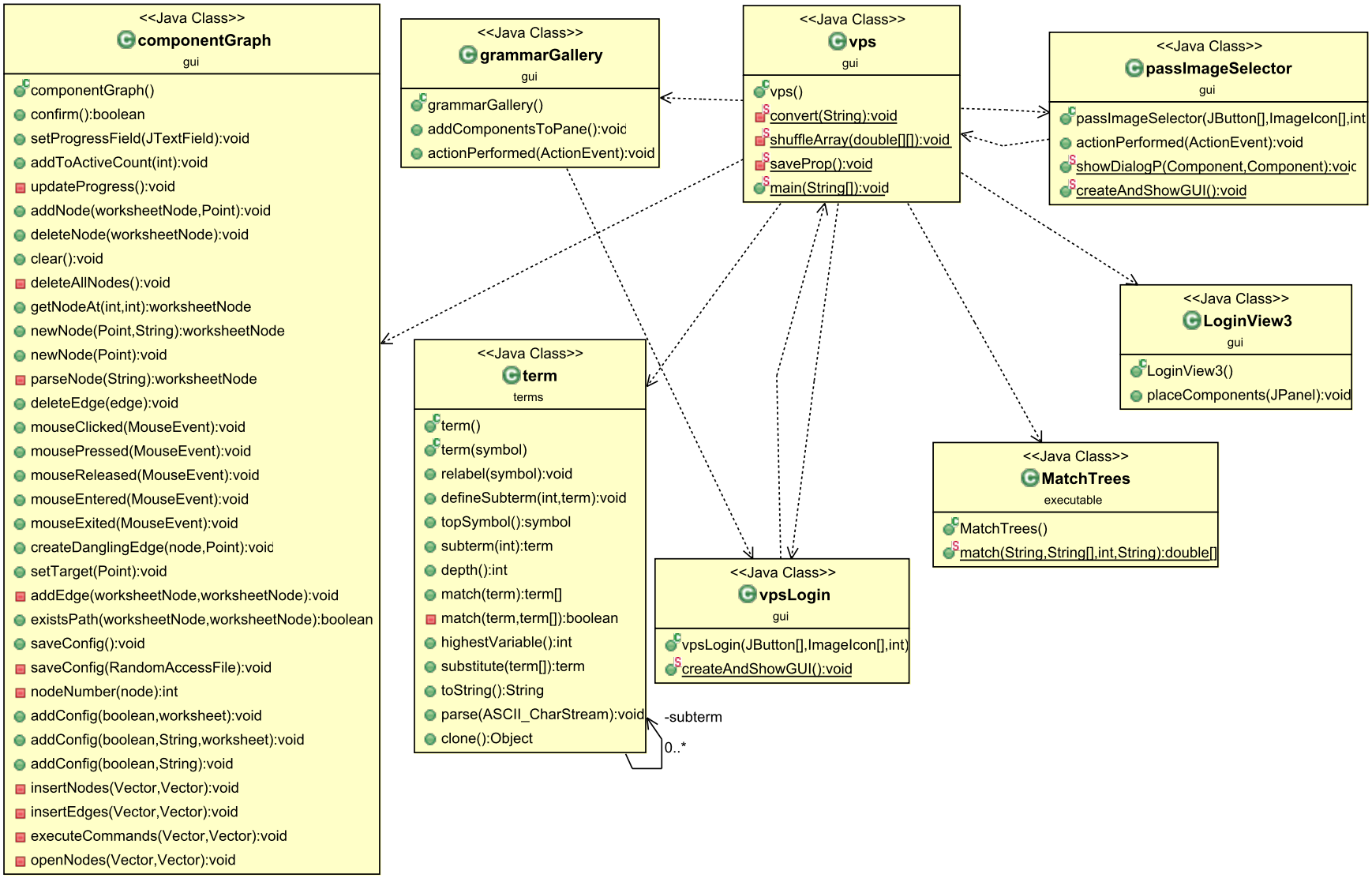


Figure 6.4: UML class diagram for class vps

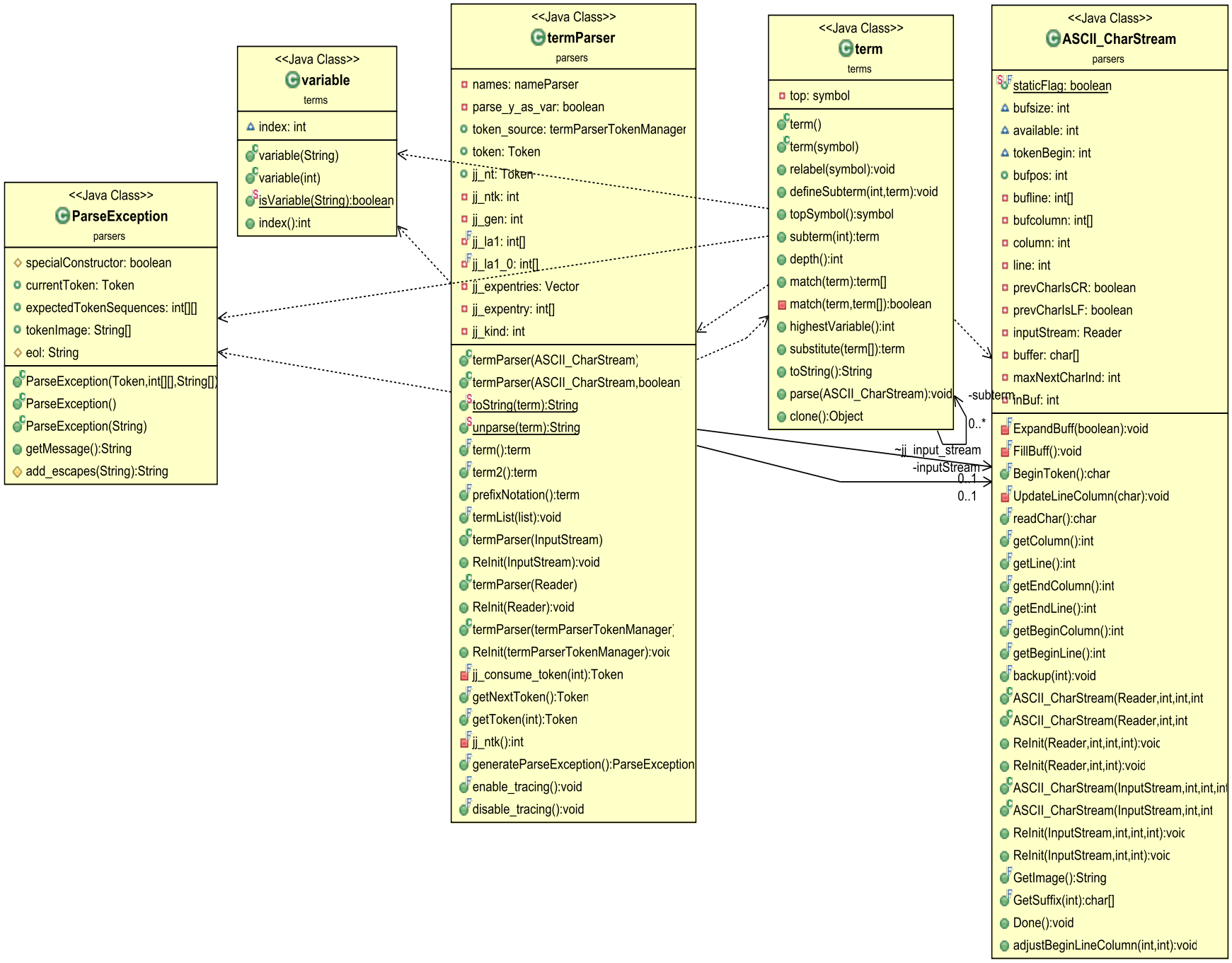


Figure 6.5: UML class diagram for class term

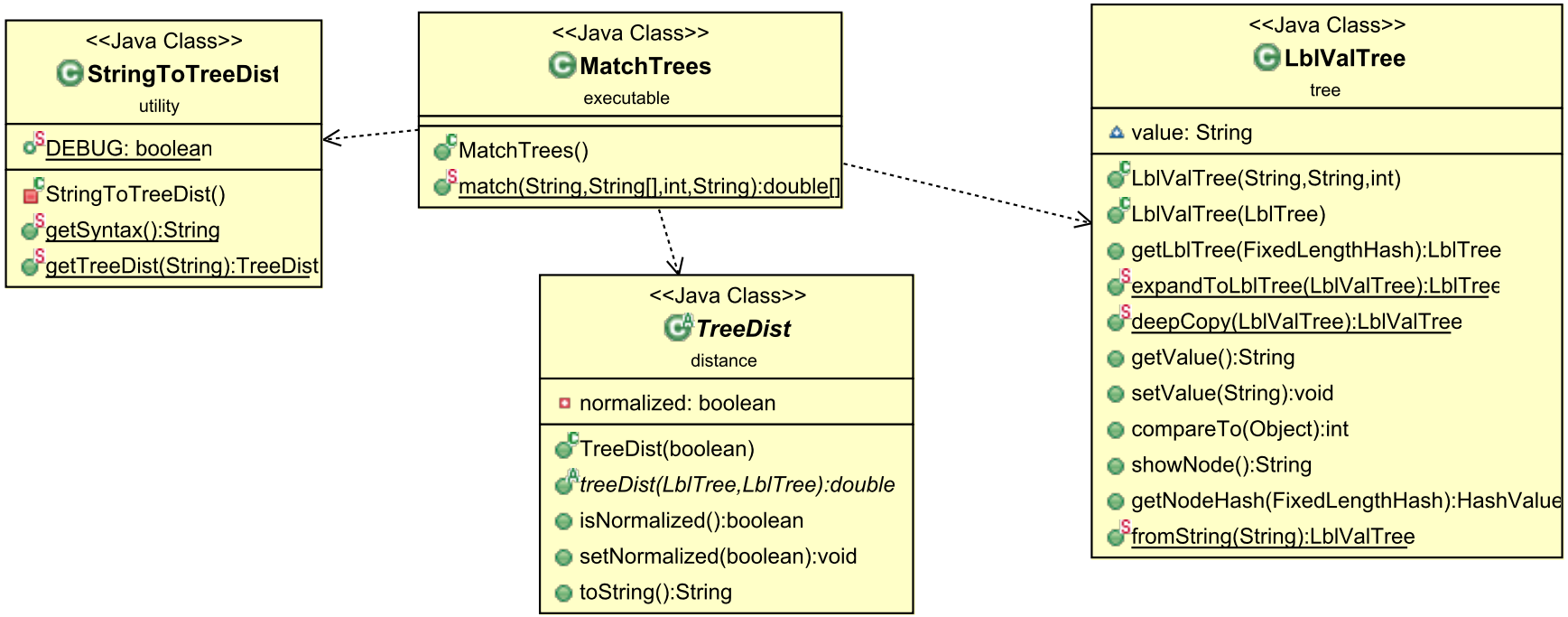


Figure 6.6: UML class diagram for class matchtrees

Algorithm 1 Algorithm for user registration process

1: Begin
2: Display of user registration screen (Figure 6.1 on page 124)
3: **while** the user has not selected up to six pass-images **do**
4: system displays image gallery (Figure 6.7a on page 136)
5: user selects an image representing a particular grammar from image gallery (Figure 6.7a on page 136)
6: system displays images from selected grammar
7: user selects pass-image and system stores grammar for the selected pass-image
8: system stores production rule sequence number
9: **end while**
10: system obtains the number of additional authentication rounds to add to the minimum three rounds required by the system (Figure 6.2 on page 124)
11: system indicates that registration was successful and user can proceed to authentication
12: End

can be completed in less than 20 seconds. The system has a main memory footprint of less than 100 MB while running.

The only user data that is stored by the system is an encrypted string consisting of the username and six pairs of numbers. Each pair of numbers contains the grammar number and the pass-image sequence number. The system is designed to work as a standalone system on desktops, laptops and high end tablets. It is also highly scalable, when used in a networked environment with a large number of computers, with appropriate configuration. For example, on a networked system (i.e. client server system) with a few grammars (i.e. 24), all the grammars can be stored on the client's computer and the only information that will be obtained from the server will be the user authentication string mentioned earlier. In this scenario, the number of users using the system is irrelevant, as the system will be as fast as the processing power of the client's computer. However, if the networked system consists of hundreds of grammars, then the appropriate configuration will be to download the grammars selected by the user during registration and "Add Pass-image" to the client's computer. Thus during authentication, the grammar definition files will be on the client's computer, and the number of people being authenticated by the network will again be irrelevant. The user authentication

Algorithm 2 Algorithm for user addition of pass-images process

1: Begin
2: Display of user pass-image addition screen (Figure 6.1 on page 124)
3: **while** the user has not selected up to three pass-images **do**
4: system display of image gallery (Figure 6.7a on page 136)
5: user selects an image representing a particular grammar from image gallery (Figure 6.7a on page 136)
6: system displays images from selected grammar
7: user selects pass-image and system stores grammar for the selected pass-image
8: system stores production rule sequence number
9: **end while**
10: system adds current pass-images selection to existing ones
11: system obtains the number of additional authentication rounds to add to the minimum of three rounds required by the system (Figure 6.2 on page 124)
12: system indicates that addition of pass-images was successful and user can proceed to authentication
13: End

string could also be stored on the client's computer, making it independent of the network. This is especially useful if the resource to be accessed is local. Therefore, the system is highly scalable and responsive, and can even be modified to run as an applet with appropriate permissions on the client's computer. The images should always be generated on the client's computer, so as not to obviate one of the most important advantages of the system, which is to eliminate the potential bottleneck that might arise from passing images back and forth over the network and to avoid introducing scalability issues.

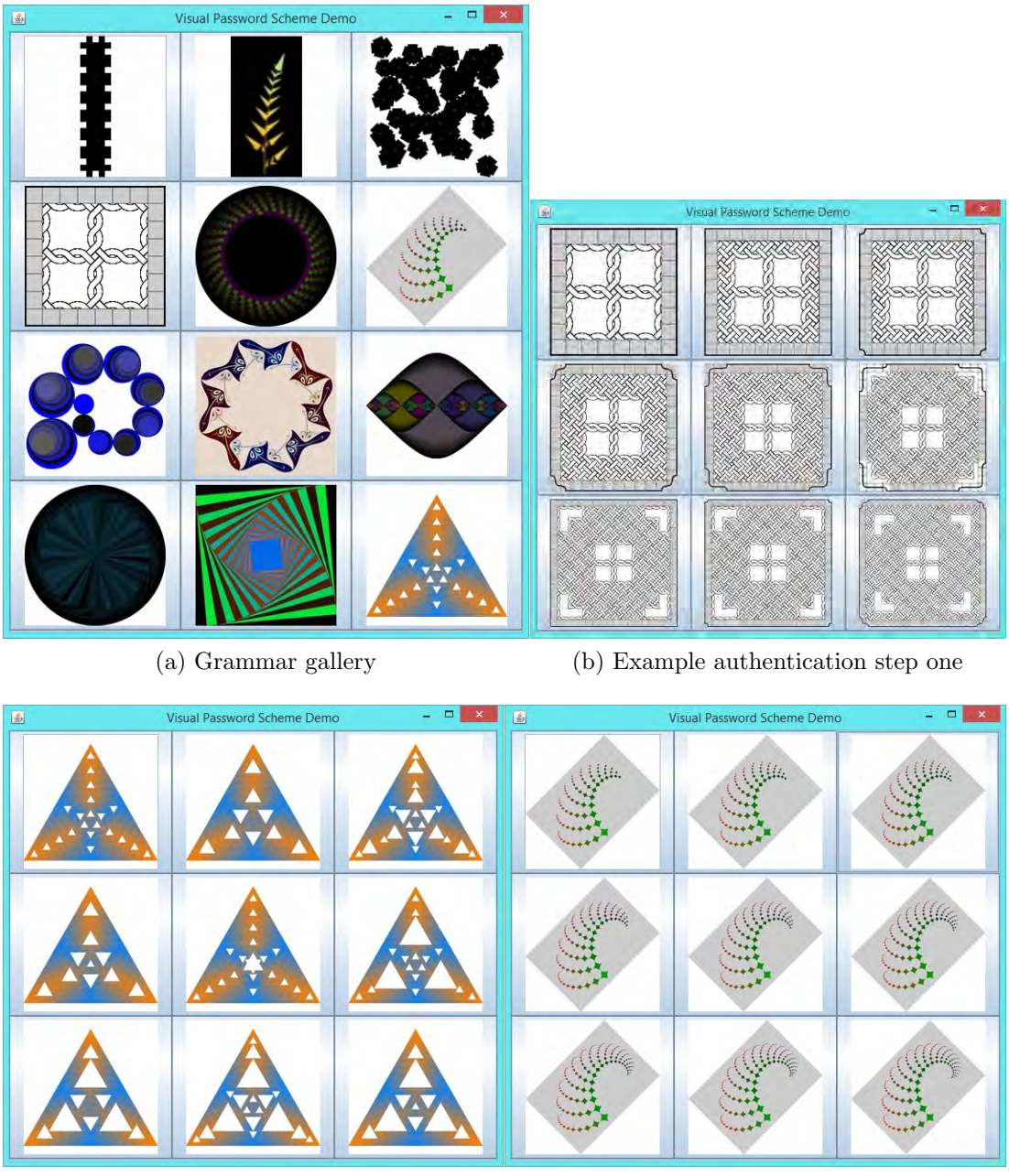
6.4 Summary

This chapter describes the design of a prototype Passfaces type of visual password scheme using grammatical picture generation. The system has the advantage of dynamically generating pass-images instead of the statically displayed images found in the literature. One advantage of this approach is that it removes the need for large storage space for the database of images and the high network bandwidth needed to sustain the high network traffic generated during authentication by the

Algorithm 3 Algorithm for user authentication process

```
1: Begin
2: Display of user login screen (Figure 6.1 on page 124)
3: system adds three to the additional no. of authentication rounds indicated
   by the user to get total no. of authentication rounds (N)
4: while the user has not made up to three authentication attempts do
5:     system shuffles user selected grammars and uses the first N grammars for
       the current authentication process
6:     while the user has not selected up to N pass-images do
7:         system generates 16 (eight images immediately before and after the pass-
           image) images and selects the most similar eight images to be used as
           distractor images in the following step
8:         display of pass-image chosen during the registration process with the eight
           distractor images picked in previous step (Figures 6.7b on the next page,
           6.7c on the following page and 6.7d on the next page)
9:         if user selects the right pass-image then
10:            if user has selected N pass-images then
11:                successful login and stop
12:            end if
13:            repeat while do
14:        else
15:            while the user has not selected up to N pass-images do
16:                system generates and displays a set of similar distractor images
                   without pass-image
17:                discard user input
18:            end while
19:        end if
20:    end while
21: end while
22: failed login and stop
23: End
```

passing of images back and forth over the network. This approach also overcomes the selection bias noted in the literature, as the images used in this study are all abstract computer generated images. The problem of limited potential password space is also eliminated with our approach as an unlimited number of images can be generated as well as variability in the number of pass-images used for authentication. The use of abstract images and the similarity of the generated images is a huge deterrent to shoulder surfing. The granularity of production rules



(a) Grammar gallery

(b) Example authentication step one

(c) Example authentication step two

(d) Example authentication step three

Figure 6.7: Grammar gallery and a three step authentication process

refers to the amount of observable change to an image made by each application of production rules. The ability to control the similarity of the generated images, by controlling the granularity of both the production rules and grammar tables

used for the generation of the images, is an additional advantage offered by this approach to the Passfaces type of visual password scheme. The method used in this study also overcomes the brute force approach and dictionary attacks common in alphanumeric and some visual password schemes.

A fully implemented working system, probably deployed in a networked corporate environment could be developed and further studied. Also, timing considerations and ease of use when compared to similar systems could be further studied. Test users of the system could be recalled and tested after a suitable period of time has elapsed, say six months, in order to determine long term memorability of this system. User experience improvements, when deployed across multiple devices like desktops, laptops and tablets with multiple display resolutions and multi touch capabilities, could be another area for further research.

The next chapter looks at how to use similarity threshold values and the similarity convergence region and point to control how similar the distractor images are to the pass-image.

6. DESIGN AND IMPLEMENTATION OF A NOVEL VISUAL PASSWORD SCHEME

Chapter 7

Experimental characterisation of tree picture grammars used in a visual password scheme

7.1 Introduction

In this chapter, experiments were conducted to determine how to further control the extent of similarity of the distractor images to the pass-image. In other words, using normalised bottom-up tree edit distance measure experiments, the various factors influencing the similarity of the distractor images to the pass-image were studied. Two approaches to further controlling the similarity of distractor images were observed: grammar design refinements and generating distractor images from the similarity convergence region. This chapter concludes with a summary of our findings.

7.2 Similarity threshold value experiments

The questions that this set of experiments addressed in this section were:

- How can we be sure we are using the most similar distractor images?
- How can we make the images more or less similar according to need?

- What are the factors affecting the similarity of the images generated by the grammars?
- Is there a generic method of controlling the similarity of the images generated by all the grammars used in this study?

During the course of this project, it was observed that certain grammars generate images to the point where they are no longer distinguishable to the human eye. This point is referred to in this study as the similarity convergence point and the region before this point as the similarity convergence region. This behaviour was observed mostly amongst deterministic picture grammars. For some other grammars, it was also observed that the images were similar at the same tree depth level. Images generated at different tree depth levels were very different from previously generated images. This can be observed with the first set of images used in the online survey (the I set of images). In the design of the prototype in the last chapter, sixteen images were generated, out of which eight distractor images were selected. In order to answer the first question above and in line with our aforementioned observations, the prototype design was made to use the generated eight images immediately before and after the pass-image, out of which eight distractor images were then selected.

In order to address the second question, it was necessary to determine the best similarity threshold values for each grammar, with the reasoning that ensuring that all selected distractor images are below this threshold value will guarantee that we have the best set of distractor images. An initial normalised distance value of 0.5 was set as the maximum that was to be accepted, since the normalised distance values are between 0 (identical) and 1 (very different). Hence the following experiments were done to both find out the current similarity threshold values being used to pick distractor images by each grammar used in the study, and to see if further adjustments to the prototype are needed to ensure that all the images selected are below our optimal similarity threshold value. The results of these experiments were also expected to help answer the third and fourth questions heretofore itemised.

A discussion of the procedure followed in the conduct of the experiments now follows. The first observed image (image 0 in the image array, in an array of

images with increasing terminals, since our array is zero based) generated by each grammar was selected for our initial set of experiments, although it had earlier been observed that the first set of images were generally more dissimilar than similar for some of the grammars. Also, the first image generated by a grammar does not have any previous image to use as a distractor image, so all the sixteen images had to come from images generated after our first image. The ninth image (image 8) and sixteenth image (image 15) were also selected for our initial set of experiments to see if there was any noticeable trend or differences between the tree edit distance values of the distractor images when these three images are used as pass-images. Notice that when both the ninth and sixteenth images were used as pass-images, it was possible to generate the eight images before and after the pass-image to use as a pool, out of which the most similar eight distractor images were chosen. The generated images were then sorted by their normalised bottom-up tree edit distance values. The tree edit distance value of the ninth image, in the sorted array of images, is referred to in this study as the similarity threshold value[1]. The pass-image and the eight distractor images were then shuffled and displayed. The sixteen images used for picking the pass-images (during user registration), the tree edit distance computed and their resultant display can be seen in Appendix D. The results of the initial set of experiments are however included in Table 7.1 on page 145 and 7.2 on page 146 and graphed in Figure 7.1 on page 142 for ease of discussion. The more comprehensive set of experiments is shown in the above table and in subsequent tables.

The next section describes the results of the more comprehensive series of experiments performed to ensure that the degree of similarity of the distractor images generated by our prototype can be determined by the researcher.

7.3 Results

Results show that for some of the grammars, referred to as symmetrical grammar type in this study, the similarity threshold values reduce (i.e. the images become more similar) for the distractor images, the further the move away from using the initially generated images as pass-images. This can be clearly seen in Figure 7.1

[1]See nomenclature for a differently worded definition.

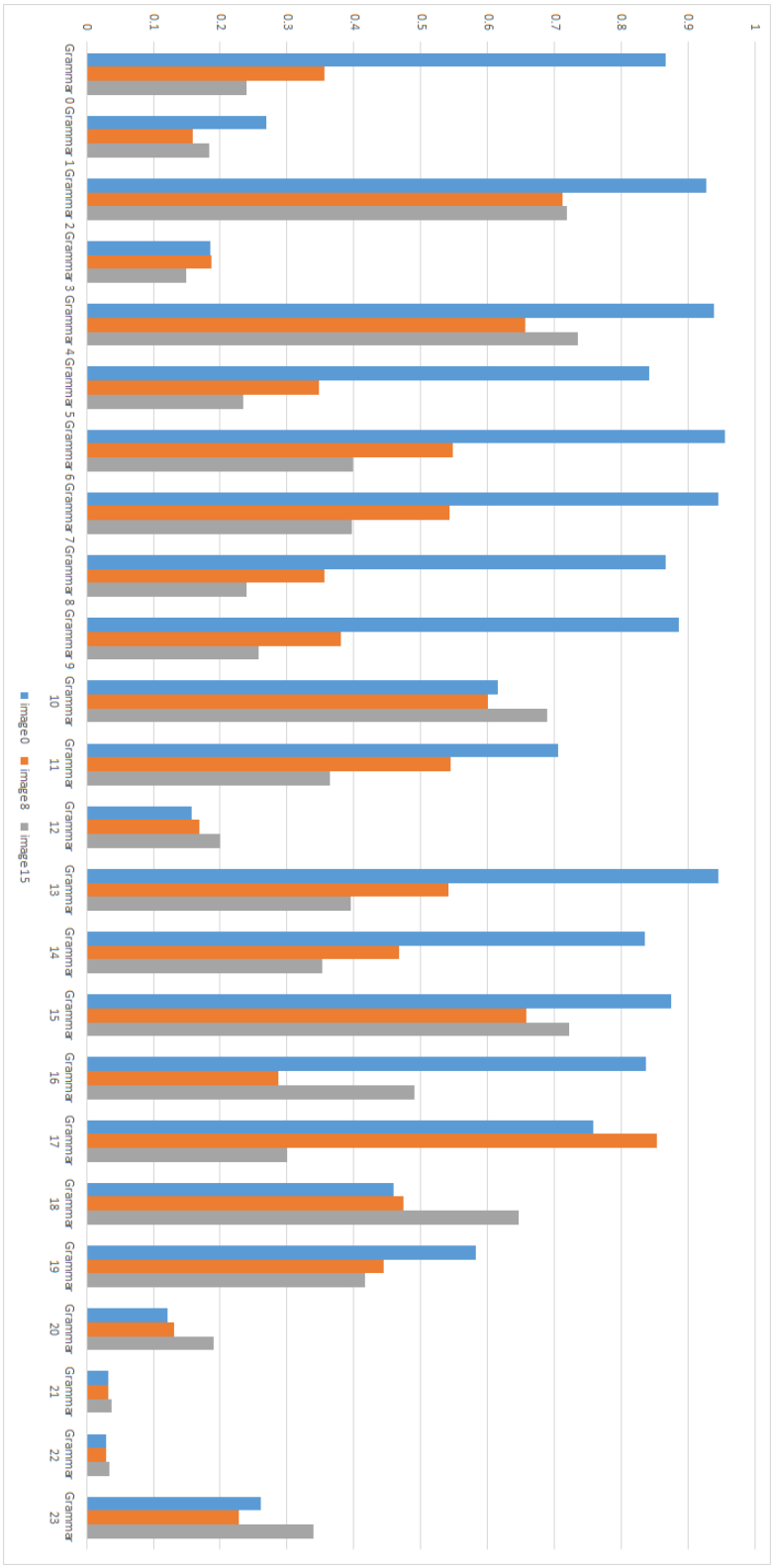


Figure 7.1: Column graph for threshold values

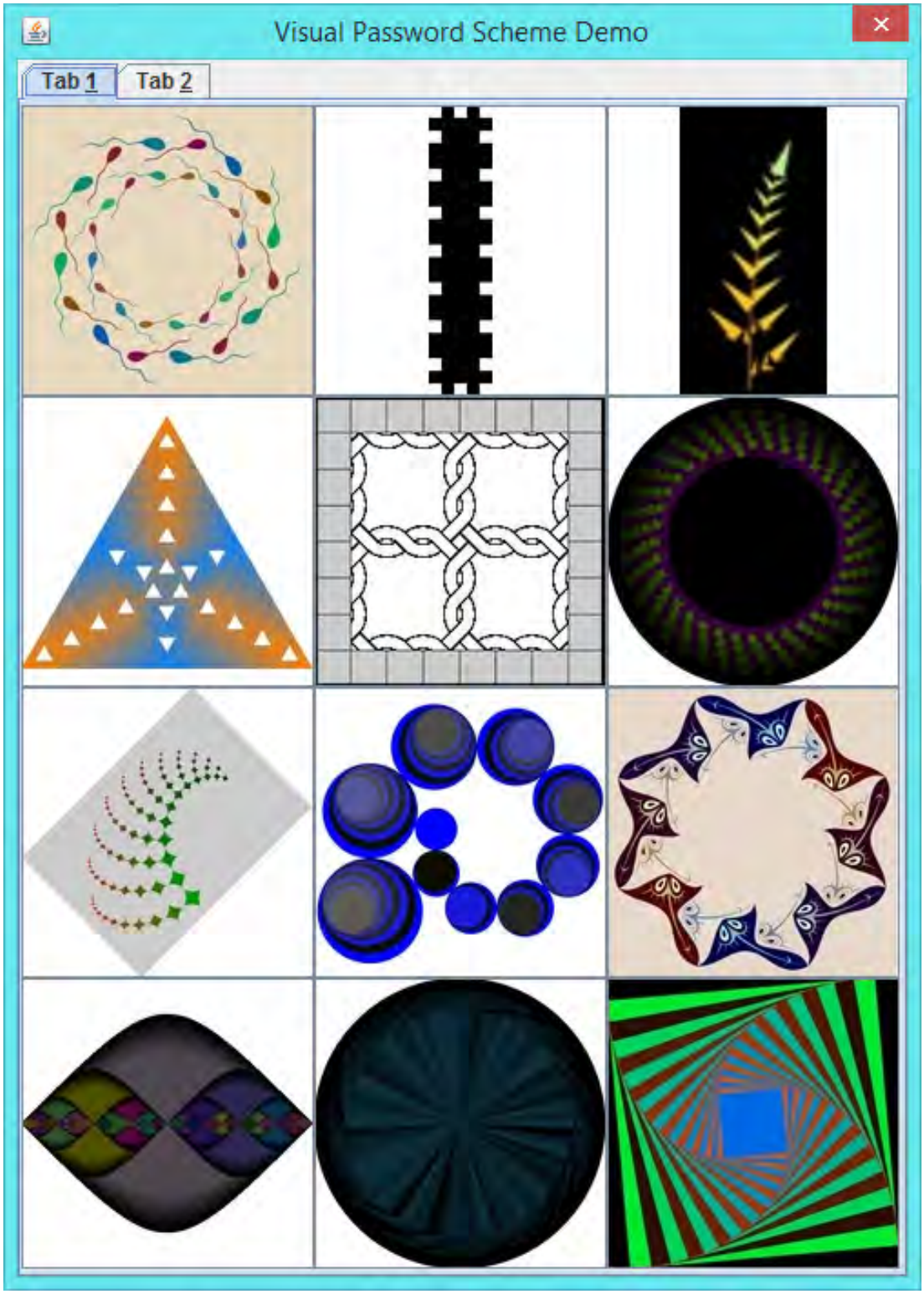


Figure 7.2: First twelve grammars gallery

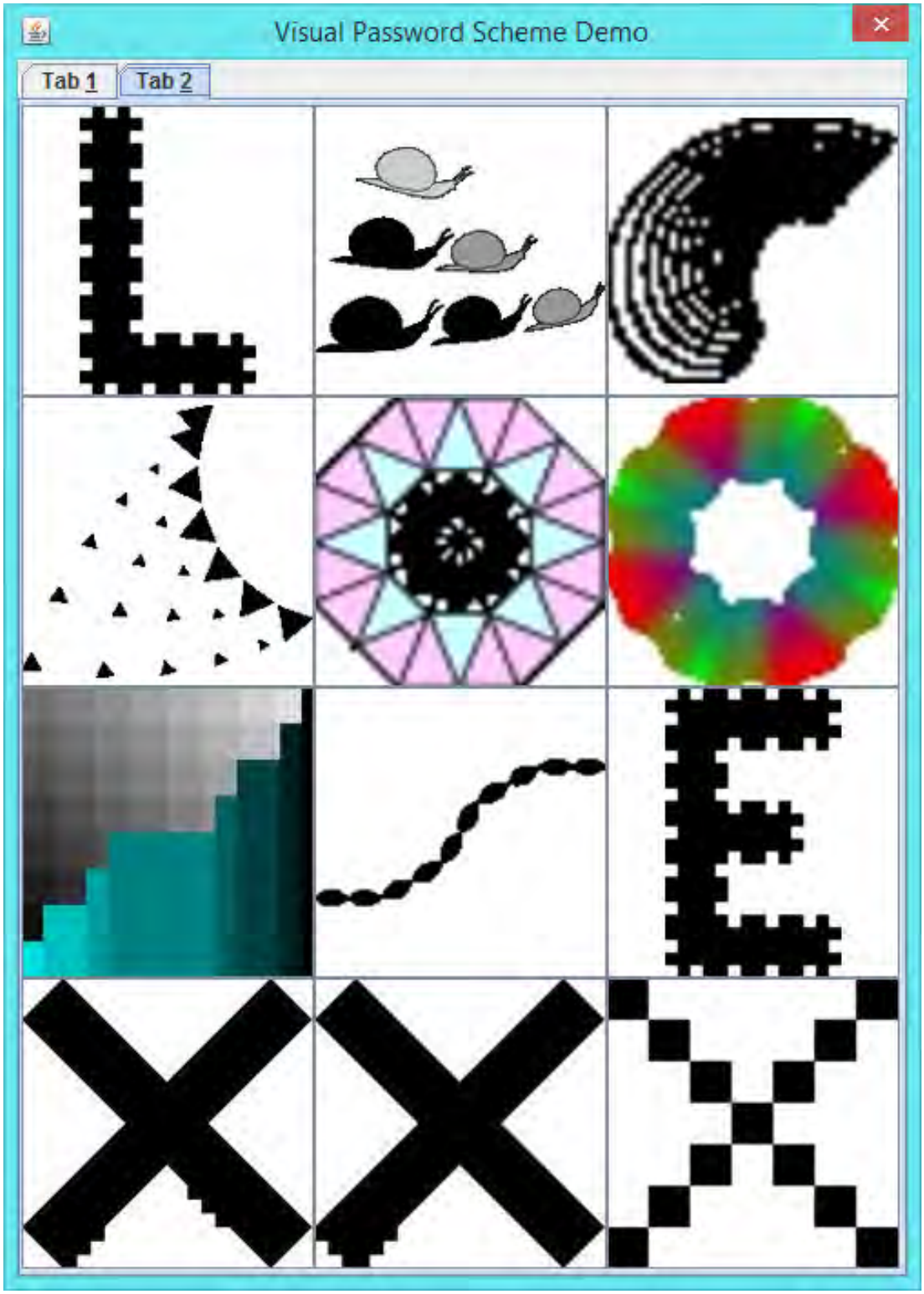


Figure 7.3: Second twelve grammars gallery

Table 7.1: Similarity threshold value for the 24 grammars

Pass-Image no.	Grammar 0	Grammar 1	Grammar 2	Grammar 3	Grammar 4	Grammar 5
Pass-image 0	0.866666667	0.269230769	0.927927928	0.185737977	0.938461538	0.842639594
Pass-image 1	0.760000000	0.159420290	0.873873874	0.176975945	0.809523810	0.738216099
Pass-image 2	0.653333333	0.222222222	0.819819820	0.186046512	0.773584906	0.633792603
Pass-image 3	0.590361446	0.183333333	0.765765766	0.154228856	0.684848485	0.573867367
Pass-image 4	0.538461538	0.159420290	0.757575758	0.208333333	0.876923077	0.524295141
Pass-image 5	0.490196078	0.183333333	0.761006289	0.154485050	0.569230769	0.466737064
Pass-image 6	0.423728814	0.158730159	0.725490196	0.187604690	0.692307692	0.405132906
Pass-image 7	0.383177570	0.159420290	0.733333333	0.181063123	0.722772277	0.373401535
Pass-image 8	0.356521739	0.159420290	0.711711712	0.187604690	0.657142857	0.347784659
Pass-image 9	0.333333333	0.183333333	0.761006289	0.153988868	0.780821918	0.325456977
Pass-image 10	0.312977099	0.174603175	0.771689498	0.152917505	0.752212389	0.305823209
Pass-image 11	0.294964029	0.183333333	0.745704467	0.153703704	0.826086957	0.288423548
Pass-image 12	0.278911565	0.159420290	0.716629381	0.150943396	0.521367521	0.272897196
Pass-image 13	0.264516129	0.300000000	0.726634252	0.201207243	0.646017699	0.258957077
Pass-image 14	0.251533742	0.183333333	0.725723345	0.148846960	0.565891473	0.246371920
Pass-image 15	0.239766082	0.183333333	0.717971269	0.144308943	0.735537190	0.234953331
Pass-image 16	0.229050279	0.260869565	0.716629381	0.156474820	0.781818182	0.224546293
Std dev.	0.187552248	0.042309560	0.057950714	0.019844048	0.108501780	0.181342004
Variance	**0.035175846**	**0.001790099**	**0.003358285**	**0.000393786**	**0.011772636**	**0.032884922**
Min.	**0.229050279**	**0.158730159**	**0.711711712**	**0.144308943**	**0.521367521**	**0.224546293**
Max.	**0.866666667**	**0.300000000**	**0.927927928**	**0.208333333**	**0.938461538**	**0.842639594**
Max. diff.	**0.637616388**	**0.141269841**	**0.216216216**	**0.064024390**	**0.417094017**	**0.618093301**

Pass-image no.	Grammar 6	Grammar 7	Grammar 8	Grammar 9	Grammar 10	Grammar 11
Pass-image 0	0.954954955	0.945945946	0.866666667	0.886629526	0.642629905	0.705882353
Pass-image 1	0.900900901	0.891891892	0.760000000	0.783577128	0.624501425	1.000000000
Pass-image 2	0.828828829	0.819819820	0.653333333	0.702797203	0.571993224	0.617647059
Pass-image 3	0.781954887	0.774436090	0.590361446	0.616965620	0.558516196	1.000000000
Pass-image 4	0.738853503	0.720930233	0.538461538	0.564264487	0.548699335	0.529411765
Pass-image 5	0.666666667	0.649122807	0.490196078	0.519735461	0.554360136	1.000000000
Pass-image 6	0.612021858	0.606557377	0.423728814	0.467770815	0.601087494	0.473684211
Pass-image 7	0.578199052	0.573459716	0.383177570	0.413936430	0.594512195	1.000000000
Pass-image 8	0.547717842	0.543568465	0.356521739	0.381000782	0.601351351	0.545454545
Pass-image 9	0.520146520	0.516483516	0.333333333	0.356759777	0.613814757	0.500000000
Pass-image 10	0.495114007	0.491856678	0.312977099	0.335476718	0.630190523	0.480000000
Pass-image 11	0.472303207	0.469387755	0.294964029	0.316478369	0.619346171	0.444444444
Pass-image 12	0.451443570	0.448818898	0.278911565	0.299594395	0.646174259	0.428571429
Pass-image 13	0.432304038	0.429928741	0.264516129	0.284309946	0.665104389	0.400000000
Pass-image 14	0.414686825	0.412526998	0.251533742	0.270603312	0.662452107	0.387096774
Pass-image 15	0.398422091	0.396449704	0.239766082	0.258174792	0.713175282	0.363636364
Pass-image 16	0.383363472	0.381555154	0.229050279	0.246917205	0.692901802	0.352941176
Std dev.	**0.177372135**	**0.174128265**	**0.187552248**	**0.190626964**	**0.046034017**	**0.237090984**
Variance	**0.031460874**	**0.030320653**	**0.035175846**	**0.036338640**	**0.002119131**	**0.056212135**
Min.	**0.383363472**	**0.381555154**	**0.229050279**	**0.246917205**	**0.548699335**	**0.352941176**
Max.	**0.954954955**	**0.945945946**	**0.866666667**	**0.886629526**	**0.713175282**	**1.000000000**
Max. diff.	**0.571591483**	**0.564390792**	**0.637616388**	**0.639712321**	**0.164475947**	**0.647058824**

on page 142 with grammars 6-9, 14 and 19. A well designed non-symmetrical grammar type, tends to have very uniform similarity threshold values, as can be seen for grammars 21 and 22 in Figure 7.1 on page 142. The behaviour of other non-symmetrical grammars may even show a slight increase in similarity

Table 7.2: Similarity threshold value for the 24 grammars - continued

Pass-Image no.	Grammar 12	Grammar 13	Grammar 14	Grammar 15	Grammar 116	Grammar 17
Pass-image 0	0.157894737	0.945454545	0.835164835	0.874251497	0.837150127	0.757575758
Pass-image 1	0.168421053	0.890909091	0.769230769	0.814371257	0.436123348	0.730639731
Pass-image 2	0.197674419	0.818181818	0.692307692	0.742514970	0.395993837	0.755958055
Pass-image 3	0.207920792	0.772727273	0.604395604	0.658682635	0.329914530	0.778836988
Pass-image 4	0.119565217	0.714285714	0.571428571	0.657276995	0.305605787	0.813197970
Pass-image 5	0.173913043	0.642857143	0.541666667	0.659340659	0.247600768	0.832221163
Pass-image 6	0.204819277	0.604395604	0.514705882	0.612612613	0.234188034	0.761194030
Pass-image 7	0.157894737	0.571428571	0.490196078	0.645669291	0.284705882	0.853807107
Pass-image 8	0.168421053	0.541666667	0.467836257	0.658682635	0.287531807	0.854103343
Pass-image 9	0.180722892	0.514705882	0.447368421	0.657276995	0.247264770	0.432372506
Pass-image 10	0.246913580	0.490196078	0.428571429	0.676156584	0.247600768	0.379363848
Pass-image 11	0.130434783	0.467836257	0.411255411	0.708661417	0.247264770	0.363564041
Pass-image 12	0.173913043	0.447368421	0.395256917	0.716369530	0.284705882	0.329421626
Pass-image 13	0.166666667	0.428571429	0.380434783	0.707015131	0.341176471	0.363564041
Pass-image 14	0.253333333	0.411255411	0.366666667	0.719840479	0.348600509	0.354538402
Pass-image 15	0.200000000	0.395256917	0.352380952	0.722100656	0.490304709	0.299920446
Pass-image 16	0.130434783	0.380434783	0.337662338	0.716369530	0.493282150	0.350835322
Std dev.	**0.035974413**	**0.173921072**	**0.142901900**	**0.062561947**	**0.144860929**	**0.220244771**
Variance	**0.001294158**	**0.030248539**	**0.020420953**	**0.003913997**	**0.020984689**	**0.048507759**
Min.	**0.119565217**	**0.380434783**	**0.337662338**	**0.612612613**	**0.234188034**	**0.299920446**
Max.	**0.253333333**	**0.945454545**	**0.835164835**	**0.874251497**	**0.837150127**	**0.854103343**
Max. diff.	**0.133768116**	**0.565019762**	**0.497502497**	**0.261638884**	**0.602962093**	**0.554182897**

Pass-Image no.	Grammar 18	Grammar 19	Grammar 20	Grammar 21	Grammar 22	Grammar 23
Pass-image 0	0.458823529	0.583333333	0.120967742	0.032954545	0.029411765	0.261363636
Pass-image 1	0.492753623	0.500000000	0.118518519	0.034090909	0.029411765	0.352272727
Pass-image 2	0.435294118	0.444444444	0.173611111	0.032954545	0.029411765	0.227272727
Pass-image 3	0.383838384	0.555555556	0.138211382	0.036363636	0.029411765	0.261363636
Pass-image 4	0.400000000	0.444444444	0.128787879	0.032954545	0.029411765	0.173913043
Pass-image 5	0.494117647	0.555555556	0.113636364	0.032954545	0.029411765	0.227272727
Pass-image 6	0.465753425	0.444444444	0.148936170	0.035169988	0.034313725	0.252173913
Pass-image 7	0.492753623	0.500000000	0.169354839	0.034090909	0.034313725	0.352272727
Pass-image 8	0.474226804	0.444444444	0.130434783	0.032954545	0.029411765	0.227272727
Pass-image 9	0.361904762	0.583333333	0.126984127	0.032954545	0.029411765	0.253164557
Pass-image 10	0.371134021	0.500000000	0.155555556	0.035108959	0.029411765	0.188679245
Pass-image 11	0.444444444	0.416666667	0.172413793	0.032954545	0.029411765	0.227272727
Pass-image 12	0.691629956	0.416666667	0.142857143	0.033997655	0.029411765	0.193548387
Pass-image 13	0.657276995	0.583333333	0.130081301	0.033997655	0.029411765	0.188679245
Pass-image 14	0.352941176	0.416666667	0.166666667	0.039951574	0.034313725	0.218045113
Pass-image 15	0.646766169	0.416666667	0.189655172	0.037500000	0.034313725	0.339622642
Pass-image 16	0.659898477	0.333333333	0.139130435	0.033997655	0.029411765	0.173913043
Std dev.	**0.107799439**	**0.071635001**	**0.021859735**	**0.001889322**	**0.002079326**	**0.056137559**
Variance	**0.011620719**	**0.005131573**	**0.000477848**	**0.000003570**	**0.000004324**	**0.003151426**
Min.	**0.352941176**	**0.333333333**	**0.113636364**	**0.032954545**	**0.029411765**	**0.173913043**
Max.	**0.691629956**	**0.583333333**	**0.189655172**	**0.039951574**	**0.034313725**	**0.352272727**
Max. diff.	**0.338688780**	**0.250000000**	**0.076018808**	**0.006997029**	**0.004901960**	**0.178359684**

threshold values as can be seen for grammars 12, 18 and 20 in the same figure. The others either show slightly increasing or decreasing similarity threshold values. All symmetrical grammars move towards a similarity convergence point. Another noticeable result of the experiment is that the rate of movement of grammars towards this similarity convergence point is different for different grammars. A

quick look at the images in appendix D.3 will show that some of the grammars have arguably gone past the similarity convergence point. The implication of this is that similarity values for different grammars are incomparable. In other words, 0.5 normalised similarity threshold value for one grammar may produce a more similar set of distractor images than say 0.3 similarity threshold value for another grammar. A well designed non-symmetrical tree grammar generates images with similarity threshold values that oscillate about a mean normalised tree edit distance value. In other words, their normalised values cluster around a central value with little variation, which produce straight line graph plots.

The results show that there are basically three types of tree picture grammars, as far as visual password schemes are concerned: symmetrical grammars, non-symmetrical grammars and hybrid tree picture grammars. Hybrid tree picture grammars cannot be further fine tuned with respect to the similarity of their generated images through their generic behaviour, and as such are the least suitable for use in a visual password scheme. The characteristics of these grammars are discussed in the next section.

7.3.1 Symmetrical tree picture grammars

These are grammars that have tree edit distance values that cluster around the pass-image tree edit distance value. These types of grammars have between "V" and "U" shaped graphs , when the similarity threshold values of the 16 distractor images are plotted against their generation sequence numbers (example Figure 7.5 on page 149). They usually have a large standard deviation, variance and a large difference between the minimum and maximum similarity threshold values (see Table 7.1 on page 145 and Table 7.2 on the facing page).

A plot of the similarity threshold values, also displays a downward left-to-right sloping curve (example Figure 7.4 on page 149) that flatten out at the similarity convergence point, with the similarity convergence region coming up immediately before this point. These can be deterministic grammars with the images becoming indistinguishable at the similarity convergence point, although the tree edit distance measure continues to distinguish between the images, but producing increasingly smaller differences in their similarity values.

The initially observed set of images generated by this type of grammar are quite distinct and are not suitable for use in a visual password scheme. However, the images become usable as the generating sequence enters the similarity threshold region. On plotting the unordered tree edit distance values for these images, one obtains a "V" shaped graph that can be fitted to a quadratic curve that is symmetrical around the pass-image tree edit distance values. The equation of a quadratic curve is,

$$y = ax^2 + bx + c. \tag{7.1}$$

Further examples of symmetrical grammars are in Appendix F.

7.3.2 Non-symmetrical tree picture grammars

These grammars are usually non-deterministic, produce horizontal straight line graphs for plots of their similarity threshold values (Figure 7.6 on page 150) and a combination of zig-zag and straight lines for a plot of their normalised tree edit distance measure values (Figure 7.7 on page 150). They also have small standard deviation values, variance values and a small difference between their minimum and maximum values (see Table 7.1 on page 145 and Table 7.2 on page 146). They also do not have similarity convergence point or similarity convergence region, as they tend to have almost uniform tree edit distance values from the very first image.

The equation of a straight line is,

$$y = mx + c. \tag{7.2}$$

where m is the gradient of the straight line and c is the intercept with the y axis. m tends towards 0 for these types of grammars, with the resulting equation being,

$$y = c. \tag{7.3}$$

c usually has values of less than 0.4.

Further examples of non-symmetrical grammars are in Appendix F.

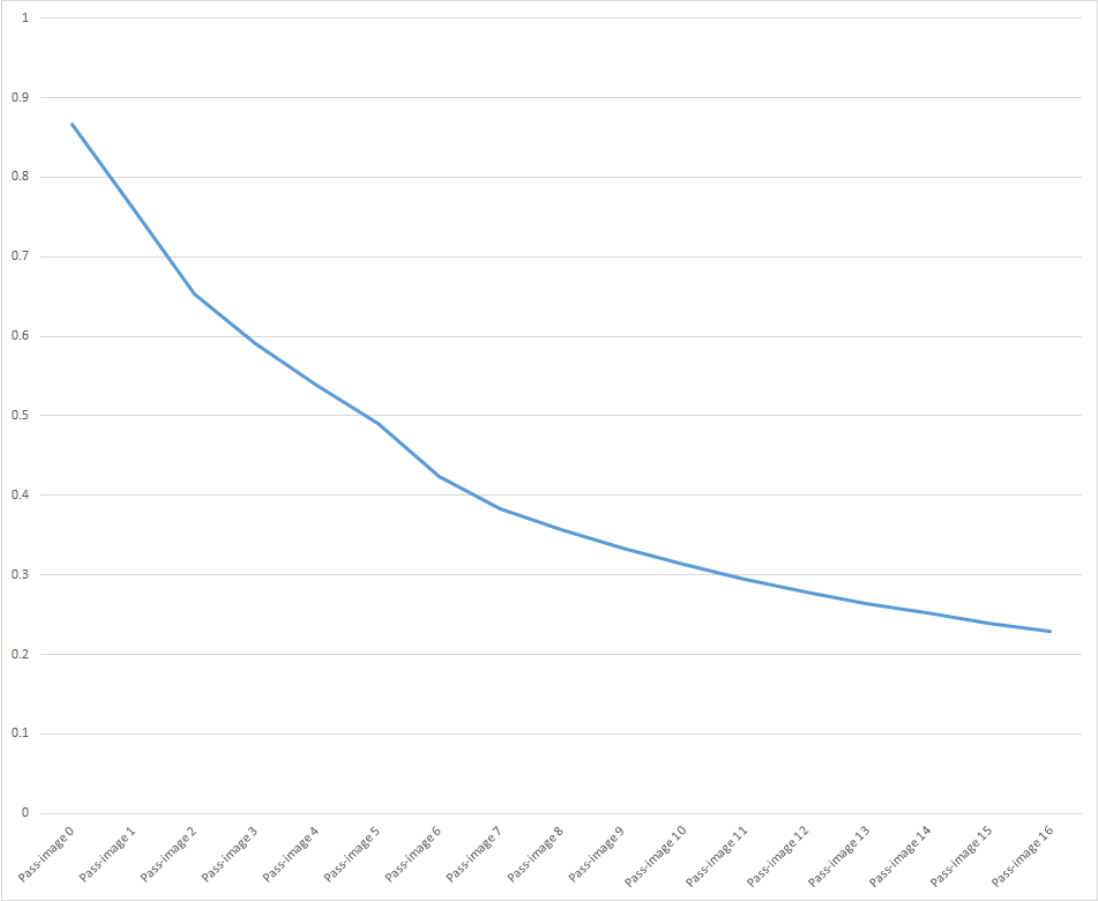


Figure 7.4: Similarity threshold values for grammar 0

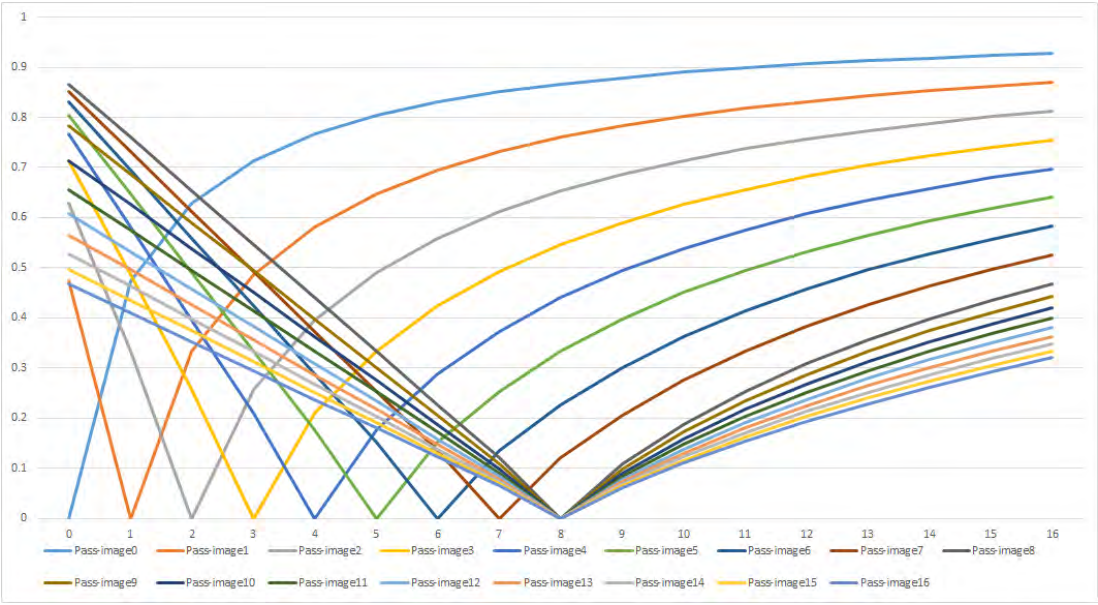


Figure 7.5: Bottom-up tree edit distance values for grammar 0 for Pass-images 0 to 16

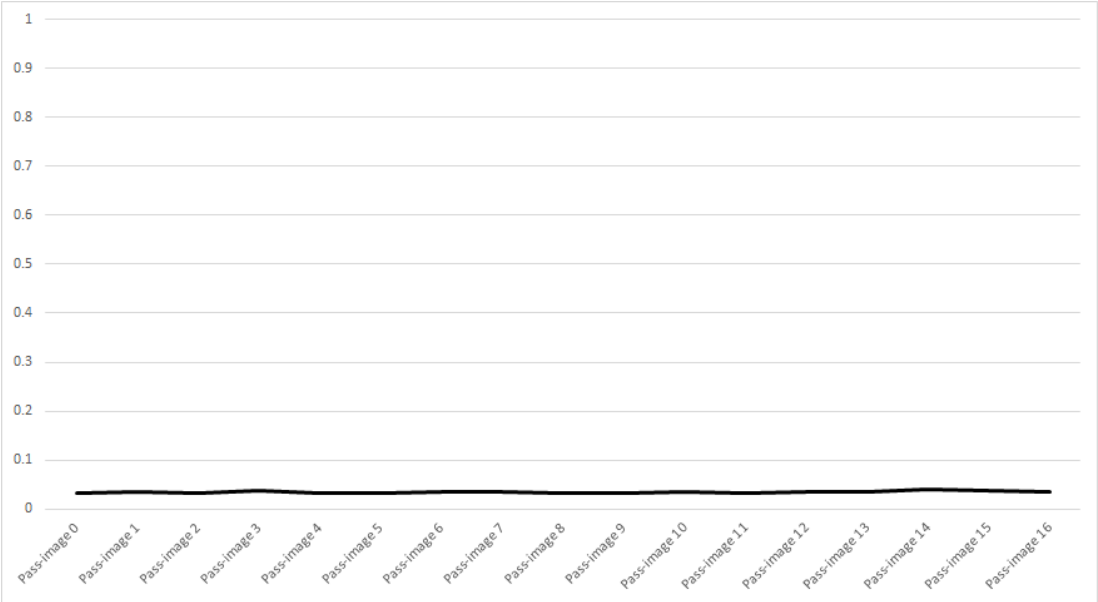


Figure 7.6: Similarity threshold values for grammar 21

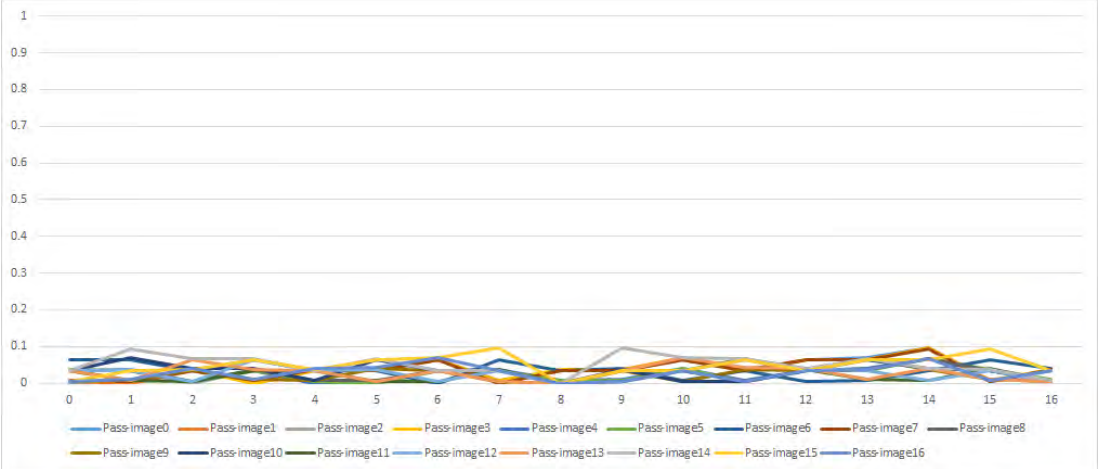


Figure 7.7: Bottom-up tree edit distance values for grammar 21 for pass-images 0 to 16

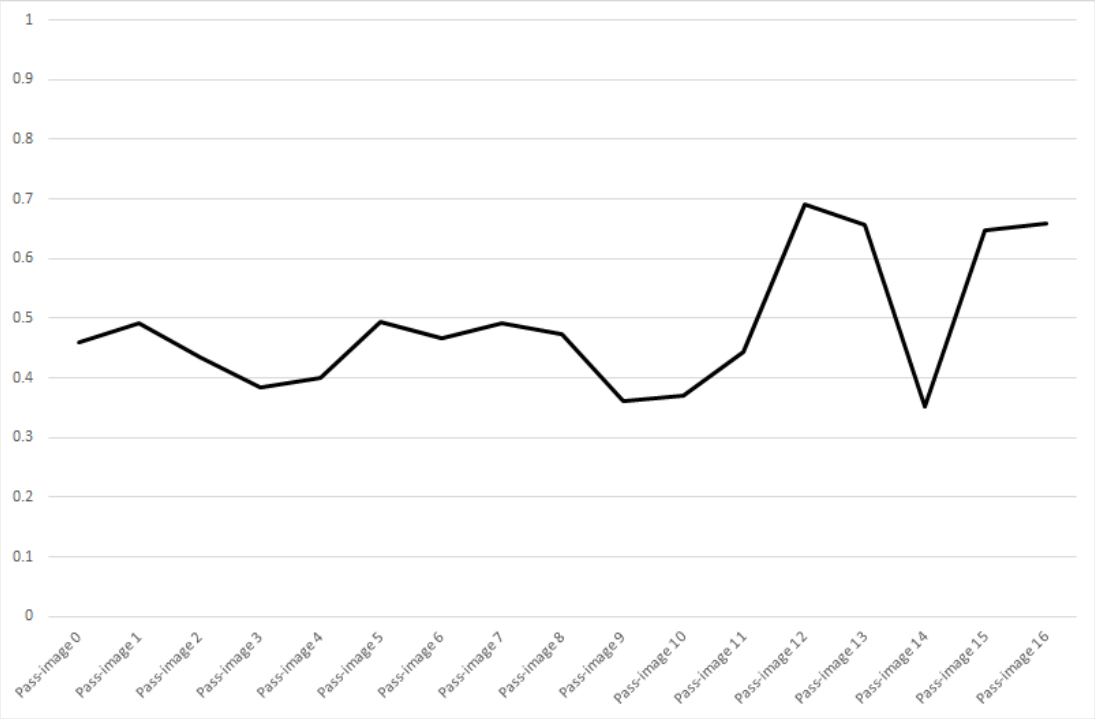


Figure 7.8: Similarity threshold values for grammar 18

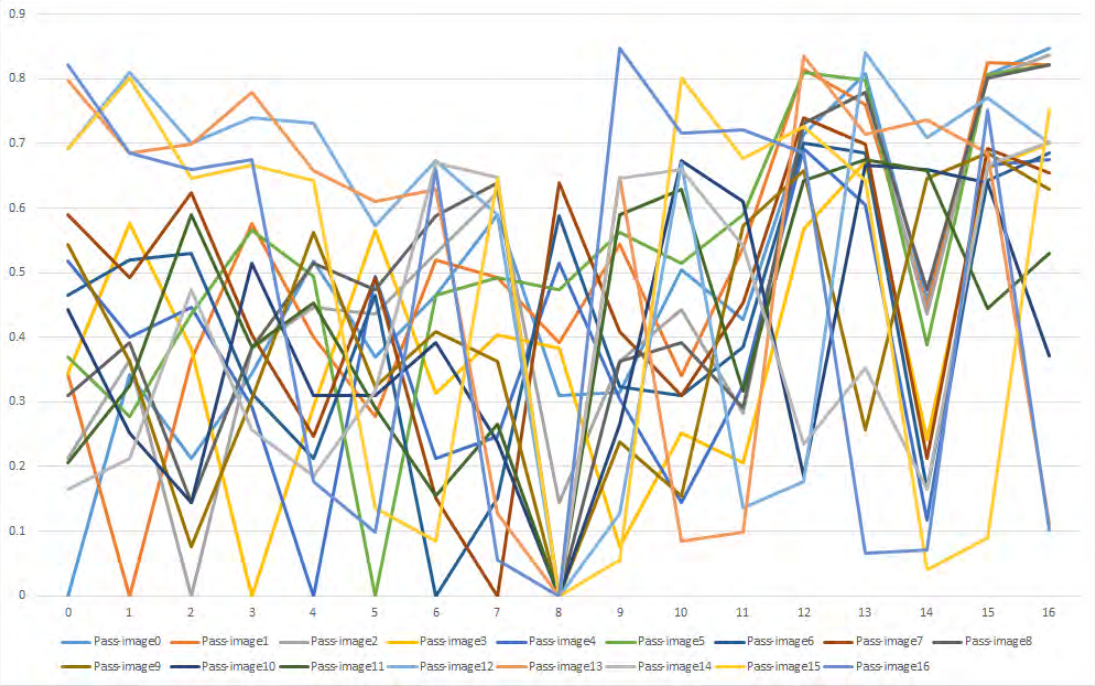


Figure 7.9: Bottom-up tree edit distance values for grammar 18 for pass-images 0 to 16

7.3.3 Hybrid tree picture grammars

Hybrid grammars are grammars which possess unpredictable similarity threshold values. They may possess some of the attributes of both symmetrical and non-symmetrical grammars. A plot of their similarity threshold values neither display the downward sloping curves of symmetrical grammars nor the straight lines of non-symmetrical grammars (see Figure 7.8 and Figure 7.9 on the preceding page). Similarly, a plot of their Bottom-up tree edit distance values do not show the "V" shaped graphs of symmetrical grammars nor the horizontal patterns of non-symmetrical grammars (see Figure 7.9 on the previous page).

Further examples of hybrid grammars are in Appendix F:

7.3.4 Goodness of fit plot for normalised tree edit distance values of grammars

Table 7.3: Quadratic goodness of fit test for symmetrical grammars

Quadratic goodness of fit test for symmetrical grammars				
grammar	a	b	c	R^2
0	0.0177	-0.2956	1.3125	0.8976
2	0.0345	-0.5454	2.3709	0.8164
5	0.0117	-0.1918	0.8375	0.9071
6	0.0213	-0.3504	1.5431	0.8821
7	0.0235	-0.3875	1.7113	0.8795
8	0.0125	-0.2067	0.9065	0.9012
9	0.0121	-0.1998	0.8948	0.8535
13	0.0194	-0.3180	1.3894	0.8933
14	0.0171	-0.2798	1.2173	0.8982
15	0.0315	-0.4863	2.0667	0.8321
16	0.0123	-0.1644	0.6315	0.8327
17	0.0339	-0.6668	2.8385	0.7120
std. dev.	**0.008194828**	**0.149734601**	**0.64227367**	**0.053323126**
Variance	**6.71552E-05**	**0.022420451**	**0.412515467**	**0.002843356**
Average	**0.0206**	**-0.3410**	**1.4767**	**0.8588**
Max.	**0.0345**	**-0.1644**	**2.8385**	**0.9071**
Min.	**0.0117**	**-0.6668**	**0.6315**	**0.7120**
Max. diff.	**0.0228**	**0.5024**	**2.2070**	**0.1951**

7.3.4 Goodness of fit plot for normalised tree edit distance values of grammars

A goodness of fit plot (R^2) was conducted on the graphs of both the symmetrical and non-symmetrical grammars in order to determine the best equation to best describe both types of grammars. R^2 values, which can range from -1 (perfect negative fit) through 0 (no fit) to 1 (perfect fit), for the nine displayed images were plotted for both the symmetrical and the non-symmetrical grammars.

The goodness of fit plots available for these data are linear, logarithmic, polynomial and moving average. The best goodness of fit plot for symmetrical grammars was polynomial of order two (quadratic) to six. Table 7.4 shows that increasing polynomial order values also increases goodness of fit values for grammar 0 and pass-image eight. The goodness of fit plot for polynomial order two (quadratic) and six can be seen in Figures 7.11 on page 155 and 7.12 on page 155 respectively. As a result of the additional analysis complexity of using higher order polynomial goodness of fit plots without providing commensurate information, quadratic "U" shaped plots were used in this study.

Table 7.4: Goodness of fit values for various polynomial order values for grammar 0 and pass-mage 8

Polynomial order value	**Goodness of fit (R^2) values**
2	0.8970
3	0.9000
4	0.9655
5	0.9657
6	0.9855

The graphs for a sample symmetrical grammar, showing both the equation of the resultant graphs and their R^2 values, for symmetrical grammar number zero for pass-images five and eight are shown in Figure 7.10 on page 155 and Figure 7.11 on page 155 respectively. A plot of all the similarity values was also done for grammar number 0 (using data from Table F.1 on page 373) in Figure 7.13 on page 156. A very good fit to the quadratic equation of approximately 0.9 for all the pass-images for grammar number zero was found. The complete table containing both the quadratic and linear plots values can be found in Appendix F.4 on page 456. The table of values for symmetrical grammars is Table 7.3 on the facing page and the complete set of graphs can be found in Appendix F.3 on page 422.

The non-symmetrical grammar number 21 plots for pass-images five and eight are depicted in Figure 7.14 on page 156 and Figure 7.15 on page 157 respectively. All the R^2 values are low for grammar number 21 (approximately 0.1), including a plot of all the displayed images for it (Figure 7.16 on page 157) from Table F.22 on page 394. The reason for the apparently low R^2 values, stem from variations in similarity values being more than that of the maximum tree edit distance similarity values, as can be seen in Figure 7.16 on page 157. Their graphs therefore, form a horizontal zig-zag pattern. However, it can also be observed in the last mentioned figure, that the resultant general equation, is that of a straight line.

7.3.5 Grammar type limitations

This section discusses the limitations of symmetrical and non-symetrical grammars and how to reduce or eliminate the impact of these limitations on a Passfaces type visual password scheme.

7.3.5.1 Symmetrical grammars

The presence of a similarity convergence point for this type of grammar means that an infinite number of images cannot be generated for use in a visual password scheme. Also, the additional fact that the first set of images has to be discarded, due to their not being very similar, means that fewer images are available for use in a visual password scheme.

To overcome these limitations, the grammar designer has to increase the granularity [1] of the production rules by increasing the number of production rule tables at the same nesting depth level or by transforming existing production rules in such a way as to make the smallest possible observable change to the generated images or by using both approaches. This will increase both the number of images generated in the similarity convergence region and the number of images available to increase potential password space. This will also have the additional benefit of increasing the speed of generation of the images and thus increase the overall responsiveness of the visual password system.

[1] See nomenclature for a more formal definition

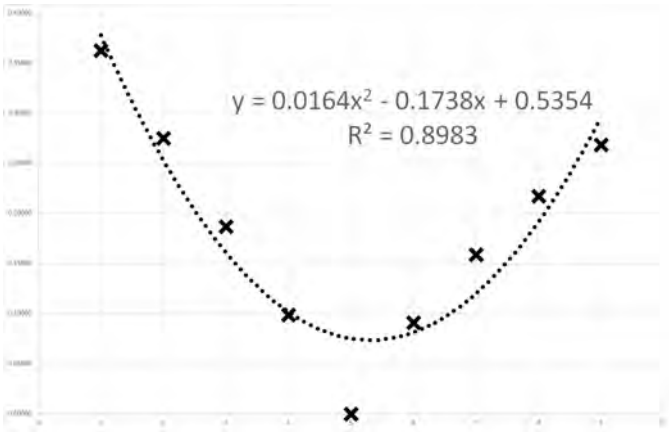


Figure 7.10: Goodness of fit graph for displayed images for grammar 0 for pass-image 5

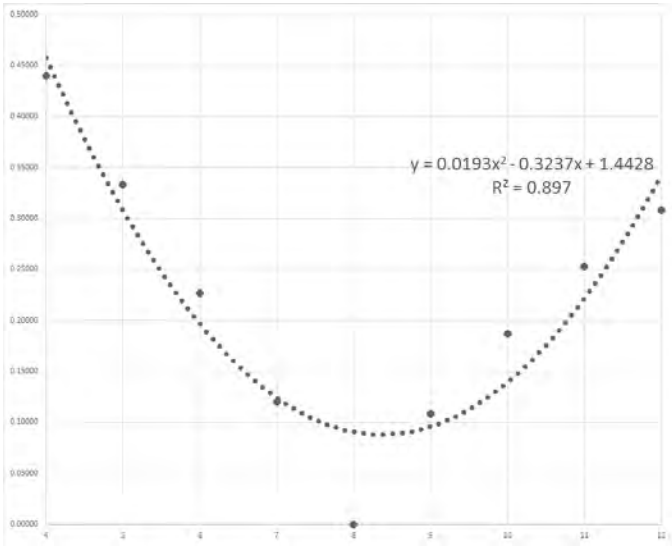


Figure 7.11: Goodness of fit graph for displayed images for grammar 0 for pass-image 8

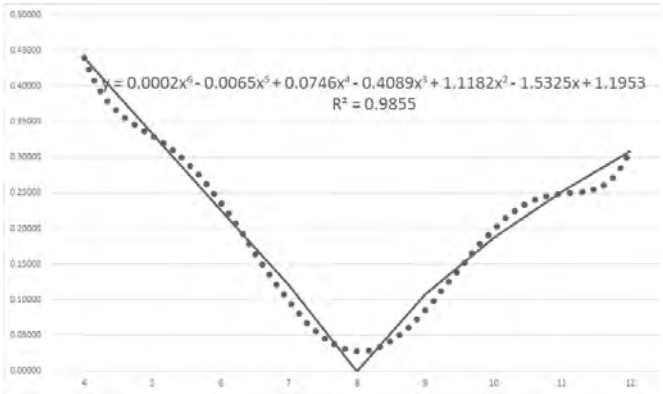


Figure 7.12: Goodness of fit graph for displayed images for grammar 0 for pass-image 8 and polynomial order 6

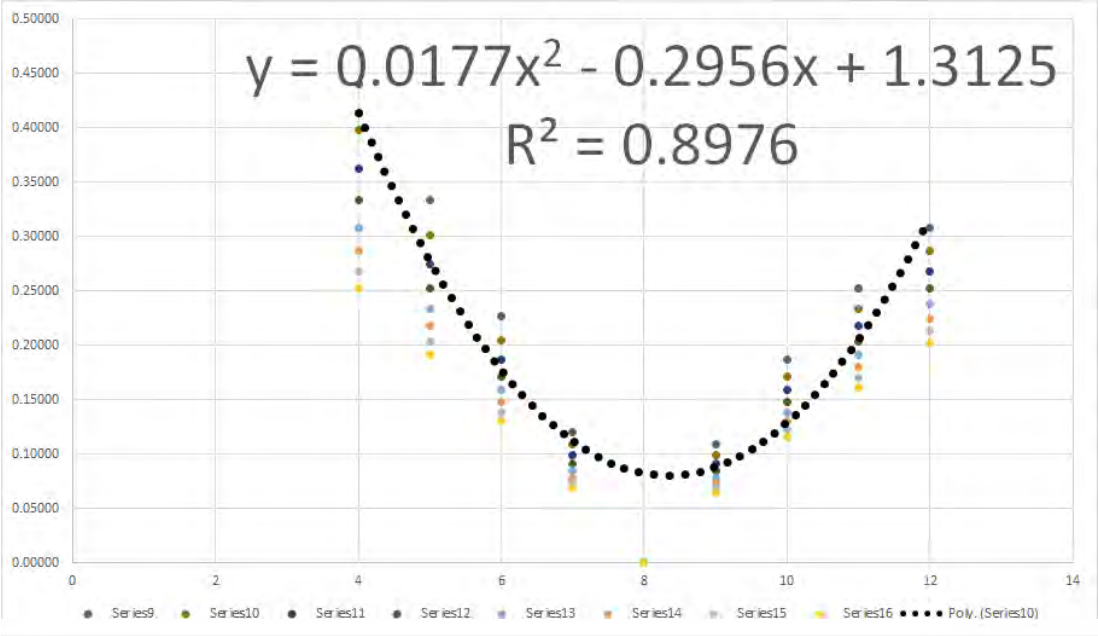


Figure 7.13: Goodness of fit graph for displayed images for grammar 0 for pass-images 9 to 16

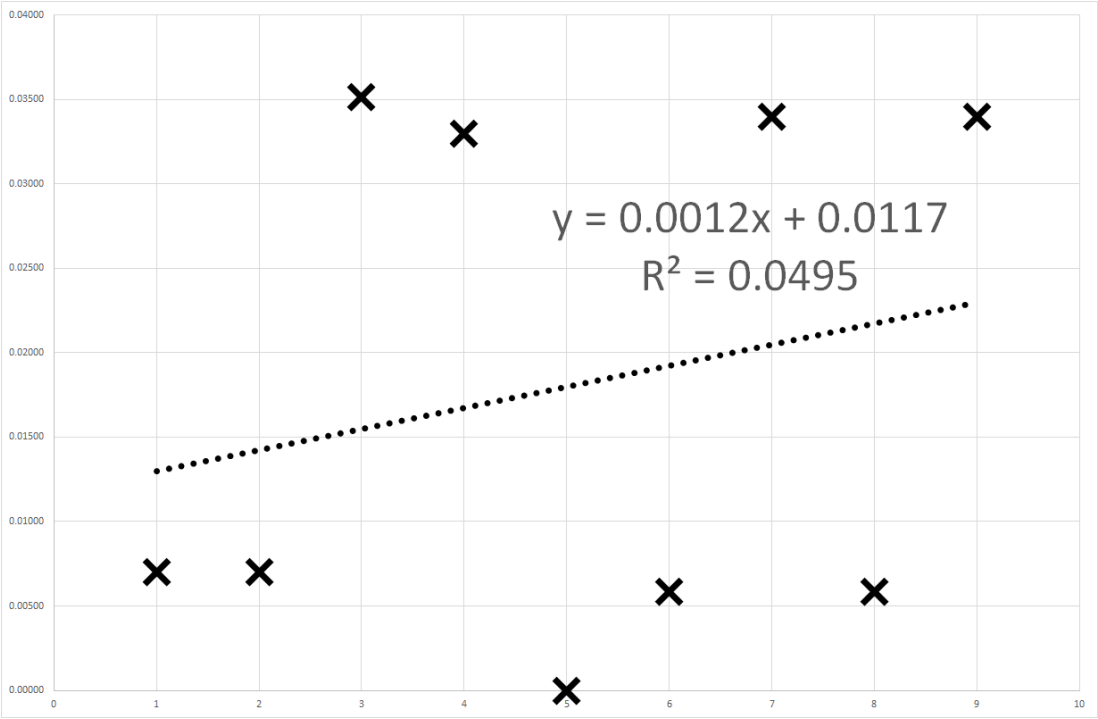


Figure 7.14: Goodness of fit graph for displayed images for grammar 21 for pass-image 5

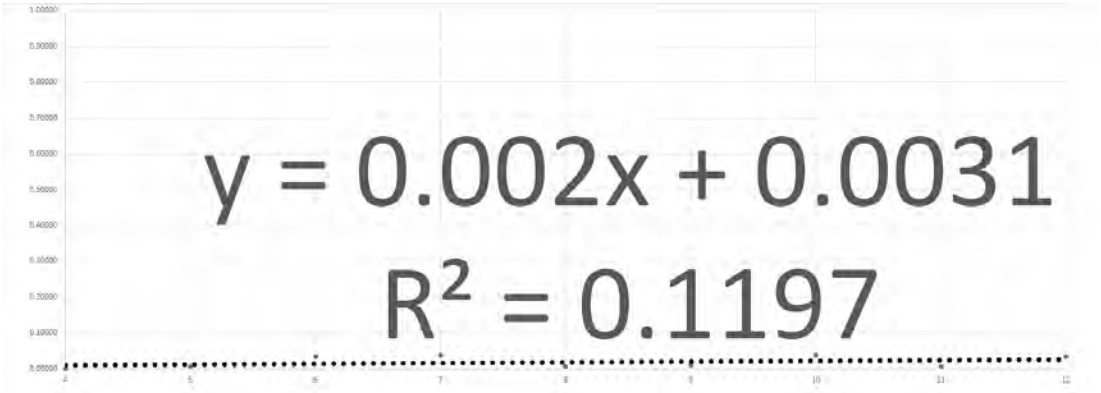


Figure 7.15: Goodness of fit graph for displayed images for grammar 21 for pass-image 8

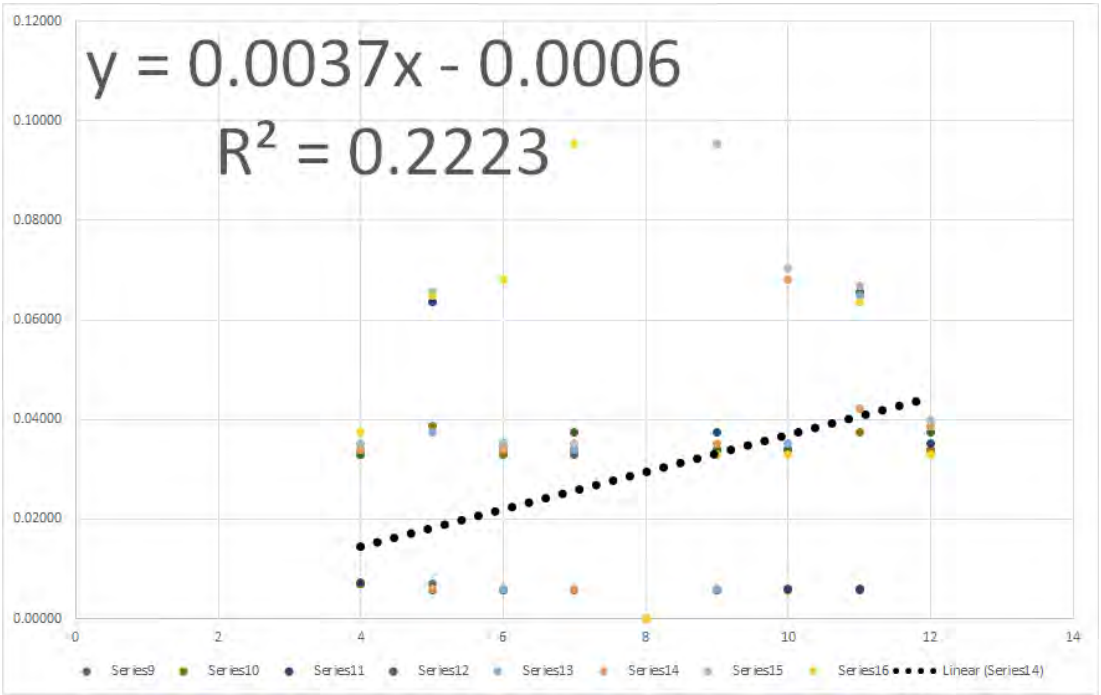


Figure 7.16: Goodness of fit graph for displayed images for grammar 21 for pass-images 9 to 16

7.3.5.2 Non-symmetrical grammars

A major limitation of this type of grammar is the tendency to repeat generated images since they are usually nondeterministic, and as such the same production rules can be applied repeatedly. Similar to what was done for symmetrical grammars, one approach to overcoming the repeating images limitation is to also increase the granularity of the production rules or to control the application of production rules through the use of regular expressions. Using this approach to solving the repeating images problem also increases the password space of the grammar. Increasing the granularity of the production rules will also speed up the system.

7.3.6 General observations for all the grammars

It is possible to deduce the following from the results of these experiments:

- For symmetrical grammars with a similarity convergence region, the best approach is to control the similarity of the distractor images is to choose an appropriate segment of the similarity convergence region to use in a visual password scheme. With these grammars the initial set of images generated tend to be less similar than successive images.
- For non-symmetrical tree grammars (see an example grammar in Appendix C.2), the best approach to controlling similarity of generated images is through grammar design refinements. The grammar designer should try to ensure that most or all of the images are generated at the same tree depth level. Images produced at different tree depth levels tend to be less similar than those produced at the same tree depth level.
- For all types of grammars, care has to be taken to ensure that each application of their production rules produces the smallest possible change in both the syntax tree and visual image. The implication for symmetrical grammars is that more images can be produced in the similarity convergence region and the resulting distractor images will be more similar. For non-symmetrical tree grammars, the similarity threshold values will be more uniform and will therefore also produce highly similar distractor images.

7.4 Summary

The following is a summary of what was done to further increase the similarity of distractor images.

There are three possible ways (of which two are feasible) of applying the normalised tree edit distance threshold values for determining which of the generated images to use as distractor images for the various grammars:

- To design grammars such that successive iterations or applications of production rules make little adjustments to the pass-image tree i.e. at the leaves. This is easily done for some grammars i.e. grid picture grammars, but may be more difficult or impossible for others. A second requirement which must be guaranteed with this approach is that identical images should not be generated i.e. with identical trees. This second requirement may be more difficult to meet for some grammars (with non-deterministic) generated images.

- A second approach is to experimentally determine a normalised tree edit distance threshold value for all or each of the grammars. This threshold value can then be stored with the individual grammars in the worst case scenario or hard coded within the program. The problem with this approach is that a lot of pairwise comparison of image trees may have to be done to determine the optimal threshold value, and it may degenerate to a situation where each pass-image in a grammar may have a different threshold value. This approach is not feasible for the aforementioned reasons and thus was not used in this study.

- A third approach is both simple and elegant, and has the additional advantage of not requiring any modification to existing code. This approach is based on the observation that during successive generation of images by symmetrical grammars, there is usually a point where the images are no longer distinguishable to the human eye i.e. the images look identical. In other words, during successive generation of the images by symmetrical grammars, the images become more and more similar until they become indistinguishable to the human eye. This is noticeable in the sample image

generation sequence in Appendix C.1 (Barnsley fern). This point is referred to in this study as the similarity convergence point. Images close to this point are guaranteed to be similar while images generated during initial iterations of the grammar are usually not very similar. In this approach, images generated after the initial set of iterations and that are close to the similarity convergence point are used.

The value of 0.5 is midway between 0 (identical) and 1 (completely different), and was a tentative choice for all 24 grammars used in this work. However, this was found to be impractical, as different grammars had different similarity threshold values, to the extent that most of the symmetrical tree grammars had different similarity threshold values for different pass-images. The similarity threshold value can be adjusted upward or downward according to the deployment scenario. A downside of this approach is a slight reduction in the number of images available for use for each symmetrical grammar. Another possible effect could be a slight reduction in speed for some symmetrical grammars that have slow generating capacity in the similarity convergence region. With these three approaches, all four questions at the beginning of this chapter can be satisfactorily addressed for each application scenario.

The ideal grammar type to use with a visual password scheme is the non-symmetrical grammar without repeated images in its generative sequence. This is because all the images generated can be used right from the very first image generated (no similarity convergence point) and the images generated also tend to have almost equal similarity measure values, hence their tendency toward producing straight line graphs when their similarity values are plotted. This is the most important result from this aspect of this research.

Table 7.5 on the next page summarises the results of the characterisation experiments done in this chapter. These results may also generalise to other grammars not used in this study.

Further work in this area could be the application of these observations to the deployment of visual password schemes in different application scenarios. For example, slightly less similar distractor images could be used for more trivial authentication scenarios, like access to grocery shopping list, birthday information and other less important (to the user) information. More similar distractor images

Table 7.5: Grammar characterisation summary.

Type of Grammar	**Best approach to generating increasingly similar images based on characterisation experiments**
Symmetrical	1. Use images in the similarity convergence region that are increasingly close to the similarity convergence point
	2. Iterative refinement of production rule granularity of the specific tree grammar
	3. Grammar design with a view to generating highly similar images for use in a visual password scheme
Non-symmetrical	2 and 3 above
Hybrid	3 above

could be used for critical resources, like banking details, access to nuclear plants and other very important company information.

The next chapter tries to put the entire research work in proper perspective.

7. EXPERIMENTAL CHARACTERISATION OF TREE PICTURE GRAMMARS USED IN A VISUAL PASSWORD SCHEME

Chapter 8

Research Discussion

8.1 Introduction

This chapter discusses the strengths and weaknesses of the research with respect to security and usability. We also compare the prototype developed in this study with the commercially successful Passfaces (Realuser, 2015). In this section, we shall also consider how the research addresses the main limitations of this category of visual password scheme.

The following features designed and implemented in this prototype are used to overcome these problems.

- Alternate passwords. This feature is peculiar to visual password schemes. The possibility of having 24 pass-images, with the potential password space of 9^{24} (each screen contains nine images repeated 24 times), while using a minimum of any three pass-images during a given authentication session is unique to Passfaces style of visual password scheme. Alphanumeric passwords characters cannot be alternatively used in this manner when entering passwords..

- The provision of self determined number of authentication rounds, and the ability of the user to change it at will, makes the system more flexible.

- The fine tuning of the degree of similarity of the images by using grammars with images close to their similarity convergence point or by appropriate

grammar design.

- The display of images that do not contain the pass-images after a wrong choice of pass-image and the use of different random selections of grammars subsequently.

- The use of keyboard as an additional input source for selection of pass-images.

We shall now discuss how the above factors overcomes the following limitations in this category of visual password scheme.

8.1.1 Shoulder surfing

The use of the recommended minimum of 12 pass-images with four rounds of authentication, means that the user can use the system for over a year without repeating the same set of images (495 pass-image sets and assuming one login per day). If however, the user uses the full set of n=24 grammars with r=4 authentication rounds, the user can use the system for several years without repeating the same set of images (10626). The computation for the possible combinations, $C(n, r)$, is shown in Equation 8.1.

$$n \; choose \; r = C(n, r) = n!/(r!(n-r)!) = 24!/((4!(24-4)!) = 10626 \qquad (8.1)$$

If the number of authentication rounds is increased to five or six (r=5 or 6 respectively in Equation 8.1), the sets of images that can be displayed increases to 42504 and 134596 respectively. This effectively defeats shoulder surfing and video recordings that could take place over several authentication sessions, as the would-be shoulder surfer would not see the same set of pass-images. The selection of highly similar distractor images through the use of Bottom-up tree edit distance and the additional methods of fine-tuning the similarity of distractor images observed in the last chapter is a further deterrent to shoulder surfing. Also, the use of keyboard for pass-image selection is an additional measure against shoulder surfing.

8.1.2 Description attack

The use of 24 pass-images with at least four rounds of authentication is recommended for medium to high security environments. Since any of the 24 pass-images could be used during a login session, the user will need to be conversant with all pass-images. The need to describe 24 pass-images is enough of a deterrent to description attack. The use of images in the similarity convergence region, also makes the images similar enough that any description of a pass-image will match at least one of the displayed distractor images. The more similar the images are, the more the number of images that will fit a given pass-image description.

8.1.3 Bias in image selection

The use of highly similar distractor images in the prototype resulting from the incorporation of

- the most accurate similarity measure from a human point of view, the Bottom-up tree edit distance, in the selection of distractor images, and
- the use of images from the similarity convergence region or grammar design

makes the system resilient against bias in image selection. The fact that the images are all abstract, eliminates most of the factors responsible for poor image selection, found in the traditional Passfaces type of visual password scheme.

8.1.4 Password space

The use of 24 pass-images increases the virtual password space considerably (9^{24}). Since any of the pass-images can be used for authentication, any would-be attacker must have the knowledge of all 24 pass-images to be sure of authentication. The ability of the user to determine the number of rounds of authentication also means that the user can increase the password space as needed. The password space can also be increased by increasing the number of displayed images on the screen. For example, for four rounds of authentication, increasing the 3×3 on screen grid to either a 4×4 or 5×5 grid will increase the password space from 9^4 to 16^4 and 25^4 respectively.

The user is allowed three attempts at authentication during any login session. This implies that for a four round unsuccessful login attempt, where the user is presented with a different grammar from the grammar gallery during each authentication round, the user is confronted with a password space of 9^{12} i.e nine images displayed on the screen three times repeated four times $(9^3)^4 = 9^{12}$ [1]. In other words, the password space increases during a failed login attempt. Also, the fact that during each round of authentication, a wrong choice of pass-image results in the display of subsequent distractor images without a pass-image, further increases the password space of the system.

8.1.5 Infrastructural and network limitations

The research has developed a stand alone system that is installed on the client's computer. The generated images are neither stored nor transmitted over the network at any time before, during and after authentication. This eliminates the need for large picture databases and high network bandwidth needed for current systems.

8.1.6 Internet and large client/server environment

Issues not directly addressed in this research are additional considerations that must be addressed in a client/server or Internet based environment. Also, usability issues were not empirically tested. These issues are discussed below.

- Two way authentication. This is where both the server and the client computers verify each other's identity.

- Two factor authentication. This is where a second authentication method, not from a knowledge based authentication system, is used along with a knowledge based one like our prototype visual password scheme.

[1] see Section 8.2.3 for further discussion on computation of password space

8.2 Security

This section discusses the security features currently implemented in the prototype and the additional security measures that could easily be implemented, when required, in a high security installation.

8.2.1 Shoulder surfing

Here we shall consider aided and unaided forms of shoulder surfing.

8.2.1.1 Shoulder surfing by simple observation

The four main ways of countering shoulder surfing in visual password schemes are:

- Reducing visibility. This is what is being done in shops where points of sale terminals have keypads with raised sides to protect the input PINs from being seen by would-be shoulder surfers. This was not implemented in the prototype as it will seriously affect usability. This could involve darkening the screen to reduce visibility of the images to everyone but the user sitting directly in front of the screen. Another approach could be to display a slightly transparent form of the pass-images within other images or coloured backgrounds in other to reduce visibility of the pass-images to everyone but the user sitting directly in front of the computer system.

- Subdividing actions. This was implemented in the prototype with the users being required to simultaneously select a letter and a number to indicate the row and column position of the pass-image respectively. The objective is to divide the attention of the would-be shoulder surfer between the two hands at the keyboard and the screen.

- Dissipating attention. This method involves confusing the shoulder surfer by making him concentrate on too many action points. With the availability of multi-touch touchscreens on laptops and tablets, the use of three fingers of both hands, to indicate the row and column numbers of the pass-image, could be implemented. In this approach, the fingers of both hands

could be placed on appropriate positions on the screens, and the row and column positions of the pass-image could be indicated using the appropriate fingers of both hands. This dissipates the attention of the would-be shoulder surfer, because the application of finger pressure on the multi-touch screen, produces involuntary movements in other fingers that prevent the identification of the actual finger pressed. This was not implemented in the prototype because the system test-bed does not have a multi-touch touch-screen and the system is meant to work on different devices with different input capabilities.

- Transforming knowledge. This involves entering credentials in isolation in a way that is difficult to reconstruct even after observing a successful login. This was not implemented in the prototype, but could be easily added and is further discussed in the next section.

8.2.1.2 Shoulder surfing through the use of image capture devices

A security issue addressed by the developed prototype is the ability to steal pass-images by taking a picture of the login screen or by video recording it. This security threat is applicable to both alphanumeric and visual password schemes. The risk is all the more prevalent by the availability of smaller cameras and video recording function on most smart phones and tablets today. One way of addressing this vulnerability is by using one of the methods for defeating conventional shoulder surfing: transforming knowledge. For example, if the user chooses six pass-images, out of which only three randomly picked pass-images are used for any one authentication session, the result is $C(n, r)$ possible combinations or 20 sets of pass-images. The computation is shown in Equation 8.2.

$$n \; choose \; r = C(n, r) = n!/(r!(n - r)!) = 6!/((3!(6 - 3)!) = 20 \tag{8.2}$$

This means that the authentication will not be compromised after a few recording sessions. This will also have the same effect on conventional shoulder surfing. This will make it more difficult to obtain all six pass-images using any video recording device, after just a few sessions, and provides an extra layer of security to the

system. Also, one could append the day of the week or month to the keys pressed when picking the pass-image or username. The video recording attack will have to be mounted at least 20 consecutive times in order to obtain all pass-images.

Another approach not directly addressed in this study is to use what is referred to as two factor authentication. This is when two completely different authentication methods are used in one login session. The prototype implements what can be referred to as a pseudo two factor authentication system. This is because our prototype hides the username during login, and as such it is a thinly disguised alphanumeric password scheme, which acts as a second authentication method. The reason for qualifying this approach with the word "pseudo" is because purist of two factor authentication will argue that both visual and alphanumeric password schemes come from the same class of knowledge based authentication method. Therefore, using a password like string as a username will provide an extra layer of security. This means that the would-be attacker must have knowledge of both the username and all pass-images to successfully access the system. Note that this suggestion will affect user registration time.

However, in a large networked environment or the Internet where the security threat is greater because of resource exposure to many more people, a better approach would-be to additionally use a One Time Password (OTP). This is where a user is sent an OTP via his phone when authenticating, in addition to the normal authentication process, and is almost mandatory for most applications by financial institutions today. This approach is therefore recommended to designers deploying this prototype in very mission critical or financial environments.

8.2.2 Social Engineering

Social engineering deals with the compromise of authentication systems that can take place in social environments. These are discussed below.

8.2.2.1 Divulging pass-images through description

The prototype system developed did not completely address one of the most desirable properties of an ideal visual password scheme: the inability to divulge pass-images by describing it to colleagues or friends. This implies that the images

are either very difficult or impossible to describe. This is to make it impossible for colleagues to share pass-images with other colleagues either willingly or under duress in high security risk situations or installations.

The greatest issue arising from this research is the dichotomy between easy to recognise and easy to describe pass-images. This is akin to the problem of long and difficult to remember passwords versus short and easy to remember, but less secure alphanumeric passwords. The prototype developed used some of the grammars from the literature deliberately, so as to cover a broad range of tree grammars with a view to determining whether some types of tree grammars were more suited to this project than others. The grammars used in this prototype were of a mixed nature, from difficult to describe generated images to easier to describe ones. This observation is especially true for most of the grammars picked from the literature. The implication of this observation is that it is better to use grammars specifically designed for visual password schemes, or at the very least modify the grammars picked from the literature, to suit the needs of a visual password system. Some of the grammars while being easy to remember are also quite easy to describe. For example, one can count the number of basic shapes in the image or the number of objects in the image i.e. snail images. The more difficult to describe images may also be difficult to distinguish between and recognise.

While the system is relatively easy to use when using the easy to describe images as pass-images, the question remains as to whether the system will be as easy to use with some of the more highly difficult to describe grammar images. This leads to the perennial debate about balancing ease of use against security of the authentication system. How long will it take to train users to use a visual password authentication system with difficult to describe pass-images? What is the long term memorability of such difficult to describe pass-images? Will users readily accept such systems? These are some the questions that have to be answered before deploying such systems.

The prototype developed can therefore be used with only grammars developed specifically to address the dichotomy between difficult to describe and easy to remember pass-images. One way of achieving this will be to conduct a survey in which the images generated by the tree grammars are described on a recording

device and the users are then asked to identify the images after listening to the recorded description. The success or failure rate with the images generated by each of the grammars will enable the grammar designer to identify which grammars are more or less susceptible to sharing or communication.

One way to address the description issue is to set a limit on the amount of description time or length needed to accurately describe an image for someone to be able to recognise the pass-image when it is displayed on the screen. The grammar designer, in designing additional grammars for the prototype, should therefore:

- Choose grammars that are not easy to describe, for example, count the number of elements (i.e. circles, blocks) in an image. This can be done by setting the image description length to, say a minimum of 150 words, to accurately describe an image enough to be recognisable by other people.
- Ensure that the images are similar enough that any description that is less than the predetermined length will match at least two of the displayed images.

Another approach to tackling the problem of sharing pass-images that can be easily implemented in the prototype is to divide the grammars into at least two groups from moderately easy to describe to difficult to describe ones. The users can then be made to select at least one pass-image from the difficult to describe group during registration, in much the same way that some authentication systems force the user to incorporate special characters in their passwords.

A study (Dunphy *et al.*, 2008) on the vulnerabilty of Passfaces to social engineering (description) attack showed the importance of choosing the right distractor images. The study showed that the more similar the distractor images are to the pass-image, the less successful this form of attack is. This underlies the importance of using the right similarity measure in our prototype and granularity of the grammar production rules.

8.2.2.2 Phishing

This is only relevant if the prototype is deployed on the world wide web. This is the situation where the would-be attackers present the user with a fake website for

authentication, in order to steal the pass-images of the user. This approach has not been reported to have worked on Passfaces style visual password scheme and cannot succeed with the prototype because of the following additional capability built into the prototype.

Firstly, during authentication the position and a representation of the pass-image is never transmitted over the network, since the tree edit distance computation is done on the client's machine. Secondly, since the pass-images are generated on the user's computer, there is nothing to steal, as the only information passed back to the server is whether or not the user passed authentication. Thirdly the fake website will have to be able to generate all the grammars and pass-images needed for authentication by the user, which implies a foreknowledge of the user's pass-images. However, the username is not protected against this form of attack, and as such a two way authentication method is recommended in addition to the pseudo two factor authentication implemented in the prototype. Two way authentication is a bidirectional authentication process where the user's computer also authenticates the server during the process of user authentication. Two way authentication is also implemented in the current version of Passfaces for finance. The reason for not implementing it in the developed prototype is because it requires a client server environment.

8.2.2.3 User input simulation

While it is conceivable that a program could be created to simulate mouse clicks on the displayed images in order to gain authentication, this is highly unlikely to succeed against Passfaces style of visual password schemes and less so against our prototype because of the security measures against guesswork implemented in the prototype, which can be read in the guesswork section below.

8.2.3 Guessing

The theoretical password space for a Passfaces style visual password scheme is N^k, where N is the number of images displayed on the screen and k is the number of authentication rounds. The prototype has $N = 9$ and $k = 3$ which gives an apparently small password space of $9^3 = 729$. For alphanumeric passwords, the

password space is 94^L, where 94 represents the number of printable characters on the keyboard, with the space key excluded, and L is the length of the password. Since the prototype uses both forms of authentication in our version of a two factor authentication, the actual minimum password space for our system is 729×94^L, where L is the length of the hidden username used by the prototype.

The phrase "minimum password space" was used because of the additional security provided by the system. On choosing the wrong pass-image, subsequent displays of generated images will not contain the pass-image, instead a randomly selected set of images is displayed. This alerts the legitimate user that he made the wrong choice of pass-image in the previous step, but will deny the would-be attacker the ability to further guess at the pass-images on the system. For example, if the would-be attacker successfully guessed the first pass-image, a wrong guess of the second pass-image will produce a randomly selected set of images on the third screen, leaving the attacker with uncertain knowledge of only the first pass-image after an unsuccessful login. Since the attacker is denied access, he may not even be sure which of the pass-images he guessed right or wrong. Also, the random display of images on the screen implies that the attacker cannot just click on all nine image positions on the screen, one after another, with the hope that after nine attempts, he would have covered all nine images, out of which one must be the correct pass-image. This is more so, since the attacker does not even get feedback on whether or not he has selected the right pass-image. This method of attack will not succeed against our prototype. This is because of the aforementioned fact that one wrong guess will virtually end the authentication session, although the attacker will not be aware of this until all three rounds of authentication have been completed.

The idea of randomly displaying the same set of images on the screen also protects the system against a form of guessing attack that is referred to as intersection attack in the literature. This is where a user can correctly identify a pass-image by looking out for the image that is always appearing on the screen during several authentication sessions either through simple observation or the use of a recording device. We are therefore confident that our prototype is protected against intersection attacks.

8.2.4 Operating System and infrastructural vulnerabilities

This section covers facilities that should be provided by either the underlying operating system or network to ensure that the system is more secure.

8.2.4.1 Local access

The prototype developed does not make use of operating system privileged execution mode normally reserved for operating system drivers and kernel services that need to be protected from user modification or access. This means that the system is not protected from other processes running in operating system user space. It is therefore possible for a hacker to write programs or use code injection tools to access the code or data segments of the prototype. Thus it may be possible to hack into the running program and trick the system into accepting that a successful authentication has taken place when in fact it is the program that has been compromised. It is also possible for a hacker with access to the executable code either on disk or in the main memory to profile the execution of the prototype code in order to perform code injection and gain access to the resource. This is only important when gaining access to a local resource on a computer system. Operating system privilege level execution is needed to protect our prototype visual password scheme from being interfered with by hackers on user machines. Although this mode of attack requires expertise beyond the ability of most users and is highly unlikely, it is still a distinct possibility that should be guarded against.

8.2.4.2 Internet access

One of the strengths of the developed prototype is the ability to download it into the user's system and execute it using either Java web start or as an applet. Although the ability to run as an applet is not yet implemented in the system, it is quite easy to extend the prototype in this direction when needed.

8.2.5 Other attack vectors

Other vulnerabilities are Brute force attack, Dictionary attack and Spyware (key-logging) attack. These have not been seen as a threat to the Passfaces style of visual password schemes.

8.3 User Experience

Usability consists of several attributes, according to an ISO standard (Jokela *et al.*, 2003; Towhidi & Masrom, 2009), with implications for visual password schemes. The usability factors were categorised into satisfaction, effectiveness and efficiency. We shall now discuss the usability of the prototype with respect to this standard. We shall also compare the developed prototype with Passfaces and other relevant password schemes.

8.3.1 Satisfaction

This refers to the sub-factors that contribute to how happy the users are with the system. These satisfaction sub-factors are discussed below with respect to our prototype.

8.3.1.1 Ease of use

A visual password scheme should be easy to use. This means that creating and selecting pass-images should be easy for the user. This also implies multiple easy means of selecting pass-images, whether through the use of multiple fingers of both hands in multi-touch systems or through the use of the mouse or keyboard. Questions to be addressed here are:

- Is it easy to register your pass-images?
- Is it easy to select your pass-images using either the keyboard or mouse?

For this sub-factor, the facilities provided by the prototype are as good or better that Passfaces.

8.3.1.2 Ease of creation

Here we look at how easy it is to create passwords. The questions to be addressed here are:

- Are the registration instructions easy to understand?
- Are the registration instructions easy to follow?
- How easy is the registration process?

The registration process should be fast and simple to the user. In the earliest incarnations of Passfaces, the users were allowed to select their own Passfaces. There was a tendency to select faces from their own races or attractive faces. These made these incarnations of Passfaces vulnerable to guessing attack.

In the current version of the commercial Passfaces visual password scheme, the system chooses pass-images for the user, and as such, the user has to be trained to recognise the pass-images through a familiarisation process where the user has to go through a simulated login twice for the registration to be regarded as successful. This feature is supposed to prevent the user from selecting predictable pass-images, but this measure increases registration time and places an additional burden on the ability of the user to recall the pass-images. In contrast to this, the developed prototype allows a user to choose his own pass-images with a resultant short registration time and a more user friendly system. However, further studies may be needed to determine whether users may prefer either some tree picture grammar images over others or some of the images of a particular tree picture grammar over other images of the same tree picture grammar, so as not increase the vulnerability of the system to guessing attacks..

The implemented selection of six pass-images during registration in our prototype may have a negative impact on user registration time and may be inconvenient to the user. This is however recommended as it is only done once and has the potential to greatly increase the security of the system.

8.3.1.3 Memorability

In this attribute, consideration is given to how easy it is to recognise the pass-images, both in the short and long term. Emphasis is on how colourful, clear and

appealing the images are. Dull or black and white images may be unappealing to the users, which may lead to a natural resistance to remembering the images. The questions of interest here are:

- Are the pass-images easy to memorize?
- Can the pass-images be easily remembered after a period of not less that six months?

8.3.1.4 Ease of execution

Using the system must consist of simple steps or activities. The relevant questions here are:

- Is it simpler or faster to execute when compared to similar systems?
- Which login process, out of multiple alternative login processes, is simpler or faster?

8.3.1.5 Execution speed

In this case we are concerned with the number of training sessions the user needs to go through to successfully make use of the system. Also, the registration and login processes should not be too long, otherwise it might be easier for the shoulder surfer to identify the pass-images. The questions of interest here are:

- Which system, out of multiple comparable systems, provides faster registration?
- Which system, out of multiple comparable systems, provides faster login?

The login time for the prototype is comparable to or better than the average for the Passfaces style of visual password schemes with stated login times in the literature. However, it is still slower than that of most alphanumeric based password schemes.

8.3.1.6 User interface

This involves the use of high quality coloured images that are more attractive and acceptable to the users as implemented in the prototype. A good user interface for a cognometric visual password scheme must use attractive images in order to be rated highly in it's user interface design.

The user interface of the prototype could be further improved for multi-touch systems by allowing the use of multiple fingers of both hands for authentication. This is because the use of the mouse for selecting a pass-image is one of the weakest links in the security of a Passfaces style of visual password scheme. Although the prototype allows the use of the keyboard when selecting pass-images, it could be further extended with the addition of randomly generated one time passwords for each of the nine images displayed on the screen. The length of this password should depend on the security context. A random password length of two or more is usually enough to provide enough security.

The use of the mouse should be especially discouraged in a social environment, as it greatly increases vulnerability of the system to shoulder surfing. Passfaces allows the use of the numeric keypad to select the pass-image from the nine displayed on the screen by using the numbers 1 to 9. On the other hand, our prototype allows the simultaneous use of both hands to indicate the row and column positions respectively of the pass-image. The use of the keyboard may not be attractive to some users of the system.

8.3.2 Effectiveness - Reliability

The question of whether the users will willingly trust and rely on the system for authentication in the real world touches on reliability, and is the main determinant of the system's effectiveness.

The system is highly reliable and scalable as discussed in Chapter 6. However, use in an internet environment will mean that it has to be deployed either as an applet or by using Java web start in user machines. This means that the authentication string has to be stored on the server because of restrictions on Java applets accessing client's hard drives without express permission from the underlying operating system. The reliability in this context will depend on the

reliability of the underlying network infrastructure.

8.3.3 Efficiency - Acceptability

Acceptability refers to whether the users will willingly agree to the deployment of the system from both usability and security points of views.

8.4 Prototype Overview

The prototype developed in this study compares favourably with the commercially successful Passfaces. The author did not have access to the Passfaces source code, since it is a closed source copyrighted authentication system, and as such had to make do with the free demo available on their website to try out their system. The registration time for Passfaces was between three to five minutes, which was also confirmed on their web pages, which gives an average registration time of about four minutes. The prototype, on the other hand, has an average registration time of 40 seconds. However, the login time for both systems was about 23 seconds. Thus the usability of the prototype is better than that of Passfaces in terms of being easier to create and will therefore be more acceptable to the average user.

Human faces are generally easier to share through verbal description than abstract images (Dhamija & Perrig, 2000). The prototype can be made invulnerable to this method of attack if the easy to describe pass-images from the relevant tree picture grammars are replaced with less easy to describe ones. Generally, the problem of whether easy to recognise equals easy to describe pass-images can be solved by carefully designing grammars for the prototype and then conducting the required survey. Another strength of the prototype is that the grammars can be changed even while the system is running. In other words, the grammars are hot swappable and can be changed on the fly. It is even possible to deploy different grammars on different user systems, such that no two users have the same set of grammars used for authentication, if the collection of tree picture grammars is large enough.

Most of the images generated by the system are very colourful and have high aesthetic appeal. However, further work has to be done on the images to find the

most acceptable grammars to use for the prototype.

A weakness of the system is that it will have to be modified to run as an applet in an internet based environment, since this feature was not implemented due to time and infrastructural constraints.

8.4.1 Grammar design

The design of grammars for our prototype is a very important aspect of this study. The grammars used in our prototype should be specially designed for the system, unlike the current prototype that has a mixture of grammars designed specifically for the system and those that were not. The design of the grammars should therefore strike a balance between difficult to describe images and easy to remember images. Ideally the grammars meant for the system should be tested on users with regard to both long and short term memorability, ease of use and visual appeal.

The images from the symmetrical grammars used for authentication should be close to the similarity convergence point for high end security systems. To ensure that there is a large number of images in this similarity convergence region, the production rules should make as little change as possible to the images during each iteration. This granularity of production rules will also increase the available password space. Therefore the two most important issues in grammar design are choice of grammars and granularity of production rules.

The right choice of grammars will produce bright and colourful images with good visual appeal, which will have an impact on memorability, ease of use and have the right balance between difficult to remember pass-images and easy to describe ones. The right level of production rules granularity will ensure that a lot of similar images can be generated, which will increase the images available to increase the password space and help defeat shoulder surfing and description attack. This is because any attempt to describe the images will result in such descriptions being applicable to a number of the generated images.

8.5 Research Overview

This research has raised a number of issues and further questions. These are addressed in this section.

8.5.1 The use of tree grammars

The questions to be addressed in this section are:

- Was the use of tree grammars a good decision or not?
- Would the researcher have used a different class of grammar, if the study was to be done all over again?
- Would any grammar be good for a visual password scheme?
- What other picture grammars could other researchers try?

The use of tree grammars was a good decision to use for this study. This is because the vast array of tree grammars available are enough to implement a visual password scheme. They allow the generation of appropriate images for use in a visual password scheme by the use of appropriate control mechanisms, both those inherent in the grammar, like the use of nested tables and those that are not (like the use of regular expressions and random generation modes). They range from simple context free grammars like regular tree grammars to the more powerful branching tree grammars with nested tables. Also the implementations had the ability to control the generation of images through the use of regular expressions. The characteristics of a good grammar for a visual password scheme are as follows.

- They should not require too much computational power to generate images in successive iterations, otherwise the user may have to wait a long time to see the generated images in the display grid.
- They should display clear and distinguishable images at low resolutions. This is important if lower resolution images are to be transmitted over a network in a large network or internet environment.

- They should provide mechanisms for fine tuning the appearance of the generated images. This was accomplished by the current set of grammars through the use of production rule granularity [1], nested tables, similarity convergence point[2] and regular expressions.

- They should be easily implemented in the system in such a way as to prune derivation trees that are dead ends i.e. will not result in an output image.

Also, the observed behaviour of symmetrical grammars [3] with regards to the similarity convergence point can be further used to fine tune the similarity of the generated images. However, there are also other tree grammars with the inherent ability to control the appearance of the generated images that could have been included amongst the grammars used in this study. Examples are Random context tree picture grammars (Ewert *et al.*, 2005) and Bag context tree picture grammars (Drewes *et al.*, 2008).

These grammars could be added to the system by extending the class treeGrammar. This is done by defining the appropriate parser syntax file in the directory parser and then running the JavaCC (Java Compiler Compiler^{TM}) program on it. The interested reader can check the JavaCC documentation for the syntax and semantics of its input files. A class specific method that must be implemented is the method currentTerm(). After this step, the would-be implementer will need to implement the following method in the algebra

```
protected Object apply(symbol op, object[] args)
```

which yields the value of an operation **op** applied to the given arguments.

[1]The granularity of production rules refers to changing the number and or order of production rules applied at any one time in order to cause the smallest observable change to an image.

[2]This is the point, after successive generation of images, where the images become indistinguishable to the human eye.

[3]Symmetrical grammars are grammars whose tree edit distance values tend to alternate between the tree edit distance values of distractor images immediately before and after the pass-image and form quadratic graphs.

8.5.2 Production rule granularity

The images generated by the tree grammars were controlled using production rule granularity as one of the options. The two methods used were nested tables and direct modification of production rules. A complete example of the steps in the process of transforming a sample grammar to the final image is shown in Appendices A.2.1 to A.2.6.

8.5.2.1 Granularity using nested tables

An example grammar is depicted below. This is a regular tree grid picture grammar.

```
generators.regularTreeGrammar:
( { S,A2,B2,C2,Ht,Hb,Vl,Vr,X1,X2 },
  { S:3,A2:7,B2:9,C2:7,Hb:3,Ht:3,Vr:3,Vl:3,X1:5,X2:5, sq:0, -:0 },
  {
    S -> S[A2,B2,C2],
    A2 -> A2[Ht,Vr,sq,Vl,Vr,sq,Vl],
    A2 -> A2[-,-,sq,-,-,sq,-],

    B2 -> B2[Vr,sq,Vl,Vr,sq,Vl,Vr,sq,Vl],
    B2 -> B2[-,sq,-,-,sq,-,-,sq,-],

    C2 -> C2[Vr,sq,Vl,Vr,sq,Vl,Hb],
    C2 -> C2[-,sq,-,-,sq,-,-],

    Hb -> Hb[Hb,Hb,Hb],
    Hb -> Hb[sq,-,sq],
    Ht -> Ht[Ht,Ht,Ht],
    Ht -> Ht[sq,-,sq],
    Vr -> Vr[Vr,Vr,Vr],
    Vr -> Vr[sq,-,sq],
    Vl -> Vl[Vl,Vl,Vl],
    Vl -> Vl[sq,-,sq]
```

```
   },
  S
)
```

To use nested tables, the grammar is promoted to an ET0L grammar and the production rules are divided into three tables as show below. Note the use of regular expressions to control the order of application of the production rules.

```
 generators.ET0LTreeGrammar:
( {S,A2,B2,C2,Ht,Hb,Vl,Vr },
  { H:1,S:3,A2:7,B2:9,C2:7,Hb:3,Ht:3,Vr:3,Vl:3,X1:5,X2:5, sq:0, -:0 },
  {
    {

    A2 -> A2[Ht,Vr,sq,Vl,Vr,sq,Vl],
    B2 -> B2[Vr,sq,Vl,Vr,sq,Vl,Vr,sq,Vl],
    C2 -> C2[Vr,sq,Vl,Vr,sq,Vl,Hb]

   },
   {
    Hb -> Hb[Hb,Hb,Hb]},
    {Ht -> Ht[Ht,Ht,Ht]},
    {Vr -> Vr[Vr,Vr,Vr]},
    {Vl -> Vl[Vl,Vl,Vl]
    },
    {
    Hb -> Hb[sq,-,sq],
    Ht -> Ht[sq,-,sq],
    Vr -> Vr[sq,-,sq],
    Vl -> Vl[sq,-,sq]
    }
   },
```

```
  S[A2,B2,C2], 1 2+ 3+ 4+ 5+ 6
)
```

A second example using a different form of regular expression can be seen.

```
  generators.ETOLTreeGrammar:
( { S,A2,B2,C2,Ht,Hb,Vl,Vr },
  { S:3,A2:7,B2:9,C2:7,Hb:3,Ht:3,Vr:3,Vl:3,X1:5,X2:5, sq:0, -:0 },
  {
    {
    A2 -> A2[Ht,Vr,sq,Vl,Vr,sq,Vl],
    A2 -> A2[-,-,sq,-,-,sq,-],
    B2 -> B2[Vr,sq,Vl,Vr,sq,Vl,Vr,sq,Vl],
    B2 -> B2[-,sq,-,-,sq,-,-,sq,-],
    C2 -> C2[Vr,sq,Vl,Vr,sq,Vl,Hb],
    C2 -> C2[-,sq,-,-,sq,-,-]
   },
   {
    Hb -> Hb[Hb,Hb,Hb]},
    {Ht -> Ht[Ht,Ht,Ht]},
    {Vr -> Vr[Vr,Vr,Vr]},
    {Vl -> Vl[Vl,Vl,Vl]
    },
    {
    Hb -> Hb[sq,-,sq],
    Ht -> Ht[sq,-,sq],
    Vr -> Vr[sq,-,sq],
    Vl -> Vl[sq,-,sq]
    }
   },
  S[A2,B2,C2], 1 2* 3* 4* 5* 6
)
```

If further nesting depths are required, then the grammar has to be promoted to branching synchronisation grammar.

8.5.2.2 Granularity using direct modification of production rules

The example grammar above can be granularised by direct modification of the production rules. Also note that in this case there was no need to use regular expressions. The images produced by this grammar are non-symmetrical[1] in nature and appropriate for use in a visual password scheme. This grammar was actually used as the second grammar in the developed prototype.

```
  generators.regularTreeGrammar:
( { S,A1,A2,B1,B2,C1,C2,Ht,Hb,Vl,Vr,X1,X2 },
  { S:6,C1:5,C2:2,C3:0, B1:6,B2:3,B3:0,A1:5,A2:2,A3:0,Hb:3,
  Ht:3,Vr:3,Vl:3,X1:5,X2:5, sq:0, -:0 },
  {
    S -> S[A1,A2,B1,B2,C1,C2],
    A1 -> A1[-,-,sq,-,sq],
    A1 -> A1[Ht,Vr,sq,Vr,sq],
    A2 -> A2[Vl,Vl],
    A2 -> A2[-,-],

    B1 -> B1[Vr,sq,Vr,sq,Vr,sq],
    B1 -> B1[-,sq,-,sq,-,sq],
    B2 -> B2[Vl,Vl,Vl],
    B2 -> B2[-,-,-],

    C1 -> C1[Vr,sq,Vr,sq,Hb],
    C1 -> C1[-,sq,-,sq,-],
    C2 -> C2[Vl,Vl],
    C2 -> C2[-,-],
```

[1]Non-symmetrical grammars used in this study have tree edit distance values of distractors that oscillates around a mean value and form straight line graphs.

```
    Hb -> Hb[Hb,Hb,Hb],
    Hb -> Hb[sq,-,sq],
    Ht -> Ht[Ht,Ht,Ht],
    Ht -> Ht[sq,-,sq],
    Vr -> Vr[Vr,Vr,Vr],
    Vr -> Vr[sq,-,sq],
    Vl -> Vl[Vl,Vl,Vl],
    Vl -> Vl[sq,-,sq],
    X1 -> X1[X1,Hb,Hb,Vl,Vl],
    X1 -> X1[sq,-,sq,-,sq]

     },
  S
)
```

8.5.3 Choice of similarity measures

There are four distinct hierarchies of tree edit distance measures found in the literature (Valiente, 2001; Wang & Zhang, 2001):

1. General tree edit distance measure,
2. Alignment tree edit distance measure,
3. Isolated subtree tree edit distance measure and
4. Top down tree edit distance measure.

when moving from the general (1) to the most restrictive (4). The general tree edit distance measure is used in this study along with a variant that uses edit costs equal to the number of child nodes called the Fanout weighted tree edit distance. The bottom up tree edit distance is a special case of the isolated subtree tree edit distance that is equal to the top down tree edit distance for isomorphic trees. A

number of tree edit distance measures in the alignment and isolated subtree tree edit distance measure categories could be used in further research. Also, the most restrictive tree edit distance measure, the top down tree edit distance measure, could be used.

The images used with the various similarity measures came from the online study. It should be possible for another researcher to use as many as possible of the images in the actual prototype, in another online perceptual similarity study. The more the number of images used in the survey, the better the reliability of the conclusion about the similarity measure that best models perceptual similarity. To implement new tree edit distance measures in VPS (the prototype), the implementation should be done in the distance package and referenced from MatchTree class in the executable package.

8.5.4 Good and bad images

In this section, we shall try to answer the following questions.

- What are good images?
- What are bad images?
- Can bad images be converted to good images by modifying the grammar?

Good images should be colourful and pleasing to the eye of the users. Most dull and complex binary images would-be bad images to use in a visual password system. Good images should have fast display times, while bad images will be slow to load and display on the screen. Good images should be clear and distinguishable at various resolutions. This is because the higher the resolution the larger the size of an image. This could become a problem in an internet based environment and also in systems without a powerful graphics card. There is also a possible conflict between how colourful an image is and its display time.

There are four classes of images that can be used in a visual password scheme.

1. Easy to recognise and easy to describe.
2. Easy to recognise and difficult to describe.

3. Difficult to recognise and easy to describe.

4. Difficult to recognise and difficult to describe.

The only one that is appropriate in a visual password scheme is option two, and can be considered to be good images. Good images can be obtained by;

1. Making the right choice of grammars with appropriate level of production rule granularity.

2. Using images close to the similarity convergence point for grammars that display this attribute.

3. Using picture grammars that provide additional facilities for control of picture generation like Random and Bag context tree picture grammars.

4. Using computer graphics generation algorithms and manually determining the good images.

In option one above, it is also possible to use images close to the similarity convergence point for symmetrical grammars, in order to ensure that good images are obtained.

8.5.5 Undone tasks

This section discusses issues not directly addressed by this research, in other words "what has not been done?".

The question of whether "easy to recognise equals easy to describe" was not directly addressed in this study and should be a major future research focus area. This actually refers to the problem of mining tacit knowledge (as opposed to formal, codified or explicit knowledge), which is the kind of knowledge that is difficult to transfer to another person by means of writing it down or verbalizing it. To this end, we would like to suggest that a pilot and full study, using the actual images in the prototype should be done to address this issue. As previously suggested, taped recordings of pass-image description should be given to other users to login and noting the number of successful logins to find out whether the above assertion is true or false. There is the example of collies, where owners

can recognise their dogs amongst hundreds, but may not be able to describe to another person how to do so. There is also the example of parents or siblings being able to identify identical twins easily but may not be able to easily describe the reason to others. In both of the above examples, familiarity with the object or person or image may aid in the ability to distinguish between the images. It may therefore not be surprising that familiarity with the images generated in a visual password scheme may make it easy for people to recognise difficult to describe generated images. This has been shown to be possible with training. For example, professional musicians can accurately distinguish between the sounds of various instruments playing but may find it difficult to explain the reason to others who are not knowledgeable in the field or are non-musicians.

This research was not designed for the internet or for multi-user client server environments or the next generation client server environment called cloud computing. For example, a multinational corporation with branches in multiple continents might not want to risk their authentication software falling into the wrong hands by allowing the download of their authentication software into the client's computer for authentication. Therefore a client server or distributed authentication system may have to be implemented, much like for large databases like Oracle or NoSQL systems. This was not the aim of the research and as such was not implemented in the system. Multiple authentication listeners will have to be created for use in large multi-user environment with this prototype. Each user can either have his own thread or a separate process can be created for each user. This will also require a multi-server system with associated architecture.

8.5.6 Research weakness

The number of images used in this research for determining the best tree edit distance measure may not have been large enough to be conclusive. A lot more tree picture grammar generated images from the prototype should have been used. However, the observation that tree edit distance measures give a better correlation with perceptual similarity than CBIR measures appears to be overwhelming enough to be regarded as conclusive. This is one of the strengths of this research.

8.5.7 Research strengths

The following are some of the strengths of this research.

- The vast array of tree picture grammars implementation used in this research can be considered a strength of this research. Regular tree grammars, ET0L tree grammars, Branching tree grammars with nested tables, Link grammars, Grid picture grammars, Collage grammars, Chain code, Turtle and other line drawing grammars are available for use in creating pass-images.

- The modular way the tree grammars were implemented makes it easy to include new grammars.

- Algorithms for eliminating dead ends in derivation trees were used in the prototype.

- The large number of similarity measure implementations used in this research can be considered as another strength of this research. Eight CBIR and eight tree edit distance measures were used in this study.

- The ease with which new tree edit distance values can be included as a result of highly modular implementations.

- The security and user interface enhancements over and above the state of the art Passfaces style visual password scheme and the ability to further enhance these features with little effort.

8.5.8 What could have been done differently?

The multithreading kernel for the tree grammars was designed to speed up execution of the system, especially in a multithreaded environment, and also to take advantage of the object oriented facilities available in Java. However, since this prototype executes actions sequentially, with no two threads actually being executed at the same time, this feature could have been removed to ease debugging and other problems associated with multithreading. However, for future use of

the prototype in multi-user environments, this feature could become indispensable as the threads could be turned to independent processes to service multiple users on multiple listener processes.

The use of context sensitive grammars, like random context tree grammars and bag context tree grammars, would have provided additional means of controlling the similarity of the generated images.

8.6 Summary

Analysing the strengths and weaknesses of the developed prototype boils down to the age old problem of balancing security and ease of use. Most of the enhancements or modifications in the security section will make the system more difficult to use. Hence, a balance must be struck between making the system more secure and or being more usable during deployment in any environment. The major attack vectors against this type of visual password scheme are susceptibility to description attacks, guessing attacks, shoulder surfing and software profiling attacks. In this chapter, we have taken a look at how the developed prototype defends itself against these attacks and what modification to make to the system in order to better handle these types of attacks. A comparison of the prototype with the Passfaces visual password scheme was also made, noting its areas of strengths and weaknesses when compared with Passfaces.

In conclusion, the decision of whether or not to use the system as is or to implement any of the additional security measures or user interface improvements should depend on further research and application deployment scenarios. The next chapter looks at how to extend this research in new ways and the resultant studies this would engender.

Chapter 9

Further Studies

9.1 Introduction

This chapter discusses further work that could be done in this area, and its relationship to the world of visual password schemes. The next section discusses how to make modifications to the prototype while still using it as a Passfaces (cognometric) type of visual password scheme. After this we discuss how the research could be used as Passpoints (Locimetric) and hybrid types of visual password scheme. The last section summarises and concludes the chapter.

9.2 The Passfaces type visual password scheme

Further studies based on the original prototype and alternative designs will be considered in this section.

9.2.1 Modifications to the prototype

The prototype could be implemented in an enterprise environment with a view to studying:

- Memorability: this is where users are brought back after a period of say six months or more to see how easy it is for them to remember these types of pass-images. Users could also be observed in order to know how difficult it

is for them to remember their pass-images by noticing how often they fail authentication before finally succeeding.

- Ease of use: various methods of pass-image selection vis-a-vis multi-touch, mouse and keyboard could be investigated with a view to finding out which approach is most suitable to the majority of users.

- Shoulder surfing susceptibility: by conducting experiments to find out how easy it is for shoulder surfers to remember one, two or all three of the pass-images when standing at various positions around the user during authentication.

- Varying the pass-image grid from the 3-by-3 grid used in this study to a 4-by-4 or a 5-by-5 grid, and studying its effect on the above three factors.

A comparative study could be done to compare our design approach with that of the traditional approach of transferring pictures of human faces back and forth over a network: in terms of speed, practicability for smaller computing systems and adaptability to and/or impact of display resolution on deployment.

The next section considers variations to the prototype that could be compared to the current study.

9.2.2 Alternative design prototype

The following variations of the visual password prototype developed in this study could be further investigated.

- A single image could be generated and various image effects could be applied to generate similar images by making small adjustments to the images to be used as distractors. Example effects are: sharpening/blurring, transparency/opacity, sepia, contrast, colour tinting or toning, embossing, image blending, etc.

- Another variation to using the same image, could be to use affine transformations like rotation, scaling, skewing, etc., to generate similar images to use as both distractors and pass-images.

- The use of coloured backgrounds for the generated image could further increase the resilience of the system to shoulder surfing. For example, using a pass-image with light blue background with the same image displayed against various shades of blue could further confuse the shoulder surfer.

- The use of tree grammars that generate grayscale or binary images could be used in smaller phones or tablets.

- With the field of CBIR coming up with newer and better similarity measures, new experiments could be conducted to find more suitable measures for abstract computer generated images. These measures could then be applied to the first two modifications to the prototype mentioned in this list, for the determination of the distractor images to be displayed. This is especially true when these images are generated using graphic transformation operations and not tree grammars.

- The use of specially designed grammars for the passfaces type of visual password scheme should be further investigated, in order to better understand and speed up the application of the best approaches to creating such grammars.

- Other methods of computer generated graphics, could be used in place of tree grammars and the results obtained compared to this one.

- Including this prototype in a boot loader program for operating systems authentication could be of some interest.

The next section considers how to modify the system to create a Passpoints type visual password scheme or hybrid systems.

9.3 Further studies in related visual password systems

9.3.1 Passpoint type visual password system

In a Passpoint type visual password system, preselected points or hotspots in an image are used for authentication. Tree grammars could be used to generate images for this type of visual password scheme. In this approach a single image with more detail could be presented to the user for pass-points selection. One of the problems with this type of visual password scheme, the selection of obvious hotspots, could also be reduced or eliminated by using this approach. Special tree grammars to generate images with a large number of interest points for selection could also be developed and investigated.

9.3.2 Hybrid visual password systems

This section covers various hybrid visual password systems that could also be further investigated. They are referred to as "hybrid" because they combine techniques from multiple visual password schemes or domains.

9.3.2.1 Partially opaque sub-images

In this approach, pass-images are actually embedded in other randomly generated images at lower opacity, such that only the user sitting in close proximity to the computer system can see the embedded images. This will make it virtually impossible for a would-be shoulder surfer to see the image. The optimal distance for the non visibility of the embedded images could be the subject of further experimentation or could be made system adjustable by varying the opacity of the embedded images.

9.3.2.2 Multiple pass-image selection

Here, multiple small images could be generated by tree grammars and used as icons. A large number of these icons could be displayed on the screen, along with multiple preselected pass-icons. The user is expected to trace a path on the

screen around all his pass-icons. The area drawn on the screen must enclose all the preselected pass-icons in order to pass authentication. This method was initially suggested to prevent a shoulder surfer from seeing a user click on a pass-image or pass-icon.

9.4 Summary

This chapter presents a bird's eye view of the possible areas of research that could be further pursued, as offshoots of this research. However, more variations could be thought of according to the interest, need and inspiration of the individual researcher.

The use of bag context grammars would, for example, provide additional means of controlling the appearance of the generated images. This should be an interesting area for further studies.

The next chapter summarises the entire project with a listing of significant contributions of this study to the computer science literature.

Chapter 10

Summary and Conclusion

Visual password schemes have gained great popularity in recent years due to their perceived advantages over alphanumeric passwords. They are resistant against stealing, divulging to colleagues, guessing, eavesdropping, dictionary attacks and automated brute force attacks.

This study uses the Passfaces type of visual password scheme. A major limitation of the Passfaces type of visual passwords is the limited number of images that are available for use, and also the fact that the selection of images may be biased by gender, race, appearance and family ties. To overcome these limitations, this research looked at using automatically generated abstract images as visual passwords. The fact that the images are highly similar and abstract eliminates both shoulder surfing and image selection bias. Their automatic generation also implies virtually infinite availability of images, thus greatly improving the potential password space or entropy.

The research was conducted in five phases. In the first phase, Chapter 4, we used tree grammars in the generation of abstract computer graphics and searched for suitable similarity measures. In the second phase, we conducted both a pilot and a full study to model the human idea of similarity vis-à-vis the generated abstract images. In the third phase, we determined which mathematical model of similarity ranking most closely models the human one and that is feasible to implement. The second and third phases are covered in Chapter 5. In the fourth phase, we incorporated this model into our tree picture grammars and then tested its applicability in a visual password environment (Chapter 6). In the

last phase, which was discussed in Chapter 7, we conducted further experiments to characterise our tree grammars and used similarity threshold values and the similarity convergence region to further fine tune the similarity between the pass-images and their distractors.

10.1 Contributions

At the end of the research, this study made significant research contributions to the following fields:

- Visual passwords: The use of syntactically generated images as visual passwords.
- Perceptual similarity: Modelling of human idea of similarity with respect to tree grammar generated abstract images.
- Modelling of perceptual similarity: Finding the similarity measure to use for abstract computer images that best matches perceptual similarity.

What follows is a more detailed list of contributions. To the knowledge of the researcher, this is the first time that:

- an online perceptual similarity survey was conducted with a view to determining how humans determine similarity of pictures generated by tree picture grammars,
- eight tree edit distance and eight CBIR measures have been used on pictures generated by tree picture grammars, with a view to finding out which one of the sixteen computer based similarity measures best models perceptual similarity,
- dynamically generated pictures have been used in a Passfaces style of visual password scheme,
- tree picture grammars have been used to dynamically generate pass-images and their distractors in a visual password scheme,

- tree edit distance measures have been used to determine similarity of images generated by tree picture grammars,
- tree edit distance has been used to determine selection of the most similar distractor images in a visual password scheme,
- comparison of CBIR and tree edit distance measures with each other in order to find out which group is better at modelling perceptual similarity,
- a collection of tree picture grammars (which include those designed by the author and others found in the literature) well suited for use in a visual password scheme has been assembled,
- the notion of the tree edit distance as a special case of Tversky's feature contrast model with a single feature space of node edit operations has been postulated,
- the success of tree edit distance measure in a wide variety of similarity applications in the literature has been attributed to the close relation between it and Tversky's model,
- a prototype visual password scheme that dynamically generates pass-images and their distractors using tree grammars has been successfully designed and implemented. The system also selects the most similar distractor images from a human point of view using a tree edit distance measure.

10.2 Conclusion

The novel visual password scheme developed in this study is suitable for use on desktops, laptops and tablets. It may still be useful on handsets with smaller screen sizes that have high screen resolutions and support Java, by using tree grammars that produce simple greyscale images.

The system developed is a novel one and addresses the five main issues with this category of visual password scheme identified in the first chapter (Section 1.7).

- The problem of limited potential password space is greatly reduced with this system as the number of syntactically generated images is almost infinite. As many images as are needed can be generated by the system. Also, the use of alternate passwords, pseudo two factor authentication, and the display of distractors without pass-image on wrong choice of pass-image increases the password space considerably.

- Susceptibility to description attack is greatly reduced in the prototype by the use of a large number of alternate passwords and highly similar distractor images.

- The use of automatically generated and highly similar distractor images is an effective deterrent to shoulder surfing as it may be very difficult for a shoulder surfer to identify and memorise which of the highly similar abstract images was the pass-image. The use of a large number of alternate pass-images, of keyboard and the hiding of username makes the system highly resistant to shoulder surfing.

- The type of images used in our prototype also greatly reduces the human tendency for biased pass-image selection, as the use of human faces as recorded in the literature has been found to lead to easily guessed pass-images due to selection bias. For example, people tend to select faces from their own race or background or attractive faces or the faces of models.

- The automatic generation of images obviates the need for storing large picture databases and the associated high bandwidth connection needed for passing images back and forth during authentication. There is also the associated security problem of storing pass-images directly. The system developed in this study does not store pass-images directly and does not need to pass images back and forth over a network as it is small enough to be installed on the local file system.

Areas of further research would be coming up with a detailed methodology on how to select or design grammars with less memory usage, which are more appropriate for smaller and less powerful handsets. Another useful research area

could be to determine which grammars produce more preferred and more easily remembered images and their characteristics. A study could also be done to determine actual effectiveness of this type of visual password scheme in a real corporate setting. Some of these ideas have earlier been discussed in Chapter 9.

References

Acosta, O., Fripp, J., Doré, V., Bourgeat, P., Favreau, J.M., Chételat, G., Rueda, A., Villemagne, V.L., Szoeke, C., Ames, D. *et al.* (2011). Cortical surface mapping using topology correction, partial flattening and 3D shape context-based non-rigid registration for use in quantifying atrophy in Alzheimer's disease. *Journal of Neuroscience Methods*. 29

Adams, A. & Sasse, M.A. (1999). Users are not the enemy. *Communications of the ACM*, **42**, 40–46. 10

Adjeroh, D.A., Lee, M. & King, I. (1998). A distance measure for video sequence similarity matching. In *Proceedings of the International Workshop on Multi-Media Database Management Systems*, 72–79. 81

Agresti, A. (2010). *Analysis of ordinal categorical data*, vol. 656. John Wiley & Sons. 95

Akutsu, T. (2010). Tree edit distance problems: Algorithms and applications to bioinformatics. *The Institute of Electronics, Information and Communication Engineers transactions on information and systems*, **93**, 208–218. 82

Alsulaiman, F. & Saddik, A. (2006). A novel 3D graphical password schema. In *Proceedings of 2006 IEEE International Conference on Virtual Environments, Human-Computer Interfaces and Measurement Systems,*, 125–128. 13

Ashby, F.G. & Perrin, N.A. (1988). Toward a unified theory of similarity and recognition. *Psychological Review*, **95**, 124–150. 42, 45

Attneave, F. (1950). Dimensions of similarity. *The American Journal of Psychology*, **63**, 516–556. 39

Augsten, N. (2008). *Approximate Matching of Hierarchial Data*. Ph.D. thesis, Institute of Computer Science, The Faculty of Engineering and Science, Aalborg University. 67

Augsten, N., Böhlen, M. & Gamper, J. (2005). Approximate matching of hierarchical data using pq-grams. In *Proceedings of the 31st international conference on Very large data bases*, 301–312, VLDB Endowment. 65, 101

Augsten, N., Böhlen, M. & Gamper, J. (2010). The pq-gram distance between ordered labeled trees. *Association for Computing Machinery Transactions on Database Systems*. 65, 82, 101

Augsten, N., Böhlen, M., Dyreson, C. & Gamper, J. (2012). Windowed pq-grams for approximate joins of data-centric XML. *The Very Large DataBases Journal*, **21**, 463–488. 67

Bellando, J. & Kothari, R. (1999). Region-based modeling and tree edit distance as a basis for gesture recognition. In *Proceedings of the International Conference on Image Analysis and Processing.*, 698–703. 82

Belongie, S. & Malik, J. (2000). Matching with shape contexts. In *IEEE Workshop Proceedings on Content-based Access of Image and Video Libraries, 2000.*, 20–26. 29

Belongie, S., Carson, C., Greenspan, H. & Malik, J. (1998). Color-and texture-based image segmentation using EM and its application to content-based image retrieval. In *Sixth International Conference on Computer Vision*, 675–682. 29

Belongie, S., Malik, J. & Puzicha, J. (2001). Shape context: A new descriptor for shape matching and object recognition. *Advances in Neural Information Processing Systems*, 831–837. 29

Belongie, S., Malik, J. & Puzicha, J. (2002). Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **24**, 509–522. 29

Ben-Shakhar, G. & Gati, I. (1992). The effect of common versus distinctive stimulus features on electrodermal orientation to significant stimuli. *Psychophysiology*, **29**, 306–314. 44

Benko, H., Wilson, A.D. & Baudisch, P. (2006). Precise selection techniques for multi-touch screens. In *Proceedings of the Conference on Human Factors in Computing Systems*, 1263–1272. 13

Berenzweig, A., Ellis, D.P.W. & Lawrence, S. (2003). Anchor space for classification and similarity measurement of music. In *Proceedings of the International Conference on Multimedia and Expo ICME'03.*, vol. 1, I–29. 48

Biederman, I. (1987). Recognition-by-components: a theory of human image understanding. *Psychological Review*, **94**, 115. xxi, 44, 45

Blonder, G.E. (1996). *Graphical password*. US Patent 5,559,961. 1, 11

Blough, D.S. (2001). The perception of similarity. *Avian Visual Cognition*. 39, 45

Buttler, D. (2004). A short survey of document structure similarity algorithms. In *International Conference on Internet Computing*, 3–9. 68, 101

Canny, J. (1986). A computational approach to edge detection. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 679–698. 29

Cao, Y., Chen, H., Yu, X., Zang, H. & Li, X. (2010). Machine Vision-Based detection system of a cleaning robot for vertical type air conditioning duct. In *2010 2nd International Conference on Advanced Computer Control (ICACC)*, vol. 1, 545–548. 81

Chang, T. & Kuo, C. (1993). Texture analysis and classification with tree-structured wavelet transform. *IEEE Transactions on Image Processing*, **2**, 429–441. 70

Chatzichristofis, S. & Boutalis, Y. (2008a). CEDD: Color and edge directivity descriptor: A compact descriptor for image indexing and retrieval. *Computer Vision Systems*, 312–322. 75, 84

Chatzichristofis, S., Boutalis, Y. & Lux, M. (2009). Img(Rummager): an interactive content based image retrieval system. In *Second International Workshop on Similarity Search and Applications, 2009. SISAP '09*, 151–153. 69, 70, 85

Chatzichristofis, S., Boutalis, Y. & Lux, M. (2011a). Combining color and spatial color distribution information in a fuzzy rule based compact composite descriptor. In J. Filipe, A. Fred & B. Sharp, eds., *Agents and Artificial Intelligence*, vol. 129 of *Communications in Computer and Information Science*, 49–60, Springer Berlin Heidelberg. 76

Chatzichristofis, S.A. & Boutalis, Y.S. (2008b). FCTH: fuzzy color and texture histogram - a low level feature for accurate image retrieval. In *Ninth International Workshop on Image Analysis for Multimedia Interactive Services, 2008. WIAMIS'08*, 191–196. 29, 75, 84

Chatzichristofis, S.A., Boutalis, Y.S. & Lux, M. (2010). SpCD-Spatial color distribution descriptor: A fuzzy rule-based compact composite descriptor appropriate for hand drawn color sketches retrieval. *Proceedings of the Second International Conference on Agents and Artificial Intelligence (ICAART)*, 58–63. 29, 76, 84

Chatzichristofis, S.A., Zagoris, K. & Arampatzis, A. (2011b). The TREC files: the (ground) truth is out there. In *Proceedings of the 34th International ACM SIGIR Conference on Research and Development in Information*, 1289–1290. 29

Chawathe, S.S. (1999). Comparing hierarchical data in external memory. In *Proceedings of the International Conference on Very Large Data Bases*, 90–101. 82

Chiasson, S., van Oorschot, P. & Biddle, R. (2007). Graphical password authentication using cued click points. *Computer Security*, 359–374. 13, 20

Chiasson, S., Forget, A., Biddle, R. & van Oorschot, P.C. (2008). Influencing users towards better passwords: persuasive cued click-points. In *Proceedings of the 22nd British Human Computer Initeraction Group Annual Conference on People and Computers: Culture, Creativity, Interaction-Volume 1*, 121–130. xxi, 13, 20, 21

Chomsky, N. (1956). Three models for the description of language. *Transactions on Information Theory*, **2**, 113–124. 34

Cieplinski, L. (2001). MPEG-7 color descriptors and their applications. In *Computer analysis of images and patterns*, 11–20, Springer. 73, 84

Cinque, L., Levialdi, S., Olsen, K.A. & Pellicano, A. (1999). Color-based image retrieval using spatial-chromatic histograms. In *IEEE International Conference on Multimedia Computing and Systems*, vol. 2, 969–973. 28

Cobena, G., Abiteboul, S. & Marian, A. (2002). Detecting changes in XML documents. In *Proceedings of the 18th International Conference on Data Engineering*, 41–52. 82

Creative Research Systems (2012). Sample size calculator. http://www.surveysystem.com/sscalc.htm, accessed: 04/01/2015. 78

Dalamagas, T., Cheng, T., Winkel, K. & Sellis, T. (2006). A methodology for clustering XML documents by structure. *Information Systems*, **31**, 187–228. 82

Datta, R., Joshi, D., Li, J. & Wang, J.Z. (2008). Image retrieval: Ideas, influences, and trends of the new age. *ACM Computing Surveys (CSUR)*, **40**, 5. xxi, 31

Davis, D., Monrose, F. & Reiter, M.K. (2004). On user choice in graphical password schemes. In *13th USENIX Security Symposium*, 151–164. 12

Deselaers, T. (2008). *Image retrieval, object recognition, and discriminative models*. Ph.D. dissertation, Rheinisch-Westfälische Technische Hochschule (RWTH) Aachen University, Germany. 69

Dhamija, R. & Perrig, A. (2000). Deja vu: A user study using images for authentication. In *Proceedings of the 9th conference on USENIX Security Symposium-Volume 9*, 4–4. 12, 179

Dopkins, S. & Ngo, C.T. (2001). The contribution of category relationships to similarity ratings. *Canadian Journal of Experimental Psychology*, **55**, 244–252. 44

Drewes, F. (2006). *Grammatical Picture Generation: A Tree-Based Approach*. Springer-Verlag GmbH, Berlin, Heidelberg. 33, 34, 54, 57, 61, 83, 85, 101, 125

Drewes, F. (2007). Links. *International Journal of Foundations of Computer Science*, **18**, 1187–1196. 125

Drewes, F. & Engelfriet, J. (2004). Branching synchronization grammars with nested tables. *Journal of Computer and System Sciences*, **68**, 611–656. 125

Drewes, F., Ewert, S., Klempien-Hinrichs, R. & Kreowski, H.J. (2003). Computing raster images from grid picture grammars. *Journal of Automata, Languages and Combinatorics*, **8**, 499–519. 36, 83

Drewes, F., du Toit, C., Ewert, S., van der Merwe, B. & van der Walt, A. (2008). Bag context tree grammars. *Fundamenta Informaticae*, **86**, 459–480. 182

Dunphy, P., Nicholson, J. & Olivier, P. (2008). Securing passfaces for description. In *Proceedings of the 4th Symposium on Usable Privacy and Security*, SOUPS '08, 24–35, ACM, New York, NY, USA. 171

Eiselen, R. (2012). Questionnaires (Part one). *EXACTICA: Thesis and Dissertation Solutions*, **9**. 78

Elftmann, P. (2006). *Secure Alternatives to Password-based Authentication Mechanisms*. Ph.D. thesis, Rheinisch-Westfaelische Technische Hochschule Aachen University, Germany. 10, 121

Estes, W.K. (1955). Statistical theory of distributional phenomena in learning. *Psychological Review*, **62**, 369. 40

Ewert, S., van der Merwe, B., du Toit, C., van der Walt, A., Drewes, F. & Högberg, J. (2005). *Random context tree grammars and tree transducers*. Umeå University. 182

Farmand, S. & Bin Zakaria, O. (2010). Improving graphical password resistant to shoulder-surfing using 4-way recognition-based sequence reproduction (rbsr4). In *The 2nd IEEE International Conference on Information Management and Engineering (ICIME)*, 644–650, IEEE. 14

Forget, A., Chiasson, S. & Biddle, R. (2010). Shoulder-surfing resistance with eye-gaze entry in cued-recall graphical passwords. In *Proceedings of the 28th International Conference on Human Factors in Computing Systems*, 1107–1110, ACM. 13

Gao, H., Ren, Z., Chang, X., Liu, X. & Aickelin, U. (2010). A new graphical password scheme resistant to Shoulder-Surfing. In *Proceedings of the 2010 International Conference on Cyberworlds*, 194–199. 13, 19

Gao, Y. (2003). Efficiently comparing face images using a modified Hausdorff distance. In *Vision, Image and Signal Processing, IEE Proceedings-*, vol. 150, 346–350, IET. 32

Garofalakis, M. & Kumar, A. (2005). XML stream processing using tree-edit distance embeddings. *Association for Computing Machinery Transactions on Database Systems*, **30**, 279–332. 67, 82, 101

Gati, I. & Tversky, A. (1984). Weighting common and distinctive features in perceptual and conceptual judgments. *Cognitive Psychology*, **16**, 341–370. 44

Goldberg, J., Hagman, J. & Sazawal, V. (2002). Doodling our way to better authentication. In *CHI'02 Extended Abstracts on Human Factors in Computing Systems*, 868–869. 12

Goldstone, R.L. & Son, J.Y. (2005). Similarity. In K.J. Holyoak & R.G. Morrison, eds., *The Cambridge Handbook of Thinking and Reasoning*, vol. 14, 13–36, Cambridge University Press, New York. 40

Gonzalez, R.C., Woods, R.E. & Eddins, S.L. (2009). *Digital image processing using MATLAB*, vol. 2. Gatesmark Publishing, Tennessee. 69

Govindarajulu, N.S. & Madhvanath, S. (2007). Password management using doodles. In *Proceedings of the 9th International Conference on Multimodal Interfaces*, 236–239. 12

Grey, J.M. (1977). Multidimensional perceptual scaling of musical timbres. *Journal of Acoustic Society of America*, **61**, 1270–1277. 48

Gross, M.H., Koch, R., Lippert, L. & Dreger, A. (1994). Multiscale image texture analysis in wavelet spaces. In *Proceedings of IEEE International Conference on Image Processing, ICIP-94*, vol. 3, 412–416. 70

Guha, S., Jagadish, H.V., Koudas, N., Srivastava, D. & Yu, T. (2002). Approximate XML joins. In *Proceedings of the 2002 Association for Computing Machinery international conference on Management of data*, 287–298, ACM, New York, NY, USA. 82

Guttman, N. & Kalish, H.I. (1956). Discriminability and stimulus generalization. *Journal of Experimental Psychology*, **51**, 79–88. 46

Habrard, A., Iñesta, J.M., Rizo, D. & Sebban, M. (2008). Melody recognition with learned edit distances. In N.d.V. Lobo, T. Kasparis, F. Roli, J.T. Kwok, M. Georgiopoulos, G.C. Anagnostopoulos & M. Loog, eds., *Structural, Syntactic, and Statistical Pattern Recognition*, no. 5342 in Lecture Notes in Computer Science, 86–96, Springer Berlin Heidelberg. 82

Hangouet, J.F. (1995). Computation of the Hausdorff distance between plane vector polylines. In *AUTOCARTO-CONFERENCE-*, 1–10. 32

Haralick, R.M. (1979). Statistical and structural approaches to texture. *Proceedings of the IEEE*, **67**, 786–804. 29

Haralick, R.M., Shanmugam, K. & Dinstein, I.H. (1973). Textural features for image classification. *IEEE Transactions on Systems, Man and Cybernetics*, 610–621. 70

Hastings, S.K. (2005). *Perceived Features and Similarity of Images: An Investigation into Their Relationships and a Test of Tversky's Contrast Model*. Ph.D. thesis, University of North Texas. 48

Hayashi, E., Dhamija, R., Christin, N. & Perrig, A. (2008). Use your illusion: secure authentication usable anywhere. In *Proceedings of the 4th Symposium on Usable Privacy and Security*, 35–45. 12

Heumann, H. & Wittum, G. (2009). The tree-edit-distance, a measure for quantifying neuronal morphology. *Neuroscience*, **10**, P89. 82

Hsieh, F.Y., Bloch, D.A. & Larsen, M.D. (1998). A simple method of sample size calculation for linear and logistic regression. *Statistics in medicine*, **17**, 1623–1634. 78

Hu, M.K. (1962). Visual pattern recognition by moment invariants. *Transactions on Information Theory*, **8**, 179–187. 29

Huang, J., Kumar, S.R., Mitra, M., Zhu, W.J. & Zabih, R. (1997). Image indexing using color correlograms. In *Proceedings of 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, 762–768. 28, 70

Hunter, P. (2004). Biometrics latest: one size does not fit all comers. *Computer Fraud & Security*, **2004**, 7–9. 11

Huttenlocher, D.P., Rucklidge, W.J. & Klanderman, G.A. (1992). Comparing images using the Hausdorff distance under translation. In *Computer Vision and Pattern Recognition, 1992. Proceedings CVPR'92., 1992 IEEE Computer Society Conference on*, 654–656, IEEE. 32

Huttenlocher, D.P., Klanderman, G.A. & Rucklidge, W.J. (1993). Comparing images using the Hausdorff distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**, 850–863. 32

Jain, A.K. & Vailaya, A. (1996). Image retrieval using color and shape. *Pattern Recognition*, **29**, 1233–1244. 29

James, W. (1890). *The Principles of Psychology*. Holt, New York. 42

Jebriel, S.M. & Poet, R. (2011). Preventing shoulder-surfing when selecting pass-images in challenge set. In *Innovations in Information Technology (IIT), 2011 International Conference on*, 437–442, IEEE. xxi, 14, 15, 16

Jermyn, I., Mayer, A., Monrose, F., Reiter, M.K. & Rubin, A.D. (1999). The design and analysis of graphical passwords. In *Proceedings of the 8th USENIX Security Symposium*, 1–14. 2, 12, 13

Johnson, M. (1981). Context effects in product perception. In *Advances in Consumer Research*, vol. 08, 112–115, Association for Consumer Research, Chicago. 44

Johnson, M.D. (1986). Consumer similarity judgments: A test of the contrast model. *Psychology and Marketing*, **3**, 47–60. 44

Johnson, M.D. & Horne, D.A. (1988). The contrast model of similarity and comparative advertising. *Psychology and Marketing*, **5**, 211–232. 44

Jokela, T., Iivari, N., Matero, J. & Karukka, M. (2003). The standard of user-centered design and the standard definition of usability: analyzing ISO 13407 against ISO 9241-11. In *Proceedings of the Latin American conference on Human-computer interaction*, 53–60, ACM. 175

Joshuva, M., Rani, T.S. & John, M.S. (2011). Implementing CHC to counter shoulder surfing attack in PassPoint–Style graphical passwords. *International Journal of Advanced Networking and Applications*, **2**, 906–910. 19

Kadam, P. & Bhalerao, S. (2010). Sample size calculation. *International Journal of Ayurveda Research*, **1**, 55–57. 78

Kendall, M.G. (1938). A new measure of rank correlation. *Biometrika*, 81–93. 95

Kim, D., Dunphy, P., Briggs, P., Hook, J., Nicholson, J.W., Nicholson, J. & Olivier, P. (2010). Multi-touch authentication on tabletops. In *Proceedings of the 28th international conference on Human factors in computing systems*, 1093–1102, ACM, Atlanta, Georgia, USA. xxi, xxxi, 14, 15, 17, 18

Kim, Y., Park, J., Kim, T. & Choi, J. (2007). Web information extraction by HTML tree edit distance matching. In *International Conference on Convergence Information Technology, 2007*, 2455–2460. 82

Klein, P., Tirthapura, S., Sharvit, D. & Kimia, B. (2000). A tree-edit-distance algorithm for comparing simple, closed shapes. In *Proceedings of the eleventh annual Society for Industrial and Applied Mathematics symposium on Discrete algorithms*, 696–704. 82

Ko, B.C. & Byun, H. (2002). Integrated region-based image retrieval using region's spatial relationships. In *16th International Conference on Pattern Recognition, 2002. Proceedings*, vol. 1, 196–199. 32

Krueger, L.E. (1978). A theory of perceptual matching. *Psychological Review*, **85**, 278–304. 38

Krumhansl, C.L. (1978). Concerning the applicability of geometric models to similarity data: The interrelationship between similarity and spatial density. *Psychological Review*, **85**, 445–463. 41

Kumar, K.D. (2011). Automated attacks on pass Point-Style graphical passwords. *Journal of Computer Applications (JCA)*, **4**. 19

Laine, A. & Fan, J. (1993). Texture classification by wavelet packet signatures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **15**, 1186–1191. 70

Lashkari, A.H., Farmand, S., Zakaria, O.B. & Saleh, R. (2009). Shoulder surfing attack in graphical password authentication. *International Journal of Computer Science*, **6**, 145–154. 14, 81, 99

Lee, K., Choy, Y. & Cho, S. (2004). An efficient algorithm to compute differences between structured documents. *Transactions on Knowledge and Data Engineering*, **16**, 965–979. 82

Li, Z., Sun, Q., Lian, Y. & Giusto, D. (2005). An Association-Based graphical password design resistant to Shoulder-Surfing attack. In *IEEE International Conference on Multimedia and Expo*, 245–248. 14, 17

Lin, Z., Wang, H. & McClean, S. (2010). Measuring tree similarity for natural language processing based information retrieval. In *Natural Language Processing and Information Systems*, 13–23, Springer. 82

Lindenmayer, A. (1968). Mathematical models for cellular interactions in development, parts i and ii. *Journal of Theoretical Biology*, **18**, 280 – 315. 36

Liu, X., Qiu, J., Ma, L., Gao, H. & Ren, Z. (2011). A novel cued-recall graphical password scheme. In *2011 Sixth International Conference on Image and Graphics (ICIG)*, 949–956, IEEE. 13

Lloyds, B.B. & Rosch, E. (1978). *Cognition and Categorization*. Hillsdale, NJ: Lawrence Erlbaum Associates. 42

Long, F., Zhang, H. & Feng, D.D. (2003). Fundamentals of content-based image retrieval. In *Multimedia Information Retrieval and Management*, 1–26, Springer. 69, 71, 84

Lux, M. & Chatzichristofis, S.A. (2008). LIRe: Lucene Image Retrieval-an extensible java CBIR library. In *Proceeding of the 16th ACM International Conference on Multimedia*, 1085–1088. 29

Lux, M. & Marques, O. (2013). Visual information retrieval using Java and LIRe. *Synthesis Lectures on Information Concepts, Retrieval, and Services*, **5**, 1–112. 69, 70, 85

Ma, B., Wang, L. & Zhang, K. (2002). Computing similarity between RNA structures. *Theoretical Computer Science*, **276**, 111–132. 82

Madigan, S. (1983). Picture memory. *Imagery, Memory and Cognition*, 65–89. 10, 121

Maetz, Y., Onno, S. & Heen, O. (2009). Recall-a-story, a story-telling graphical password system. In *Proceedings of the 5th Symposium on Usable Privacy and Security*, 1–1, ACM, Mountain View, California. 12

Mallows, C.L. (1972). A note on asymptotic joint normality. *The Annals of Mathematical Statistics*, **43**, 508–515. 32

Martinez-Diaz, M., Fierrez, J., Martin-Diaz, C. & Ortega-Garcia, J. (2010). DooDB: a graphical password database containing doodles and Pseudo-Signatures. In *2010 International Conference on Frontiers in Handwriting Recognition (ICFHR)*, 339–344, IEEE. 12

Masin, S.C., Zudini, V. & Antonelli, M. (2009). Early alternative derivations of Fechner's law. *Journal of the History of the Behavioral Sciences*, **45**, 56–65. 39

McAdams, S., Winsberg, S., Donnadieu, S., Soete, G. & Krimphoff, J. (1995). Perceptual scaling of synthesized musical timbres: Common dimensions, specificities, and latent subject classes. *Psychological Research*, **58**, 177–192. 48

Melara, R.D. (1992a). The concept of perceptual similarity: From psychophysics to cognitive psychology. *Psychophysical Approaches to Cognition*, 303–388. 39

Melara, R.D. (1992b). How important are dimensions to perception? *Behavioral and Brain Sciences*, **15**, 576–577. 39

Mémoli, F. (2008). Gromov-Hausdorff distances in euclidean spaces. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops, CVPRW'08.*, 1–8. 32

Miyachi, T., Takahashi, K., Hasegawa, M., Tanaka, Y. & Kato, S. (2010). A study on memorability and shoulder-surfing robustness of graphi-

cal password using DWT-based image blending. In *Picture Coding Symposium (PCS), 2010*, 134–137, IEEE. 14, 17

Nair, A.A., Joseph, T.R. & Johny, J.M. (2013). A proficient multilevel graphical authentication system. *International Journal of Science, Engineering and Technology Research (IJSETR)*, **2**. 24

Nicholson, J. (2009). *Design of a Multi-Touch Shoulder Surfing Resilient Graphical Password*. B.Sc in information systems, Newcastle University, Newcastle. 10

Nicholson, J., Dunphy, P., Coventry, L., Briggs, P. & Olivier, P. (2012). A security assessment of tiles: A new portfolio-based graphical authentication system. In *CHI '12 Extended Abstracts on Human Factors in Computing Systems*, CHI EA '12, 1967–1972, ACM, New York, NY, USA. 23

Nikolić, D., Mureşan, R.C., Feng, W. & Singer, W. (2012). Scaled correlation analysis: a better way to compute a cross-correlogram. *European Journal of Neuroscience*, **35**, 742–762. 95

Nosofsky, R.M. (1992). Similarity scaling and cognitive process models. *Annual Review of Psychology*, **43**, 25–53. 46

O'Gorman, L. (2003). Comparing passwords, tokens, and biometrics for user authentication. *Proceedings of the IEEE*, **91**, 2021–2040. 11

Ojala, T., Pietikäinen, M. & Harwood, D. (1996). A comparative study of texture measures with classification based on featured distributions. *Pattern Recognition*, **29**, 51–59. 28

Okundaye, B., Ewert, S. & Sanders, I. (2013). Determining image similarity from pattern matching of abstract syntax trees of tree picture grammars. In *Proceedings of the Twenty-Fourth Annual Symposium of the Pattern Recognition Association of South Africa*, 83–90, Johannesburg, South Africa. iii, 81, 124

Okundaye, B., Ewert, S. & Sanders, I. (2014a). A novel approach to visual password schemes using tree picture grammars. In *Proceedings of the 2014 PRASA, RobMech and AfLaT International Joint Symposium*, 247–252, Johannesburg, South Africa. iii, 121

Okundaye, B., Ewert, S. & Sanders, I. (2014b). Perceptual similarity of images generated using tree grammars. In *Empowered by Technology - Proceedings of South African Institute of Computer Scientists and Information Technologists 2014*, 286–296, The Association for Computing Machinery, Irene (Pretoria) , South Africa. iii, 99, 124

Oorschot, P.C. & Wan, T. (2009). TwoStep: an authentication method combining text and graphical passwords. *E-Technologies: Innovation in an Open World*, 233–239. 13

Pass, G., Zabih, R. & Miller, J. (1997). Comparing images using color coherence vectors. In *Proceedings of the Fourth ACM International Conference on Multimedia*, 65–73. 28

Pawlik, M. & Augsten, N. (2011). RTED: A robust algorithm for the tree edit distance. *Proceedings of the Very Large Database Endowment*, **5**, 334–345. xxi, 64, 66, 67, 84

Pearson, K. (1895). Note on regression and inheritance in the case of two parents. *Proceedings of the Royal Society of London*, **58**, 240–242. 95

Rasekgala, M., Ewert, S., Sanders, I. & Fogwill, T. (2014a). Requirements for secure graphical password schemes. In *IST-Africa Conference Proceedings, 2014*, 1–10, IEEE. 25

Rasekgala, M., Ewert, S., Sanders, I. & Fogwill, T. (2014b). Shape grammar model generating secure visual passwords: The move towards completely grammar based images. In *Second International Conference on Advances in Computing, Communication and Information Technology - CCIT 2014*, pp Accepted, Birmingham, UK. 25

Ray, P.P. (2012). Ray's scheme: Graphical password based hybrid authentication system for smart hand held devices. *Journal of Information Engineering and Applications*, **2**, 1–11. 11, 13

Realuser (2012). The science behind passfaces. http://www.realuser.com/published/The Science Behind Passfaces.pdf. 12, 14

Realuser (2015). Two factor authentication, graphical passwords - passfaces. http://www.realuser.com/demo/. 163

Reis, D.D.C., Golgher, P.B., Silva, A. & Laender, A.F. (2004). Automatic web news extraction using tree edit distance. In *Proceedings of the 13th international conference on World Wide Web*, 502–511. 82

Rogowitz, B.E., Frese, T., Smith, J.R., Bouman, C.A. & Kalin, E. (1998). Perceptual image similarity experiments. *Human Vision and Electronic Imaging III*, **3299**, 576–590. 47

Rorissa, A. (2005). *Perceived Features and Similarity of Images: An Investigation into Their Relationships and a Test of Tversky's Contrast Model*. Ph.D. thesis, University of North Texas. 39, 40, 44

Rosman, B. (2007). *Similarity of Abstract Images*. B.Sc (Honours) research report, University of the Witwatersrand, Johannesburg. 48, 50

Rubner, Y. & Tomasi, C. (2000). *Perceptual metrics for image database navigation*, vol. 1. Springer. 31

Rubner, Y., Tomasi, C. & Guibas, L.J. (2000). The Earth Mover's Distance as a metric for image retrieval. *International Journal of Computer Vision*, **40**, 99–121. 27, 40

Russell, R. & Sinha, P. (2001). Perceptually-based comparison of image similarity metrics. 48

Salehi-Abari, A., Thorpe, J. & Van Oorschot, P.C. (2008). On purely automated attacks and click-based graphical passwords. In *Annual Computer Security Applications Conference*, 111–120. 19

Sangore, M.R., Patil, G., Ramani, S. & Pasare, S. (2014). Authentication using images and pattern. *International Journal of Advanced Research in Electrical, Electronics and Instrumentation Engineering*, **3**, 8661—8667. 21

Santini, S. & Jain, R. (1999). Similarity measures. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **21**, 871–883. 38, 48

Sarohi, H.K. & Khan, F.U. (2013). Graphical password authentication schemes: Current status and key issues. *International Journal of Computer Science Issues (IJCSI)*, **10**. 24

Scassellati, B.M., Alexopoulos, S. & Flickner, M.D. (1994). Retrieving images by 2D shape: a comparison of computation methods with human perceptual judgments. In *Society of Photo-Optical Instrumentation Engineers (SPIE) Conference Series*, vol. 2185, 2–14. 49

Sebe, N., Tian, Q., Loupias, E., Lew, M.S. & Huang, T.S. (2000). Color indexing using wavelet-based salient points. In *Proceedings of the IEEE Workshop on Content-based Access of Image and Video Libraries*, 15–19. 29

Seng, L., Ithnin, N. & Mammi, H. (2011). User affinity of choice-features of mobile device graphical password scheme Anti-Shoulder surfing mechanism. *International Journal of Computer Science*, **8**, 255–261. 14

Shapiro, L. & Stockman, G. (2001). *Computer Vision*. Prentice Hall. 70

Shepard, R.N. (1962). The analysis of proximities: Multidimensional scaling with an unknown distance function. II. *Psychometrika*, **27**, 125–140. 39

Shepard, R.N. (1980). Multidimensional scaling, tree-fitting, and clustering. *Science*, **210**, 390–398. 39, 47

Shepard, R.N. (1987). Toward a universal law of generalization for psychological science. *Science*, **237**, 1317–1323. 39, 46

Sikora, T. (2001). The MPEG-7 visual standard for content description: an overview. *IEEE Transactions on Circuits and Systems for Video Technology*, **11**, 696–702. 74, 84

Sobrado, L. & Birget, J.C. (2002). Graphical passwords. *The Rutgers Scholar, an electronic Bulletin for undergraduate research*, **4**. 10, 121

Sonkar, S.K. & SB, G. (2011). Minimum space and huge security in 3D password scheme. *International Journal*, **29**, 35–40. 13

Squire, D.M. & Pun, T. (1997). A comparison of human and machine assessments of image similarity for the organization of image databases. In *Proceedings of the Scandinavian Conference on Image Analysis*, vol. 1, 51–58. 49

Srinivasa Ravi Kiran, T., Samabasiva Rao, K.V., Kameswara Rao, M. & Srisaila, A. (2013). A symbol based graphical schema resistant to peeping attack. *International Journal of Computer Science Issues (IJCSI)*, **10**. 22

Stricker, M. & Orengo, M. (1995). Similarity of color images. In *Proceedings of SPIE Storage and Retrieval for Image and Video Databases*, vol. 2420, 381–392. 27

Stubblefield, A. & Simon, D. (2004). Inkblot authentication. Technical Report MSR-TR-2004-85, Microsoft Research, Redmond, USA. 13

Subha, A. (2011). *Graphical Password Authentication using click cued points*. M.Sc, Cochin University of Science and Technology, Kerala. 13

Suo, X., Zhu, Y. & Owen, G.S. (2005). Graphical passwords: A survey. In *Computer Security Applications Conference, 21st Annual*, 10–pp. 10, 13

Swain, M.J. & Ballard, D.H. (1991). Color indexing. *International Journal of Computer Vision*, **7**, 11–32. 27, 69, 70, 84

Tamura, H., Mori, S. & Yamawaki, T. (1978). Textural features corresponding to visual perception. *IEEE Transactions on Systems, Man and Cybernetics*, **8**, 460–473. 70, 71, 84

Tao, H. (2006). *Pass-Go, a new graphical password scheme*. Ph.D. thesis, University of Ottawa. 12

Tao, H. & Adams, C. (2008). Pass-Go: a proposal to improve the usability of graphical passwords. *International Journal of Network Security*, **7**, 273–292. 12

Tarr, M.J. & Bülthoff, H.H. (1998). Image-based object recognition in man, monkey and machine. *Cognition*, **67**, 1–20. 40

The Research Advisors (2006). Sample size table*. http://www.research-advisors.com/tools/SampleSize.htm, accessed: 04/01/2015. 78

Thorpe, J. & Van Oorschot, P.C. (2004). Towards secure design choices for implementing graphical passwords. In *20th Annual Computer Security Applications Conference*, 50–60. 19

Thorpe, J. & van Oorschot, P.C. (2007). Human-seeded attacks and exploiting hot-spots in graphical passwords. In *Proceedings of 16th USENIX Security Symposium on USENIX Security Symposium*, 8. 19

Thurstone, L.L. (1927). A law of comparative judgment. *Psychological Review*, **34**, 273–286. 39

Towhidi, F. & Masrom, M. (2009). A survey on recognition based graphical user authentication algorithms. *arXiv:0912.0942 [cs]*, arXiv: 0912.0942. 175

Towhidi, F., Masrom, M. & Manaf, A.A. (2013). An enhancement on pass-face graphical password authentication. *Journal of Basic and Applied Scientific Research*, **2**. 24, 25

Tversky, A. (1977). Features of similarity. *Psychological Review*, **84**, 327. 39, 42, 43, 44, 46

Tversky, A. & Gati, I. (1978). Studies of similarity. *Cognition and Categorization*, **1**, 79–98. 43, 44

Tversky, A. & Gati, I. (1982). Similarity, separability, and the triangle inequality. *Psychological Review*, **89**, 123. 42, 43

Ulhaque, E. & Bahn, K.D. (1992). A spreading activation model of consumers' asymmetric similarity judgment. *Advances in Consumer Research*, **19**, 782–786. 44

Valiente, G. (2001). An efficient Bottom-Up distance between trees. In *Eighth Symposium on String Processing and Information Retrieval SPIRE'01*, 212–219. 68, 101, 187

Van Oorschot, P.C. & Thorpe, J. (2008). On predictive models and user-drawn graphical passwords. *ACM Transactions on Information and System Security*, **10**, 5. 19

van Oorschot, P.C. & Thorpe, J. (2011). Exploiting predictability in click-based graphical passwords. *Journal of Computer Security*, **19**, 669–702. 19

Van Oorschot, P.C., Salehi-Abari, A. & Thorpe, J. (2010). Purely automated attacks on passpoints-style graphical passwords. *IEEE Transactions on Information Forensics and Security*, **5**, 393–405. 19

Varenhorst, C. (2004). Passdoodles: A lightweight authentication method. *Research Science Institute*, 1–11. 12

Vivek, E.P. & Sudha, N. (2006). Gray Hausdorff distance measure for comparing face images. *IEEE Transactions on Information Forensics and Security*, **1**, 342–349. 32

Wang, J.T. & Zhang, K. (2001). Finding similar consensus between trees: an algorithm and a distance hierarchy. *Pattern Recognition*, **34**, 127–137. 187

Wang, J.Z., Li, J. & Wiederhold, G. (2001). SIMPLIcity: semantics-sensitive integrated matching for picture libraries. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, **23**, 947–963. 31, 32

Weiss, R. & De Luca, A. (2008). PassShapes: utilizing stroke based authentication to increase password memorability. In *Proceedings of the 5th Nordic Conference on Human-Computer Interaction: Building Bridges*, 383–392, ACM. 12

Wiedenbeck, S., Waters, J., Birget, J., Brodskiy, A. & Memon, N. (2005). PassPoints: design and longitudinal evaluation of a graphical password system. *International Journal of Human-Computer Studies*, **63**, 102–127. 10, 12, 13

Wiedenbeck, S., Waters, J., Sobrado, L. & Birget, J. (2006). Design and evaluation of a shoulder-surfing resistant graphical password scheme. In *Proceedings of the Working Conference on Advanced Visual Interfaces*, 177–184, ACM, Venezia, Italy. 12, 13, 14, 20

Wixted, J.T. (2004). The psychology and neuroscience of forgetting. *Annual Review of Psychology*, **55**, 235–269, pMID: 14744216. 121

Wixted, J.T. (2005). A theory about why we forget what we once knew. *Current Directions in Psychological Science*, **14**, 6–9. 121

Yang, R., Kalnis, P. & Tung, A.K. (2005). Similarity evaluation on tree-structured data. In *Proceedings of the 2005 Association for Computing Machinery's Special Interest Group on Management of Data*, 754–765, ACM. 67, 101

Zhang, K. & Shasha, D. (1989). Simple fast algorithms for the editing distance between trees and related problems. *Society for Industrial and Applied Mathematics journal on computing*, **18**, 1245–1262. 64, 84, 101

Zhao, H. & Li, X. (2007). S3PAS: a scalable shoulder-surfing resistant textual-graphical password authentication scheme. In *21st International Conference on Advanced Information Networking and Applications Workshops, AINAW'07*, vol. 2, 467–472. 20

Appendix A

Sample Tree Grammars, Images and Trees

A.1 Sample Grid Picture Grammar (1)

This appendix provides a sample of the different grammars used in the study.

A.1.1 Grammar

This is a sample Grid picture grammar implemented using regular tree grammar.

```
generators.regularTreeGrammar:
( { S,A1,A2,B1,B2,C1,C2,Ht,Hb,Vl,Vr,X1,X2 },
  { S:6,C1:5,C2:2,C3:0, B1:6,B2:3,B3:0,A1:5,
  A2:2,A3:0,Hb:3,Ht:3,Vr:3,Vl:3,X1:5,X2:5, sq:0, -:0 },
  {
    S -> S[A1,A2,B1,B2,C1,C2],
    A1 -> A1[-,-,sq,-,sq],
```

```
    A1 -> A1[Ht,Vr,sq,Vr,sq],
    A2 -> A2[Vl,Vl],
    A2 -> A2[-,-],

    B1 -> B1[Vr,sq,Vr,sq,Vr,sq],
    B1 -> B1[-,sq,-,sq,-,sq],
    B2 -> B2[Vl,Vl,Vl],
    B2 -> B2[-,-,-],

    C1 -> C1[Vr,sq,Vr,sq,Hb],
    C1 -> C1[-,sq,-,sq,-],
    C2 -> C2[Vl,Vl],
    C2 -> C2[-,-],

    Hb -> Hb[Hb,Hb,Hb],
    Hb -> Hb[sq,-,sq],
    Ht -> Ht[Ht,Ht,Ht],
    Ht -> Ht[sq,-,sq],
    Vr -> Vr[Vr,Vr,Vr],
    Vr -> Vr[sq,-,sq],
    Vl -> Vl[Vl,Vl,Vl],
    Vl -> Vl[sq,-,sq],
    X1 -> X1[X1,Hb,Hb,Vl,Vl],
    X1 -> X1[sq,-,sq,-,sq]
  },
```

```
  S
)
```

A.1.2 Algebra

This is the algebra for the previous grammar.

```
applications.collages.gridAlgebra:
( 3,
  {
    S = 0ab-1ab-3ab-4ab-6ab-7ab,
    A1 = 2ab-4ab-5ab-7ab-8ab,
    A2 = 3ab-6ab,

    B1 = 1ab-2ab-4ab-5ab-7ab-8ab,
    B2 = 0ab-3ab-6ab,

    C1 = 1ab-2ab-4ab-5ab-8ab,
    C2 = 0ab-3ab,

    Hb = 0ab-1ab-2ab,
    Ht = 6ab-7ab-8ab,
    Vr = 2ab-5ab-8ab,
    Vl = 0ab-3ab-6ab,
    X1 = 0ab-1ab-2ab-3ab-6ab

   }
```

```
)
```

A.2 Another Sample Grid Picture Grammar

This Grid picture grammar is implemented using ET0L grammar.

A.2.1 Grammar

```
generators.ET0LTreeGrammar:
( {S,A2,B2,C2,Ht,Hb,Vl,Vr },
  { H:1,S:3,A2:7,B2:9,C2:7,Hb:3,Ht:3,Vr:3,Vl:3,X1:5,X2:5, sq:0, -:0 },
  {
    {

     A2 -> A2[Ht,Vr,sq,Vl,Vr,sq,Vl],
     B2 -> B2[Vr,sq,Vl,Vr,sq,Vl,Vr,sq,Vl],
     C2 -> C2[Vr,sq,Vl,Vr,sq,Vl,Hb]
    },

    {Hb -> Hb[sq,-,sq],
     Hb -> Hb[-,sq,sq],
     Hb -> Hb[sq,sq,-]},

    { Ht -> Ht[sq,-,sq],
     Ht -> Ht[sq,sq,-],
     Ht -> Ht[-,sq,sq]},
```

```
    {Vr -> Vr[sq,-,sq],
    Vr -> Vr[sq,sq,-],
    Vr -> Vr[-,sq,sq]},



    {Vl -> Vl[-,sq,sq],
    Vl -> Vl[sq,sq,-],
    Vl -> Vl[sq,-,sq]
   }
   },
  S[A2,B2,C2], (1 2+ 3+ 4+ 5+)
)
```

A.2.2 Algebra

This is the algebra for the previous ET0L grammar.

```
applications.collages.gridAlgebra:
( 3,
  {


    S = 1ab-4ab-7ab,
    A2 = 1ab-3ab-4ab-5ab-6ab-7ab-8ab,
    B2 = 0ab-1ab-2ab-3ab-4ab-5ab-6ab-7ab-8ab,
    C2 = 0ab-1ab-2ab-3ab-4ab-5ab-7ab,
```

```
    Hb = 0ab-1ab-2ab,
    Ht = 6ab-7ab-8ab,
    Vr = 2ab-5ab-8ab,
    Vl = 0ab-3ab-6ab
   }
)
```

A.2.3 Generated grammar

Since the previous grammar is an ET0L grammar, it is translated into an equivalent regular tree grammar and a top-down transducer by Treebag. The next two sections show the translated grammar and transducer respectively.

```
generators.stubbornRegularTreeGrammar:
  (
    { S0, S, S8, S6, S4, S2, S1 },
    { t5:1, t4:1, t3:1, t2:1, t1:1, init:1, bot:0 },
    {
      S -> init[S0],
      S0 -> t1[S1],
      S1 -> t2[S2],
      S2 -> t3[S4],
      S2 -> t2[S2],
      S4 -> t4[S6],
      S4 -> t3[S4],
      S6 -> t5[S8],
      S6 -> t4[S6],
```

```
    S8 -> bot weight 0,
    S8 -> t5[S8]
  },
  S
)
```

A.2.4 Generated transducer

```
generators.tdTransducer:
(
  { S0:0, t5:1, S:0, t4:1, t3:1, t2:1, t1:1, bot:0, S8:0,
   S6:0, S4:0, S2:0, init:1, S1:0 },
  { Ht:3, Ht:0, -:0, A2:0, A2:7, S:0, S:3, B2:9, B2:0, Hb:3,
  Hb:0, H:1, C2:0, C2:7, sq:0, Vr:0, Vr:3, X2:5, X1:5, Vl:0, Vl:3 },
  { Vr, A2, Vl, S, C2, q0, Hb, B2, Ht },
  {
    q0[init[x1]] -> S[A2[x1],B2[x1],C2[x1]],
    A2[t1[x1]] -> A2[Ht[x1],Vr[x1],sq,Vl[x1],Vr[x1],sq,Vl[x1]],
    B2[t1[x1]] -> B2[Vr[x1],sq,Vl[x1],Vr[x1],sq,Vl[x1],Vr[x1],sq,Vl[x1]],
    C2[t1[x1]] -> C2[Vr[x1],sq,Vl[x1],Vr[x1],sq,Vl[x1],Hb[x1]],
    Hb[t1[x1]] -> Hb[x1],
    Vr[t1[x1]] -> Vr[x1],
    S[t1[x1]] -> S[x1],
    Ht[t1[x1]] -> Ht[x1],
```

```
Vl[t1[x1]] -> Vl[x1],
Hb[t2[x1]] -> Hb[sq,-,sq],
Hb[t2[x1]] -> Hb[-,sq,sq],
Hb[t2[x1]] -> Hb[sq,sq,-],
A2[t2[x1]] -> A2[x1],
Vr[t2[x1]] -> Vr[x1],
S[t2[x1]] -> S[x1],
B2[t2[x1]] -> B2[x1],
Ht[t2[x1]] -> Ht[x1],
C2[t2[x1]] -> C2[x1],
Vl[t2[x1]] -> Vl[x1],
Ht[t3[x1]] -> Ht[sq,-,sq],
Ht[t3[x1]] -> Ht[sq,sq,-],
Ht[t3[x1]] -> Ht[-,sq,sq],
Hb[t3[x1]] -> Hb[x1],
A2[t3[x1]] -> A2[x1],
Vr[t3[x1]] -> Vr[x1],
S[t3[x1]] -> S[x1],
B2[t3[x1]] -> B2[x1],
C2[t3[x1]] -> C2[x1],
Vl[t3[x1]] -> Vl[x1],
Vr[t4[x1]] -> Vr[sq,-,sq],
Vr[t4[x1]] -> Vr[sq,sq,-],
Vr[t4[x1]] -> Vr[-,sq,sq],
Hb[t4[x1]] -> Hb[x1],
```

```
A2[t4[x1]] -> A2[x1],
S[t4[x1]] -> S[x1],
B2[t4[x1]] -> B2[x1],
Ht[t4[x1]] -> Ht[x1],
C2[t4[x1]] -> C2[x1],
Vl[t4[x1]] -> Vl[x1],
Vl[t5[x1]] -> Vl[-,sq,sq],
Vl[t5[x1]] -> Vl[sq,sq,-],
Vl[t5[x1]] -> Vl[sq,-,sq],
Hb[t5[x1]] -> Hb[x1],
A2[t5[x1]] -> A2[x1],
Vr[t5[x1]] -> Vr[x1],
S[t5[x1]] -> S[x1],
B2[t5[x1]] -> B2[x1],
Ht[t5[x1]] -> Ht[x1],
C2[t5[x1]] -> C2[x1],
Hb[S0] -> Hb,
Hb[S] -> Hb,
Hb[S8] -> Hb,
Hb[S6] -> Hb,
Hb[S4] -> Hb,
Hb[S2] -> Hb,
Hb[S1] -> Hb,
A2[S0] -> A2,
A2[S] -> A2,
```

```
A2[S8] -> A2,
A2[S6] -> A2,
A2[S4] -> A2,
A2[S2] -> A2,
A2[S1] -> A2,
Vr[S0] -> Vr,
Vr[S] -> Vr,
Vr[S8] -> Vr,
Vr[S6] -> Vr,
Vr[S4] -> Vr,
Vr[S2] -> Vr,
Vr[S1] -> Vr,
S[S0] -> S,
S[S] -> S,
S[S8] -> S,
S[S6] -> S,
S[S4] -> S,
S[S2] -> S,
S[S1] -> S,
B2[S0] -> B2,
B2[S] -> B2,
B2[S8] -> B2,
B2[S6] -> B2,
B2[S4] -> B2,
B2[S2] -> B2,
```

```
    B2[S1] -> B2,
    Ht[S0] -> Ht,
    Ht[S] -> Ht,
    Ht[S8] -> Ht,
    Ht[S6] -> Ht,
    Ht[S4] -> Ht,
    Ht[S2] -> Ht,
    Ht[S1] -> Ht,
    C2[S0] -> C2,
    C2[S] -> C2,
    C2[S8] -> C2,
    C2[S6] -> C2,
    C2[S4] -> C2,
    C2[S2] -> C2,
    C2[S1] -> C2,
    Vl[S0] -> Vl,
    Vl[S] -> Vl,
    Vl[S8] -> Vl,
    Vl[S6] -> Vl,
    Vl[S4] -> Vl,
    Vl[S2] -> Vl,
    Vl[S1] -> Vl
  },
  q0
)
```

A.2.5 Sample trees for previous grammar

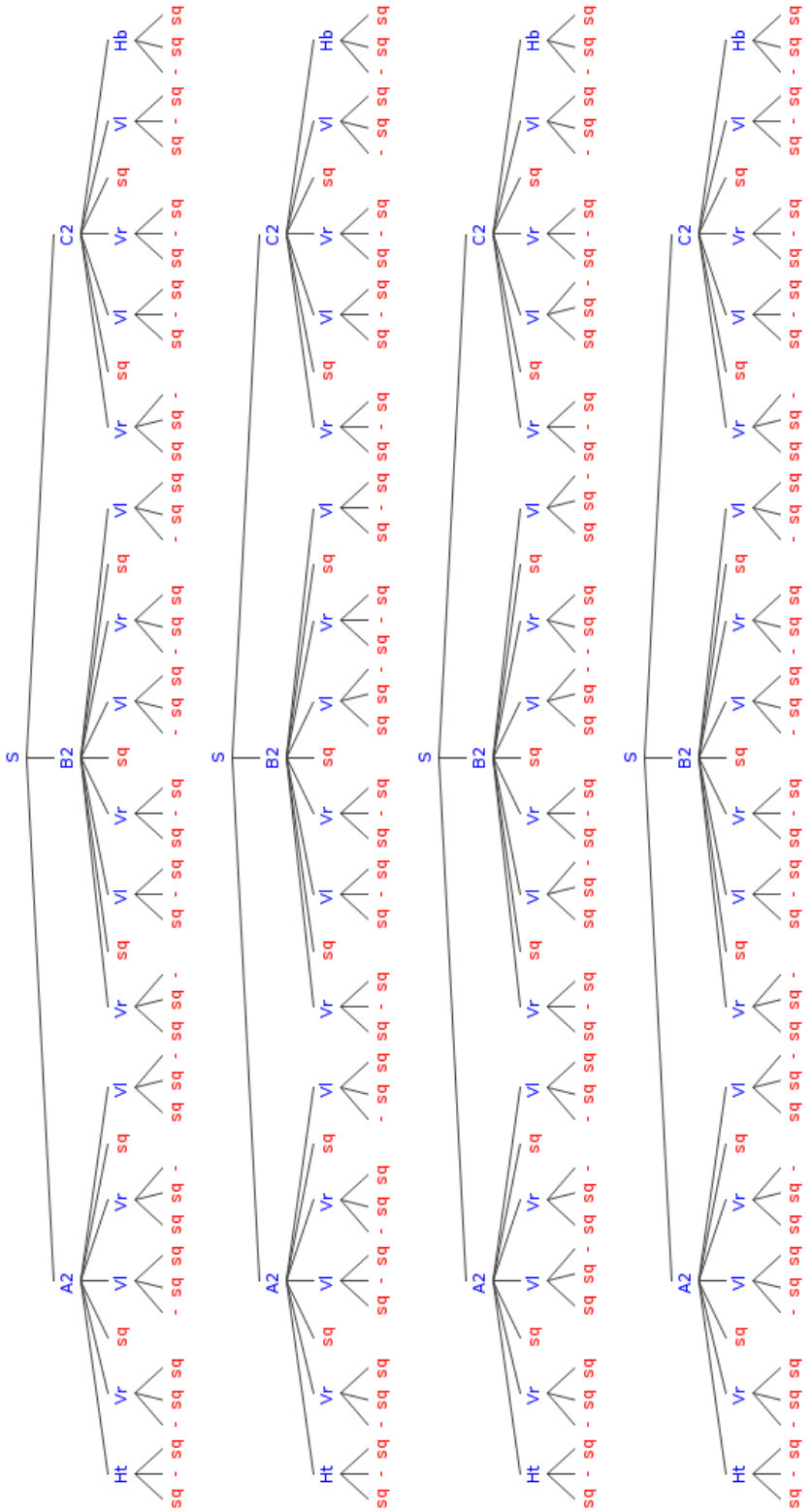
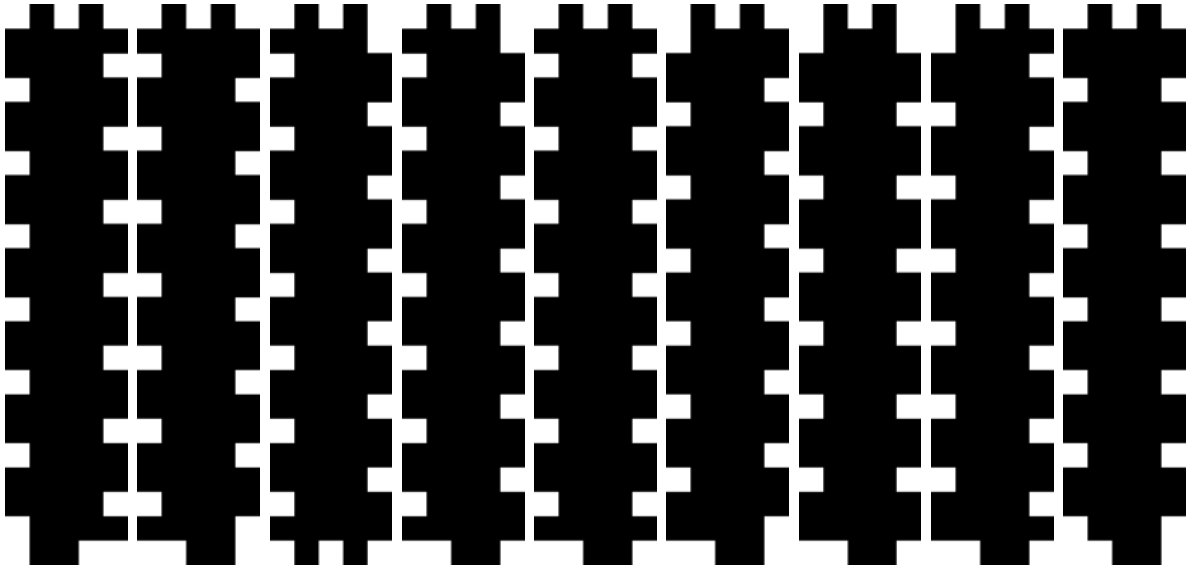
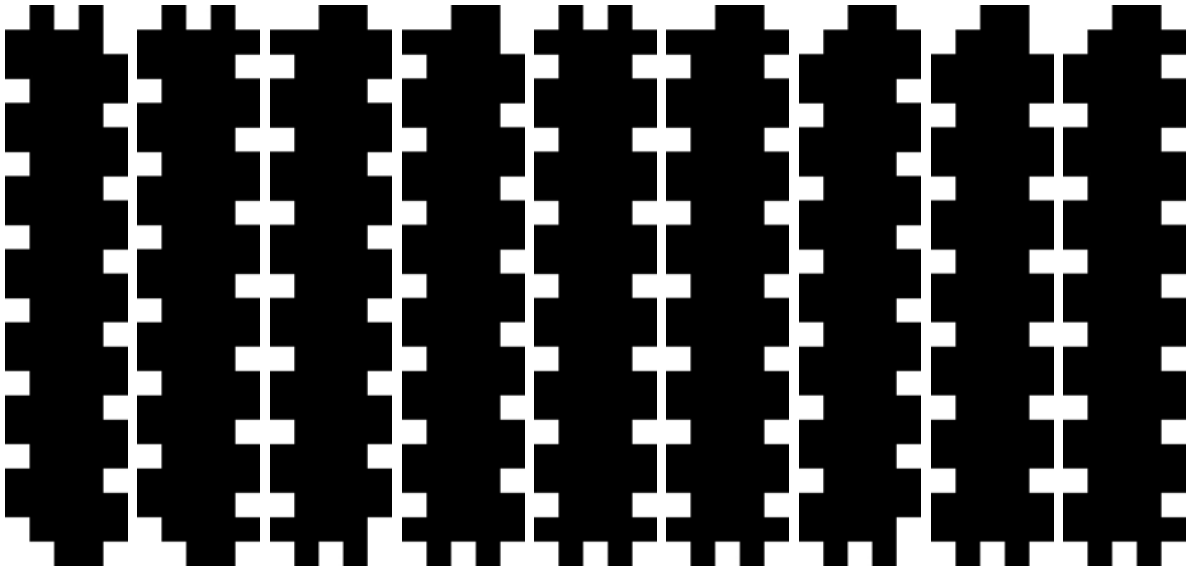


Figure A.1: Sample trees for previous grammar

A.2.6 Sample images for previous grammar



(a) I (b) II (c) III (d) IV (e) V (f) VI (g) VII (h) VIII (i) IX



(j) X (k) XI (l) XII (m) XIII (n) XIV (o) XV (p) XVI (q) XVII (r) XVIII

Figure A.2: ET0L Grammar images

A.3 Sample Branching Synchronization Grammar (16)

A.3.1 Grammar

```
generators.BSTGrammar:
  ( { tri1, tri2 },
    { F:8, G:4, p1:0, p2:0 },
    { 1, 2 },
    1,
    {
      {
        tri1 -> G[tri1<1>,tri1<2>,tri2<2>,tri1<2>],
        tri2 -> G[tri2<2>,tri2<1>,tri1<1>,tri2<1>]
      },
      {
        tri1 -> p1,
        tri2 -> p2
      }
    },
    F[tri1<1>,tri1<1>,tri1<1>,tri1<1>,tri1<1>,tri1<1>,tri1<1>,tri1<1>] )
```

A.3.2 Algebra

```
applications.collages.collageAlgebra:
  {
```

```
ntColour = const(.8),
w = const(tan(22.5)),

f1 = rotate(0),
f2 = rotate(45),
f3 = rotate(90),
f4 = rotate(135),
f5 = rotate(180),
f6 = rotate(225),
f7 = rotate(270),
f8 = rotate(315),
F = <f1, f2, f3, f4, f5, f6, f7, f8>,

g1 = scale(.5),
g2 = scale(.5) . translate(-(sqrt(1/2) - 1/2),.5),
g3 = scale(.5,-.5) . translate(0,1),
g4 = scale(.5) . translate(sqrt(1/2) - 1/2,.5),
G = <g1, g2, g3, g4>,

p1 = { filledPolygon((-\# w / 3,1/3), (\# w / 3,1/3),
         (2/3*\# w,2/3), (0,1), (-2/3*\# w,2/3)),
       polygon((0,0),(-\# w,1),(\# w,1))
     },
```

```
    p2 = { filledPolygon((-\# w / 3,1/3), (\# w / 3,1/3),
                 (2/3*\# w,2/3), (0,1), (-2/3*\# w,2/3)),
           polygon((0,0),(-\# w,1),(\# w,1))
         },

    tri1 = { filledPolygon((0,0),(-\# w,1),(\# w,1))[1,\# ntColour,1],
             polygon((0,0),(-\# w,1),(\# w,1)) },
    tri2 = { filledPolygon((0,0),(-\# w,1),(\# w,1))[\# ntColour,1,1],
             polygon((0,0),(-\# w,1),(\# w,1)) }

  }
```

A.3.3 Generated Grammar

```
generators.stubbornRegularTreeGrammar:    (
    { S0, Start0 },
    { TBL_axiom_table_:2, TBL_2:2, init:1, TBL_1:2, bot:0 },
    {
S0 -> TBL_1[S0,S0] weight 1.0,
S0 -> TBL_2[bot,bot] weight 1.0,
Start0 -> init[TBL_axiom_table_[S0,bot]]

    },
    Start0
  )
```

A.3.4 Generated Transducer

```
generators.tdTransducer:
  (
    { TBL_axiom_table_:2, Start0:0, S0:0, TBL_2:2, init:1,
    TBL_1:2, bot:0 },
    { tri2:0, tri1:0, G:4, F:8, p2:0, NEW_AXIOM:0, p1:0 },
    { q0, tri2, tri1, NEW_AXIOM },
    {
      q0[init[x1]] -> NEW_AXIOM[x1]
    ,
      tri1[TBL_1[x1,x2]] -> G[tri1[x1],tri1[x2],tri2[x2],
      tri1[x2]] weight 1.0,
      tri2[TBL_1[x1,x2]] -> G[tri2[x2],tri2[x1],tri1[x1],
      tri2[x1]] weight 1.0
      ,
      tri1[TBL_2[x1,x2]] -> p1 weight 1.0,
      tri2[TBL_2[x1,x2]] -> p2 weight 1.0
      ,
      NEW_AXIOM[TBL_axiom_table_[x1,x2]] ->
      F[tri1[x1],tri1[x1],tri1[x1],tri1[x1],tri1[x1],tri1[x1],
      tri1[x1],tri1[x1]]
       weight 1.0
```

```
      ,
    tri2[S0] -> tri2,
    tri2[Start0] -> tri2
    ,
    tri1[S0] -> tri1,
    tri1[Start0] -> tri1
    ,
    NEW_AXIOM[S0] -> NEW_AXIOM,
    NEW_AXIOM[Start0] -> NEW_AXIOM},
  q0
)
```

Appendix B

Sample On-line Survey

Perceptual Similarity Study

Welcome Page

School of Computer Science

University of the Witwatersrand, Johannesburg.

Tel: + 27 837586901

November 11, 2013

Dear Sir / Madam

I am doing research to determine how people perceive the similarity of images. To this end, I request that you kindly complete the following survey. This survey should take no longer than 30 minutes. Although your response is of great importance to us, your participation in this survey is entirely voluntary. The survey is anonymous. Any information provided by you is confidential and will be reported in summary format only.

There are a number of questions where you are asked about your view of the similarity of the given images. For each question you will be given a “target image” and a number of "answer images”. You are required to drag and drop the answer images from the left column to the right column in order of similarity to the image in the question you are currently on. You can drag and drop images in any order, and later rearrange them into, what in your opinion, is the "right" order. Please, note that there are no right or wrong answers in this survey.

Additional explanatory images and a video are available on the next page, should you need them.

The first two image questions are compulsory, and you will not be able to continue with the survey without answering them. After that, you can skip a survey question by doing the following:
1. Click on next.
2. At the top of your webpage, a message that ends with "….Go to the first error" will appear. This message simply states that a required question has been skipped.
3. Click on next a second time.

You can save your work at any time and resume it later. The save and continue survey bar is at the top of the web page. Please click on the submit button after you have completed the survey.

Summary results of this research will be published in my PhD thesis and will be available on the Wits University website (www.wits.ac.za) as from December 2014.

Should you have any queries or comments regarding this survey, you are welcome to phone me at +27837586901 or e-mail me at benjaminokundaye@gmail.com.

Yours sincerely,

Benjamin. O. Okundaye	Prof. Sigrid Ewert	Prof. Ian Sanders
School of Computer Science	PhD Supervisor	PhD Co-Supervisor
University of the Witwatersrand, Johannesburg	University of the Witwatersrand, Johannesburg	University of South Africa

Next

0%

This student research survey is powered by surveygizmo

Planning or conducting an academic research project?


Save and continue survey later
Perceptual Similarity Study
Background information
1. What is your Gender?
Male
Female
2. What is your Age?
Under 18
18-24
25-34
35-54
55+
3. What is your Ethnicity?
Asian/Indian
Black
Caucasian/White
Coloured
Native American/Alaska Native
Hispanic
Other/Multi-Racial
4. How would you rate your computer literacy?
Excellent
Good
Fair
Poor
5. What is your highest educational qualification?
Did not complete secondary education
Completed high school or equivalent
Completed tertiary education
6. From which country are you taking this survey?
South Africa
Back
Next
This student research survey is powered by
surveygizmo
Planning or conducting
an academic research project?

Save and continue survey later

Perceptual Similarity Study

How to Answer Image Questions Page

1. At the beginning all the answer images are in the left column



2. Images are dragged from the left column to the right column



3. This is the result after dragging I to the right column



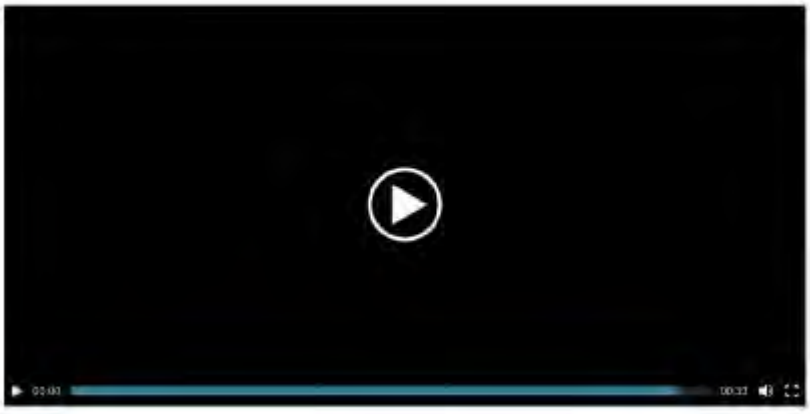
4. Images in the right column can be reordered by dragging an image up or down in this column. Here the image I is being moved up



5. This is the result after image I has been moved up from the third to the second position



Here is a short video demonstration of the drag and drop ordering that you are expected to do in this survey.



Back Next

This student research survey is powered by surveygizmo

Planning or conducting an academic research project?

Save and continue survey later

Perceptual Similarity Study

7. Order the followeg images according to their similarity to the image below:





Back Next

This student research survey is powered by surveygizmo

Planning or conducting an academic research project?

Save and continue survey later

Perceptual Similarity Study

Similarity of Some Letters to E

8. Order the following images according to their similarity to the image below



*

Drag items from the left-hand list into the right-hand list to order them.



Back Next

44%

This student research survey is powered by surveygizmo

Planning or conducting an academic research project?

Save and continue survey later

Perceptual Similarity Study

Similarity of Some Barnsley Ferns

9. Order the following images according to their similarity to the image below





















Save and continue survey later

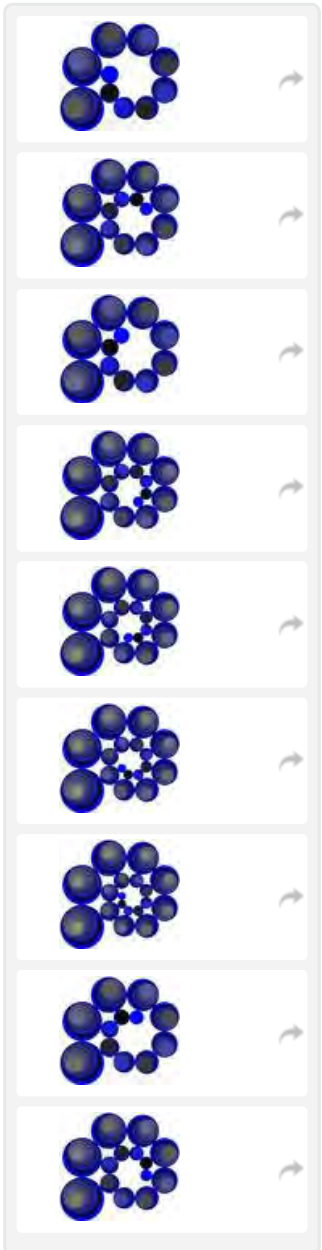
Perceptual Similarity Study

Spirals

10. Order the following images according to their similarity to the image below



Drag items from the left-hand list into the right-hand list to order them.



Save and continue survey later

Perceptual Similarity Study

New Page

11. Order the following images according to their similarity to the image below:





Back Next

This student research survey is powered by surveygizmo

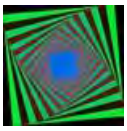
Planning or conducting an academic research project?

Save and continue survey later

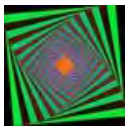
Perceptual Similarity Study

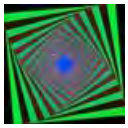
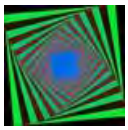
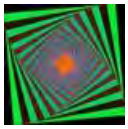
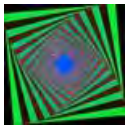
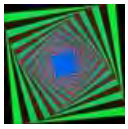
Some Rotating Squares

12. Order the following images according to their similarity to the image below



Drag items from the left-hand list into the right-hand list to order them.



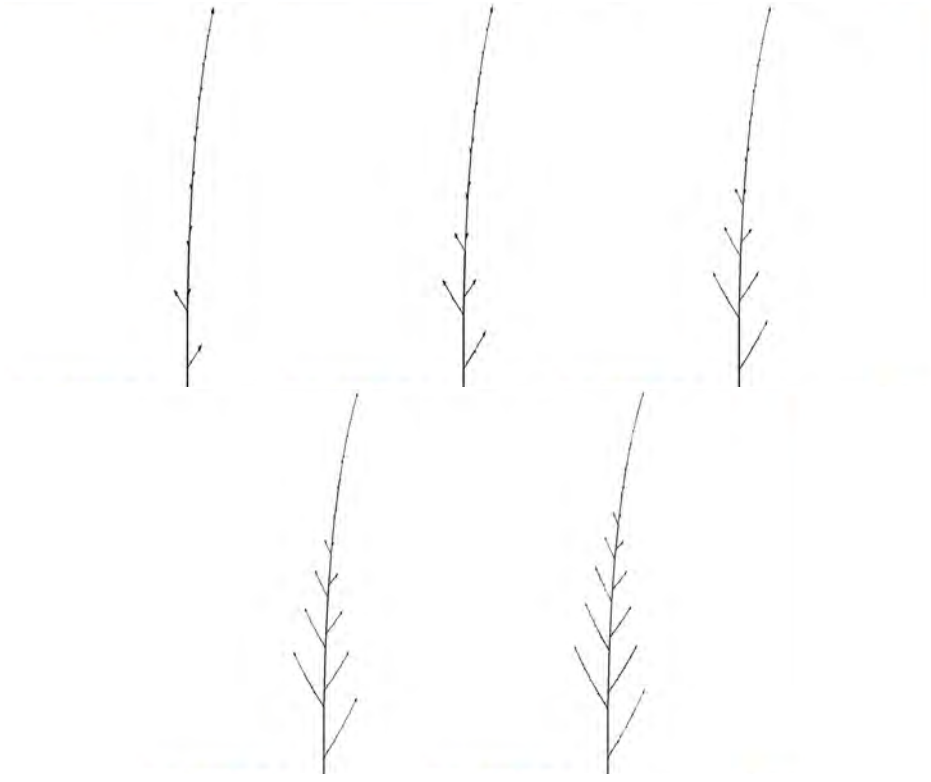






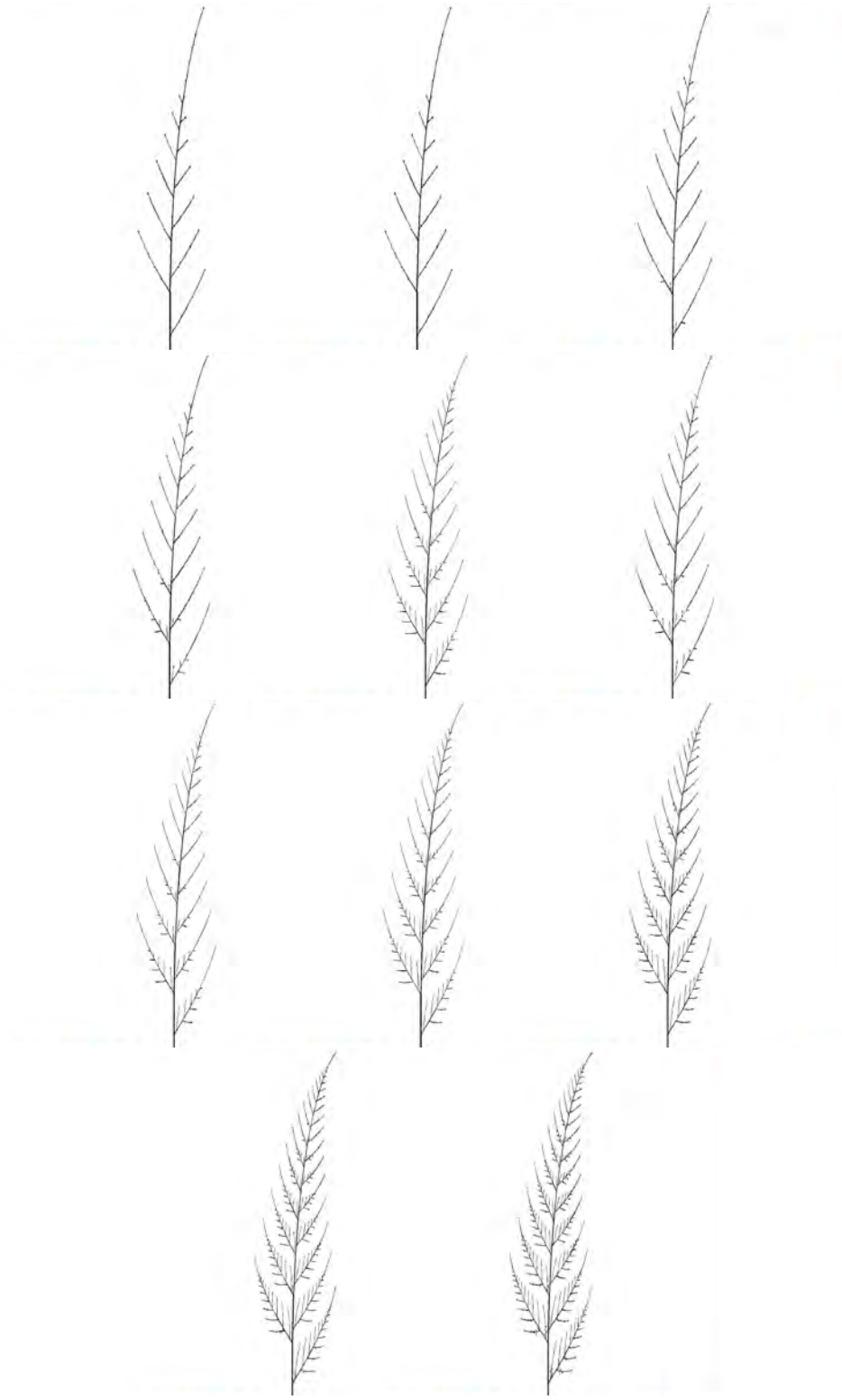


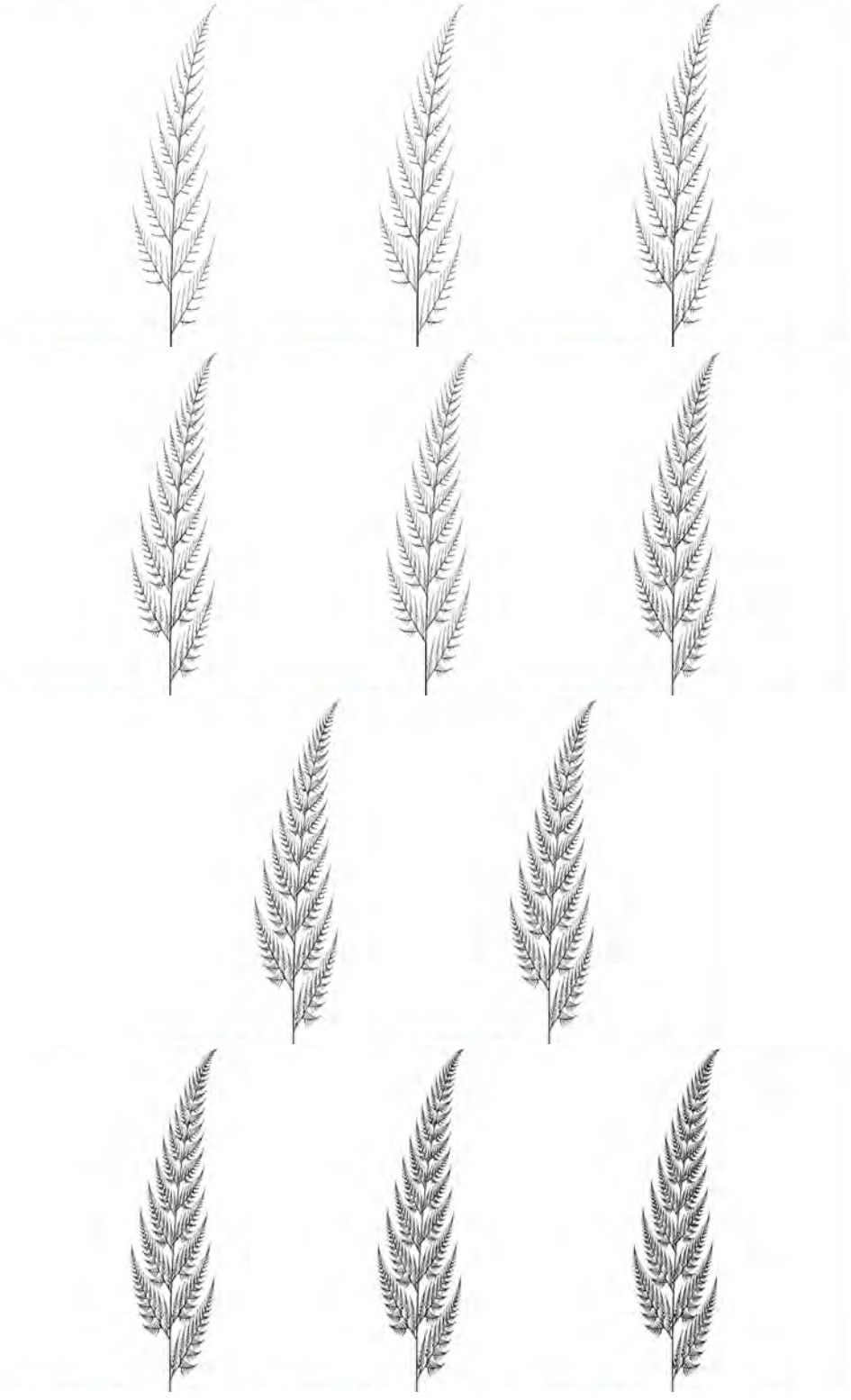
Appendix C

Tree Grammar-based Generated Pictures

C.1 Sample Barnsley Images

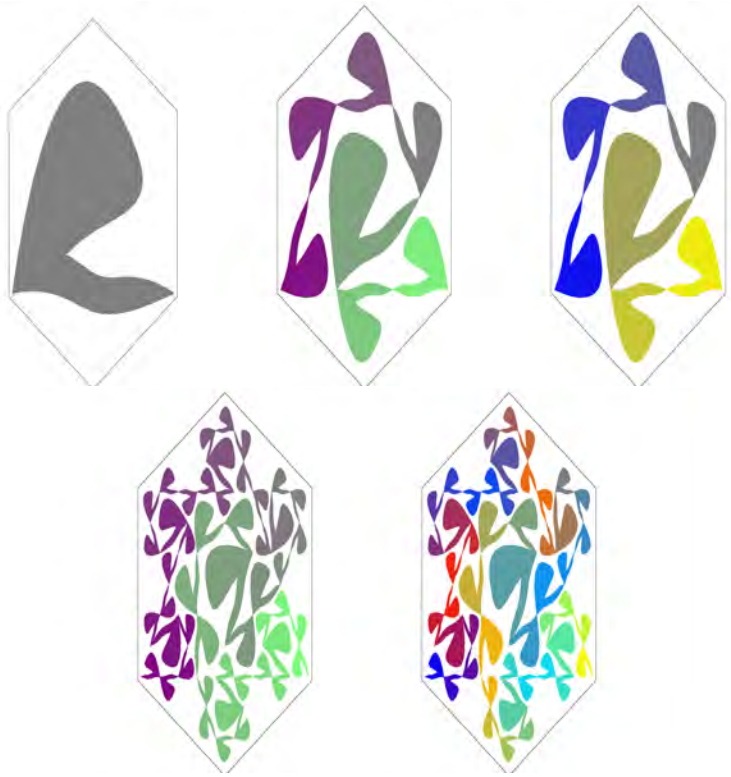


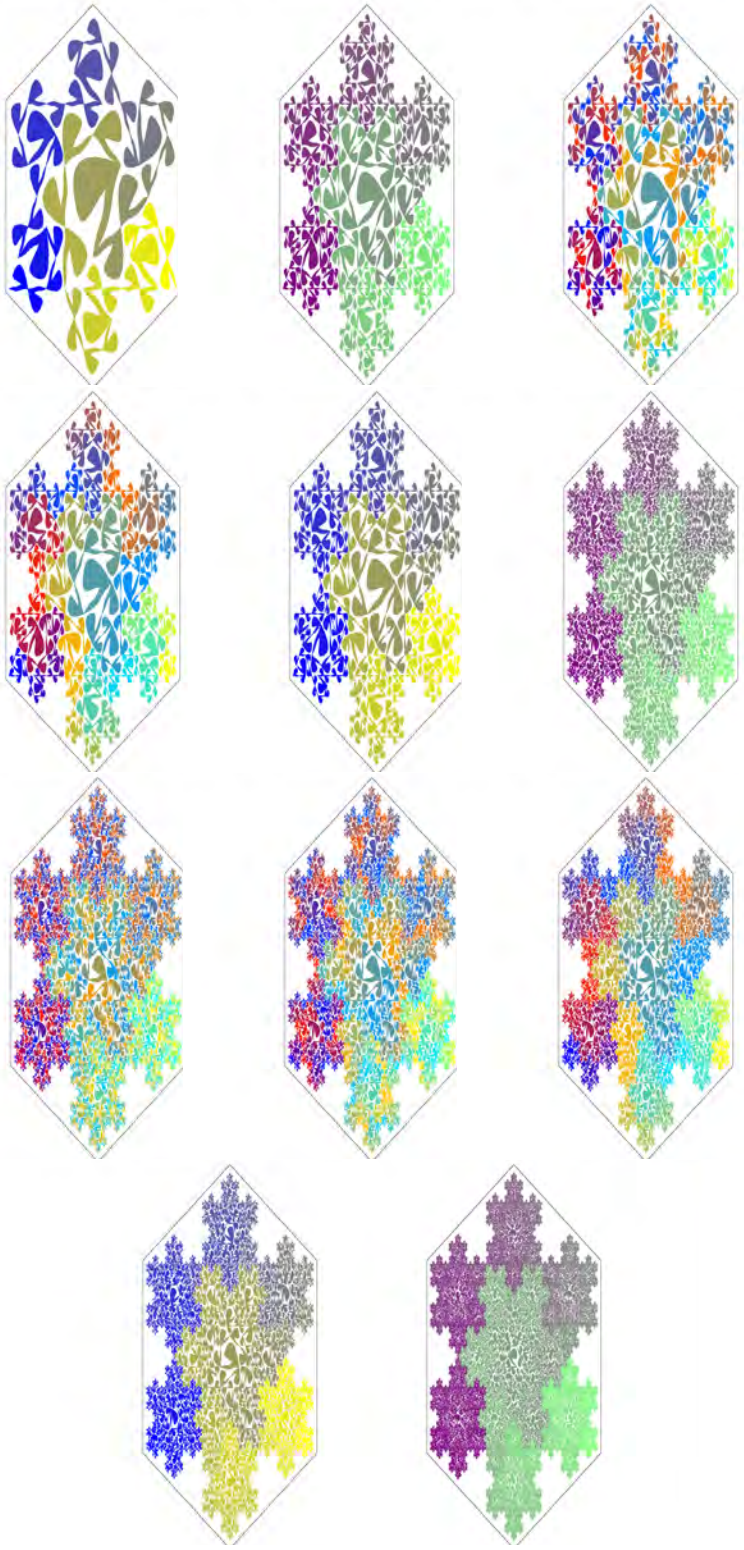


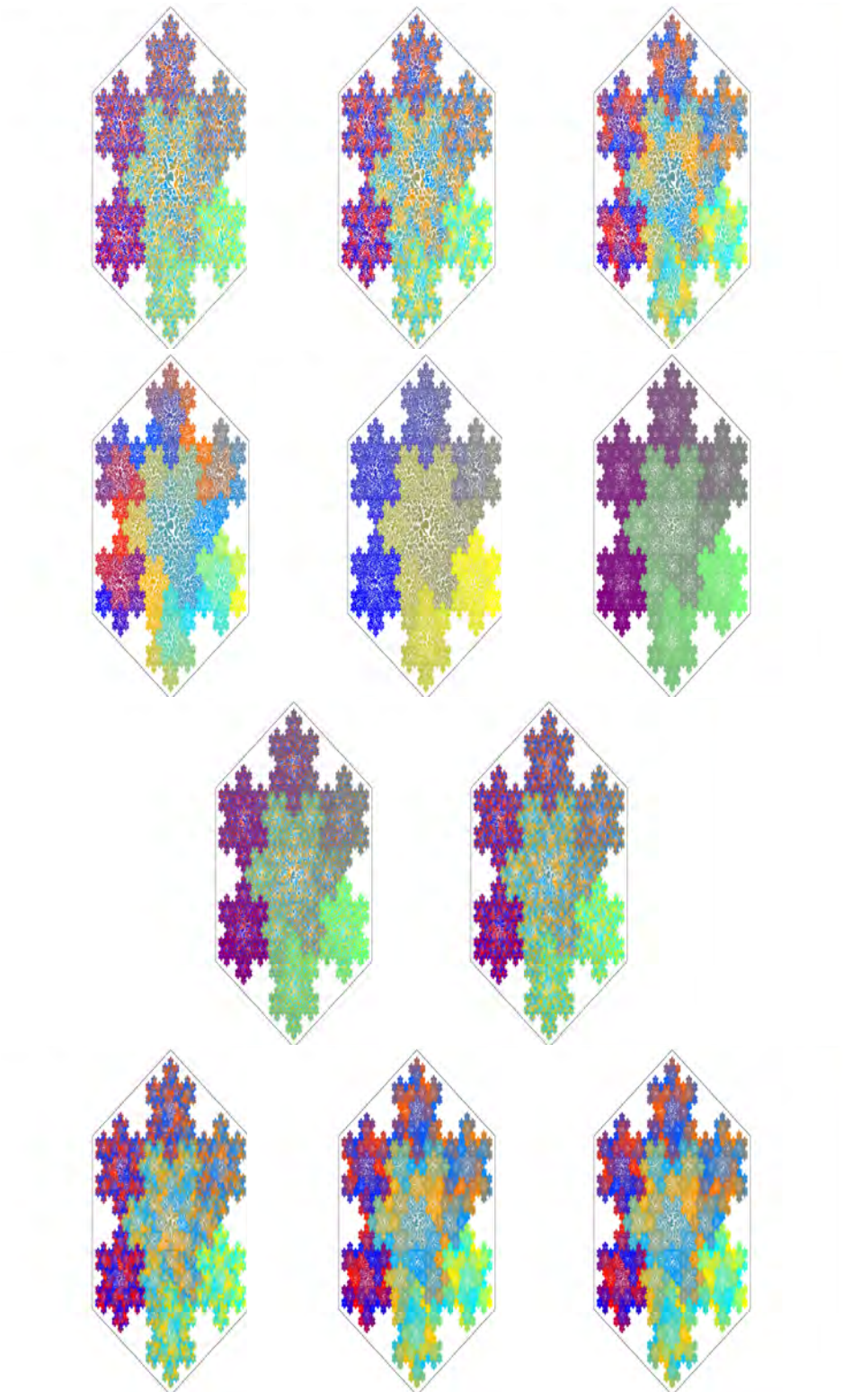


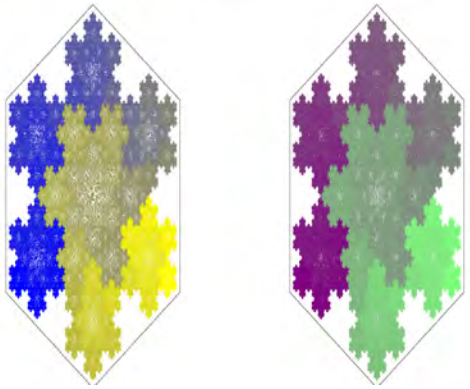


C.2 Sample Mandelbrot Curves









Appendix D

Images for Execution of Prototype in Order to get Similarity Threshold Values

D.1 First Sixteen Images For Each Grammar

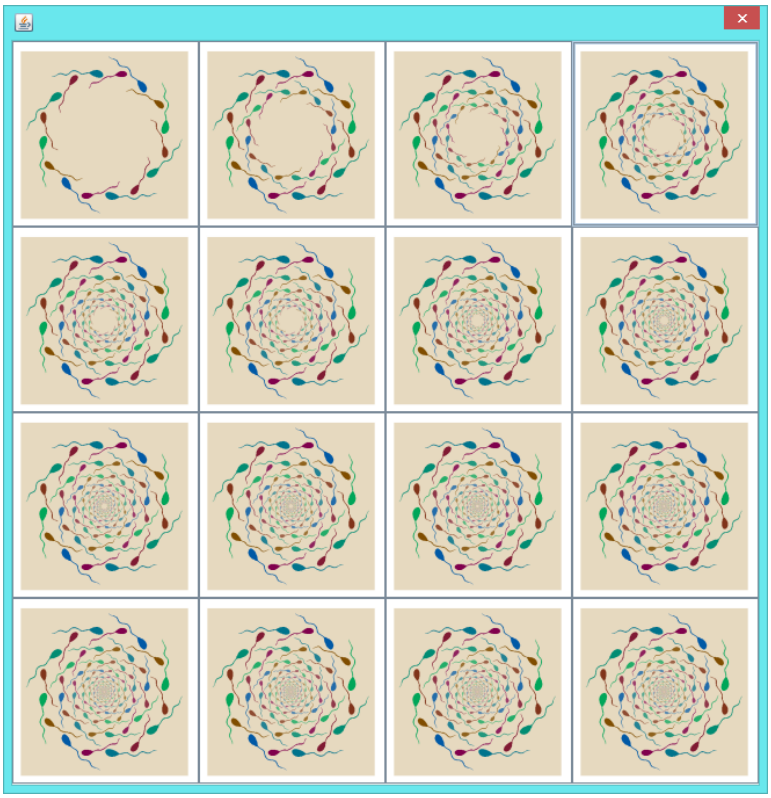


Figure D.1: First sixteen images for grammar 0

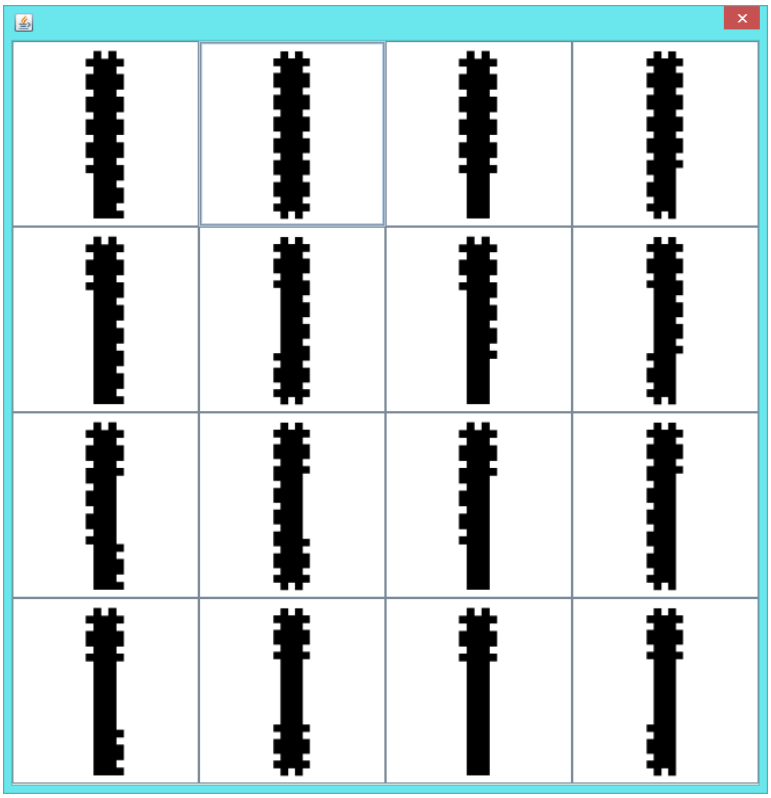


Figure D.2: First sixteen images for grammar 1

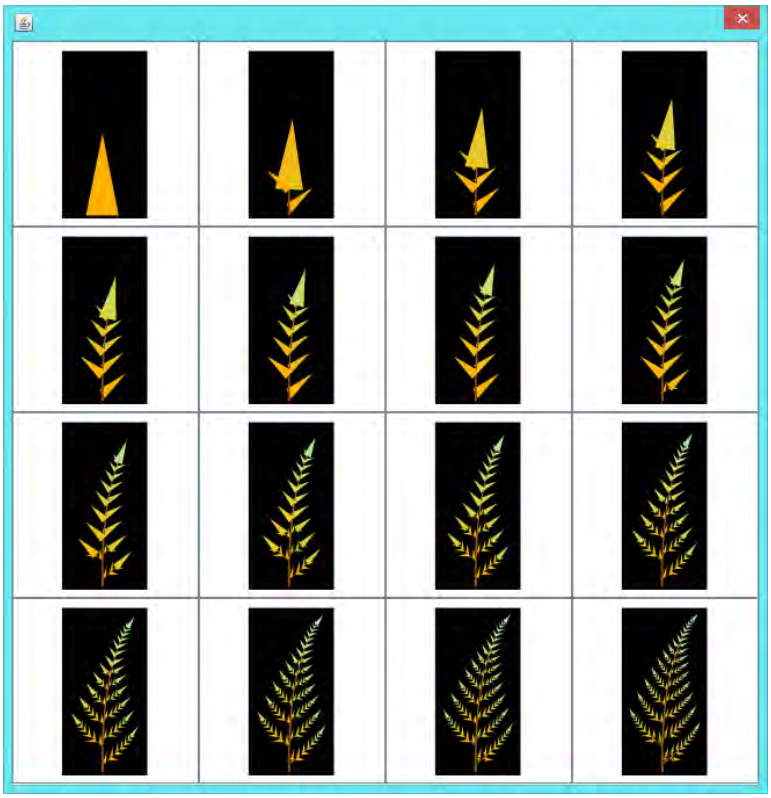


Figure D.3: First sixteen images for grammar 2

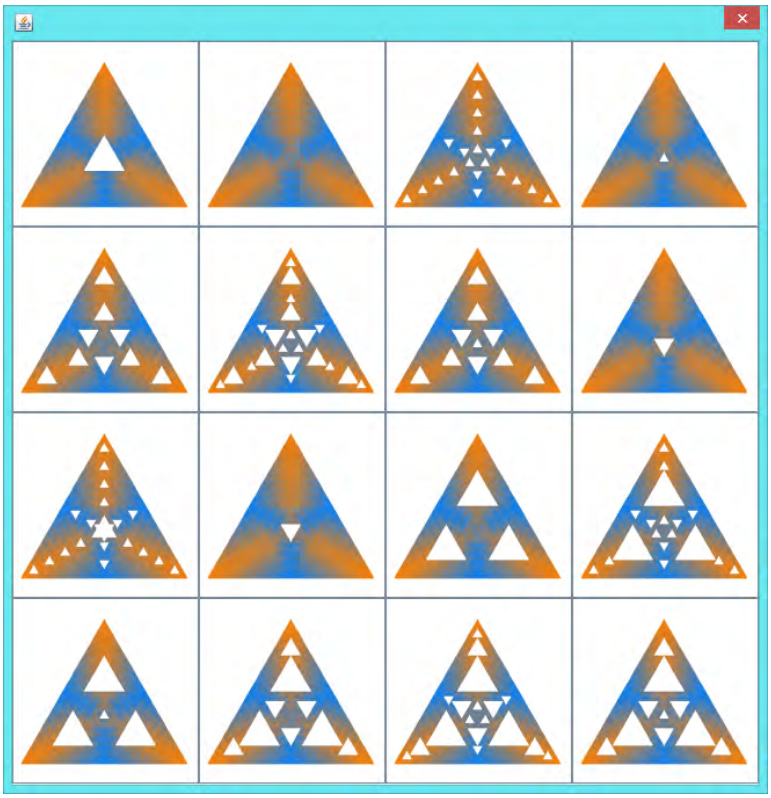


Figure D.4: First sixteen images for grammar 3

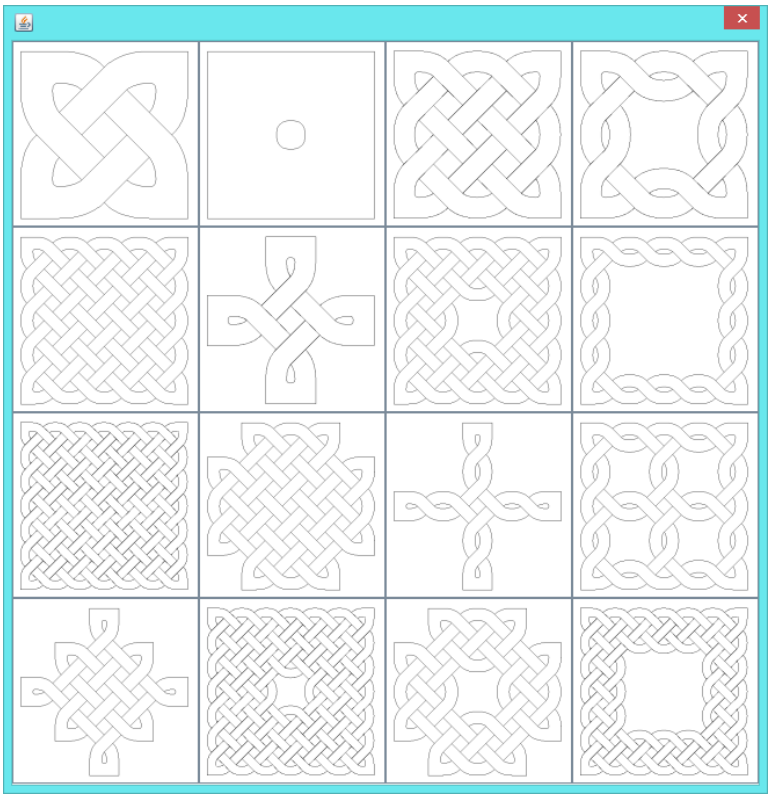


Figure D.5: First sixteen images for grammar 4

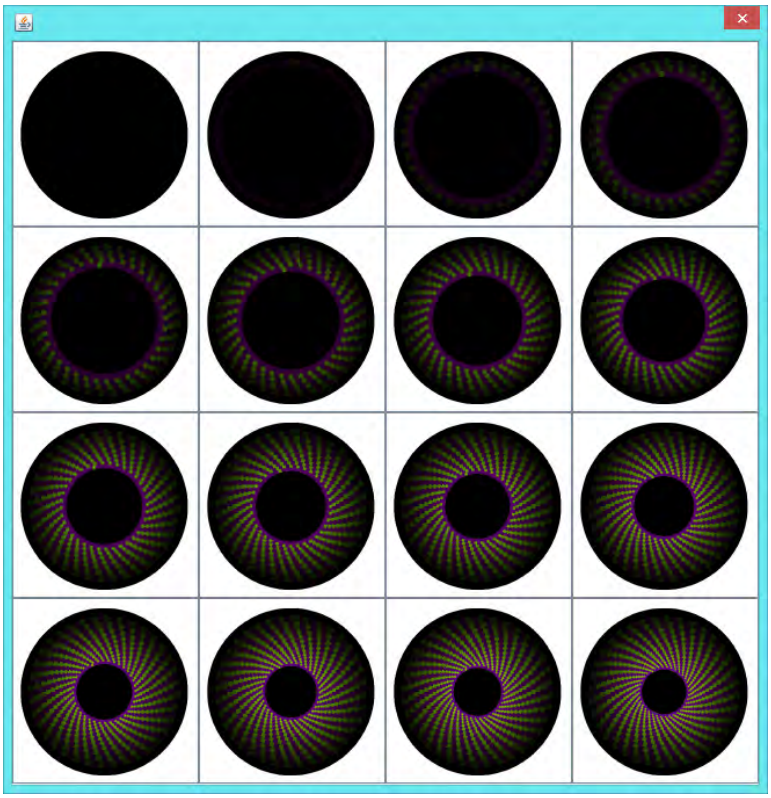


Figure D.6: First sixteen images for grammar 5

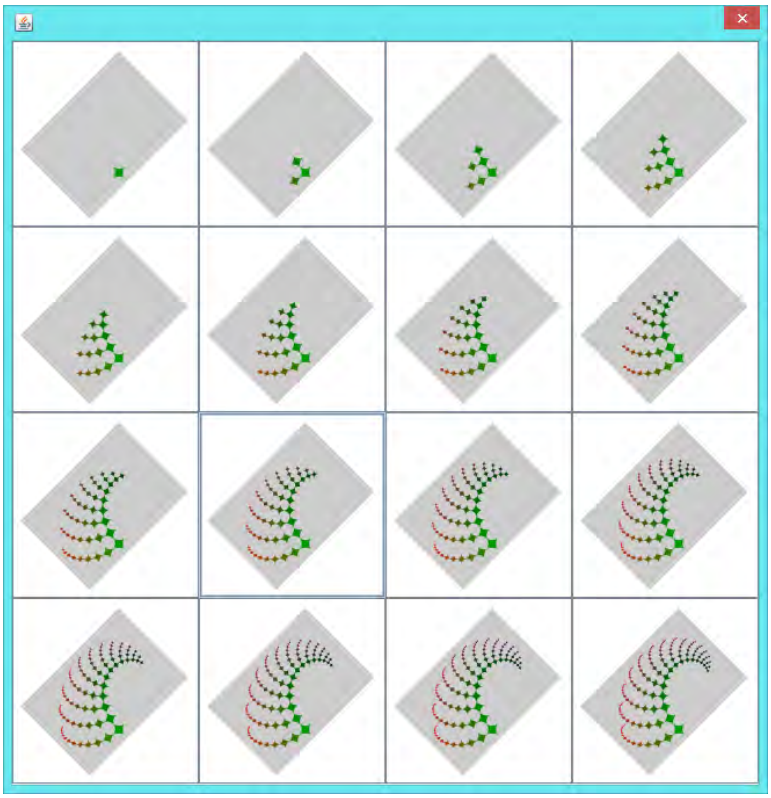


Figure D.7: First sixteen images for grammar 6

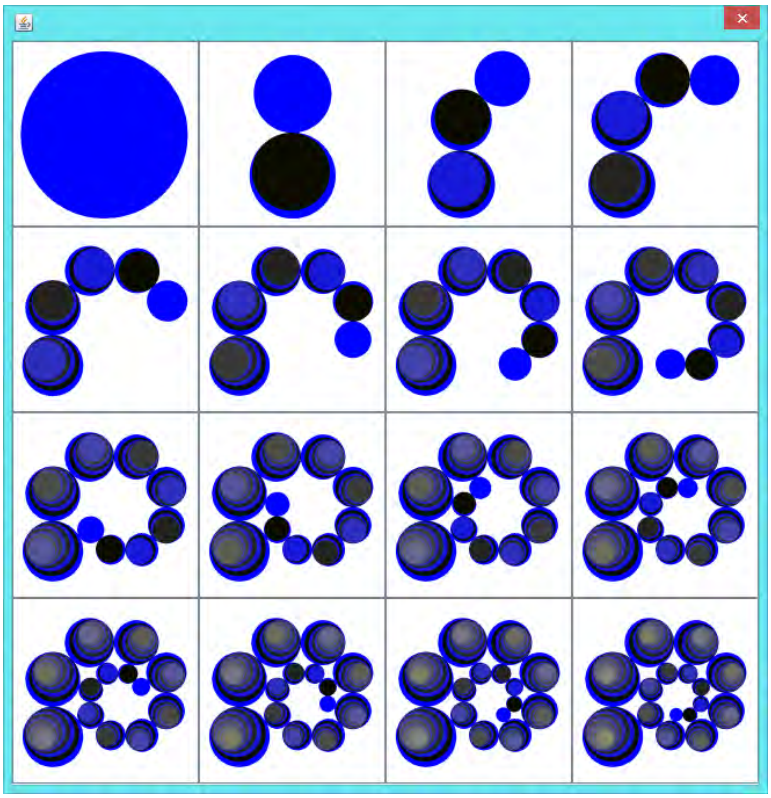


Figure D.8: First sixteen images for grammar 7

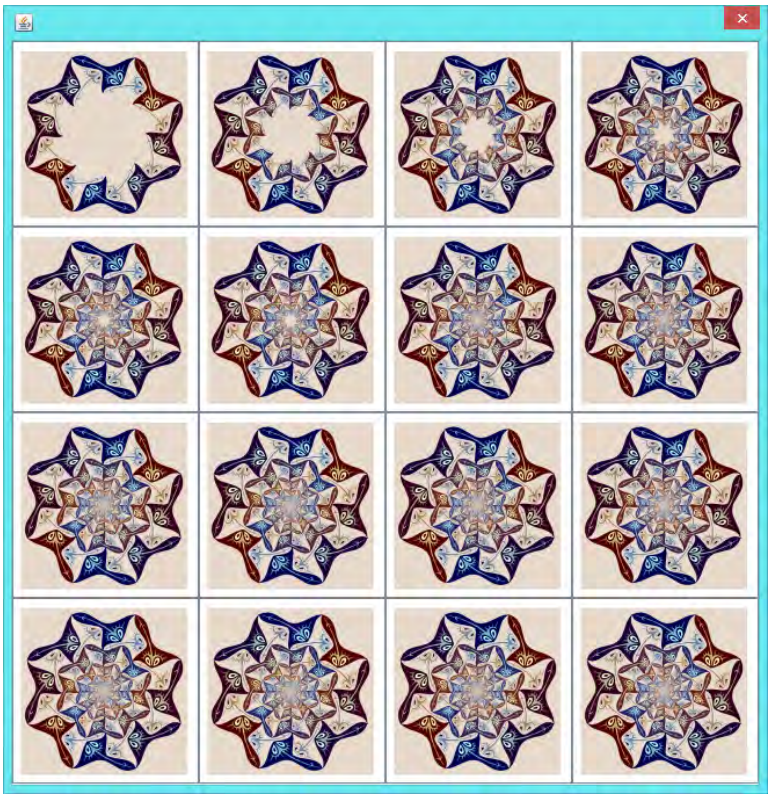


Figure D.9: First sixteen images for grammar 8

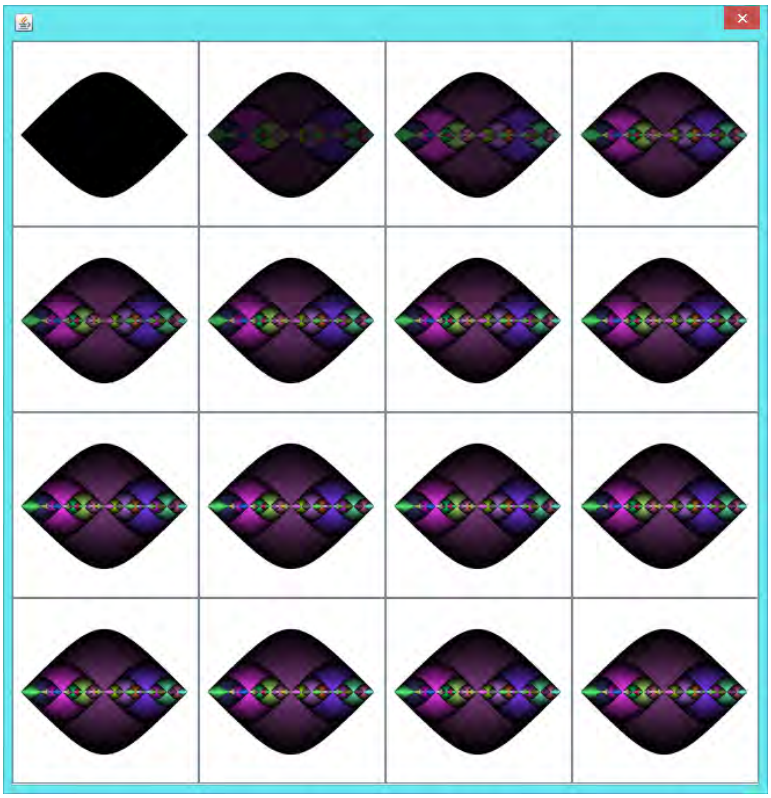


Figure D.10: First sixteen images for grammar 9

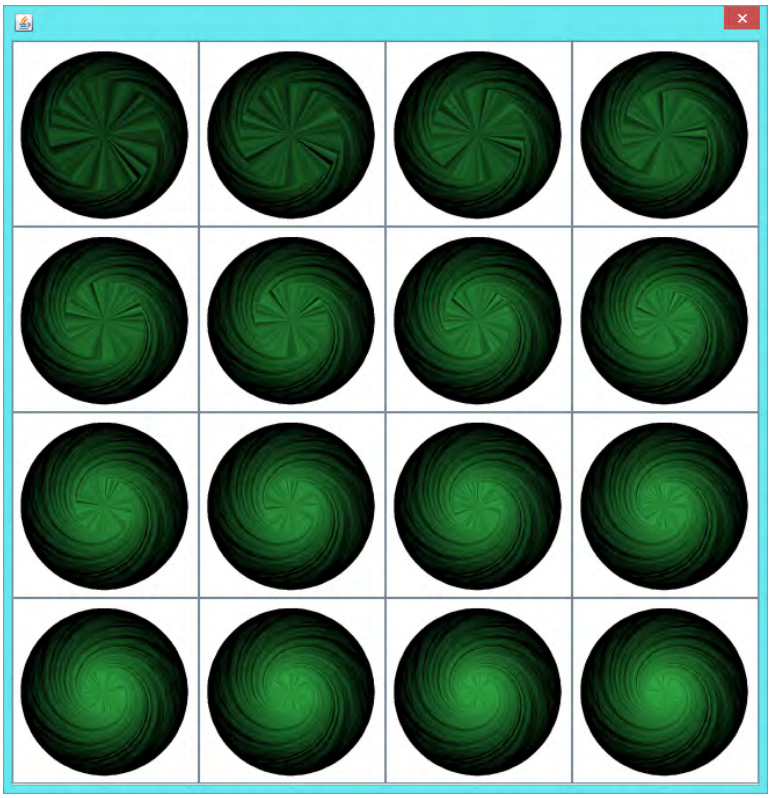


Figure D.11: First sixteen images for grammar 10

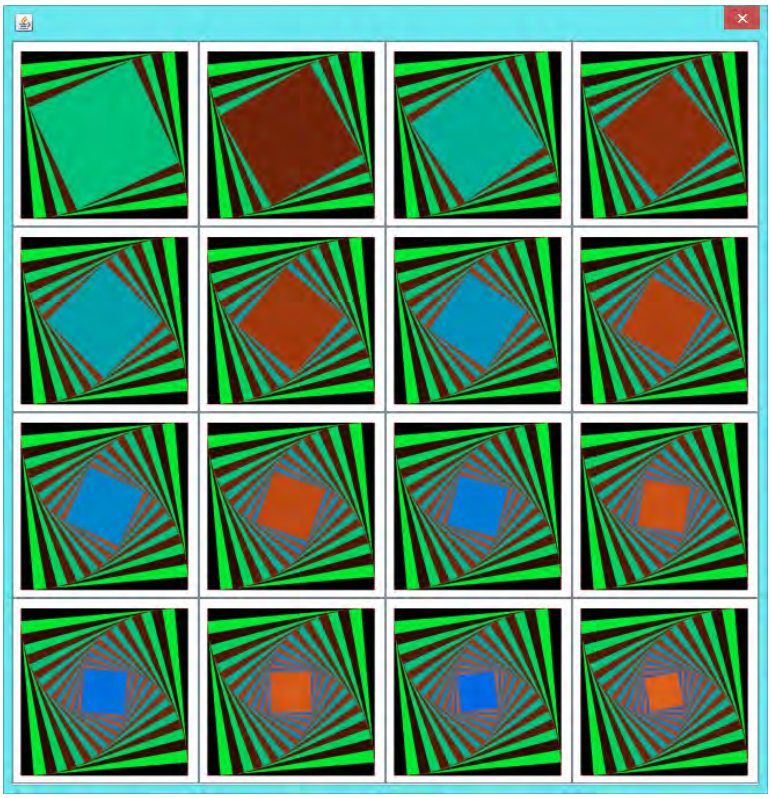


Figure D.12: First sixteen images for grammar 11

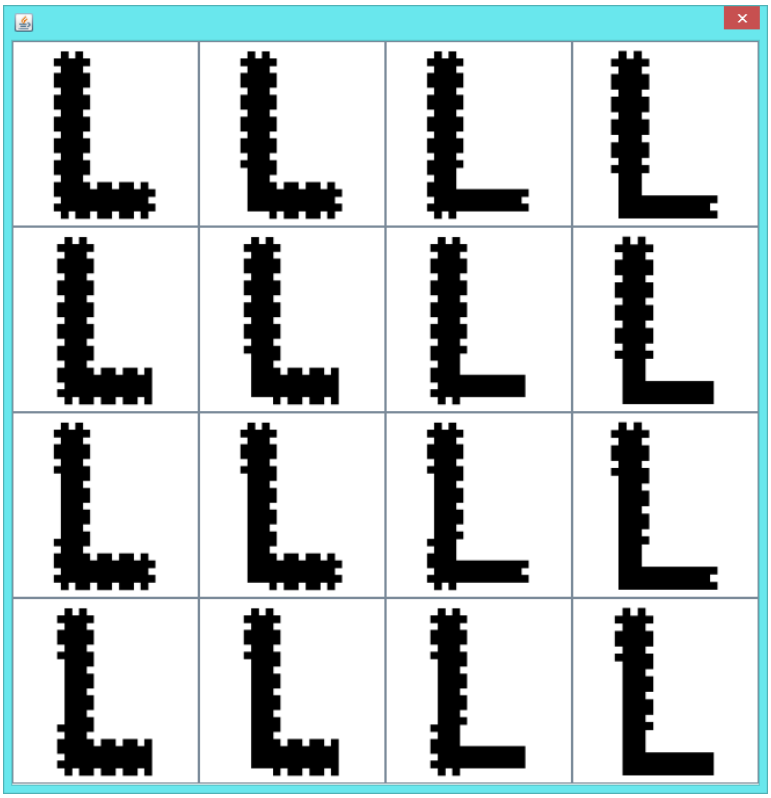


Figure D.13: First sixteen images for grammar 12

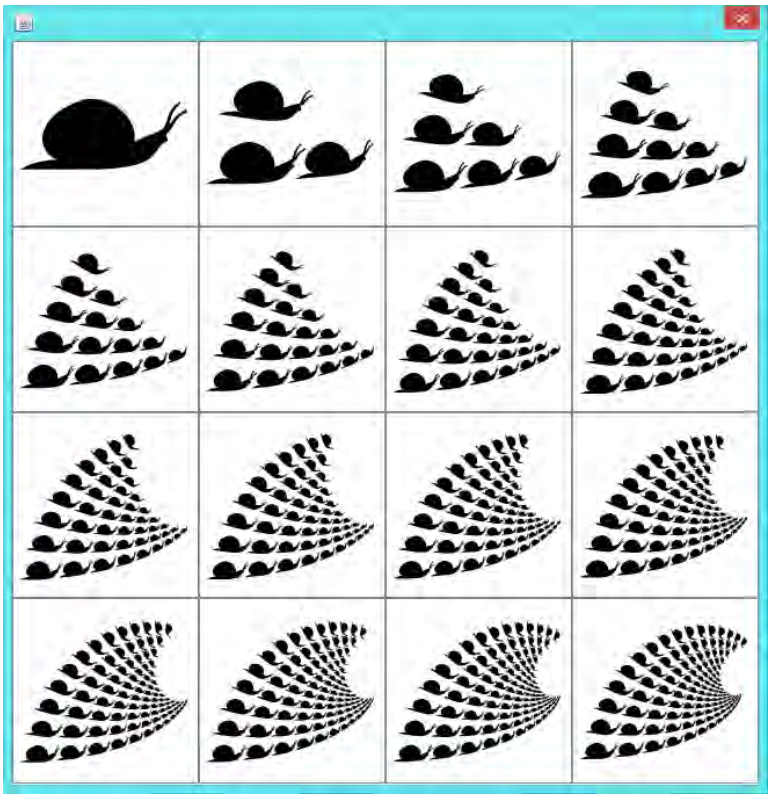


Figure D.14: First sixteen images for grammar 13

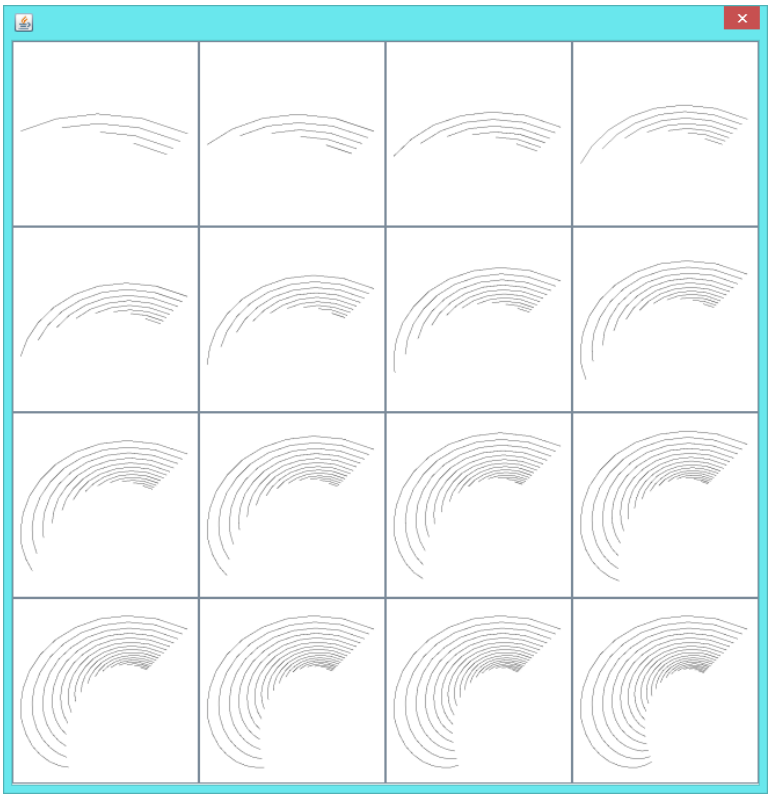


Figure D.15: First sixteen images for grammar 14

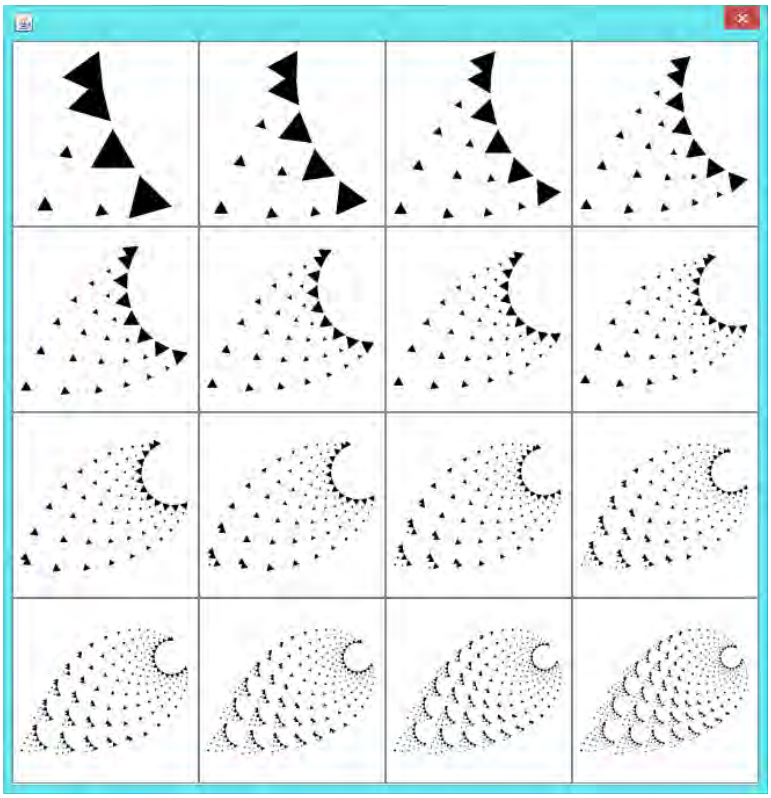


Figure D.16: First sixteen images for grammar 15

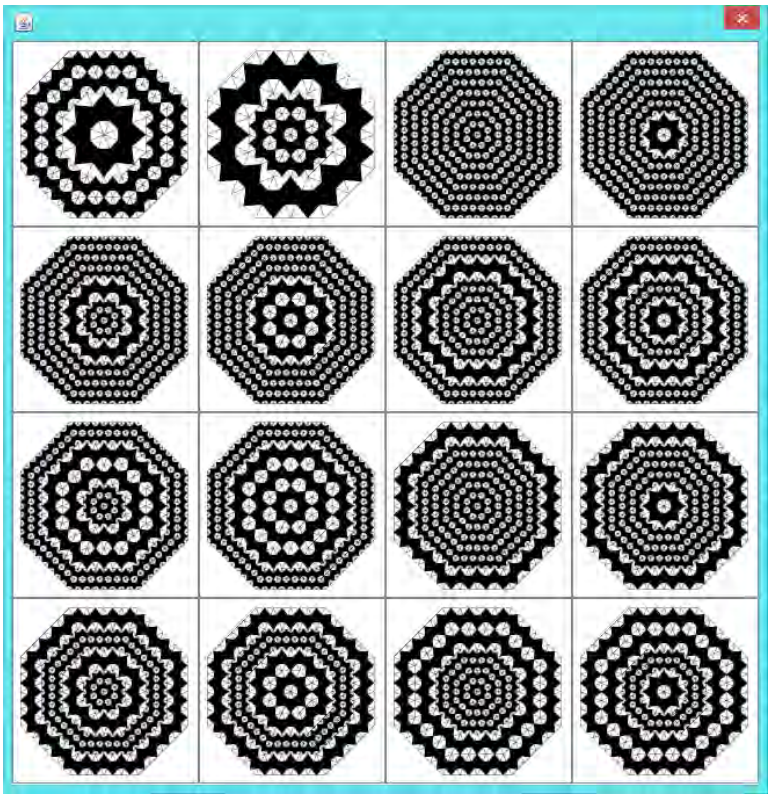


Figure D.17: First sixteen images for grammar 16

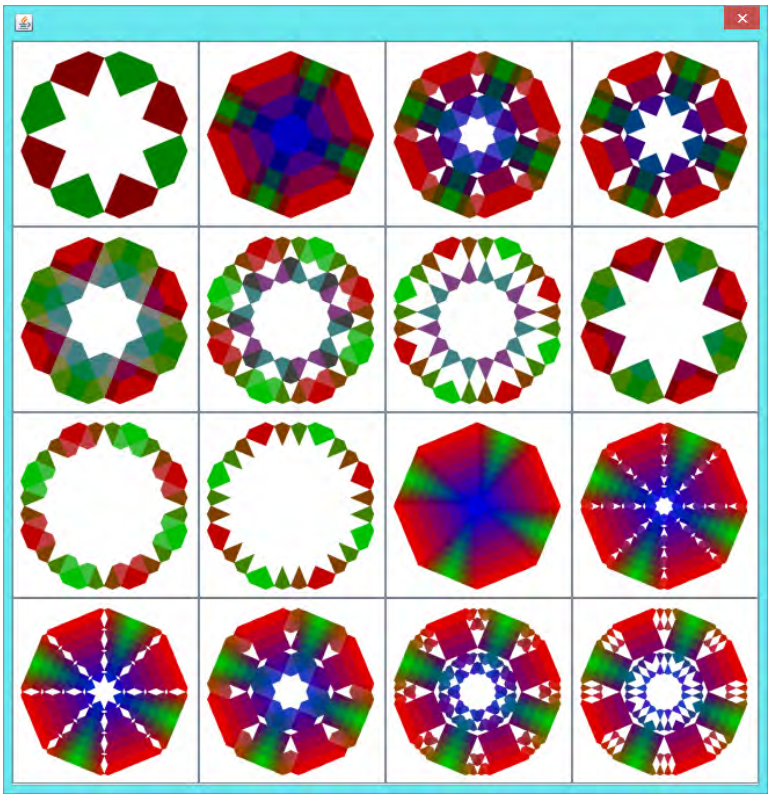


Figure D.18: First sixteen images for grammar 17

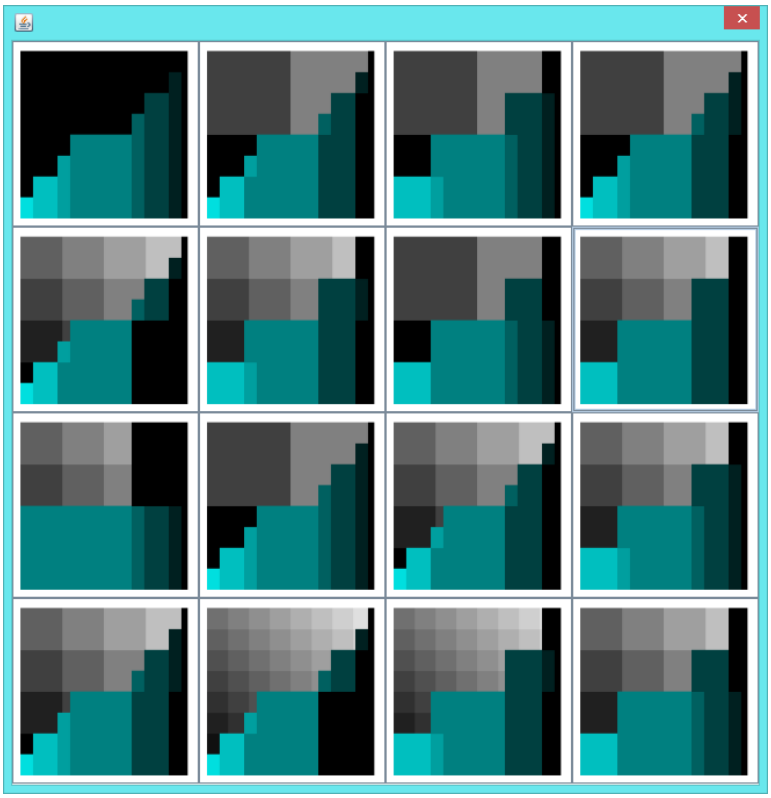


Figure D.19: First sixteen images for grammar 18

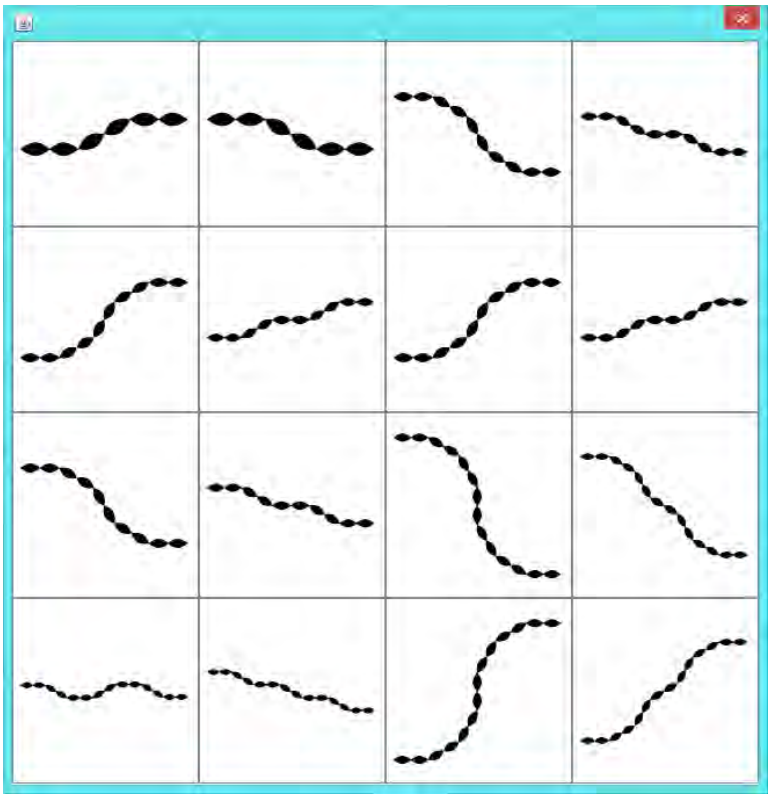


Figure D.20: First sixteen images for grammar 19

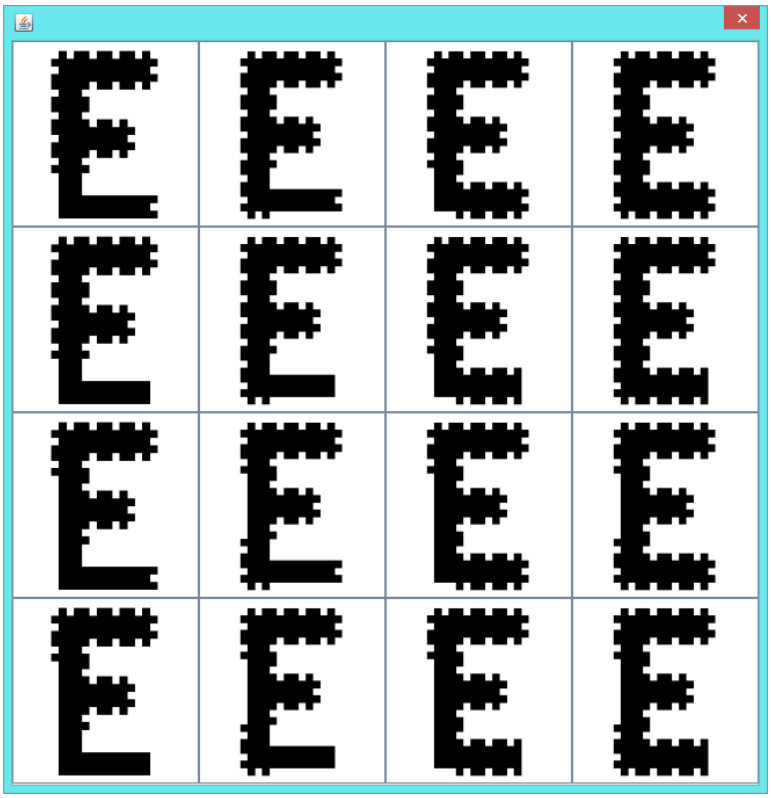


Figure D.21: First sixteen images for grammar 20

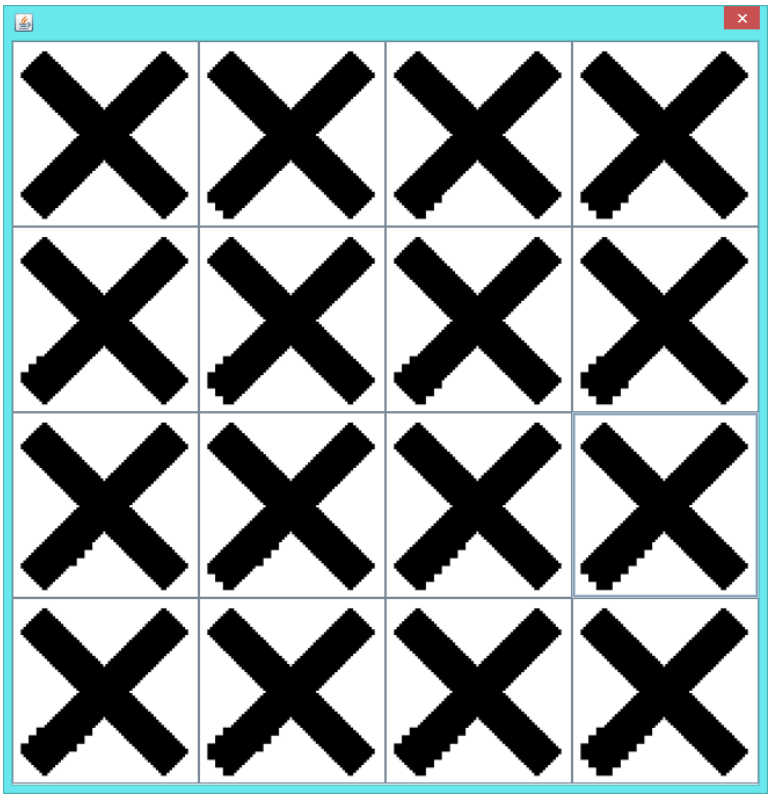


Figure D.22: First sixteen images for grammar 21

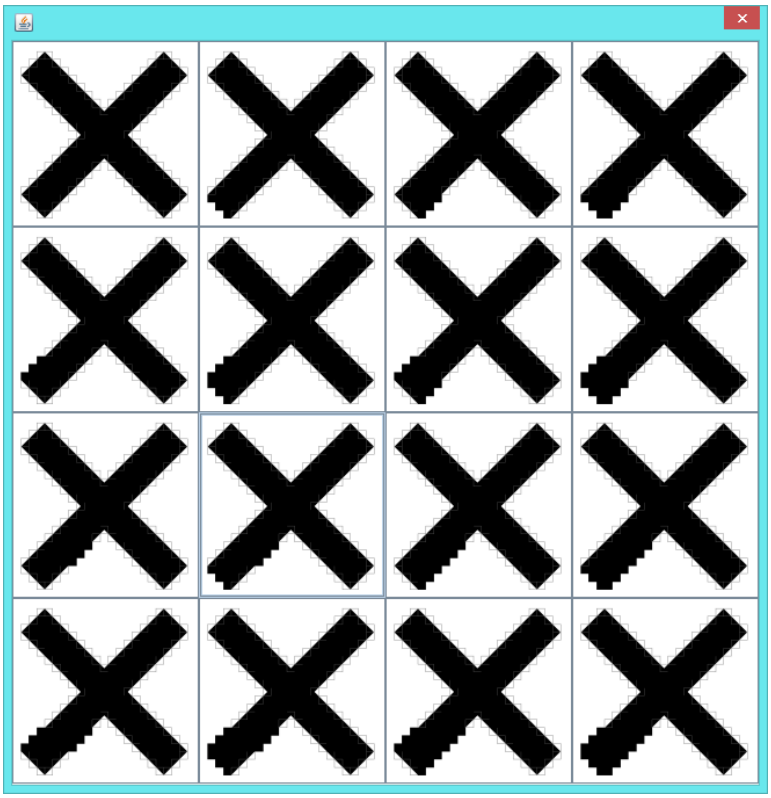


Figure D.23: First sixteen images for grammar 22

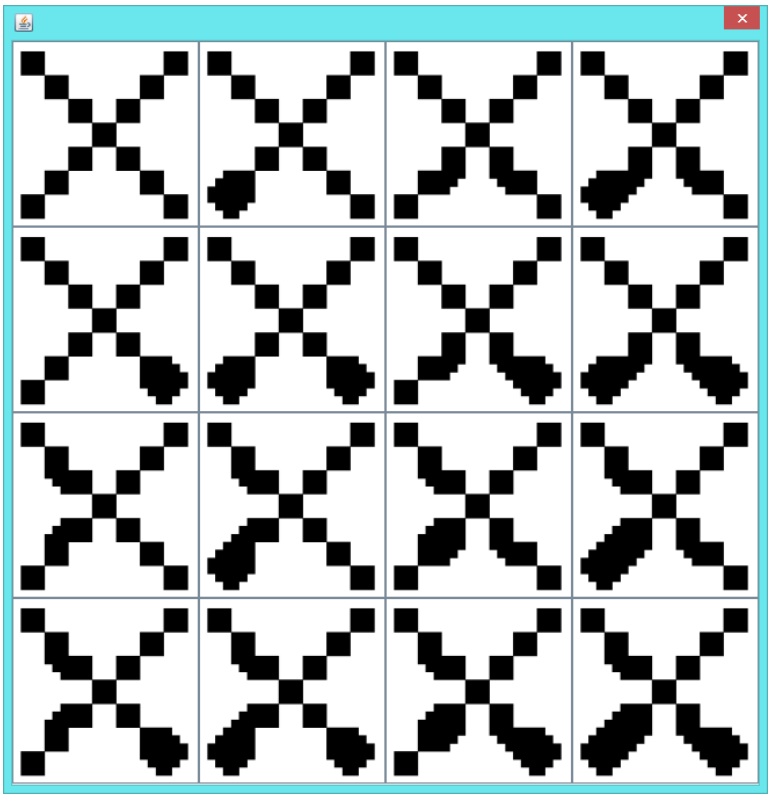


Figure D.24: First sixteen images for grammar 23

D.2 Threshold run values

D.2.1 Normalised similarity values for Grammar 0

D.2.1.1 After Sorting

0.0 0.0
0.47368421052631576 1.0
0.6296296296296297 2.0
0.7142857142857143 3.0
0.7674418604651163 4.0
0.803921568627451 5.0
0.8305084745762712 6.0
0.8507462686567164 7.0
0.8666666666666667 8.0
0.8795180722891566 9.0
0.8901098901098901 10.0
0.898989898989899 11.0
0.9065420560747663 12.0
0.9130434782608695 13.0
0.9186991869918699 14.0
0.9236641221374046 15.0
0.9280575539568345 16.0
= 0.0 0.0

D.2.1.2 After Shuffling

0.47368421052631576 1.0
0.8305084745762712 6.0
0.8507462686567164 7.0
0.8666666666666667 8.0
0.6296296296296297 2.0
0.7142857142857143 3.0
0.803921568627451 5.0
0.7674418604651163 4.0
0.8795180722891566 9.0
0.8901098901098901 10.0
0.898989898989899 11.0
0.9065420560747663 12.0
0.9130434782608695 13.0
0.9186991869918699 14.0
0.9236641221374046 15.0
0.9280575539568345 16.0

D.2.1.3 After Sorting

0.0 8.0
0.10843373493975904 9.0
0.12 7.0
0.18681318681318682 10.0
0.22666666666666666 6.0
0.25252525252525254 11.0
0.308411214953271 12.0
0.3333333333333333 5.0
0.3565217391304348 13.0
0.3983739837398374 14.0
0.4351145038167939 15.0
0.44 4.0
0.4676258992805755 16.0
0.5466666666666666 3.0
0.6533333333333333 2.0

0.76 1.0
0.8666666666666667 0.0

D.2.1.4 After Shuffling

0.12 7.0
0.18681318681318682 10.0
0.3565217391304348 13.0
0.0 8.0
0.3333333333333333 5.0
0.10843373493975904 9.0
0.308411214953271 12.0
0.25252525252525254 11.0
0.22666666666666666 6.0
0.3983739837398374 14.0
0.4351145038167939 15.0
0.44 4.0
0.4676258992805755 16.0
0.5466666666666666 3.0
0.6533333333333333 2.0
0.76 1.0
0.8666666666666667 0.0

D.2.1.5 After Sorting

0.0 8.0
0.06474820143884892 9.0
0.06870229007633588 7.0
0.11564625850340136 10.0
0.1297709923664122 6.0
0.16129032258064516 11.0
0.19083969465648856 5.0
0.20245398773006135 12.0
0.23976608187134502 13.0
0.25190839694656486 4.0
0.2737430167597765 14.0
0.3048128342245989 15.0
0.31297709923664124 3.0
0.3333333333333333 16.0
0.37404580152671757 2.0
0.4351145038167939 1.0
0.4961832061068702 0.0

D.2.1.6 After Shuffling

0.20245398773006135 12.0
0.19083969465648856 5.0
0.06870229007633588 7.0
0.0 8.0
0.11564625850340136 10.0
0.06474820143884892 9.0
0.16129032258064516 11.0
0.1297709923664122 6.0
0.23976608187134502 13.0
0.25190839694656486 4.0
0.2737430167597765 14.0
0.3048128342245989 15.0
0.31297709923664124 3.0
0.3333333333333333 16.0
0.37404580152671757 2.0
0.4351145038167939 1.0
0.4961832061068702 0.0

D.2.2 Normalised similarity values for Grammar 1

D.2.2.1 After Sorting

0.0 0.0
0.10256410256410256 2.0
0.14102564102564102 4.0
0.14102564102564102 8.0
0.14102564102564102 16.0
0.23076923076923078 1.0
0.23076923076923078 6.0
0.23076923076923078 10.0
0.2692307692307692 3.0
0.2692307692307692 7.0
0.2692307692307692 12.0
0.2692307692307692 15.0
0.358974358974359 5.0
0.358974358974359 9.0
0.358974358974359 14.0
0.3974358974358974 11.0
0.48717948717948717 13.0

D.2.2.2 After Shuffling

0.0 0.0
0.2692307692307692 3.0
0.14102564102564102 4.0
0.14102564102564102 16.0
0.10256410256410256 2.0
0.23076923076923078 6.0
0.23076923076923078 10.0
0.23076923076923078 1.0
0.14102564102564102 8.0
0.2692307692307692 7.0
0.2692307692307692 12.0
0.2692307692307692 15.0
0.358974358974359 5.0
0.358974358974359 9.0
0.358974358974359 14.0
0.3974358974358974 11.0
0.48717948717948717 13.0

D.2.2.3 After Sorting

0.0 8.0
0.08333333333333333 2.0
0.08695652173913043 4.0
0.08695652173913043 16.0
0.11594202898550725 10.0
0.14102564102564102 0.0
0.15942028985507245 1.0
0.15942028985507245 6.0
0.15942028985507245 7.0
0.15942028985507245 12.0
0.2318840579710145 3.0
0.2318840579710145 15.0
0.2608695652173913 9.0
0.2608695652173913 14.0
0.30434782608695654 5.0
0.30434782608695654 11.0
0.4057971014492754 13.0

D.2.2.4 After Shuffling

0.15942028985507245 7.0
0.14102564102564102 0.0
0.08695652173913043 16.0
0.08333333333333333 2.0
0.11594202898550725 10.0
0.15942028985507245 1.0
0.08695652173913043 4.0
0.15942028985507245 6.0
0.0 8.0
0.15942028985507245 12.0
0.2318840579710145 3.0
0.2318840579710145 15.0
0.2608695652173913 9.0
0.2608695652173913 14.0
0.30434782608695654 5.0
0.30434782608695654 11.0
0.4057971014492754 13.0

D.2.2.5 After Sorting

0.0 8.0
0.06666666666666667 13.0
0.1 0.0
0.12698412698412698 11.0
0.13333333333333333 5.0
0.13333333333333333 10.0
0.15942028985507245 9.0
0.18333333333333332 12.0
0.18333333333333332 15.0
0.18333333333333332 16.0
0.20634920634920634 3.0
0.21666666666666667 2.0
0.2318840579710145 1.0
0.23333333333333334 4.0
0.25 7.0
0.3 14.0
0.35 6.0

D.2.2.6 After Shuffling

0.12698412698412698 11.0
0.13333333333333333 10.0
0.18333333333333332 12.0
0.15942028985507245 9.0
0.1 0.0
0.13333333333333333 5.0
0.18333333333333332 15.0
0.0 8.0
0.06666666666666667 13.0
0.18333333333333332 16.0
0.20634920634920634 3.0
0.21666666666666667 2.0
0.2318840579710145 1.0
0.23333333333333334 4.0
0.25 7.0
0.3 14.0
0.35 6.0

D.2.3 Normalised similarity values for Grammar 2

D.2.3.1 After Sorting

0.0 0.0
0.4666666666666667 1.0
0.6190476190476191 2.0
0.7037037037037037 3.0
0.7575757575757576 4.0
0.7948717948717948 5.0
0.8431372549019608 6.0
0.8933333333333333 7.0
0.9279279279279279 8.0
0.949685534591195 9.0
0.9634703196347032 10.0
0.9725085910652921 11.0
0.979002624671916 12.0
0.9842209072978304 13.0
0.9884559884559885 14.0
0.9916926272066459 15.0
0.994034302759135 16.0

D.2.3.2 After Shuffling

0.6190476190476191 2.0
0.8933333333333333 7.0
0.0 0.0
0.9279279279279279 8.0
0.7948717948717948 5.0
0.4666666666666667 1.0
0.8431372549019608 6.0
0.7575757575757576 4.0
0.7037037037037037 3.0
0.949685534591195 9.0
0.9634703196347032 10.0
0.9725085910652921 11.0
0.979002624671916 12.0
0.9842209072978304 13.0
0.9884559884559885 14.0
0.9916926272066459 15.0
0.994034302759135 16.0

D.2.3.3 After Sorting

0.0 8.0
0.3081761006289308 9.0
0.3333333333333333 7.0
0.4977168949771689 10.0
0.5495495495495496 6.0
0.6219931271477663 11.0
0.6576576576576577 5.0
0.7112860892388452 12.0
0.7117117117117117 4.0
0.7657657657657657 3.0
0.7830374753451677 13.0
0.8198198198198198 2.0
0.8412698412698413 14.0
0.8738738738738738 1.0
0.885773624091381 15.0
0.9179716629381058 16.0
0.9279279279279279 0.0

D.2.3.4 After Shuffling

0.7117117117117117 4.0
0.0 8.0
0.3333333333333333 7.0
0.5495495495495496 6.0
0.3081761006289308 9.0
0.6219931271477663 11.0
0.6576576576576577 5.0
0.7112860892388452 12.0
0.4977168949771689 10.0
0.7657657657657657 3.0
0.7830374753451677 13.0
0.8198198198198198 2.0
0.8412698412698413 14.0
0.8738738738738738 1.0
0.885773624091381 15.0
0.9179716629381058 16.0
0.9279279279279279 0.0

D.2.3.5 After Sorting

0.0 8.0
0.2814122533748702 7.0
0.2826249067859806 9.0
0.4745586708203531 6.0
0.48028092922744464 10.0
0.6053997923156802 5.0
0.6187078874355926 11.0
0.6988577362409139 4.0
0.7179712694224567 12.0
0.7736240913811008 3.0
0.7913684667100412 13.0
0.8359293873312564 2.0
0.8464975267273017 14.0
0.885773624091381 1.0
0.8877610547194026 15.0
0.9182181416305364 16.0
0.9231568016614745 0.0

D.2.3.6 After Shuffling

0.48028092922744464 10.0
0.6187078874355926 11.0
0.2814122533748702 7.0
0.6053997923156802 5.0
0.2826249067859806 9.0
0.4745586708203531 6.0
0.6988577362409139 4.0
0.7179712694224567 12.0
0.0 8.0
0.7736240913811008 3.0
0.7913684667100412 13.0
0.8359293873312564 2.0
0.8464975267273017 14.0
0.885773624091381 1.0
0.8877610547194026 15.0
0.9182181416305364 16.0
0.9231568016614745 0.0

D.2.4 Normalised similarity values for Grammar 3

D.2.4.1 After Sorting

0.0 0.0
0.014925373134328358 2.0
0.01990049751243781 6.0
0.01990049751243781 8.0
0.15422885572139303 3.0
0.16417910447761194 5.0
0.1724709784411277 1.0
0.18407960199004975 7.0
0.1857379767827529 9.0
0.19900497512437812 11.0
0.20398009950248755 15.0
0.23217247097844113 4.0
0.2835820895522388 12.0
0.2935323383084577 14.0
0.3018242122719735 10.0
0.31343283582089554 16.0
0.3416252072968491 13.0

D.2.4.2 After Shuffling

0.0 0.0
0.1724709784411277 1.0
0.1857379767827529 9.0
0.01990049751243781 6.0
0.16417910447761194 5.0
0.01990049751243781 8.0
0.18407960199004975 7.0
0.15422885572139303 3.0
0.014925373134328358 2.0
0.19900497512437812 11.0
0.20398009950248755 15.0
0.23217247097844113 4.0
0.2835820895522388 12.0
0.2935323383084577 14.0
0.3018242122719735 10.0
0.31343283582089554 16.0
0.3416252072968491 13.0

D.2.4.3 After Sorting

0.0 6.0
0.0 8.0
0.016611295681063124 2.0
0.01990049751243781 0.0
0.1541038525963149 3.0
0.1541038525963149 5.0
0.16917922948073702 1.0
0.17420435510887772 7.0
0.18760469011725292 15.0
0.18927973199329984 9.0
0.18927973199329984 11.0
0.22948073701842547 4.0
0.2847571189279732 12.0
0.2847571189279732 14.0
0.2998324958123953 10.0
0.304857621440536 16.0
0.34003350083752093 13.0

D.2.4.4 After Shuffling

0.0 8.0
0.18760469011725292 15.0
0.0 6.0
0.01990049751243781 0.0
0.016611295681063124 2.0
0.17420435510887772 7.0
0.1541038525963149 5.0
0.16917922948073702 1.0
0.1541038525963149 3.0
0.18927973199329984 9.0
0.18927973199329984 11.0
0.22948073701842547 4.0
0.2847571189279732 12.0
0.2847571189279732 14.0
0.2998324958123953 10.0
0.304857621440536 16.0
0.34003350083752093 13.0

D.2.4.5 After Sorting

0.0 8.0
0.019736842105263157 6.0
0.11428571428571428 7.0
0.12601626016260162 9.0
0.12601626016260162 11.0
0.1267605633802817 5.0
0.1350293542074364 15.0
0.13855421686746988 3.0
0.1488469601677149 10.0
0.15320910973084886 4.0
0.20318725099601595 16.0
0.25609756097560976 12.0
0.25609756097560976 14.0
0.2643884892086331 13.0
0.2847222222222222 1.0
0.2847571189279732 0.0
0.2847571189279732 2.0

D.2.4.6 After Shuffling

0.0 8.0
0.1350293542074364 15.0
0.12601626016260162 11.0
0.1267605633802817 5.0
0.12601626016260162 9.0
0.11428571428571428 7.0
0.019736842105263157 6.0
0.1488469601677149 10.0
0.13855421686746988 3.0
0.15320910973084886 4.0
0.20318725099601595 16.0
0.25609756097560976 12.0
0.25609756097560976 14.0
0.2643884892086331 13.0
0.2847222222222222 1.0
0.2847571189279732 0.0
0.2847571189279732 2.0

D.2.5 Normalised similarity values for Grammar 4

D.2.5.1 After Sorting

0.0 0.0
0.8095238095238095 1.0
0.84 2.0
0.8787878787878788 4.0
0.9245283018867925 3.0
0.9298245614035088 5.0
0.9384615384615385 6.0
0.9384615384615385 8.0
0.9384615384615385 10.0
0.9420289855072463 11.0
0.9420289855072463 13.0
0.9452054794520548 9.0
0.9603960396039604 7.0
0.9619047619047619 12.0
0.9646017699115044 14.0
0.9669421487603306 15.0
0.9757575757575757 16.0

D.2.5.2 After Shuffling

0.9245283018867925 3.0
0.8787878787878788 4.0
0.9384615384615385 10.0
0.8095238095238095 1.0
0.9384615384615385 8.0
0.9298245614035088 5.0
0.0 0.0
0.84 2.0
0.9384615384615385 6.0
0.9420289855072463 11.0
0.9420289855072463 13.0
0.9452054794520548 9.0
0.9603960396039604 7.0
0.9619047619047619 12.0
0.9646017699115044 14.0
0.9669421487603306 15.0
0.9757575757575757 16.0

D.2.5.3 After Sorting

0.0 8.0
0.18840579710144928 13.0
0.4461538461538462 5.0
0.5068493150684932 9.0
0.5076923076923077 3.0
0.5076923076923077 4.0
0.5362318840579711 11.0
0.6460176991150443 14.0
0.6571428571428571 12.0
0.6831683168316832 7.0
0.6923076923076923 6.0
0.7538461538461538 1.0
0.7538461538461538 2.0
0.7538461538461538 10.0
0.806060606060606 16.0
0.8347107438016529 15.0
0.9384615384615385 0.0

D.2.5.4 After Shuffling

0.18840579710144928 13.0
0.6571428571428571 12.0
0.5076923076923077 4.0
0.5068493150684932 9.0
0.4461538461538462 5.0
0.5362318840579711 11.0
0.5076923076923077 3.0
0.6460176991150443 14.0
0.0 8.0
0.6831683168316832 7.
0.6923076923076923 6.0
0.7538461538461538 1.0
0.7538461538461538 2.0
0.7538461538461538 10.0
0.806060606060606 16.0
0.8347107438016529 15.0
0.9384615384615385 0.0

D.2.5.5 After Sorting

0.0 8.0
0.3798449612403101 15.0
0.5371900826446281 7.0
0.6694214876033058 11.0
0.6694214876033058 12.0
0.689922480620155 14.0
0.7024793388429752 5.0
0.7209302325581395 16.0
0.7355371900826446 2.0
0.7355371900826446 3.0
0.768595041322314 0.0
0.768595041322314 10.0
0.768595041322314 13.0
0.7818181818181819 9.0
0.8016528925619835 6.0
0.8347107438016529 1.0
0.8347107438016529 4.0

D.2.5.6 After Shuffling

0.7209302325581395 16.0
0.5371900826446281 7.0
0.7355371900826446 2.0
0.689922480620155 14.0
0.6694214876033058 12.0
0.0 8.0
0.3798449612403101 15.0
0.6694214876033058 11.0
0.7024793388429752 5.0
0.7355371900826446 3.0
0.768595041322314 0.0
0.768595041322314 10.0
0.768595041322314 13.0
0.7818181818181819 9.0
0.8016528925619835 6.0
0.8347107438016529 1.0
0.8347107438016529 4.0

D.2.6 Normalised similarity values for Grammar 5

D.2.6.1 After Sorting

0.0 0.0
0.41509433962264153 1.0
0.5786407766990291 2.0
0.6707132018209409 3.0
0.7297633872976339 4.0
0.7708553326293559 5.0
0.8010999083409716 6.0
0.8242914979757086 7.0
0.8426395939086294 8.0
0.8575180564674983 9.0
0.8698260347930414 10.0
0.8801766979569299 11.0
0.889002557544757 12.0
0.8966174368747022 13.0
0.903254569772626 14.0
0.9090909090909091 15.0
0.9142631370999605 16.0

D.2.6.2 After Shuffling

0.8426395939086294 8.0
0.8242914979757086 7.0
0.6707132018209409 3.0
0.7708553326293559 5.0
0.41509433962264153 1.0
0.0 0.0
0.7297633872976339 4.0
0.8010999083409716 6.0
0.5786407766990291 2.0
0.8575180564674983 9.0
0.8698260347930414 10.0
0.8801766979569299 11.0
0.889002557544757 12.0
0.8966174368747022 13.0
0.903254569772626 14.0
0.9090909090909091 15.0
0.9142631370999605 16.0

D.2.6.3 After Sorting

0.0 8.0
0.10111621799080761 9.0
0.1116751269035533 7.0
0.17876424715056988 10.0
0.21609862218999276 6.0
0.24406405300938708 11.0
0.2997442455242967 12.0
0.3205221174764322 5.0
0.34778465936160075 13.0
0.3896567097637093 14.0
0.4249456127628716 4.0
0.42647674905739424 15.0
0.4591070723034374 16.0
0.5293691080493111 3.0
0.6337926033357505 2.0
0.73821609862219 1.0
0.8426395939086294 0.0

D.2.6.4 After Shuffling

0.34778465936160075 13.0
0.1116751269035533 7.0
0.3205221174764322 5.0
0.17876424715056988 10.0
0.2997442455242967 12.0
0.10111621799080761 9.0
0.21609862218999276 6.0
0.0 8.0
0.24406405300938708 11.0
0.3896567097637093 14.0
0.4249456127628716 4.0
0.42647674905739424 15.0
0.4591070723034374 16.0
0.5293691080493111 3.0
0.6337926033357505 2.0
0.73821609862219 1.0
0.8426395939086294 0.0

D.2.6.5 After Sorting

0.0 8.0
0.060845515606479654 9.0
0.06451612903225806 7.0
0.11140186915887851 10.0
0.12484289903644742 6.0
0.15679318907413978 11.0
0.18516966904063678 5.0
0.19777252784340196 12.0
0.2349533311876408 13.0
0.24549643904482615 4.0
0.2688403568132882 14.0
0.2998527245949926 15.0
0.3058232090490155 3.0
0.32834133936140153 16.0
0.36614997905320484 2.0
0.42647674905739424 1.0
0.4868035190615836 0.0

D.2.6.6 After Shuffling

0.15679318907413978 11.0
0.060845515606479654 9.0
0.18516966904063678 5.0
0.11140186915887851 10.0
0.12484289903644742 6.0
0.06451612903225806 7.0
0.2349533311876408 13.0
0.19777252784340196 12.0
0.0 8.0
0.24549643904482615 4.0
0.2688403568132882 14.0
0.2998527245949926 15.0
0.3058232090490155 3.0
0.32834133936140153 16.0
0.36614997905320484 2.0
0.42647674905739424 1.0
0.4868035190615836 0.0

D.2.7 Normalised similarity values for Grammar 6

D.2.7.1 After Sorting

0.0 0.0
0.6153846153846154 1.0
0.7619047619047619 2.0
0.8387096774193549 3.0
0.8837209302325582 4.0
0.9122807017543859 5.0
0.9315068493150684 6.0
0.945054945054945 7.0
0.954954954954955 8.0
0.9624060150375939 9.0
0.9681528662420382 10.0
0.9726775956284153 11.0
0.976303317535545 12.0
0.979253112033195 13.0
0.9816849816849816 14.0
0.9837133550488599 15.0
0.9854227405247813 16.0

D.2.7.2 After Shuffling

0.6153846153846154 1.0
0.945054945054945 7.0
0.7619047619047619 2.0
0.9315068493150684 6.0
0.8837209302325582 4.0
0.0 0.0
0.9122807017543859 5.0
0.954954954954955 8.0
0.8387096774193549 3.0
0.9624060150375939 9.0
0.9681528662420382 10.0
0.9726775956284153 11.0
0.976303317535545 12.0
0.979253112033195 13.0
0.9816849816849816 14.0
0.9837133550488599 15.0
0.9854227405247813 16.0

D.2.7.3 After Sorting

0.0 8.0
0.18045112781954886 9.0
0.1981981981981982 7.0
0.3057324840764331 10.0
0.36036036036036034 6.0
0.40437158469945356 11.0
0.4834123222748815 12.0
0.5045045045045045 5.0
0.5477178423236515 13.0
0.6007326007326007 14.0
0.6306306306306306 4.0
0.6449511400651465 15.0
0.6822157434402333 16.0
0.7387387387387387 3.0
0.8288288288288288 2.0
0.9009009009009009 1.0
0.954954954954955 0.0

D.2.7.4 After Shuffling

0.0 8.0
0.36036036036036034 6.0
0.5477178423236515 13.0
0.3057324840764331 10.0
0.18045112781954886 9.0
0.40437158469945356 11.0
0.5045045045045045 5.0
0.4834123222748815 12.0
0.1981981981981982 7.0
0.6007326007326007 14.0
0.6306306306306306 4.0
0.6449511400651465 15.0
0.6822157434402333 16.0
0.7387387387387387 3.0
0.8288288288288288 2.0
0.9009009009009009 1.0
0.954954954954955 0.0

D.2.7.5 After Sorting

0.0 8.0
0.11078717201166181 9.0
0.11726384364820847 7.0
0.1994750656167979 10.0
0.22149837133550487 6.0
0.2755344418052256 11.0
0.31921824104234525 5.0
0.3412526997840173 12.0
0.398422090729783 13.0
0.41042345276872966 4.0
0.4484629294755877 14.0
0.4925124792013311 15.0
0.495114006514658 3.0
0.5314900153609831 16.0
0.5732899022801303 2.0
0.6449511400651465 1.0
0.7100977198697068 0.0

D.2.7.6 After Shuffling

0.398422090729783 13.0
0.31921824104234525 5.0
0.0 8.0
0.22149837133550487 6.0
0.11726384364820847 7.0
0.2755344418052256 11.0
0.1994750656167979 10.0
0.3412526997840173 12.0
0.11078717201166181 9.0
0.41042345276872966 4.0
0.4484629294755877 14.0
0.4925124792013311 15.0
0.495114006514658 3.0
0.5314900153609831 16.0
0.5732899022801303 2.0
0.6449511400651465 1.0
0.7100977198697068 0.0

D.2.8 Normalised similarity values for Grammar 7

D.2.8.1 After Sorting

0.0 0.0
0.5384615384615384 1.0
0.7142857142857143 2.0
0.8064516129032258 3.0
0.8604651162790697 4.0
0.8947368421052632 5.0
0.9178082191780822 6.0
0.9340659340659341 7.0
0.9459459459459459 8.0
0.9548872180451128 9.0
0.9617834394904459 10.0
0.9672131147540983 11.0
0.9715639810426541 12.0
0.975103734439834 13.0
0.978021978021978 14.0
0.9804560260586319 15.0
0.9825072886297376 16.0

D.2.8.2 After Shuffling

0.7142857142857143 2.0
0.8947368421052632 5.0
0.9459459459459459 8.0
0.0 0.0
0.9340659340659341 7.0
0.8604651162790697 4.0
0.9178082191780822 6.0
0.8064516129032258 3.0
0.5384615384615384 1.0
0.9548872180451128 9.0
0.9617834394904459 10.0
0.9672131147540983 11.0
0.9715639810426541 12.0
0.975103734439834 13.0
0.978021978021978 14.0
0.9804560260586319 15.0
0.9825072886297376 16.0

D.2.8.3 After Sorting

0.0 8.0
0.17293233082706766 9.0
0.1891891891891892 7.0
0.29936305732484075 10.0
0.35135135135135137 6.0
0.3989071038251366 11.0
0.4786729857819905 12.0
0.4954954954954955 5.0
0.5435684647302904 13.0
0.5970695970695971 14.0
0.6216216216216216 4.0
0.6416938110749185 15.0
0.6793002915451894 16.0
0.7297297297297297 3.0
0.8198198198198198 2.0
0.8918918918918919 1.0
0.9459459459459459 0.0

D.2.8.4 After Shuffling

0.1891891891891892 7.0
0.0 8.0
0.3989071038251366 11.0
0.29936305732484075 10.0
0.4954954954954955 5.0
0.5435684647302904 13.0
0.4786729857819905 12.0
0.17293233082706766 9.0
0.35135135135135137 6.0
0.5970695970695971 14.0
0.6216216216216216 4.0
0.6416938110749185 15.0
0.6793002915451894 16.0
0.7297297297297297 3.0
0.8198198198198198 2.0
0.8918918918918919 1.0
0.9459459459459459 0.0

D.2.8.5 After Sorting

0.0 8.0
0.10787172011661808 9.0
0.11400651465798045 7.0
0.1968503937007874 10.0
0.2182410423452769 6.0
0.27315914489311166 11.0
0.31596091205211724 5.0
0.3390928725701944 12.0
0.39644970414201186 13.0
0.40716612377850164 4.0
0.44665461121157324 14.0
0.4908485856905158 15.0
0.49185667752442996 3.0
0.5299539170506913 16.0
0.5700325732899023 2.0
0.6416938110749185 1.0
0.7068403908794788 0.0

D.2.8.6 After Shuffling

0.2182410423452769 6.0
0.3390928725701944 12.0
0.27315914489311166 11.0
0.11400651465798045 7.0
0.1968503937007874 10.0
0.31596091205211724 5.0
0.10787172011661808 9.0
0.39644970414201186 13.0
0.0 8.0
0.40716612377850164 4.0
0.44665461121157324 14.0
0.4908485856905158 15.0
0.49185667752442996 3.0
0.5299539170506913 16.0
0.5700325732899023 2.0
0.6416938110749185 1.0
0.7068403908794788 0.0

D.2.9 Normalised similarity values for Grammar 8

D.2.9.1 After Sorting

0.0 0.0
0.47368421052631576 1.0
0.6296296296296297 2.0
0.7142857142857143 3.0
0.7674418604651163 4.0
0.803921568627451 5.0
0.8305084745762712 6.0
0.8507462686567164 7.0
0.8666666666666667 8.0
0.8795180722891566 9.0
0.8901098901098901 10.0
0.898989898989899 11.0
0.9065420560747663 12.0
0.9130434782608695 13.0
0.9186991869918699 14.0
0.9236641221374046 15.0
0.9280575539568345 16.0

D.2.9.2 After Shuffling

0.8507462686567164 7.0
0.7674418604651163 4.0
0.0 0.0
0.47368421052631576 1.0
0.8666666666666667 8.0
0.803921568627451 5.0
0.8305084745762712 6.0
0.7142857142857143 3.0
0.6296296296296297 2.0
0.8795180722891566 9.0
0.8901098901098901 10.0
0.898989898989899 11.0
0.9065420560747663 12.0
0.9130434782608695 13.0
0.9186991869918699 14.0
0.9236641221374046 15.0
0.9280575539568345 16.0

D.2.9.3 After Sorting

0.0 8.0
0.10843373493975904 9.0
0.12 7.0
0.18681318681318682 10.0
0.22666666666666666 6.0
0.25252525252525254 11.0
0.308411214953271 12.0
0.3333333333333333 5.0
0.3565217391304348 13.0
0.3983739837398374 14.0
0.4351145038167939 15.0
0.44 4.0
0.4676258992805755 16.0
0.5466666666666666 3.0
0.6533333333333333 2.0
0.76 1.0
0.8666666666666667 0.0

D.2.9.4 After Shuffling

0.18681318681318682 10.0
0.10843373493975904 9.0
0.3333333333333333 5.0
0.0 8.0
0.12 7.0
0.3565217391304348 13.0
0.25252525252525254 11.0
0.22666666666666666 6.0
0.308411214953271 12.0
0.3983739837398374 14.0
0.4351145038167939 15.0
0.44 4.0
0.4676258992805755 16.0
0.5466666666666666 3.0
0.6533333333333333 2.0
0.76 1.0
0.8666666666666667 0.0

D.2.9.5 After Sorting

0.0 8.0
0.06474820143884892 9.0
0.06870229007633588 7.0
0.11564625850340136 10.0
0.1297709923664122 6.0
0.16129032258064516 11.0
0.19083969465648856 5.0
0.20245398773006135 12.0
0.23976608187134502 13.0
0.25190839694656486 4.0
0.2737430167597765 14.0
0.3048128342245989 15.0
0.31297709923664124 3.0
0.3333333333333333 16.0
0.37404580152671757 2.0
0.4351145038167939 1.0
0.4961832061068702 0.0

D.2.9.6 After Shuffling

0.23976608187134502 13.0
0.06474820143884892 9.0
0.06870229007633588 7.0
0.20245398773006135 12.0
0.11564625850340136 10.0
0.16129032258064516 11.0
0.1297709923664122 6.0
0.0 8.0
0.19083969465648856 5.0
0.25190839694656486 4.0
0.2737430167597765 14.0
0.3048128342245989 15.0
0.31297709923664124 3.0
0.3333333333333333 16.0
0.37404580152671757 2.0
0.4351145038167939 1.0
0.4961832061068702 0.0

D.2.10 Normalised similarity values for Grammar 9

D.2.10.1 After Sorting

0.0 0.0
0.5931506849315068 1.0
0.703 2.0
0.7661417322834646 3.0
0.8071428571428572 4.0
0.8359116022099448 5.0
0.8572115384615384 6.0
0.8736170212765958 7.0
0.8866412213740458 8.0
0.8972318339100346 9.0
0.9060126582278482 10.0
0.9134110787172012 11.0
0.9197297297297298 12.0
0.9251889168765743 13.0
0.9299528301886792 14.0
0.9341463414634147 15.0
0.9378661087866109 16.0

D.2.10.2 After Shuffling

0.8071428571428572 4.0
0.8866412213740458 8.0
0.0 0.0
0.5931506849315068 1.0
0.8736170212765958 7.0
0.703 2.0
0.8572115384615384 6.0
0.7661417322834646 3.0
0.8359116022099448 5.0
0.8972318339100346 9.0
0.9060126582278482 10.0
0.9134110787172012 11.0
0.9197297297297298 12.0
0.9251889168765743 13.0
0.9299528301886792 14.0
0.9341463414634147 15.0
0.9378661087866109 16.0

D.2.10.3 After Sorting

0.0 8.0
0.14965497412305923 9.0
0.16508087535680305 7.0
0.22232448067315277 10.0
0.26815731049793845 6.0
0.28355135658914726 11.0
0.3358410060633281 12.0
0.3712337456390739 5.0
0.3810171619924655 13.0
0.4204389574759945 14.0
0.4551400147383935 15.0
0.4743101807802093 4.0
0.48592038936207194 16.0
0.5773866159213448 3.0
0.6804630510624802 2.0
0.7835394862036156 1.0
0.886615921344751 0.0

D.2.10.4 After Shuffling

0.0 8.0
0.22232448067315277 10.0
0.3358410060633281 12.0
0.14965497412305923 9.0
0.16508087535680305 7.0
0.26815731049793845 6.0
0.3810171619924655 13.0
0.3712337456390739 5.0
0.28355135658914726 11.0
0.4204389574759945 14.0
0.4551400147383935 15.0
0.4743101807802093 4.0
0.48592038936207194 16.0
0.5773866159213448 3.0
0.6804630510624802 2.0
0.7835394862036156 1.0
0.886615921344751 0.0

D.2.10.5 After Sorting

0.0 8.0
0.09050966608084358 9.0
0.09592868547099521 7.0
0.13913973384030418 10.0
0.15580095795635976 6.0
0.18283329573652154 11.0
0.21567323044172432 5.0
0.22230571060541005 12.0
0.2581404874052836 13.0
0.2755455029270889 4.0
0.2908183241973375 14.0
0.3207387961747609 15.0
0.33541777541245343 3.0
0.3482367758186398 16.0
0.39529004789781796 2.0
0.45516232038318255 1.0
0.5150345928685471 0.0

D.2.10.6 After Shuffling

0.15580095795635976 6.0
0.21567323044172432 5.0
0.09592868547099521 7.0
0.13913973384030418 10.0
0.2581404874052836 13.0
0.09050966608084358 9.0
0.0 8.0
0.18283329573652154 11.0
0.22230571060541005 12.0
0.2755455029270889 4.0
0.2908183241973375 14.0
0.3207387961747609 15.0
0.33541777541245343 3.0
0.3482367758186398 16.0
0.39529004789781796 2.0
0.45516232038318255 1.0
0.5150345928685471 0.0

D.2.11 Normalised similarity values for Grammar 10

D.2.11.1 After Sorting

0.0 0.0
0.25187566988210075 1.0
0.36631779257849667 2.0
0.4295473953885568 3.0
0.48099301784328935 4.0
0.5319602272727273 5.0
0.5494433529796987 6.0
0.5843446601941747 7.0
0.6159502262443439 8.0
0.6486486486486487 9.0
0.6711376055638351 10.0
0.686826066572902 11.0
0.7012871726586773 12.0
0.7663438256658596 14.0
0.7730556736081597 13.0
0.7733026467203682 15.0
0.78742184626701 16.0

D.2.11.2 After Shuffling

0.5843446601941747 7.0
0.36631779257849667 2.0
0.5494433529796987 6.0
0.5319602272727273 5.0
0.0 0.0
0.25187566988210075 1.0
0.4295473953885568 3.0
0.6159502262443439 8.0
0.48099301784328935 4.0
0.6486486486486487 9.0
0.6711376055638351 10.0
0.686826066572902 11.0
0.7012871726586773 12.0
0.7663438256658596 14.0
0.7730556736081597 13.0
0.7733026467203682 15.0
0.78742184626701 16.0

D.2.11.3 After Sorting

0.0 8.0
0.5258426966292135 7.0
0.5370467682606411 9.0
0.5516853932584269 6.0
0.5702247191011236 4.0
0.5764044943820225 1.0
0.5803370786516854 5.0
0.5910112359550562 2.0
0.601123595505618 3.0
0.6158415841584158 10.0
0.6362023070097604 12.0
0.6410112359550562 0.0
0.6423562412342216 11.0
0.644808743169399 13.0
0.6514788169464428 14.0
0.6907492354740061 15.0
0.7468864468864469 16.0

D.2.11.4 After Shuffling

0.601123595505618 3.0
0.5764044943820225 1.0
0.5803370786516854 5.0
0.5370467682606411 9.0
0.5516853932584269 6.0
0.5258426966292135 7.0
0.0 8.0
0.5910112359550562 2.0
0.5702247191011236 4.0
0.6158415841584158 10.0
0.6362023070097604 12.0
0.6410112359550562 0.0
0.6423562412342216 11.0
0.644808743169399 13.0
0.6514788169464428 14.0
0.6907492354740061 15.0
0.7468864468864469 16.0

D.2.11.5 After Sorting

0.0 8.0
0.6590733590733591 7.0
0.6602316602316602 6.0
0.6737451737451737 0.0
0.6818532818532819 1.0
0.6819357222016993 9.0
0.6853281853281853 3.0
0.6868725868725869 5.0
0.6888030888030888 4.0
0.6938223938223939 2.0
0.7064447592067988 10.0
0.71875 11.0
0.7355722204108249 12.0
0.7537065052950076 14.0
0.7537641154328732 13.0
0.7612994350282486 16.0
0.761612620508326 15.0

D.2.11.6 After Shuffling

0.6888030888030888 4.0
0.0 8.0
0.6819357222016993 9.0
0.6868725868725869 5.0
0.6853281853281853 3.0
0.6590733590733591 7.0
0.6737451737451737 0.0
0.6602316602316602 6.0
0.6818532818532819 1.0
0.6938223938223939 2.0
0.7064447592067988 10.0
0.71875 11.0
0.7355722204108249 12.0
0.7537065052950076 14.0
0.7537641154328732 13.0
0.7612994350282486 16.0
0.761612620508326 15.0

D.2.12 Normalised similarity values for Grammar 11

D.2.12.1 After Sorting

0.0 0.0
0.23076923076923078 2.0
0.375 4.0
0.47368421052631576 6.0
0.5454545454545454 8.0
0.6 10.0
0.6428571428571429 12.0
0.6774193548387096 14.0
0.7058823529411765 16.0
1.0 1.0
1.0 3.0
1.0 5.0
1.0 7.0
1.0 9.0
1.0 11.0
1.0 13.0
1.0 15.0

D.2.12.2 After Shuffling

0.375 4.0
0.5454545454545454 8.0
0.7058823529411765 16.0
0.6774193548387096 14.0
0.6428571428571429 12.0
0.0 0.0
0.47368421052631576 6.0
0.6 10.0
0.23076923076923078 2.0
1.0 1.0
1.0 3.0
1.0 5.0
1.0 7.0
1.0 9.0
1.0 11.0
1.0 13.0
1.0 15.0

D.2.12.3 After Sorting

0.0 8.0
0.12 10.0
0.13636363636363635 6.0
0.21428571428571427 12.0
0.2727272727272727 4.0
0.2903225806451613 14.0
0.35294117647058826 16.0
0.4090909090909091 2.0
0.5454545454545454 0.0
1.0 1.0
1.0 3.0
1.0 5.0
1.0 7.0
1.0 9.0
1.0 11.0
1.0 13.0
1.0 15.0

D.2.12.4 After Shuffling

0.21428571428571427 12.0
0.35294117647058826 16.0
0.2727272727272727 4.0
0.2903225806451613 14.0
0.4090909090909091 2.0
0.12 10.0
0.5454545454545454 0.0
0.0 8.0
0.13636363636363635 6.0
1.0 1.0
1.0 3.0
1.0 5.0
1.0 7.0
1.0 9.0
1.0 11.0
1.0 13.0
1.0 15.0

D.2.12.5 After Sorting

0.0 8.0
0.08333333333333333 10.0
0.09090909090909091 6.0
0.15384615384615385 12.0
0.18181818181818182 4.0
0.21428571428571427 14.0
0.26666666666666666 16.0
0.2727272727272727 2.0
0.36363636363636365 0.0
1.0 1.0
1.0 3.0
1.0 5.0
1.0 7.0
1.0 9.0
1.0 11.0
1.0 13.0
1.0 15.0

D.2.12.6 After Shuffling

0.0 8.0
0.21428571428571427 14.0
0.26666666666666666 16.0
0.15384615384615385 12.0
0.36363636363636365 0.0
0.08333333333333333 10.0
0.18181818181818182 4.0
0.2727272727272727 2.0
0.09090909090909091 6.0
1.0 1.0
1.0 3.0
1.0 5.0
1.0 7.0
1.0 9.0
1.0 11.0
1.0 13.0
1.0 15.0

D.2.13 Normalised similarity values for Grammar 12

D.2.13.1 After Sorting

0.0 0.0
0.042105263157894736 7.0
0.05263157894736842 4.0
0.06315789473684211 15.0
0.08421052631578947 11.0
0.09900990099009901 3.0
0.11578947368421053 8.0
0.11578947368421053 16.0
0.15789473684210525 12.0
0.16842105263157894 1.0
0.21052631578947367 5.0
0.23157894736842105 2.0
0.25263157894736843 9.0
0.2736842105263158 6.0
0.29473684210526313 13.0
0.3368421052631579 10.0
0.37894736842105264 14.0

D.2.13.2 After Shuffling

0.09900990099009901 3.0
0.11578947368421053 16.0
0.06315789473684211 15.0
0.08421052631578947 11.0
0.11578947368421053 8.0
0.042105263157894736 7.0
0.0 0.0
0.05263157894736842 4.0
0.15789473684210525 12.0
0.16842105263157894 1.0
0.21052631578947367 5.0
0.23157894736842105 2.0
0.25263157894736843 9.0
0.2736842105263158 6.0
0.29473684210526313 13.0
0.3368421052631579 10.0
0.37894736842105264 14.0

D.2.13.3 After Sorting

0.0 8.0
0.05813953488372093 12.0
0.06976744186046512 16.0
0.10869565217391304 4.0
0.10869565217391304 11.0
0.11578947368421053 0.0
0.11578947368421053 7.0
0.12790697674418605 1.0
0.16842105263157894 15.0
0.1744186046511628 5.0
0.18604651162790697 9.0
0.19767441860465115 2.0
0.19801980198019803 3.0
0.23255813953488372 13.0
0.2441860465116279 6.0
0.2558139534883721 10.0
0.3023255813953488 14.0

D.2.13.4 After Shuffling

0.16842105263157894 15.0
0.06976744186046512 16.0
0.10869565217391304 4.0
0.10869565217391304 11.0
0.11578947368421053 7.0
0.12790697674418605 1.0
0.0 8.0
0.11578947368421053 0.0
0.05813953488372093 12.0
0.1744186046511628 5.0
0.18604651162790697 9.0
0.19767441860465115 2.0
0.19801980198019803 3.0
0.23255813953488372 13.0
0.2441860465116279 6.0
0.2558139534883721 10.0
0.3023255813953488 14.0

D.2.13.5 After Sorting

0.0 8.0
0.05263157894736842 12.0
0.06315789473684211 0.0
0.09473684210526316 4.0
0.11578947368421053 9.0
0.11578947368421053 16.0
0.15789473684210525 13.0
0.16842105263157894 1.0
0.2 5.0
0.23157894736842105 10.0
0.2736842105263158 14.0
0.28421052631578947 2.0
0.3157894736842105 6.0
0.3368421052631579 11.0
0.37894736842105264 15.0
0.3894736842105263 3.0
0.42105263157894735 7.0

D.2.13.6 After Shuffling

0.09473684210526316 4.0
0.06315789473684211 0.0
0.05263157894736842 12.0
0.0 8.0
0.2 5.0
0.11578947368421053 16.0
0.15789473684210525 13.0
0.11578947368421053 9.0
0.16842105263157894 1.0
0.23157894736842105 10.0
0.2736842105263158 14.0
0.28421052631578947 2.0
0.3157894736842105 6.0
0.3368421052631579 11.0
0.37894736842105264 15.0
0.3894736842105263 3.0
0.42105263157894735 7.0

D.2.14 Normalised similarity values for Grammar 13

D.2.14.1 After Sorting

0.0 0.0
0.5 1.0
0.7 2.0
0.8 3.0
0.8571428571428571 4.0
0.8928571428571429 5.0
0.9166666666666666 6.0
0.9333333333333333 7.0
0.9454545454545454 8.0
0.9545454545454546 9.0
0.9615384615384616 10.0
0.967032967032967 11.0
0.9714285714285714 12.0
0.975 13.0
0.9779411764705882 14.0
0.9803921568627451 15.0
0.9824561403508771 16.0

D.2.14.2 After Shuffling

0.9454545454545454 8.0
0.8 3.0
0.8928571428571429 5.0
0.0 0.0
0.8571428571428571 4.0
0.9166666666666666 6.0
0.5 1.0
0.9333333333333333 7.0
0.7 2.0
0.9545454545454546 9.0
0.9615384615384616 10.0
0.967032967032967 11.0
0.9714285714285714 12.0
0.975 13.0
0.9779411764705882 14.0
0.9803921568627451 15.0
0.9824561403508771 16.0

D.2.14.3 After Sorting

0.0 8.0
0.16666666666666666 9.0
0.18181818181818182 7.0
0.2948717948717949 10.0
0.34545454545454546 6.0
0.3956043956043956 11.0
0.47619047619047616 12.0
0.4909090909090909 5.0
0.5416666666666666 13.0
0.5955882352941176 14.0
0.6181818181818182 4.0
0.6405228758169934 15.0
0.6783625730994152 16.0
0.7272727272727273 3.0
0.8181818181818182 2.0
0.8909090909090909 1.0
0.9454545454545454 0.0

D.2.14.4 After Shuffling

0.2948717948717949 10.0
0.47619047619047616 12.0
0.3956043956043956 11.0
0.16666666666666666 9.0
0.0 8.0
0.34545454545454546 6.0
0.4909090909090909 5.0
0.5416666666666666 13.0
0.18181818181818182 7.0
0.5955882352941176 14.0
0.6181818181818182 4.0
0.6405228758169934 15.0
0.6783625730994152 16.0
0.7272727272727273 3.0
0.8181818181818182 2.0
0.8909090909090909 1.0
0.9454545454545454 0.0

D.2.14.5 After Sorting

0.0 8.0
0.10526315789473684 9.0
0.1111111111111111 7.0
0.19473684210526315 10.0
0.21568627450980393 6.0
0.2714285714285714 11.0
0.3137254901960784 5.0
0.33766233766233766 12.0
0.3952569169960474 13.0
0.40522875816993464 4.0
0.44565217391304346 14.0
0.49 15.0
0.49019607843137253 3.0
0.5292307692307693 16.0
0.5686274509803921 2.0
0.6405228758169934 1.0
0.7058823529411765 0.0

D.2.14.6 After Shuffling

0.1111111111111111 7.0
0.0 8.0
0.10526315789473684 9.0
0.19473684210526315 10.0
0.33766233766233766 12.0
0.3137254901960784 5.0
0.21568627450980393 6.0
0.3952569169960474 13.0
0.2714285714285714 11.0
0.40522875816993464 4.0
0.44565217391304346 14.0
0.49 15.0
0.49019607843137253 3.0
0.5292307692307693 16.0
0.5686274509803921 2.0
0.6405228758169934 1.0
0.7058823529411765 0.0

D.2.15 Normalised similarity values for Grammar 14

D.2.15.1 After Sorting

0.0 0.0
0.2857142857142857 1.0
0.4642857142857143 2.0
0.5833333333333334 3.0
0.6666666666666666 4.0
0.7272727272727273 5.0
0.7727272727272727 6.0
0.8076923076923077 7.0
0.8351648351648352 8.0
0.8571428571428571 9.0
0.875 10.0
0.8897058823529411 11.0
0.9019607843137255 12.0
0.9122807017543859 13.0
0.9210526315789473 14.0
0.9285714285714286 15.0
0.935064935064935 16.0

D.2.15.2 After Shuffling

0.2857142857142857 1.0
0.8351648351648352 8.0
0.7727272727272727 6.0
0.5833333333333334 3.0
0.6666666666666666 4.0
0.8076923076923077 7.0
0.7272727272727273 5.0
0.0 0.0
0.4642857142857143 2.0
0.8571428571428571 9.0
0.875 10.0
0.8897058823529411 11.0
0.9019607843137255 12.0
0.9122807017543859 13.0
0.9210526315789473 14.0
0.9285714285714286 15.0
0.935064935064935 16.0

D.2.15.3 After Sorting

0.0 8.0
0.13333333333333333 9.0
0.14285714285714285 7.0
0.24166666666666667 10.0
0.27472527472527475 6.0
0.33088235294117646 11.0
0.3956043956043956 5.0
0.40522875816993464 12.0
0.4678362573099415 13.0
0.5054945054945055 4.0
0.5210526315789473 14.0
0.5666666666666667 15.0
0.6043956043956044 3.0
0.6060606060606061 16.0
0.6923076923076923 2.0
0.7692307692307693 1.0
0.8351648351648352 0.0

D.2.15.4 After Shuffling

0.24166666666666667 10.0
0.13333333333333333 9.0
0.40522875816993464 12.0
0.3956043956043956 5.0
0.0 8.0
0.33088235294117646 11.0
0.27472527472527475 6.0
0.14285714285714285 7.0
0.4678362573099415 13.0
0.5054945054945055 4.0
0.5210526315789473 14.0
0.5666666666666667 15.0
0.6043956043956044 3.0
0.6060606060606061 16.0
0.6923076923076923 2.0
0.7692307692307693 1.0
0.8351648351648352 0.0

D.2.15.5 After Sorting

0.0 8.0
0.09090909090909091 9.0
0.09523809523809523 7.0
0.16996047430830039 10.0
0.18571428571428572 6.0
0.2391304347826087 11.0
0.2714285714285714 5.0
0.3 12.0
0.3523809523809524 4.0
0.35384615384615387 13.0
0.4017094017094017 14.0
0.42857142857142855 3.0
0.4444444444444444 15.0
0.4827586206896552 16.0
0.5 2.0
0.5666666666666667 1.0
0.6285714285714286 0.0

D.2.15.6 After Shuffling

0.3523809523809524 4.0
0.18571428571428572 6.0
0.09090909090909091 9.0
0.3 12.0
0.2391304347826087 11.0
0.16996047430830039 10.0
0.2714285714285714 5.0
0.09523809523809523 7.0
0.0 8.0
0.35384615384615387 13.0
0.4017094017094017 14.0
0.42857142857142855 3.0
0.4444444444444444 15.0
0.4827586206896552 16.0
0.5 2.0
0.5666666666666667 1.0
0.6285714285714286 0.0

D.2.16 Normalised similarity values for Grammar 15

.0 0.0
0.3225806451612903 1.0
0.5116279069767442 2.0
0.631578947368421 3.0
0.7123287671232876 4.0
0.7692307692307693 5.0
0.8108108108108109 6.0
0.8444444444444444 7.0
0.874251497005988 8.0
0.9014084507042254 9.0
0.9252669039145908 10.0
0.9448818897637795 11.0
0.96 12.0
0.9711141678129298 13.0
0.9790628115653041 14.0
0.9846827133479212 15.0
0.9886547811993517 16.0

D.2.16.1 After Shuffling

0.8108108108108109 6.0
0.631578947368421 3.0
0.874251497005988 8.0
0.7123287671232876 4.0
0.8444444444444444 7.0
0.0 0.0
0.5116279069767442 2.0
0.3225806451612903 1.0
0.7692307692307693 5.0
0.9014084507042254 9.0
0.9252669039145908 10.0
0.9448818897637795 11.0
0.96 12.0
0.9711141678129298 13.0
0.9790628115653041 14.0
0.9846827133479212 15.0
0.9886547811993517 16.0

D.2.16.2 After Sorting

0.0 8.0
0.19161676646706588 7.0
0.215962441314554 9.0
0.33532934131736525 6.0
0.40569395017793597 10.0
0.4550898203592814 5.0
0.5616797900262467 11.0
0.562874251497006 4.0
0.6586826347305389 3.0
0.6819047619047619 12.0
0.7425149700598802 2.0
0.7702888583218707 13.0
0.8143712574850299 1.0
0.8334995014955134 14.0
0.874251497005988 0.0
0.8781911013858498 15.0
0.9097784981091303 16.0

D.2.16.3 After Shuffling

0.40569395017793597 10.0
0.215962441314554 9.0
0.33532934131736525 6.0
0.5616797900262467 11.0
0.0 8.0
0.6586826347305389 3.0
0.4550898203592814 5.0
0.19161676646706588 7.0
0.562874251497006 4.0
0.6819047619047619 12.0
0.7425149700598802 2.0
0.7702888583218707 13.0
0.8143712574850299 1.0
0.8334995014955134 14.0
0.874251497005988 0.0
0.8781911013858498 15.0
0.9097784981091303 16.0

D.2.16.4 After Sorting

0.0 8.0
0.2593192868719611 9.0
0.2684172137126185 7.0
0.44426428860964734 10.0
0.4697301239970824 6.0
0.5782836050446016 11.0
0.6170678336980306 5.0
0.677335843727936 12.0
0.7221006564551422 4.0
0.7520347259902334 13.0
0.7950401167031363 3.0
0.8093450146015854 14.0
0.8446389496717724 2.0
0.8538223691225078 15.0
0.8781911013858498 1.0
0.8884912566083774 16.0
0.9015317286652079 0.0

D.2.16.5 After Shuffling

0.2684172137126185 7.0
0.2593192868719611 9.0
0.4697301239970824 6.0
0.44426428860964734 10.0
0.7221006564551422 4.0
0.677335843727936 12.0
0.6170678336980306 5.0
0.0 8.0
0.5782836050446016 11.0
0.7520347259902334 13.0
0.7950401167031363 3.0
0.8093450146015854 14.0
0.8446389496717724 2.0
0.8538223691225078 15.0
0.8781911013858498 1.0
0.8884912566083774 16.0
0.9015317286652079 0.0

D.2.17 Normalised similarity values for Grammar 16

D.2.17.1 After Sorting

0.0 0.0
0.6417910447761194 15.0
0.7584905660377359 14.0
0.7811550151975684 12.0
0.7845117845117845 13.0
0.8005540166204986 11.0
0.8305882352941176 7.0
0.8305882352941176 10.0
0.8371501272264631 8.0
0.8424507658643327 9.0
0.869120654396728 6.0
0.8769230769230769 3.0
0.8771593090211133 5.0
0.8771593090211133 16.0
0.8842676311030742 4.0
0.901386748844376 2.0
0.9060205580029369 1.0

D.2.17.2 After Shuffling

0.7845117845117845 13.0
0.8305882352941176 7.0
0.7584905660377359 14.0
0.6417910447761194 15.0
0.8371501272264631 8.0
0.8305882352941176 10.0
0.0 0.0
0.8005540166204986 11.0
0.7811550151975684 12.0
0.8424507658643327 9.0
0.869120654396728 6.0
0.8769230769230769 3.0
0.8771593090211133 5.0
0.8771593090211133 16.0
0.8842676311030742 4.0
0.901386748844376 2.0
0.9060205580029369 1.0

D.2.17.3 After Sorting

0.0 8.0
0.09647058823529411 7.0
0.13411764705882354 10.0
0.18575063613231552 11.0
0.212253829321663 9.0
0.2147239263803681 6.0
0.24681933842239187 12.0
0.2629558541266795 5.0
0.2875318066157761 13.0
0.30902111324376197 16.0
0.3486005089058524 14.0
0.34900542495479203 4.0
0.38461538461538464 3.0
0.43297380585516176 2.0
0.4713656387665198 1.0
0.5318066157760815 15.0
0.8371501272264631 0.0

D.2.17.4 After Shuffling

0.0 8.0
0.09647058823529411 7.0
0.212253829321663 9.0
0.13411764705882354 10.0
0.18575063613231552 11.0
0.2629558541266795 5.0
0.24681933842239187 12.0
0.2147239263803681 6.0
0.2875318066157761 13.0
0.30902111324376197 16.0
0.3486005089058524 14.0
0.34900542495479203 4.0
0.38461538461538464 3.0
0.43297380585516176 2.0
0.4713656387665198 1.0
0.5318066157760815 15.0
0.8371501272264631 0.0

D.2.17.5 After Sorting

0.0 8.0
0.0845771144278607 12.0
0.24378109452736318 13.0
0.30566037735849055 7.0
0.35353535353535354 11.0
0.38047138047138046 6.0
0.41641337386018235 5.0
0.46814404432132967 4.0
0.4903047091412742 10.0
0.5223880597014925 14.0
0.5318066157760815 1.0
0.548235294117647 0.0
0.548235294117647 3.0
0.5798687089715536 2.0
0.6468330134357005 9.0
0.9317760474601409 16.0
0.9325760351777208 15.0

D.2.17.6 After Shuffling

0.0845771144278607 12.0
0.38047138047138046 6.0
0.24378109452736318 13.0
0.46814404432132967 4.0
0.0 8.0
0.41641337386018235 5.0
0.35353535353535354 11.0
0.4903047091412742 10.0
0.30566037735849055 7.0
0.5223880597014925 14.0
0.5318066157760815 1.0
0.548235294117647 0.0
0.548235294117647 3.0
0.5798687089715536 2.0
0.6468330134357005 9.0
0.9317760474601409 16.0
0.9325760351777208 15.0

D.2.18 Normalised similarity values for Grammar 17

D.2.18.1 After Sorting

0.0 0.0
0.3677811550151976 1.0
0.3677811550151976 2.0
0.46504559270516715 3.0
0.5379939209726444 6.0
0.7102001906577693 15.0
0.7451802179379715 12.0
0.7482233502538072 16.0
0.7575757575757576 9.0
0.763584366062917 14.0
0.7739288969917958 13.0
0.7963404932378679 11.0
0.8013964313421257 10.0
0.8541033434650456 4.0
0.8541033434650456 5.0
0.8541033434650456 7.0
0.8541033434650456 8.0

D.2.18.2 After Shuffling

0.7575757575757576 9.0
0.7451802179379715 12.0
0.7482233502538072 16.0
0.3677811550151976 1.0
0.46504559270516715 3.0
0.0 0.0
0.3677811550151976 2.0
0.7102001906577693 15.0
0.5379939209726444 6.0
0.763584366062917 14.0
0.7739288969917958 13.0
0.7963404932378679 11.0
0.8013964313421257 10.0
0.8541033434650456 4.0
0.8541033434650456 5.0
0.8541033434650456 7.0
0.8541033434650456 8.0

D.2.18.3 After Sorting

0.0 8.0
0.3220338983050847 5.0
0.503448275862069 7.0
0.6298932384341637 2.0
0.681592039800995 4.0
0.6836158192090396 6.0
0.7845117845117845 1.0
0.8072289156626506 3.0
0.8541033434650456 0.0
0.8856053384175405 14.0
0.9108989657915673 11.0
0.9187817258883249 16.0
0.9389895138226882 15.0
0.9416590701914311 13.0
0.9503491078355314 10.0
0.9597652975691534 12.0
0.9645232815964523 9.0

D.2.18.4 After Shuffling

0.3220338983050847 5.0
0.7845117845117845 1.0
0.681592039800995 4.0
0.0 8.0
0.8541033434650456 0.0
0.6836158192090396 6.0
0.6298932384341637 2.0
0.503448275862069 7.0
0.8072289156626506 3.0
0.8856053384175405 14.0
0.9108989657915673 11.0
0.9187817258883249 16.0
0.9389895138226882 15.0
0.9416590701914311 13.0
0.9503491078355314 10.0
0.9597652975691534 12.0
0.9645232815964523 9.0

D.2.18.5 After Sorting

0.0 8.0
0.16848281642917015 5.0
0.23736892278360344 11.0
0.2526215443279314 7.0
0.2526215443279314 9.0
0.2526215443279314 10.0
0.278640059127864 2.0
0.2999088422971741 6.0
0.29992044550517105 4.0
0.3297129557796742 3.0
0.4661582459485224 13.0
0.48141086749285034 12.0
0.5805529075309819 14.0
0.8932316491897044 16.0
0.9084842707340324 15.0
0.9313632030505243 0.0
0.9389895138226882 1.0

D.2.18.6 After Shuffling

0.2526215443279314 10.0
0.29992044550517105 4.0
0.2526215443279314 7.0
0.278640059127864 2.0
0.2526215443279314 9.0
0.23736892278360344 11.0
0.16848281642917015 5.0
0.2999088422971741 6.0
0.0 8.0
0.3297129557796742 3.0
0.4661582459485224 13.0
0.48141086749285034 12.0
0.5805529075309819 14.0
0.8932316491897044 16.0
0.9084842707340324 15.0
0.9313632030505243 0.0
0.9389895138226882 1.0

D.2.19 Normalised similarity values for Grammar 18

D.2.19.1 After Sorting

0.0 0.0
0.21176470588235294 2.0
0.30927835051546393 8.0
0.3142857142857143 9.0
0.3424657534246575 1.0
0.3434343434343434 3.0
0.3698630136986301 5.0
0.42735042735042733 11.0
0.4588235294117647 14.0
0.4657534246575342 6.0
0.5051546391752577 10.0
0.5176470588235295 4.0
0.589041095890411 7.0
0.7136563876651982 12.0
0.8059701492537313 15.0
0.8075117370892019 13.0
0.8477157360406091 16.0

D.2.19.2 After Shuffling

0.3424657534246575 1.0
0.30927835051546393 8.0
0.21176470588235294 2.0
0.3142857142857143 9.0
0.42735042735042733 11.0
0.3434343434343434 3.0
0.0 0.0
0.4588235294117647 14.0
0.3698630136986301 5.0
0.4657534246575342 6.0
0.5051546391752577 10.0
0.5176470588235295 4.0
0.589041095890411 7.0
0.7136563876651982 12.0
0.8059701492537313 15.0
0.8075117370892019 13.0
0.8477157360406091 16.0

D.2.19.3 After Sorting

0.0 8.0
0.14432989690721648 2.0
0.2905982905982906 11.0
0.30927835051546393 0.0
0.3619047619047619 9.0
0.3838383838383838 3.0
0.3917525773195876 1.0
0.3917525773195876 10.0
0.4742268041237113 5.0
0.4742268041237113 14.0
0.5154639175257731 4.0
0.5876288659793815 6.0
0.6391752577319587 7.0
0.7312775330396476 12.0
0.7793427230046949 13.0
0.8009950248756219 15.0
0.8223350253807107 16.0

D.2.19.4 After Shuffling

0.3838383838383838 3.0
0.4742268041237113 5.0
0.3917525773195876 10.0
0.14432989690721648 2.0
0.30927835051546393 0.0
0.0 8.0
0.2905982905982906 11.0
0.3619047619047619 9.0
0.3917525773195876 1.0
0.4742268041237113 14.0
0.5154639175257731 4.0
0.5876288659793815 6.0
0.6391752577319587 7.0
0.7312775330396476 12.0
0.7793427230046949 13.0
0.8009950248756219 15.0
0.8223350253807107 16.0

D.2.19.5 After Sorting

0.0 8.0
0.03980099502487562 14.0
0.05472636815920398 9.0
0.08450704225352113 6.0
0.08955223880597014 15.0
0.13656387665198239 5.0
0.6417910447761194 4.0
0.6417910447761194 13.0
0.6467661691542289 2.0
0.6467661691542289 7.0
0.6666666666666666 3.0
0.6766169154228856 11.0
0.6915422885572139 0.0
0.7263681592039801 12.0
0.7512437810945274 16.0
0.8009950248756219 1.0
0.8009950248756219 10.0

D.2.19.6 After Shuffling

0.08955223880597014 15.0
0.6417910447761194 4.0
0.0 8.0
0.13656387665198239 5.0
0.05472636815920398 9.0
0.6417910447761194 13.0
0.6467661691542289 2.0
0.08450704225352113 6.0
0.03980099502487562 14.0
0.6467661691542289 7.0
0.6666666666666666 3.0
0.6766169154228856 11.0
0.6915422885572139 0.0
0.7263681592039801 12.0
0.7512437810945274 16.0
0.8009950248756219 1.0
0.8009950248756219 10.0

D.2.20 Normalised similarity values for Grammar 19

D.2.20.1 After Sorting

0.0 0.0
0.4444444444444444 1.0
0.4444444444444444 4.0
0.4444444444444444 6.0
0.4444444444444444 7.0
0.5555555555555556 2.0
0.5555555555555556 8.0
0.5833333333333334 9.0
0.5833333333333334 12.0
0.5833333333333334 14.0
0.5833333333333334 15.0
0.6666666666666666 10.0
0.6666666666666666 11.0
0.6666666666666666 16.0
0.7777777777777778 3.0
0.7777777777777778 5.0
0.8333333333333334 13.0

D.2.20.2 After Shuffling

0.4444444444444444 7.0
0.4444444444444444 4.0
0.0 0.0
0.4444444444444444 1.0
0.5833333333333334 12.0
0.5555555555555556 8.0
0.5555555555555556 2.0
0.4444444444444444 6.0
0.5833333333333334 9.0
0.5833333333333334 14.0
0.5833333333333334 15.0
0.6666666666666666 10.0
0.6666666666666666 11.0
0.6666666666666666 16.0
0.7777777777777778 3.0
0.7777777777777778 5.0
0.8333333333333334 13.0

D.2.20.3 After Sorting

0.0 2.0
0.0 8.0
0.3333333333333333 4.0
0.3333333333333333 6.0
0.3333333333333333 10.0
0.3333333333333333 16.0
0.4166666666666667 11.0
0.4444444444444444 3.0
0.4444444444444444 5.0
0.5 12.0
0.5 14.0
0.5 15.0
0.5555555555555556 0.0
0.5555555555555556 1.0
0.5555555555555556 7.0
0.5833333333333334 13.0
0.6666666666666666 9.0

D.2.20.4 After Shuffling

0.3333333333333333 10.0
0.4444444444444444 3.0
0.3333333333333333 4.0
0.3333333333333333 6.0
0.0 8.0
0.4444444444444444 5.0
0.3333333333333333 16.0
0.0 2.0
0.4166666666666667 11.0
0.5 12.0
0.5 14.0
0.5 15.0
0.5555555555555556 0.0
0.5555555555555556 1.0
0.5555555555555556 7.0
0.5833333333333334 13.0
0.6666666666666666 9.0

D.2.20.5 After Sorting

0.0 8.0
0.0 12.0
0.25 5.0
0.3333333333333333 0.0
0.3333333333333333 2.0
0.3333333333333333 3.0
0.3333333333333333 14.0
0.3333333333333333 15.0
0.4166666666666667 7.0
0.4166666666666667 11.0
0.5 1.0
0.5 4.0
0.5 9.0
0.5 13.0
0.5 16.0
0.6666666666666666 6.0
0.6666666666666666 10.0

D.2.20.6 After Shuffling

0.3333333333333333 14.0
0.0 12.0
0.0 8.0
0.3333333333333333 15.0
0.3333333333333333 2.0
0.3333333333333333 0.0
0.3333333333333333 3.0
0.4166666666666667 7.0
0.25 5.0
0.4166666666666667 11.0
0.5 1.0
0.5 4.0
0.5 9.0
0.5 13.0
0.5 16.0
0.6666666666666666 6.0
0.6666666666666666 10.0

D.2.21 Normalised similarity values for Grammar 20

D.2.21.1 After Sorting

0.0 0.0
0.04032258064516129 4.0
0.0873015873015873 9.0
0.08870967741935484 8.0
0.0967741935483871 13.0
0.11363636363636363 5.0
0.11851851851851852 1.0
0.12096774193548387 3.0
0.12096774193548387 12.0
0.12121212121212122 14.0
0.1259259259259259 10.0
0.14893617021276595 6.0
0.1527777777777778 2.0
0.1693548387096774 7.0
0.1693548387096774 16.0
0.20161290322580644 11.0
0.25 15.0

D.2.21.2 After Shuffling

0.0 0.0
0.12096774193548387 12.0
0.12096774193548387 3.0
0.11363636363636363 5.0
0.04032258064516129 4.0
0.08870967741935484 8.0
0.0967741935483871 13.0
0.0873015873015873 9.0
0.11851851851851852 1.0
0.12121212121212122 14.0
0.1259259259259259 10.0
0.14893617021276595 6.0
0.1527777777777778 2.0
0.1693548387096774 7.0
0.1693548387096774 16.0
0.20161290322580644 11.0
0.25 15.0

D.2.21.3 After Sorting

0.0 8.0
0.043478260869565216 12.0
0.06956521739130435 3.0
0.08264462809917356 4.0
0.08870967741935484 0.0
0.09565217391304348 7.0
0.12195121951219512 13.0
0.12698412698412698 9.0
0.13043478260869565 11.0
0.1391304347826087 16.0
0.1590909090909091 14.0
0.16296296296296298 10.0
0.17424242424242425 5.0
0.17777777777777778 1.0
0.20869565217391303 15.0
0.2198581560283688 6.0
0.2222222222222222 2.0

D.2.21.4 After Shuffling

0.12698412698412698 9.0
0.06956521739130435 3.0
0.08264462809917356 4.0
0.12195121951219512 13.0
0.0 8.0
0.08870967741935484 0.0
0.13043478260869565 11.0
0.09565217391304348 7.0
0.043478260869565216 12.0
0.1391304347826087 16.0
0.1590909090909091 14.0
0.16296296296296298 10.0
0.17424242424242425 5.0
0.17777777777777778 1.0
0.20869565217391303 15.0
0.2198581560283688 6.0
0.2222222222222222 2.0

D.2.21.5 After Sorting

0.0 8.0
0.052083333333333336 12.0
0.09803921568627451 13.0
0.10476190476190476 9.0
0.11458333333333333 16.0
0.13592233009708737 4.0
0.1509433962264151 0.0
0.18584070796460178 14.0
0.1896551724137931 10.0
0.19642857142857142 5.0
0.20869565217391303 1.0
0.2540983606557377 15.0
0.256 11.0
0.2845528455284553 6.0
0.29365079365079366 2.0
0.32575757575757575 7.0
0.3333333333333333 3.0

D.2.21.6 After Shuffling

0.0 8.0
0.1509433962264151 0.0
0.052083333333333336 12.0
0.11458333333333333 16.0
0.1896551724137931 10.0
0.09803921568627451 13.0
0.18584070796460178 14.0
0.10476190476190476 9.0
0.13592233009708737 4.0
0.19642857142857142 5.0
0.20869565217391303 1.0
0.2540983606557377 15.0
0.256 11.0
0.2845528455284553 6.0
0.29365079365079366 2.0
0.32575757575757575 7.0
0.3333333333333333 3.0

D.2.22 Normalised similarity values for Grammar 21

D.2.22.1 After Sorting

0.0 0.0
0.003409090909090909 15.0
0.005681818181818182 1.0
0.005681818181818182 3.0
0.006818181818181818 7.0
0.03295454545454545 2.0
0.03295454545454545 4.0
0.03295454545454545 8.0
0.03295454545454545 16.0
0.03636363636363636 5.0
0.0375 9.0
0.0375 11.0
0.06363636363636363 6.0
0.06477272727272727 10.0
0.06477272727272727 12.0
0.06818181818181818 13.0
0.09545454545454546 14.0

D.2.22.2 After Shuffling

0.03295454545454545 4.0
0.006818181818181818 7.0
0.005681818181818182 1.0
0.03295454545454545 2.0
0.03295454545454545 8.0
0.005681818181818182 3.0
0.03295454545454545 16.0
0.0 0.0
0.003409090909090909 15.0
0.03636363636363636 5.0
0.0375 9.0
0.0375 11.0
0.06363636363636363 6.0
0.06477272727272727 10.0
0.06477272727272727 12.0
0.06818181818181818 13.0
0.09545454545454546 14.0

D.2.22.3 After Sorting

0.0 8.0
0.0035169988276670576 2.0
0.005861664712778429 9.0
0.005861664712778429 11.0
0.005861664712778429 16.0
0.007033997655334115 4.0
0.007033997655334115 5.0
0.03295454545454545 0.0
0.03295454545454545 7.0
0.03399765533411489 10.0
0.03399765533411489 12.0
0.03409090909090909 1.0
0.03409090909090909 15.0
0.035169988276670575 6.0
0.0375 3.0
0.03751465416178194 13.0
0.06565064478311841 14.0

D.2.22.4 After Shuffling

0.007033997655334115 4.0
0.005861664712778429 9.0
0.03295454545454545 0.0
0.007033997655334115 5.0
0.005861664712778429 11.0
0.0 8.0
0.005861664712778429 16.0
0.03295454545454545 7.0
0.0035169988276670576 2.0
0.03399765533411489 10.0
0.03399765533411489 12.0
0.03409090909090909 1.0
0.03409090909090909 15.0
0.035169988276670575 6.0
0.0375 3.0
0.03751465416178194 13.0
0.06565064478311841 14.0

D.2.22.5 After Sorting

0.0 8.0
0.005681818181818182 0.0
0.03295454545454545 9.0
0.03295454545454545 10.0
0.03295454545454545 12.0
0.03295454545454545 16.0
0.03409090909090909 1.0
0.0375 2.0
0.0375 4.0
0.06363636363636363 11.0
0.06363636363636363 13.0
0.06363636363636363 14.0
0.06477272727272727 3.0
0.06477272727272727 5.0
0.06818181818181818 6.0
0.09431818181818181 15.0
0.09545454545454546 7.0

D.2.22.6 After Shuffling

0.005681818181818182 0.0
0.03295454545454545 9.0
0.03295454545454545 10.0
0.0375 4.0
0.0375 2.0
0.03295454545454545 12.0
0.0 8.0
0.03295454545454545 16.0
0.03409090909090909 1.0
0.06363636363636363 11.0
0.06363636363636363 13.0
0.06363636363636363 14.0
0.06477272727272727 3.0
0.06477272727272727 5.0
0.06818181818181818 6.0
0.09431818181818181 15.0
0.09545454545454546 7.0

D.2.23 Normalised similarity values for Grammar 22

D.2.23.1 After Sorting

0.0 0.0
0.00980392156862745 1.0
0.00980392156862745 3.0
0.014705882352941176 7.0
0.014705882352941176 15.0
0.029411764705882353 2.0
0.029411764705882353 4.0
0.029411764705882353 5.0
0.029411764705882353 8.0
0.029411764705882353 16.0
0.03431372549019608 9.0
0.03431372549019608 11.0
0.049019607843137254 6.0
0.05392156862745098 10.0
0.05392156862745098 12.0
0.05392156862745098 13.0
0.07352941176470588 14.0

D.2.23.2 After Shuffling

0.029411764705882353 5.0
0.0 0.0
0.029411764705882353 4.0
0.029411764705882353 2.0
0.029411764705882353 8.0
0.00980392156862745 3.0
0.014705882352941176 15.0
0.014705882352941176 7.0
0.00980392156862745 1.0
0.029411764705882353 16.0
0.03431372549019608 9.0
0.03431372549019608 11.0
0.049019607843137254 6.0
0.05392156862745098 10.0
0.05392156862745098 12.0
0.05392156862745098 13.0
0.07352941176470588 14.0

D.2.23.3 After Sorting

0.0 8.0
0.00980392156862745 9.0
0.00980392156862745 11.0
0.00980392156862745 16.0
0.014705882352941176 2.0
0.014705882352941176 4.0
0.014705882352941176 5.0
0.029411764705882353 0.0
0.029411764705882353 7.0
0.029411764705882353 10.0
0.029411764705882353 12.0
0.029411764705882353 13.0
0.03431372549019608 1.0
0.03431372549019608 3.0
0.03431372549019608 6.0
0.03431372549019608 15.0
0.049019607843137254 14.0

D.2.23.4 After Shuffling

0.029411764705882353 0.0
0.00980392156862745 11.0
0.00980392156862745 16.0
0.0 8.0
0.014705882352941176 2.0
0.00980392156862745 9.0
0.014705882352941176 4.0
0.029411764705882353 7.0
0.014705882352941176 5.0
0.029411764705882353 10.0
0.029411764705882353 12.0
0.029411764705882353 13.0
0.03431372549019608 1.0
0.03431372549019608 3.0
0.03431372549019608 6.0
0.03431372549019608 15.0
0.049019607843137254 14.0

D.2.23.5 After Sorting

0.0 8.0
0.00980392156862745 0.0
0.029411764705882353 9.0
0.029411764705882353 10.0
0.029411764705882353 12.0
0.029411764705882353 16.0
0.03431372549019608 1.0
0.03431372549019608 2.0
0.03431372549019608 4.0
0.049019607843137254 11.0
0.049019607843137254 13.0
0.049019607843137254 14.0
0.05392156862745098 3.0
0.05392156862745098 5.0
0.05392156862745098 6.0
0.06862745098039216 15.0
0.07352941176470588 7.0

D.2.23.6 After Shuffling

0.029411764705882353 10.0
0.029411764705882353 16.0
0.029411764705882353 9.0
0.029411764705882353 12.0
0.03431372549019608 1.0
0.03431372549019608 4.0
0.00980392156862745 0.0
0.03431372549019608 2.0
0.0 8.0
0.049019607843137254 11.0
0.049019607843137254 13.0
0.049019607843137254 14.0
0.05392156862745098 3.0
0.05392156862745098 5.0
0.05392156862745098 6.0
0.06862745098039216 15.0
0.07352941176470588 7.0

D.2.24 Normalised similarity values for Grammar 23

D.2.24.1 After Sorting

0.0 0.0
0.05714285714285714 3.0
0.16455696202531644 9.0
0.17142857142857143 1.0
0.17142857142857143 7.0
0.22727272727272727 2.0
0.22727272727272727 8.0
0.25 11.0
0.26136363636363635 5.0
0.27848101265822783 15.0
0.29896907216494845 4.0
0.3584905660377358 16.0
0.36792452830188677 10.0
0.3867924528301887 13.0
0.41739130434782606 6.0
0.41739130434782606 12.0
0.5037593984962406 14.0

D.2.24.2 After Shuffling

0.17142857142857143 1.0
0.16455696202531644 9.0
0.26136363636363635 5.0
0.05714285714285714 3.0
0.17142857142857143 7.0
0.0 0.0
0.22727272727272727 2.0
0.25 11.0
0.22727272727272727 8.0
0.27848101265822783 15.0
0.29896907216494845 4.0
0.3584905660377358 16.0
0.36792452830188677 10.0
0.3867924528301887 13.0
0.41739130434782606 6.0
0.41739130434782606 12.0
0.5037593984962406 14.0

D.2.24.3 After Sorting

0.0 8.0
0.045454545454545456 2.0
0.045454545454545456 11.0
0.07954545454545454 5.0
0.13402061855670103 4.0
0.13636363636363635 9.0
0.18867924528301888 10.0
0.2169811320754717 13.0
0.22727272727272727 0.0
0.22727272727272727 15.0
0.25217391304347825 12.0
0.25471698113207547 16.0
0.26136363636363635 3.0
0.2782608695652174 6.0
0.32954545454545453 7.0
0.3522727272727273 1.0
0.3609022556390977 14.0

D.2.24.4 After Shuffling

0.2169811320754717 13.0
0.22727272727272727 0.0
0.045454545454545456 2.0
0.13636363636363635 9.0
0.045454545454545456 11.0
0.18867924528301888 10.0
0.07954545454545454 5.0
0.0 8.0
0.13402061855670103 4.0
0.22727272727272727 15.0
0.25217391304347825 12.0
0.25471698113207547 16.0
0.26136363636363635 3.0
0.2782608695652174 6.0
0.32954545454545453 7.0
0.3522727272727273 1.0
0.3609022556390977 14.0

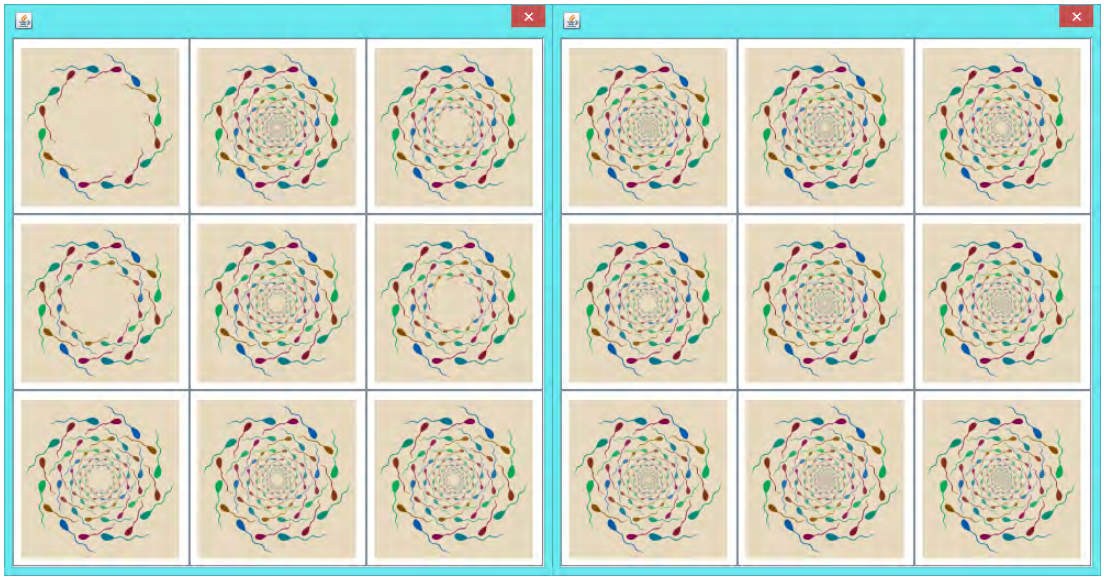
D.2.24.5 After Sorting

0.0 8.0
0.20253164556962025 2.0
0.20618556701030927 10.0
0.20618556701030927 16.0
0.22727272727272727 1.0
0.22727272727272727 4.0
0.27358490566037735 9.0
0.27358490566037735 12.0
0.33962264150943394 3.0
0.33962264150943394 6.0
0.34177215189873417 0.0
0.3870967741935484 11.0
0.3870967741935484 14.0
0.391304347826087 5.0
0.42857142857142855 13.0
0.48120300751879697 7.0
0.5033112582781457 15.0

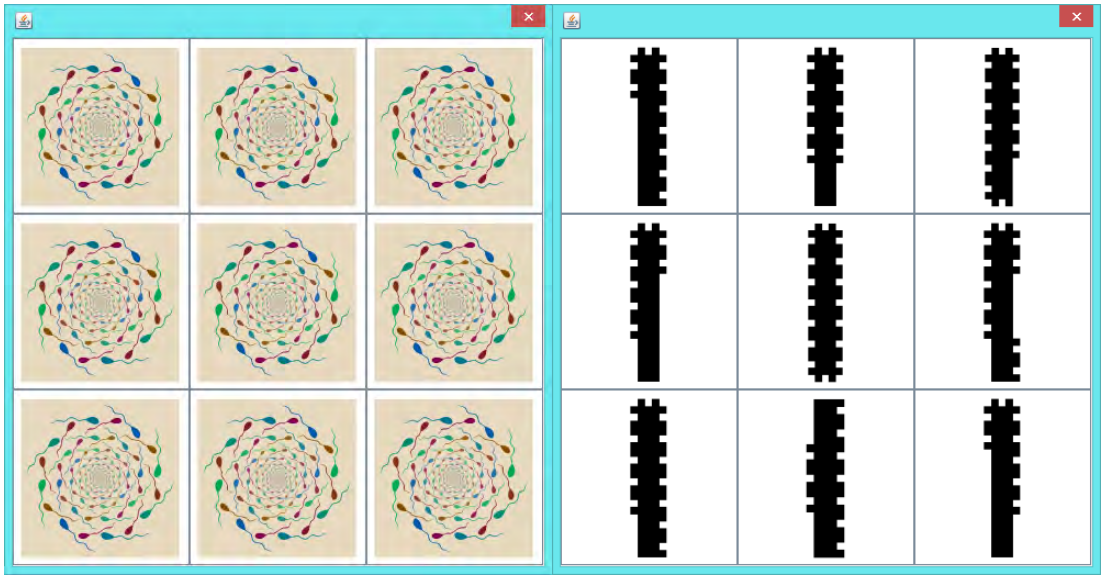
D.2.24.6 After Shuffling

0.33962264150943394 3.0
0.20253164556962025 2.0
0.27358490566037735 9.0
0.20618556701030927 10.0
0.22727272727272727 1.0
0.27358490566037735 12.0
0.0 8.0
0.22727272727272727 4.0
0.20618556701030927 16.0
0.33962264150943394 6.0
0.34177215189873417 0.0
0.3870967741935484 11.0
0.3870967741935484 14.0
0.391304347826087 5.0
0.42857142857142855 13.0
0.48120300751879697 7.0
0.5033112582781457 15.0

D.3 Threshold run images

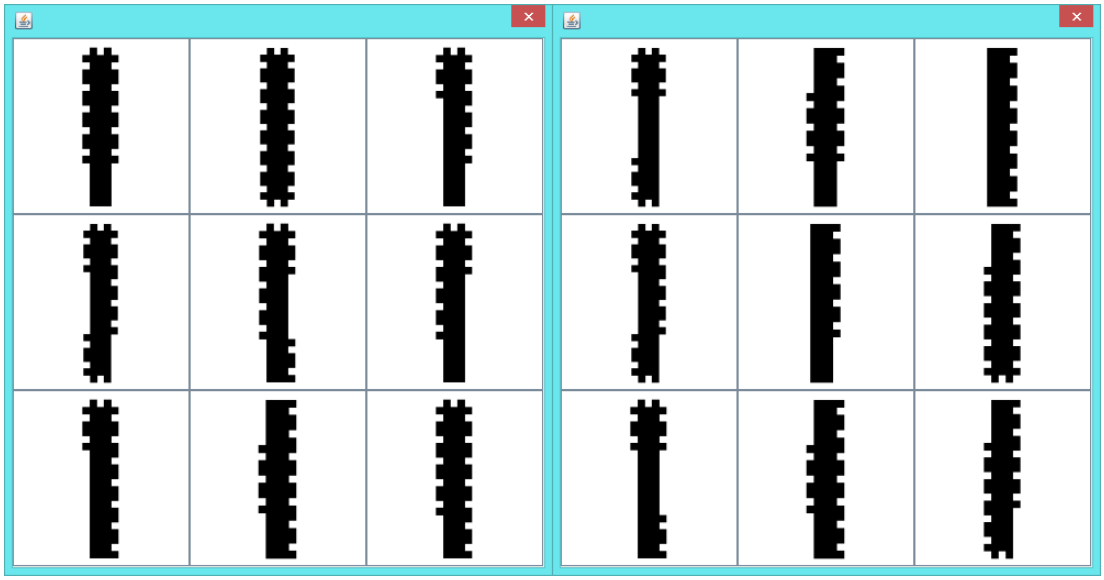


(a) Threshold run for grammar 0 and image 0 (b) Threshold run for grammar 0 and image 8

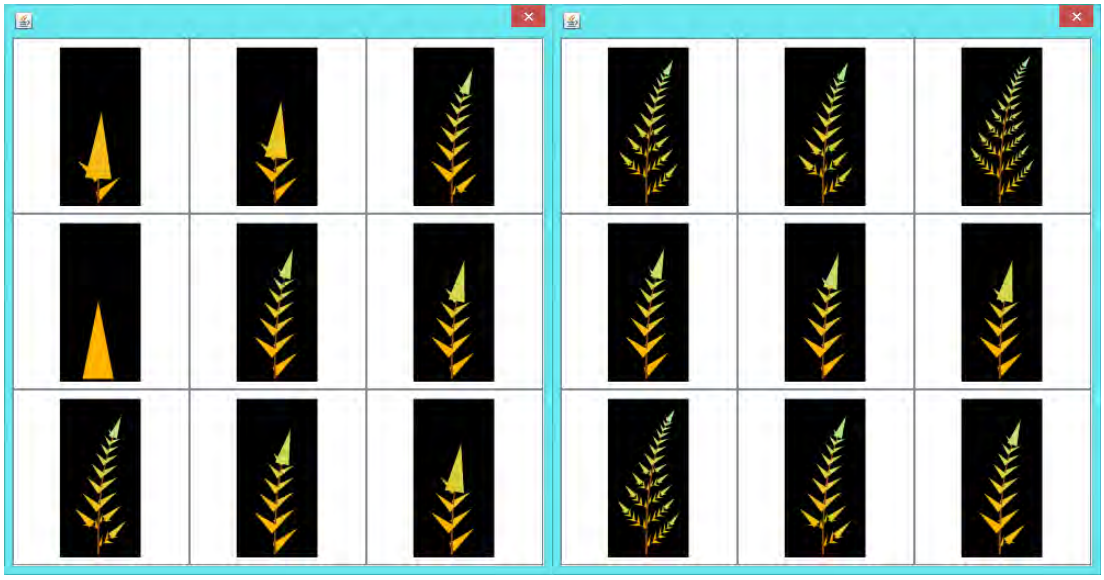


(c) Threshold run for grammar 0 and image 15 (d) Threshold run for grammar 1 and image 0

Figure D.25: Run of prototype to get similarity threshold values

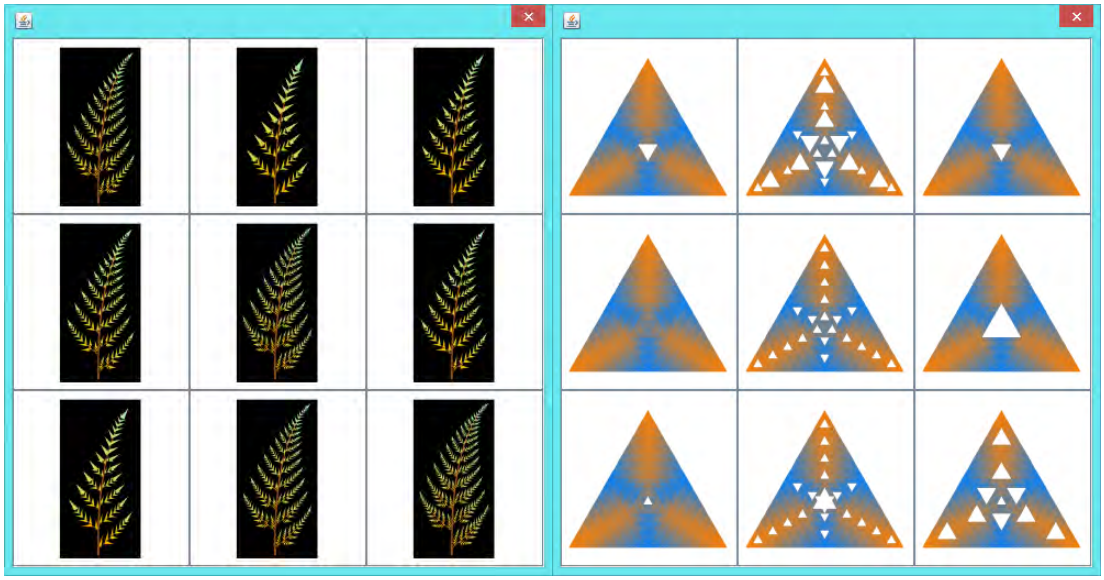


(a) Threshold run for grammar 1 and image 8 (b) Threshold run for grammar 1 and image 15

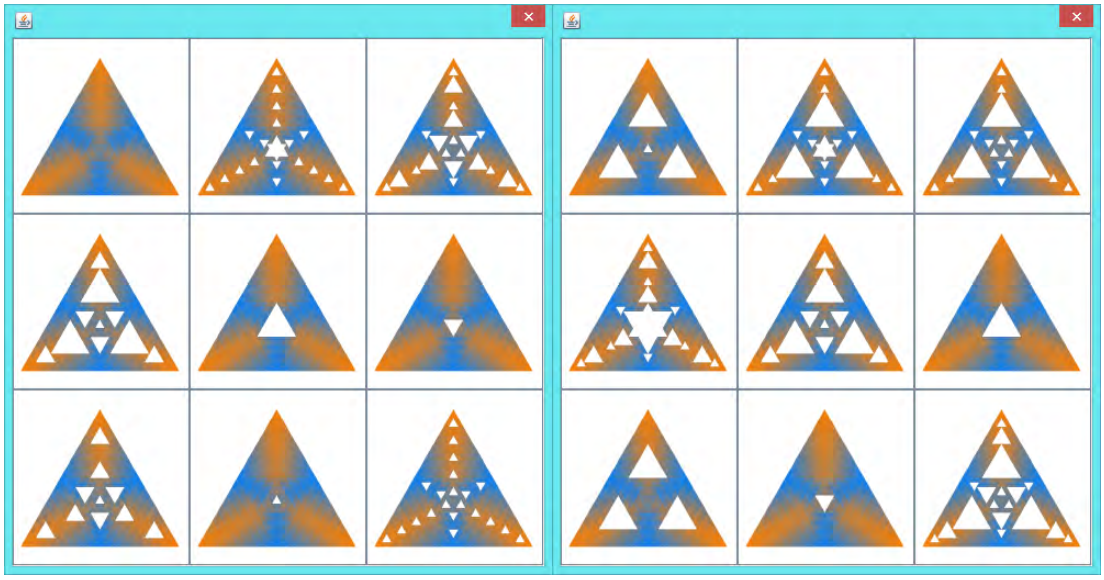


(c) Threshold run for grammar 2 and image 0 (d) Threshold run for grammar 2 and image 8

Figure D.26: Run of prototype to get similarity threshold values

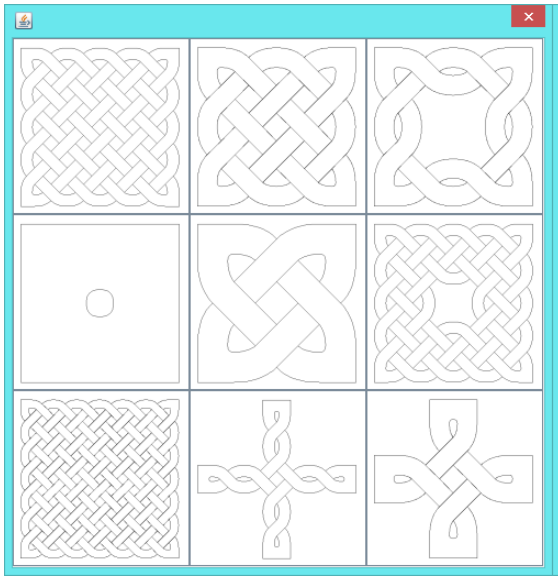


(a) Threshold run for grammar 2 and image 15 (b) Threshold run for grammar 3 and image 0

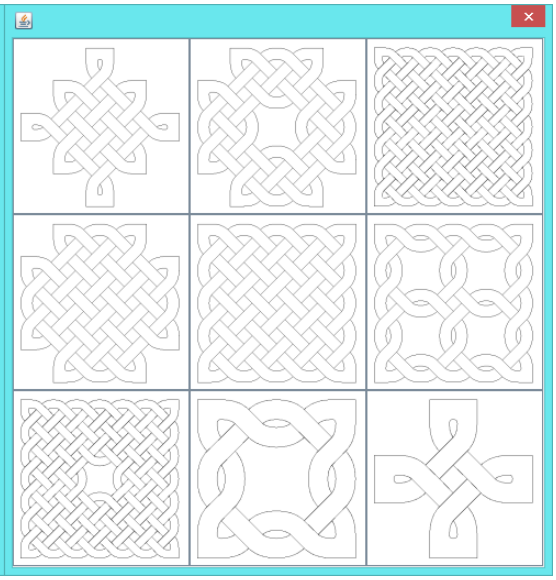


(c) Threshold run for grammar 3 and image 8 (d) Threshold run for grammar 3 and image 15

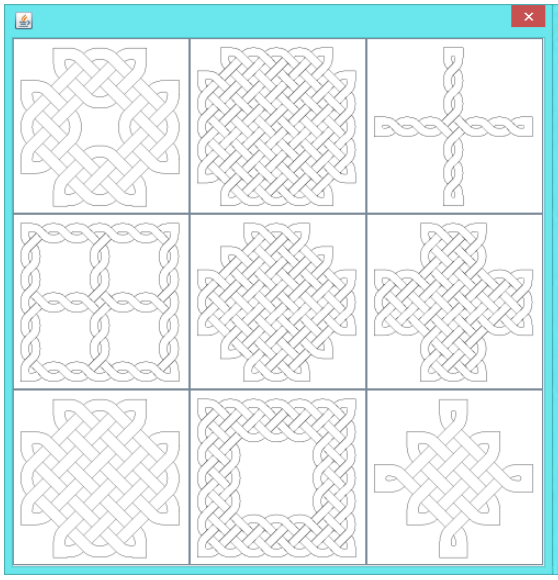
Figure D.27: Run of prototype to get similarity threshold values



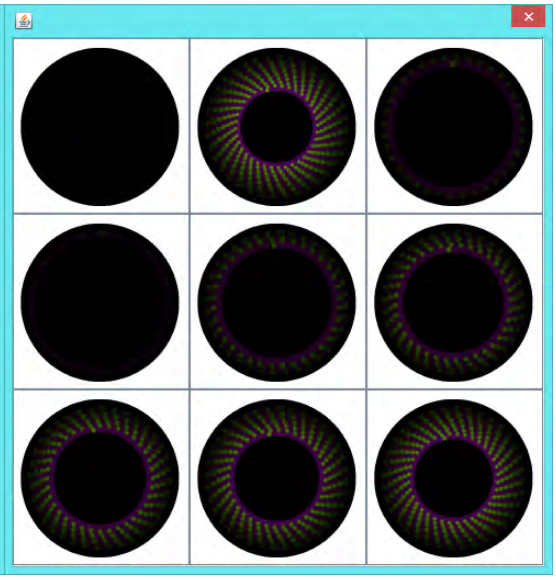
(a) Threshold run for grammar 4 and image 0



(b) Threshold run for grammar 4 and image 8

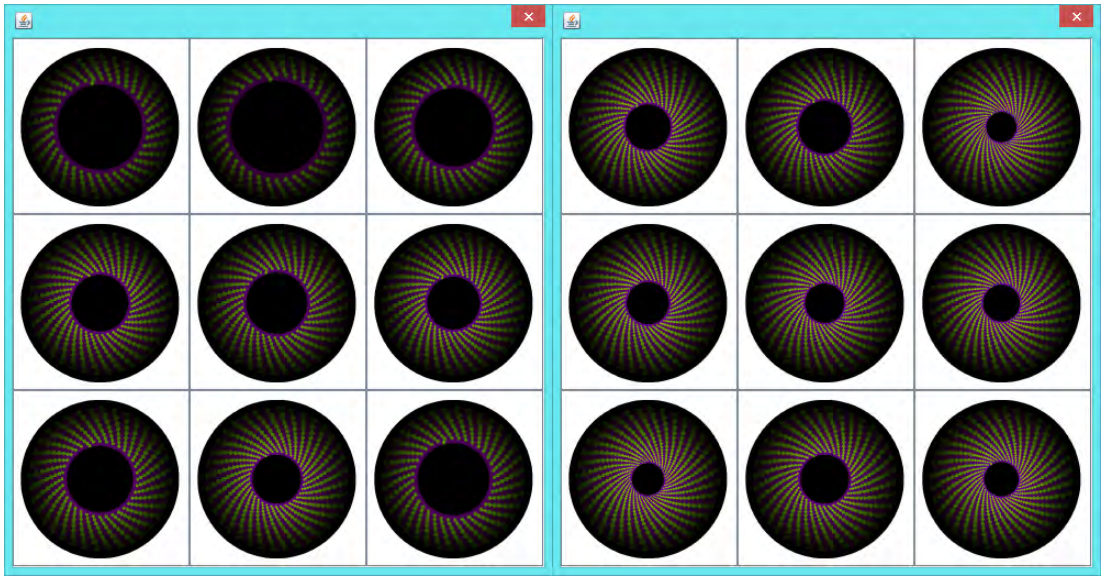


(c) Threshold run for grammar 4 and image 15

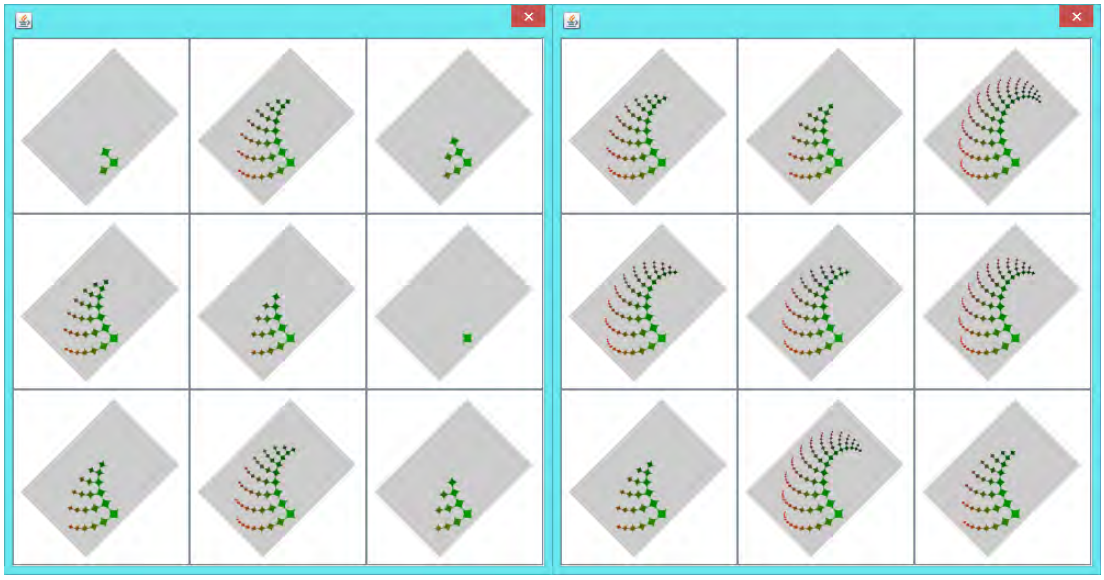


(d) Threshold run for grammar 5 and image 0

Figure D.28: Run of prototype to get similarity threshold values

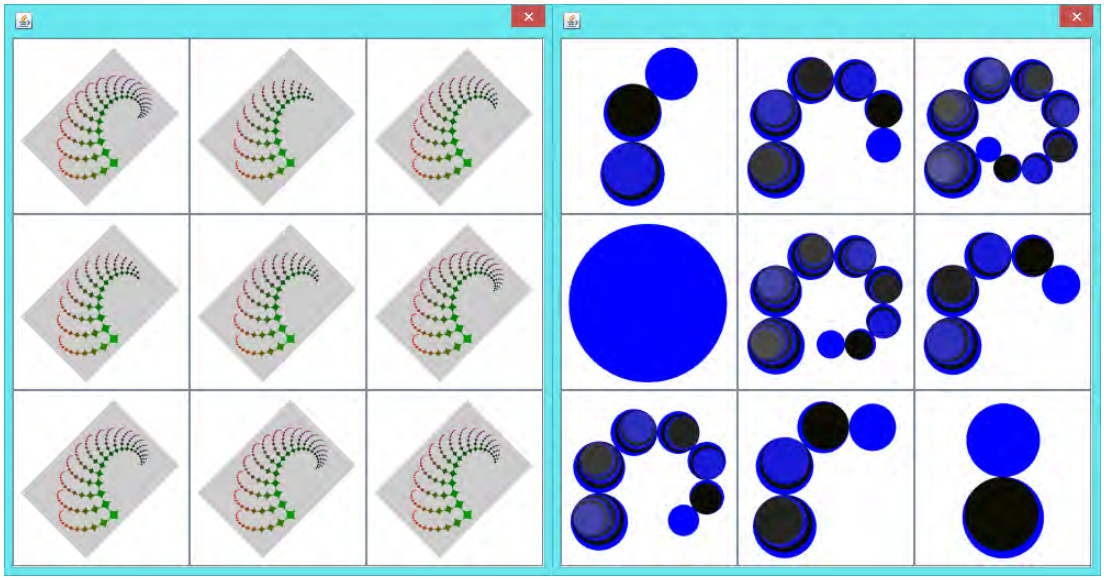


(a) Threshold run for grammar 5 and image 8 (b) Threshold run for grammar 5 and image 15

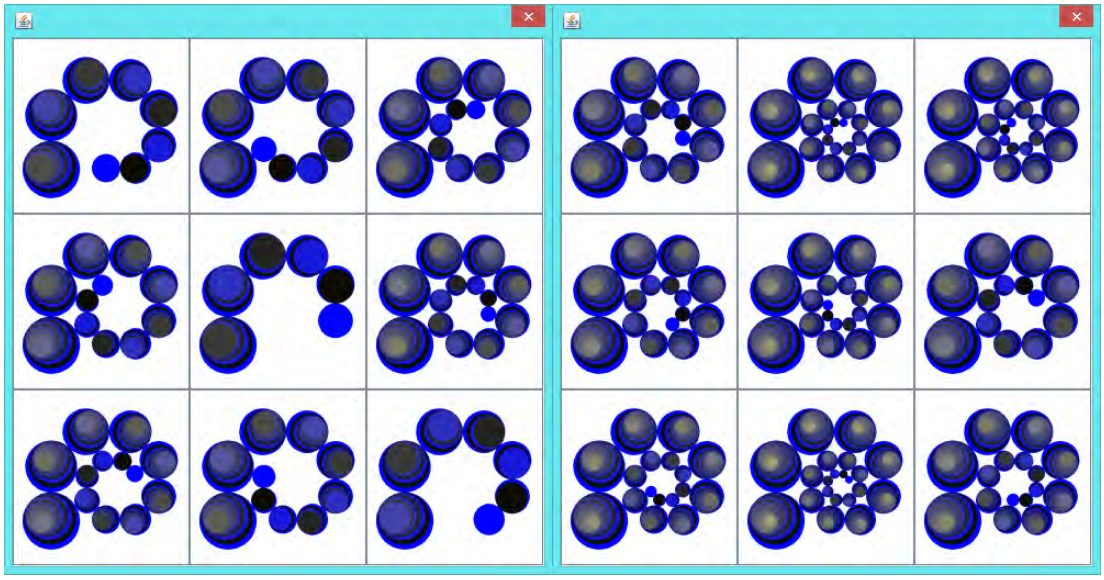


(c) Threshold run for grammar 6 and image 0 (d) Threshold run for grammar 6 and image 8

Figure D.29: Run of prototype to get similarity threshold values

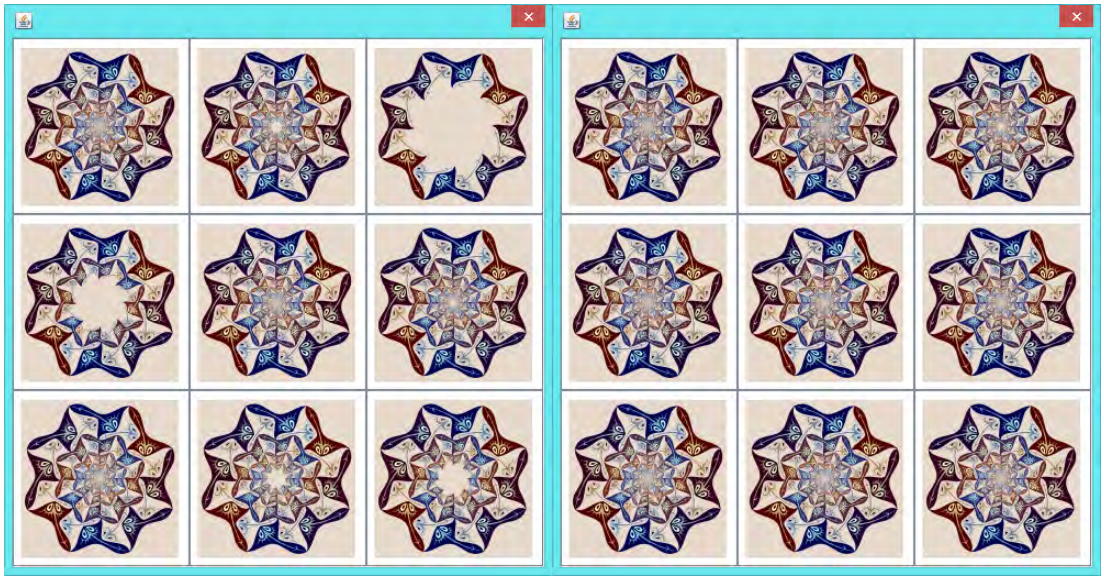


(a) Threshold run for grammar 6 and image 15 (b) Threshold run for grammar 7 and image 0

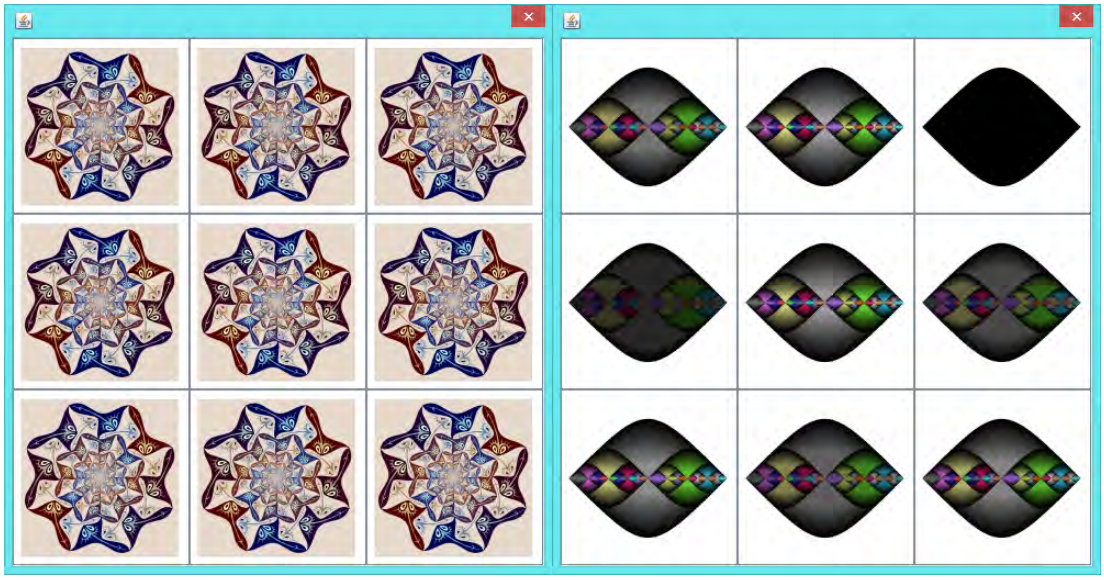


(c) Threshold run for grammar 7 and image 8 (d) Threshold run for grammar 7 and image 15

Figure D.30: Run of prototype to get similarity threshold values

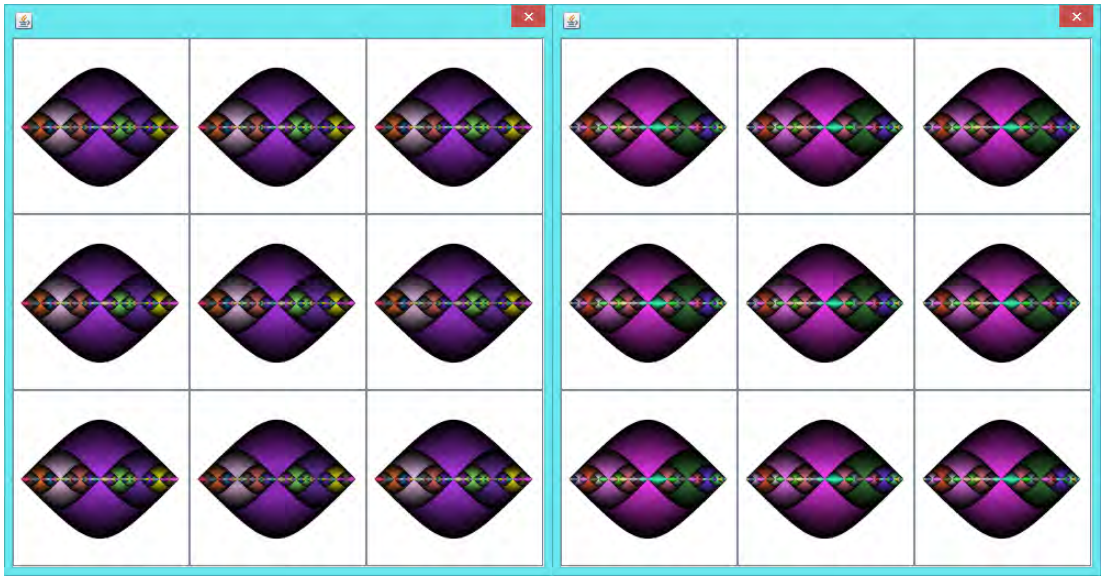


(a) Threshold run for grammar 8 and image 0 (b) Threshold run for grammar 8 and image 8

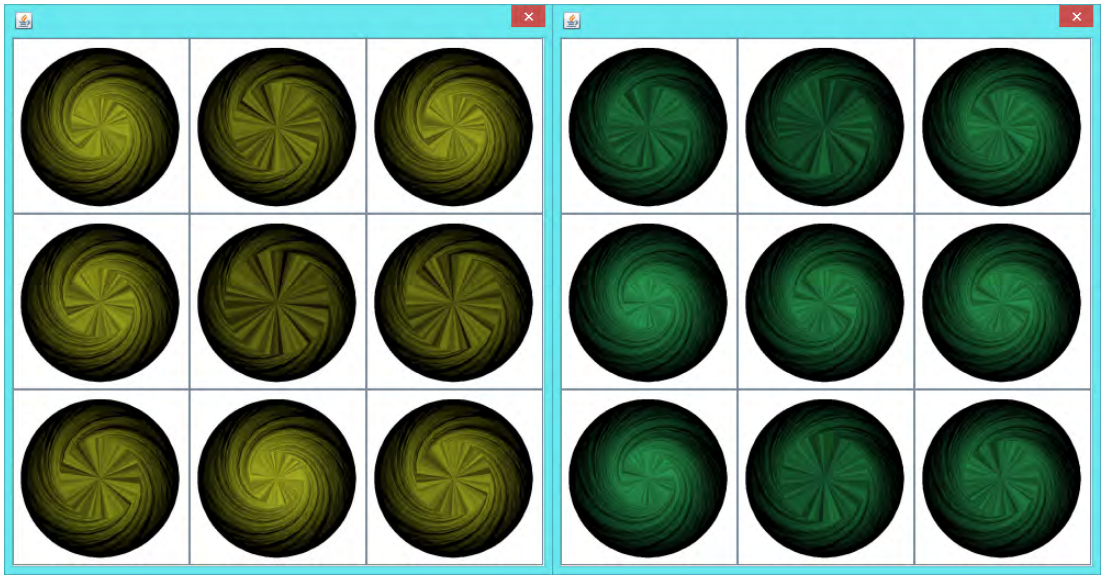


(c) Threshold run for grammar 8 and image 15 (d) Threshold run for grammar 9 and image 0

Figure D.31: Run of prototype to get similarity threshold values

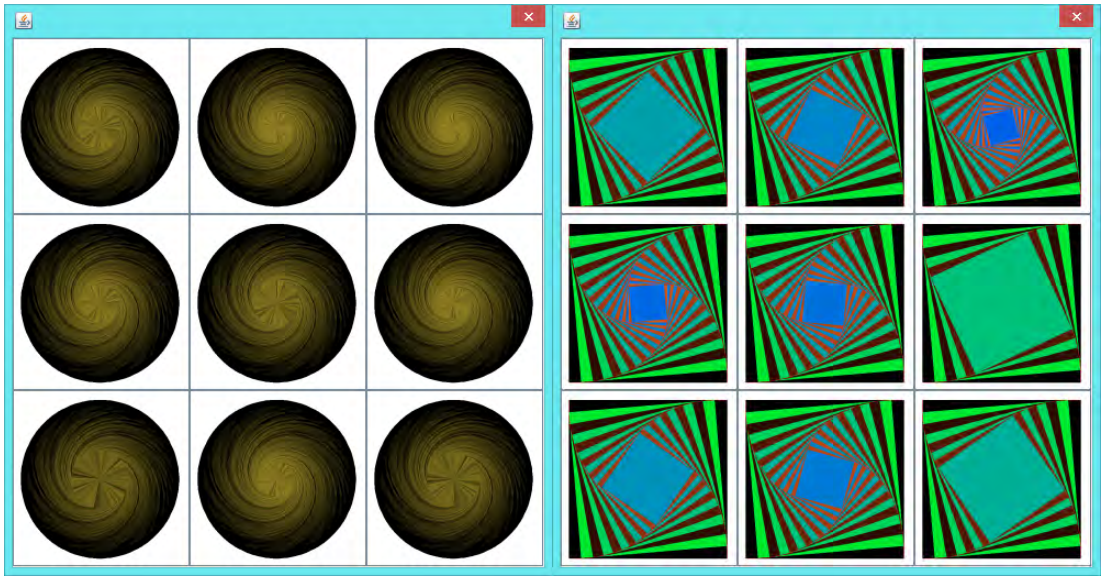


(a) Threshold run for grammar 9 and image 8 (b) Threshold run for grammar 9 and image 15

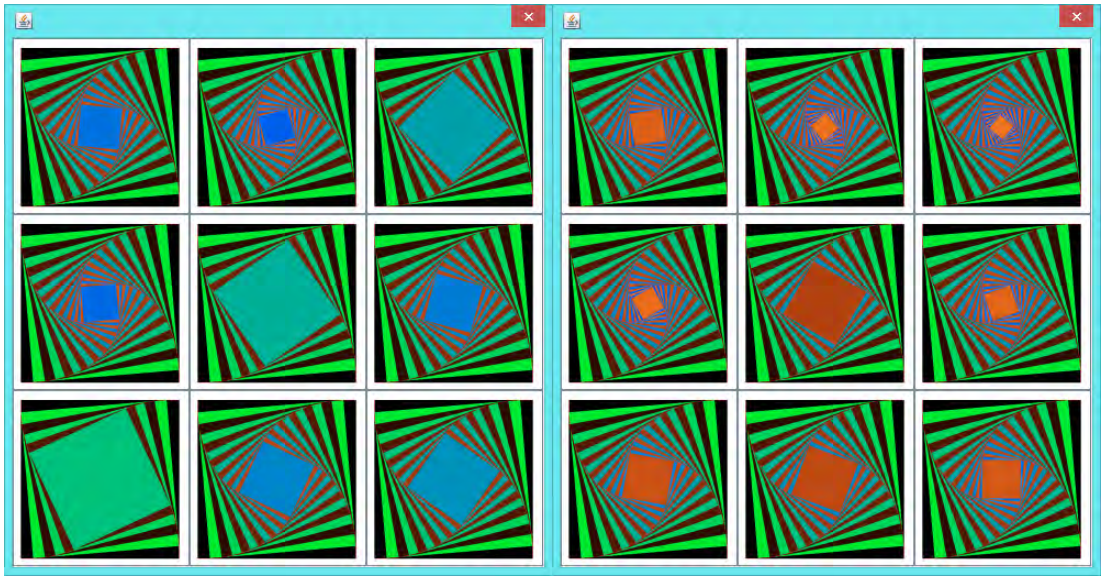


(c) Threshold run for grammar 10 and image 0(d) Threshold run for grammar 10 and image 8

Figure D.32: Run of prototype to get similarity threshold values

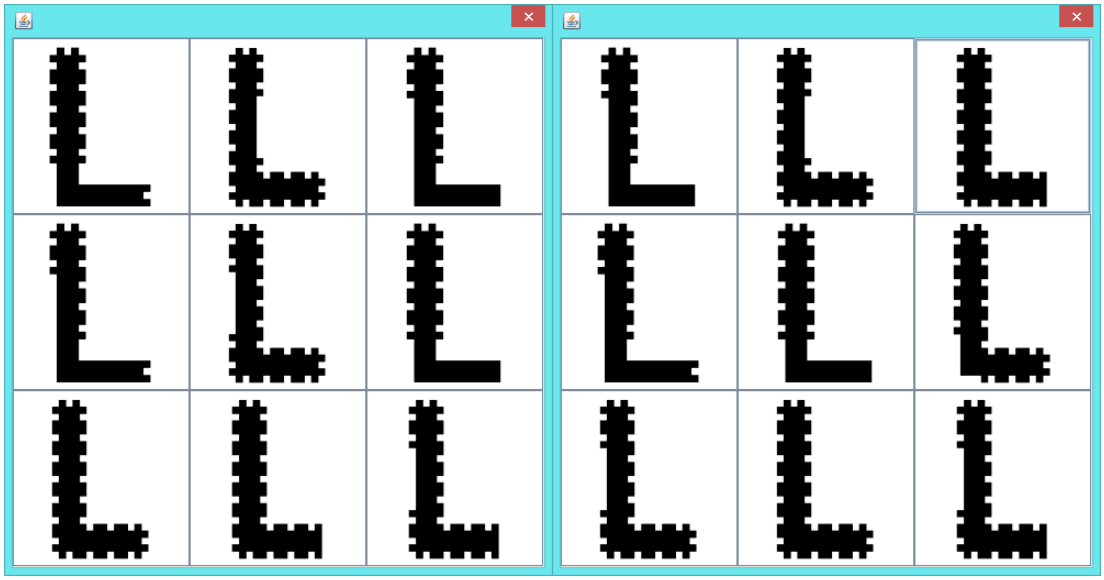


(a) Threshold run for grammar 10 and image 15(b) Threshold run for grammar 11 and image 0

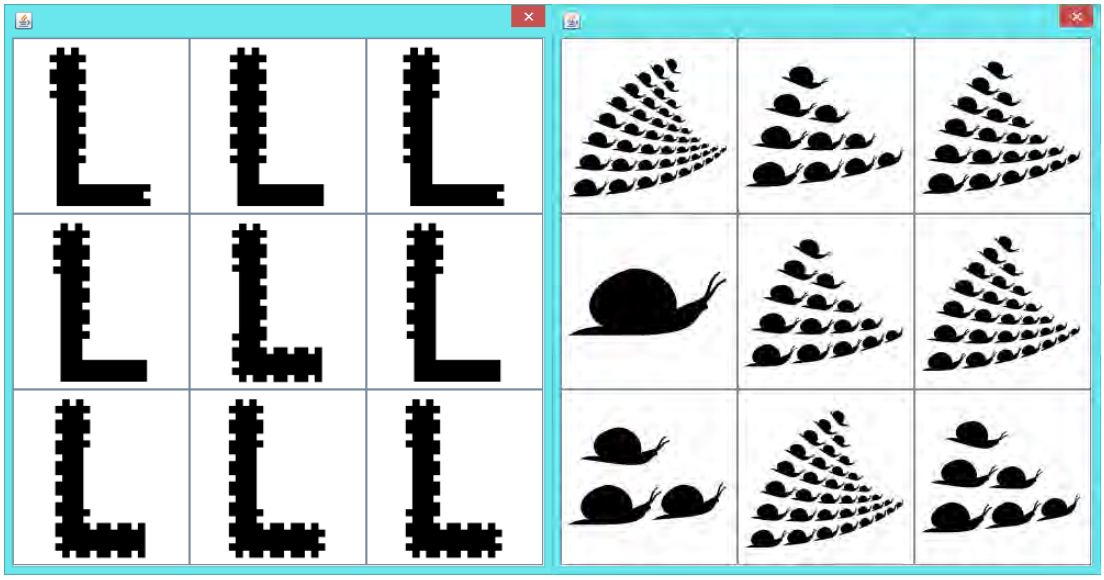


(c) Threshold run for grammar 11 and image 8(d) Threshold run for grammar 11 and image 15

Figure D.33: Run of prototype to get similarity threshold values

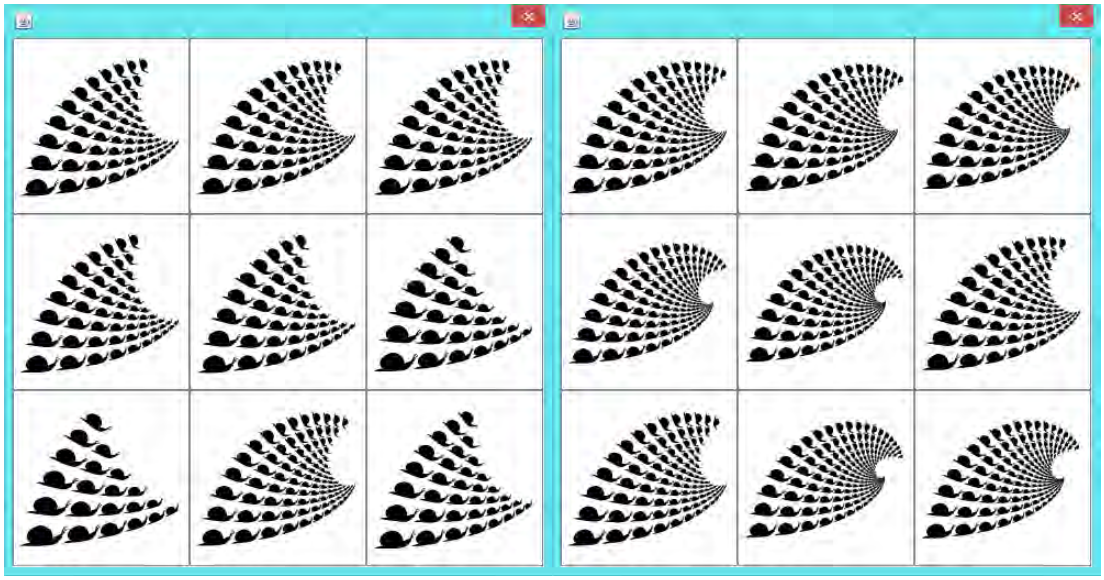


(a) Threshold run for grammar 12 and image 0(b) Threshold run for grammar 12 and image 8

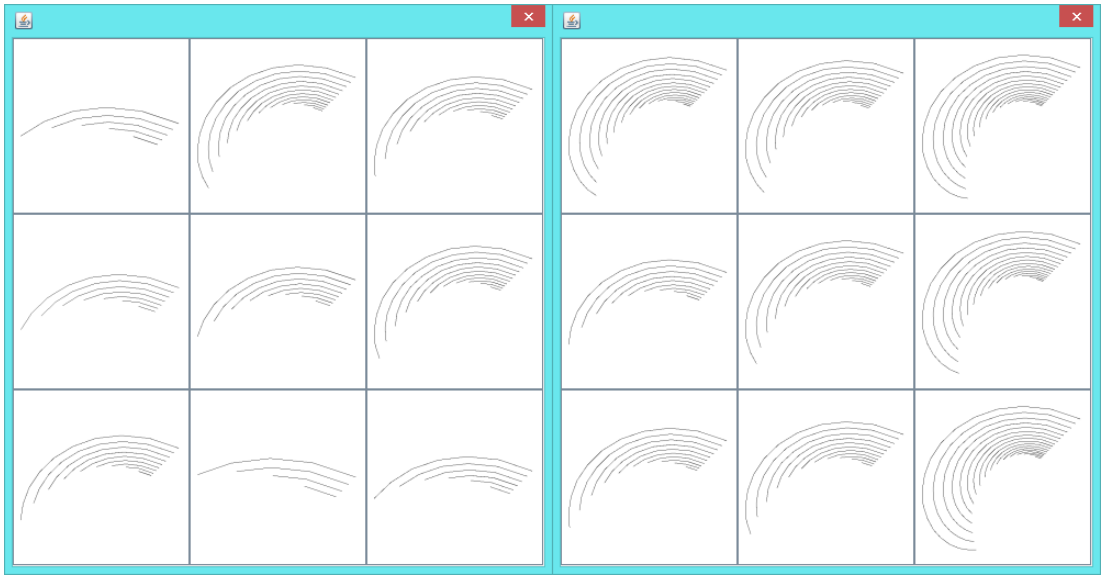


(c) Threshold run for grammar 12 and image 15(d) Threshold run for grammar 13 and image 0

Figure D.34: Run of prototype to get similarity threshold values

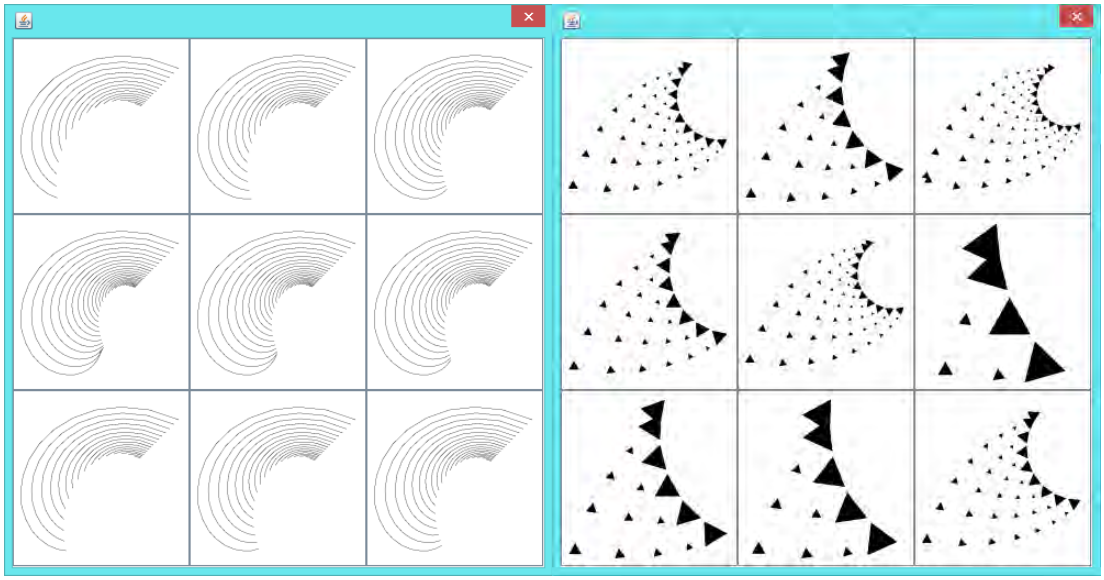


(a) Threshold run for grammar 13 and image 8(b) Threshold run for grammar 13 and image 15

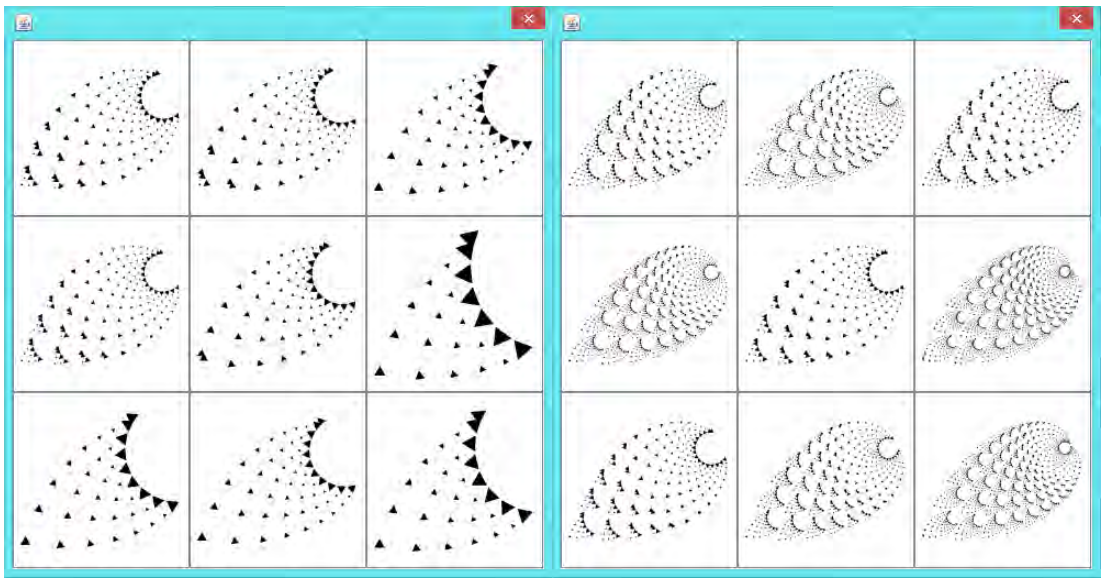


(c) Threshold run for grammar 14 and image 0(d) Threshold run for grammar 14 and image 8

Figure D.35: Run of prototype to get similarity threshold values

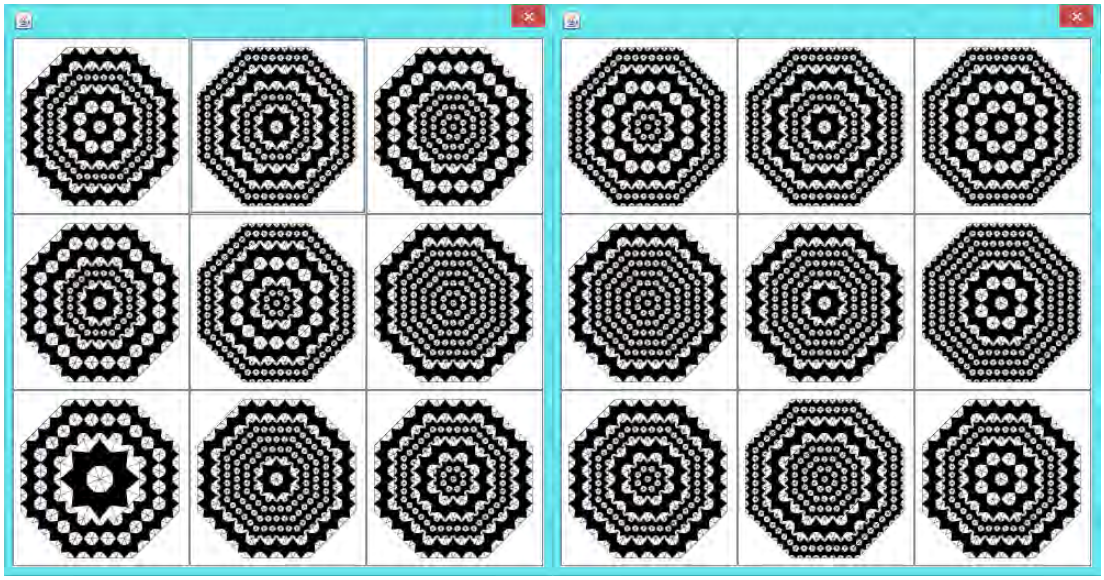


(a) Threshold run for grammar 14 and image 15(b) Threshold run for grammar 15 and image 0

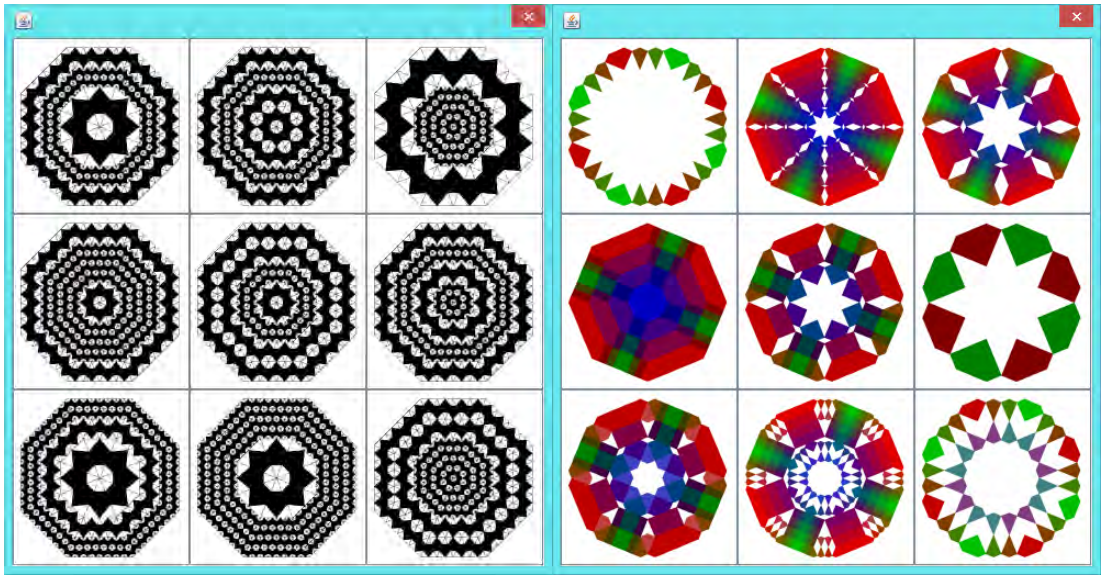


(c) Threshold run for grammar 15 and image 8(d) Threshold run for grammar 15 and image 15

Figure D.36: Run of prototype to get similarity threshold values

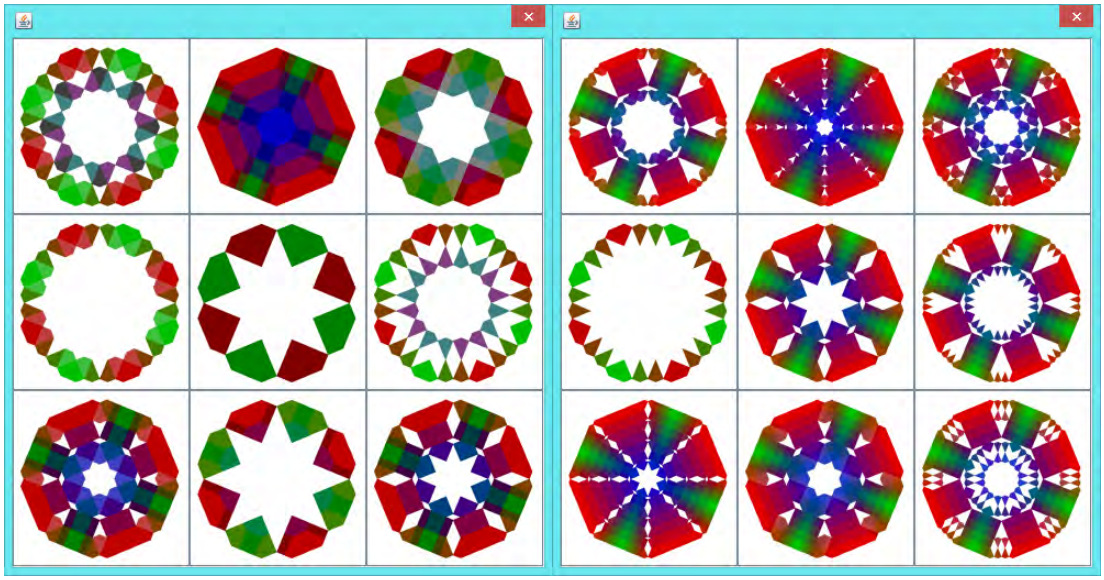


(a) Threshold run for grammar16 and image 0 (b) Threshold run for grammar 16 and image 8

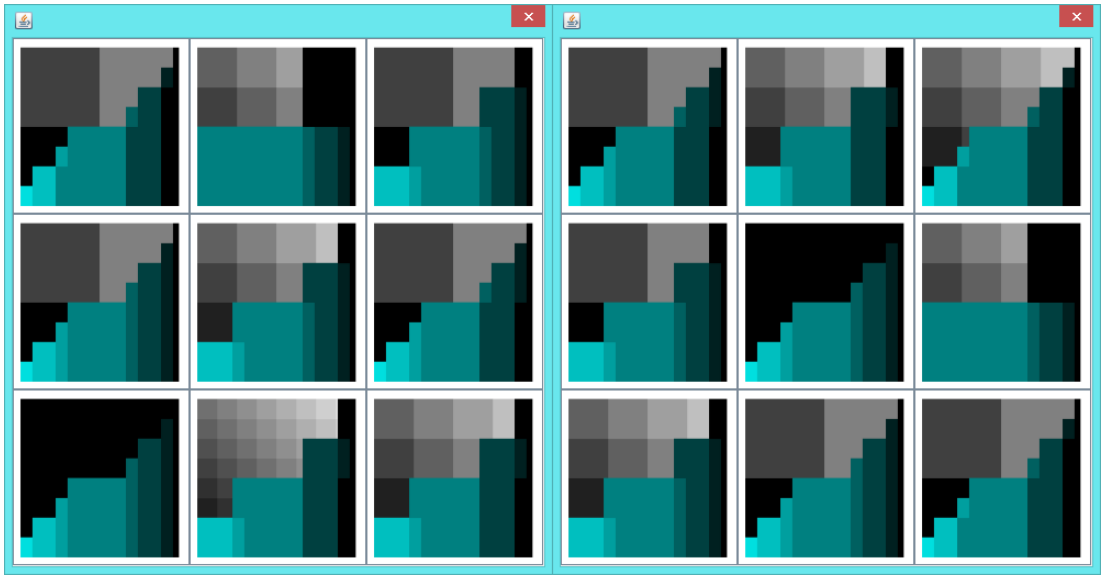


(c) Threshold run for grammar 16 and image 15(d) Threshold run for grammar 17 and image 0

Figure D.37: Run of prototype to get similarity threshold values

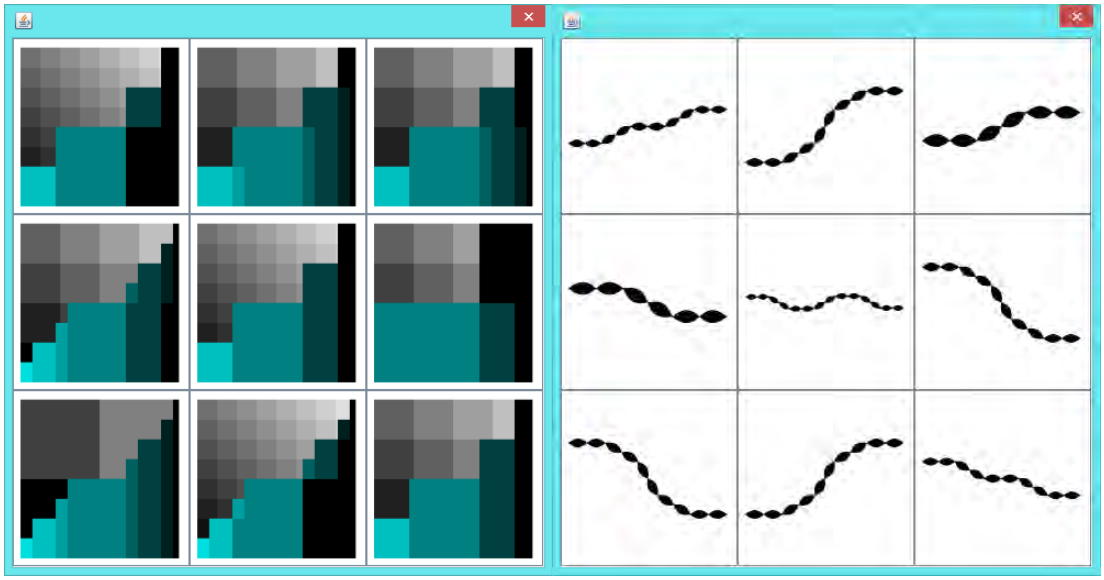


(a) Threshold run for grammar 17 and image 8(b) Threshold run for grammar 17 and image 15

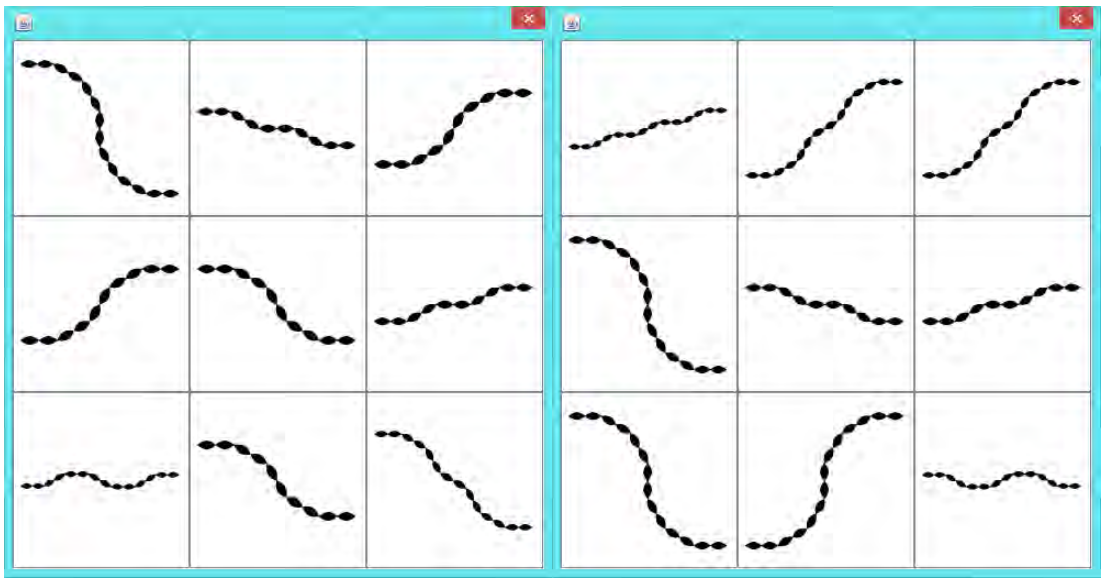


(c) Threshold run for grammar 18 and image 0(d) Threshold run for grammar 18 and image 8

Figure D.38: Run of prototype to get similarity threshold values

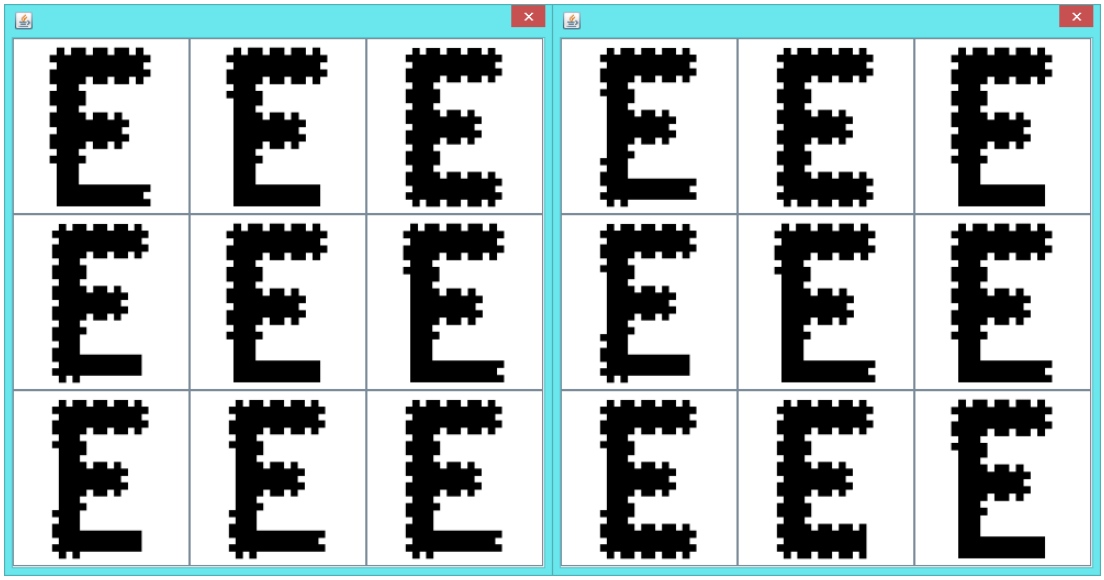


(a) Threshold run for grammar 18 and image 15(b) Threshold run for grammar 19 and image 0

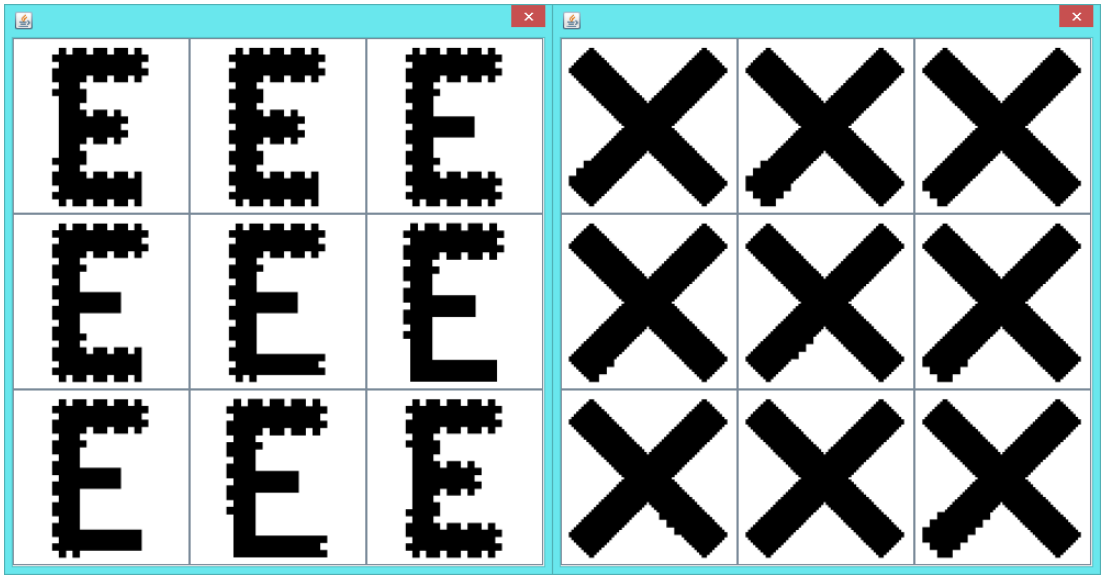


(c) Threshold run for grammar 19 and image 8(d) Threshold run for grammar 19 and image 15

Figure D.39: Run of prototype to get similarity threshold values

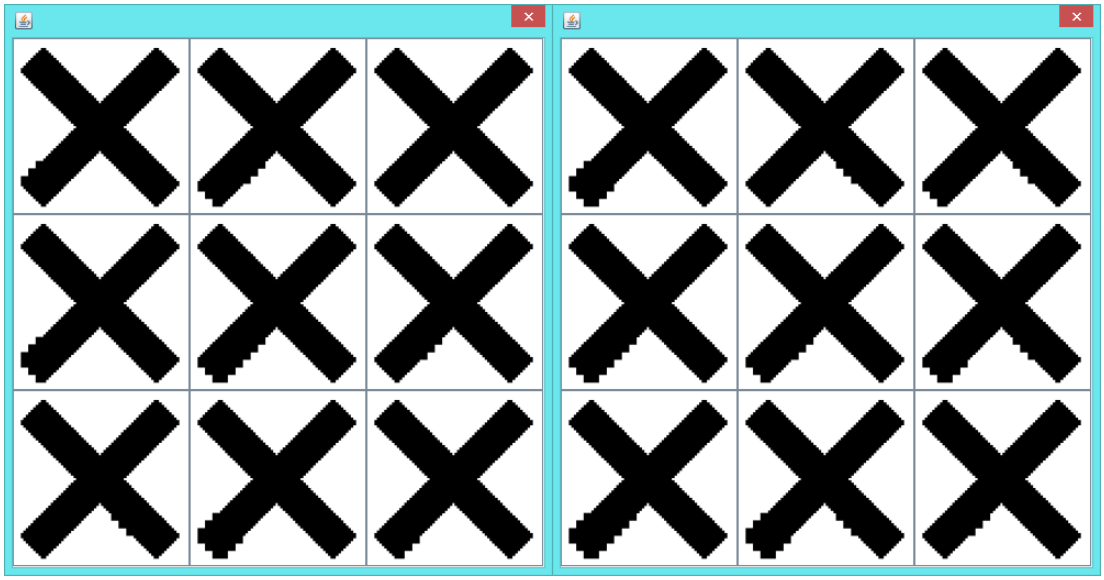


(a) Threshold run for grammar 20 and image 0(b) Threshold run for grammar 20 and image 8

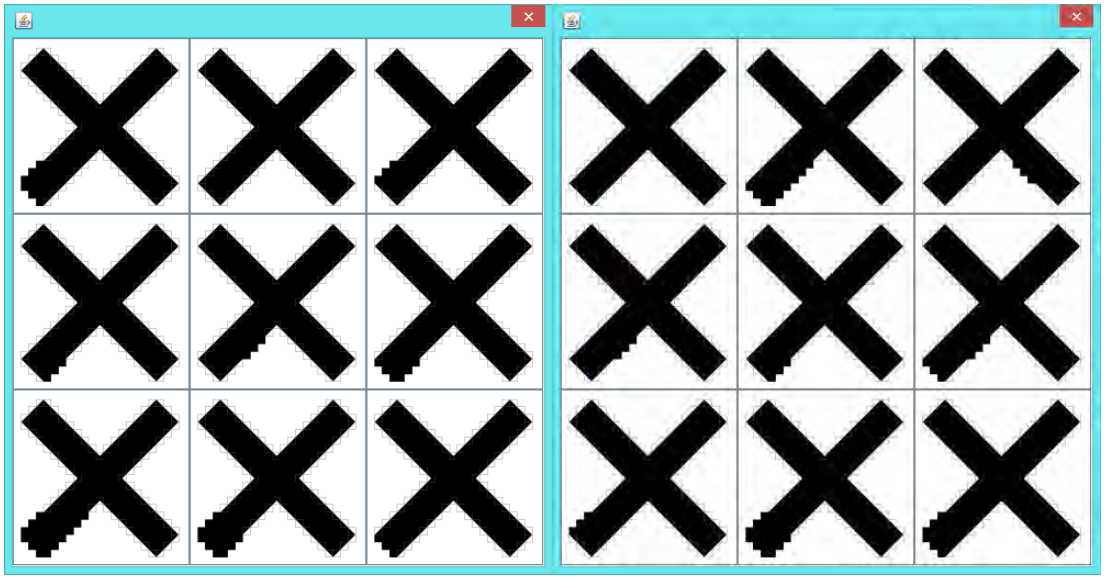


(c) Threshold run for grammar 20 and image 15(d) Threshold run for grammar 21 and image 0

Figure D.40: Run of prototype to get similarity threshold values

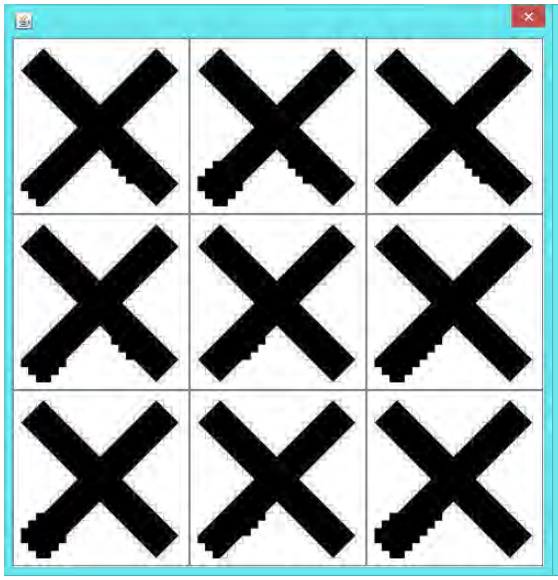


(a) Threshold run for grammar 21 and image 8(b) Threshold run for grammar 21 and image 15

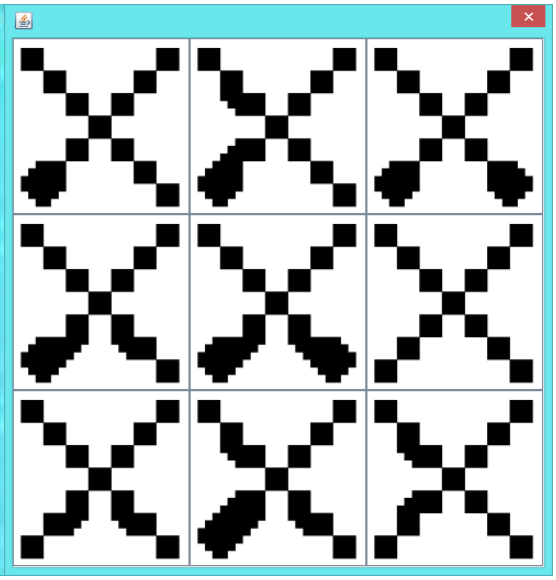


(c) Threshold run for grammar 22 and image 0(d) Threshold run for grammar 22 and image 8

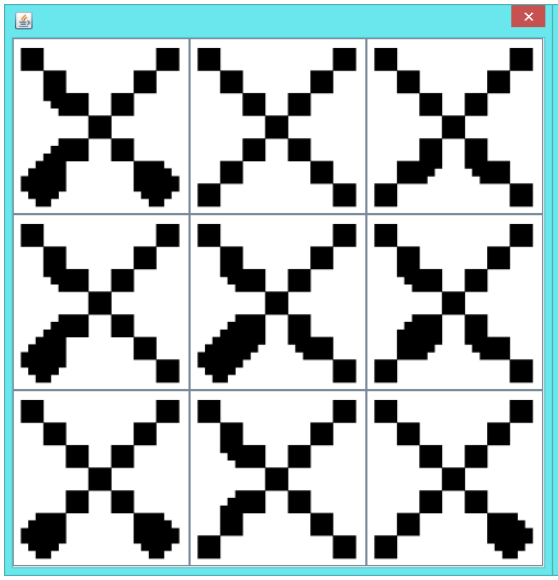
Figure D.41: Run of prototype to get similarity threshold values



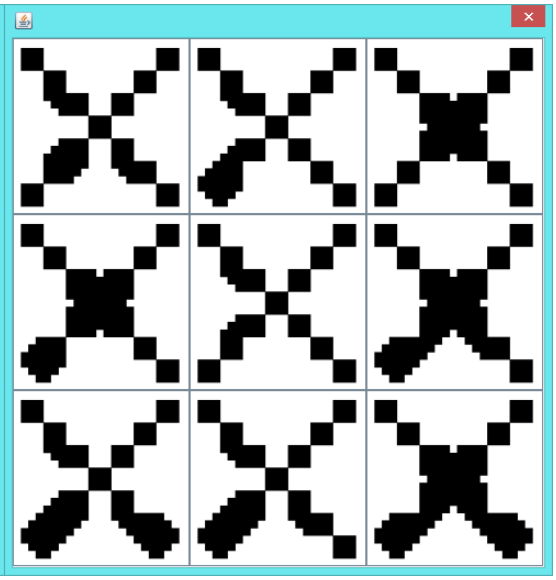
(a) Threshold run for grammar 22 and image 15



(b) Threshold run for grammar 23 and image 0



(c) Threshold run for grammar 23 and image 8



(d) Threshold run for grammar 23 and image 15

Figure D.42: Run of prototype to get similarity threshold values

Appendix E

Clustering of Online similarity Data

E.1 I Images Clusters

Table E.1: Image I cluster analysis

No.	I1	I2	I3	I4	I5	I6	I7	I8	Count	Percentage
1	1	2	4	6	3	5	7	8	132	19.96974281
2	1	2	5	6	3	4	7	8	96	14.52344932
3	1	2	4	7	3	5	6	8	25	3.78214826
4	1	3	4	6	2	5	7	8	23	3.479576399
5	1	3	5	7	4	6	8	2	21	3.177004539
6	1	3	6	7	4	5	8	2	19	2.874432678
7	1	2	5	7	3	4	6	8	18	2.723146747
8	1	2	4	5	3	7	6	8	17	2.571860817
9	1	2	4	5	3	6	7	8	17	2.571860817
10	1	2	4	7	3	5	8	6	13	1.966717095

Continued on next page

Table E.1 – continued from previous page

No.	I1	I2	I3	I4	I5	I6	I7	I8	Count	Percentage
11	1	3	5	6	2	4	7	8	9	1.361573374
12	1	3	4	7	2	5	6	8	9	1.361573374
13	1	2	7	4	3	5	6	8	7	1.059001513
14	1	2	5	7	3	6	8	4	7	1.059001513
15	1	2	6	7	3	5	8	4	7	1.059001513
16	1	2	7	5	3	4	6	8	6	0.907715582
17	1	2	3	6	4	5	7	8	6	0.907715582
18	1	2	4	6	3	7	5	8	5	0.756429652
19	1	2	4	7	3	6	8	5	5	0.756429652
20	1	3	2	6	4	5	7	8	5	0.756429652
21	1	2	5	6	4	3	7	8	4	0.605143722
22	1	2	5	7	3	4	8	6	4	0.605143722
23	1	5	3	7	6	4	8	2	3	0.453857791
24	1	3	5	6	4	7	8	2	3	0.453857791
25	1	2	6	7	3	4	8	5	3	0.453857791
26	1	2	5	6	3	8	7	4	3	0.453857791
27	1	4	5	8	3	6	7	2	3	0.453857791
28	1	2	5	6	3	7	4	8	3	0.453857791
29	1	4	6	7	3	5	8	2	3	0.453857791
30	1	2	6	5	3	4	7	8	3	0.453857791
31	1	4	3	7	5	6	8	2	3	0.453857791
32	1	2	4	7	3	6	5	8	3	0.453857791
33	1	2	6	4	5	7	3	8	2	0.302571861
34	1	2	6	3	4	7	5	8	2	0.302571861
35	1	3	7	5	2	4	6	8	2	0.302571861
36	1	2	4	3	5	6	7	8	2	0.302571861
37	1	2	4	3	6	7	5	8	2	0.302571861
38	1	3	5	8	2	4	7	6	2	0.302571861
39	1	2	3	4	6	5	7	8	2	0.302571861

Continued on next page

Table E.1 – continued from previous page

No.	I1	I2	I3	I4	I5	I6	I7	I8	Count	Percentage
40	1	4	2	6	5	3	7	8	2	0.302571861
41	1	2	3	5	4	6	7	8	2	0.302571861
42	1	3	6	7	2	4	5	8	2	0.302571861
43	1	2	6	4	3	7	5	8	2	0.302571861
44	2	3	6	7	4	5	8	1	2	0.302571861
45	1	2	7	3	6	4	5	8	2	0.302571861
46	1	2	6	7	3	4	5	8	2	0.302571861
47	1	3	5	7	2	4	6	8	2	0.302571861
48	1	3	2	5	4	7	6	8	2	0.302571861
49	1	2	8	6	3	5	7	4	2	0.302571861
50	1	2	3	4	5	6	7	8	2	0.302571861
51	1	4	3	8	5	6	7	2	2	0.302571861
52	1	3	5	8	4	6	7	2	2	0.302571861
53	1	3	5	7	4	8	6	2	2	0.302571861
54	1	4	3	6	5	2	7	8	2	0.302571861
55	1	3	4	7	2	5	8	6	2	0.302571861
56	1	2	6	8	3	5	7	4	2	0.302571861
57	1	3	4	5	2	6	7	8	2	0.302571861
58	1	3	7	6	2	4	5	8	2	0.302571861
59	1	2	5	7	3	6	4	8	2	0.302571861
60	1	2	7	3	4	5	6	8	2	0.302571861
61	1	4	6	8	3	5	7	2	2	0.302571861
62	1	2	5	8	3	6	7	4	2	0.302571861
63	1	3	8	6	4	5	7	2	2	0.302571861
64	1	3	4	7	5	6	8	2	2	0.302571861
65	1	3	4	5	2	7	6	8	2	0.302571861
66	1	2	3	7	4	5	6	8	2	0.302571861
67	1	4	5	7	3	6	8	2	1	0.15128593
68	1	6	2	3	7	8	4	5	1	0.15128593

Continued on next page

Table E.1 – continued from previous page

No.	I1	I2	I3	I4	I5	I6	I7	I8	Count	Percentage
69	2	3	6	8	4	5	7	1	1	0.15128593
70	1	3	5	4	2	6	7	8	1	0.15128593
71	1	3	4	7	2	6	5	8	1	0.15128593
72	1	2	4	7	3	8	5	6	1	0.15128593
73	1	5	6	3	8	2	7	4	1	0.15128593
74	1	2	4	8	3	5	7	6	1	0.15128593
75	1	4	8	5	3	6	7	2	1	0.15128593
76	1	3	7	8	4	5	6	2	1	0.15128593
77	1	5	2	7	6	3	8	4	1	0.15128593
78	1	2	4	3	5	7	6	8	1	0.15128593
79	1	7	2	5	6	3	8	4	1	0.15128593
80	1	3	6	7	2	4	8	5	1	0.15128593
81	1	6	2	4	5	3	7	8	1	0.15128593
82	1	2	6	3	5	4	7	8	1	0.15128593
83	1	5	2	8	6	4	7	3	1	0.15128593
84	3	2	6	1	7	4	8	5	1	0.15128593
85	1	6	3	4	5	2	7	8	1	0.15128593
86	1	5	4	6	7	3	8	2	1	0.15128593
87	2	3	7	6	4	8	5	1	1	0.15128593
88	1	2	7	4	3	6	5	8	1	0.15128593
89	1	6	5	4	2	7	3	8	1	0.15128593
90	7	1	5	2	4	6	3	8	1	0.15128593
91	7	5	3	1	6	4	2	8	1	0.15128593
92	5	8	3	1	6	2	4	7	1	0.15128593
93	1	3	4	5	2	7	8	6	1	0.15128593
94	1	5	2	3	6	7	4	8	1	0.15128593
95	3	2	5	7	1	4	6	8	1	0.15128593
96	1	3	5	6	2	8	7	4	1	0.15128593
97	1	5	8	2	4	3	6	7	1	0.15128593

Continued on next page

Table E.1 – continued from previous page

No.	I1	I2	I3	I4	I5	I6	I7	I8	Count	Percentage
98	1	2	7	3	5	4	6	8	1	0.15128593
99	2	3	5	6	4	8	7	1	1	0.15128593
100	1	2	4	6	7	5	3	8	1	0.15128593
101	1	3	4	7	6	5	8	2	1	0.15128593
102	1	3	5	4	2	7	8	6	1	0.15128593
103	1	3	2	7	4	5	6	8	1	0.15128593
104	1	2	6	7	4	3	5	8	1	0.15128593
105	2	4	6	3	7	5	8	1	1	0.15128593
106	1	7	6	2	8	4	3	5	1	0.15128593
107	1	2	6	8	3	7	4	5	1	0.15128593
108	1	2	7	6	3	5	4	8	1	0.15128593
109	1	2	5	8	3	4	7	6	1	0.15128593
110	1	4	3	6	5	7	8	2	1	0.15128593
111	7	5	4	1	6	2	3	8	1	0.15128593
112	1	2	8	6	7	4	3	5	1	0.15128593
113	2	1	4	5	3	7	6	8	1	0.15128593
114	1	5	3	6	4	8	7	2	1	0.15128593
115	1	3	7	2	5	4	6	8	1	0.15128593
116	4	7	6	8	2	3	1	5	1	0.15128593
117	1	6	7	8	3	2	5	4	1	0.15128593
118	1	3	6	5	4	7	8	2	1	0.15128593
119	1	5	2	6	4	3	7	8	1	0.15128593
120	1	2	6	7	3	8	4	5	1	0.15128593
121	7	3	5	1	2	6	4	8	1	0.15128593
122	1	2	3	5	7	6	4	8	1	0.15128593
123	1	3	5	6	4	2	7	8	1	0.15128593
124	1	4	6	2	3	5	7	8	1	0.15128593
125	1	3	5	4	6	2	7	8	1	0.15128593
126	1	3	7	4	2	6	5	8	1	0.15128593

Continued on next page

Table E.1 – continued from previous page

No.	I1	I2	I3	I4	I5	I6	I7	I8	Count	Percentage
127	1	3	5	7	2	6	8	4	1	0.15128593
128	3	5	6	1	4	7	8	2	1	0.15128593
129	1	3	4	7	2	6	8	5	1	0.15128593
130	1	3	7	6	4	2	8	5	1	0.15128593
131	8	3	6	1	2	5	4	7	1	0.15128593
132	1	2	4	6	5	7	3	8	1	0.15128593
133	1	8	6	4	7	5	2	3	1	0.15128593
134	1	2	7	3	5	6	4	8	1	0.15128593
135	1	2	5	6	7	4	3	8	1	0.15128593
136	1	5	4	8	7	2	3	6	1	0.15128593
137	1	5	7	8	3	4	6	2	1	0.15128593
138	1	4	5	6	2	3	7	8	1	0.15128593
139	1	2	6	8	3	4	7	5	1	0.15128593
140	1	2	6	4	3	5	7	8	1	0.15128593
141	1	2	7	6	3	4	5	8	1	0.15128593
142	1	2	6	4	5	3	7	8	1	0.15128593
143	1	4	8	6	5	3	7	2	1	0.15128593
144	1	2	8	6	4	3	7	5	1	0.15128593
145	1	2	7	5	4	3	6	8	1	0.15128593
146	1	5	8	4	2	7	6	3	1	0.15128593
147	7	6	3	2	5	4	1	8	1	0.15128593
148	1	6	2	3	7	4	5	8	1	0.15128593
149	1	2	4	3	6	5	7	8	1	0.15128593
150	1	3	4	8	7	5	6	2	1	0.15128593
151	1	2	8	7	4	3	6	5	1	0.15128593
152	1	5	2	8	4	6	7	3	1	0.15128593
153	5	6	3	1	2	4	7	8	1	0.15128593
154	1	2	7	4	3	8	6	5	1	0.15128593
155	1	2	3	4	6	7	5	8	1	0.15128593

Continued on next page

Table E.1 – continued from previous page

No.	I1	I2	I3	I4	I5	I6	I7	I8	Count	Percentage
156	1	3	2	7	4	5	8	6	1	0.15128593
157	1	2	3	5	4	7	6	8	1	0.15128593
158	1	3	5	7	2	4	8	6	1	0.15128593
159	1	5	3	7	4	2	6	8	1	0.15128593
160	1	2	3	6	5	4	7	8	1	0.15128593
161	1	2	6	8	4	5	7	3	1	0.15128593
162	1	2	5	3	6	4	7	8	1	0.15128593
163	1	4	3	7	5	2	6	8	1	0.15128593
164	1	2	6	7	4	5	8	3	1	0.15128593
165	1	2	3	6	4	8	5	7	1	0.15128593
166	3	1	8	5	2	4	6	7	1	0.15128593
167	1	4	3	6	2	5	7	8	1	0.15128593
168	1	4	5	7	6	3	8	2	1	0.15128593
169	1	2	4	8	6	5	7	3	1	0.15128593
170	1	4	3	7	6	8	2	5	1	0.15128593
171	1	3	4	6	2	7	5	8	1	0.15128593
172	1	4	2	6	7	3	5	8	1	0.15128593
173	1	4	3	6	5	8	7	2	1	0.15128593
174	1	6	7	4	3	2	5	8	1	0.15128593
Total									661	100 %

E.2 E to T Images Clusters

Table E.2: Images E to T cluster analysis

No.	E	F	L	I	H	T	Count	Percentage
1	1	2	3	4	6	5	194	29.3494705
2	1	2	3	4	5	6	97	14.67473525
3	1	2	3	5	6	4	63	9.531013616
4	1	2	3	6	5	4	44	6.656580938
5	1	2	3	5	4	6	40	6.051437216
6	1	2	3	6	4	5	37	5.597579425
7	1	2	4	6	3	5	37	5.597579425
8	1	2	5	6	3	4	31	4.689863843
9	1	2	4	5	3	6	19	2.874432678
10	1	2	4	5	6	3	16	2.420574887
11	1	2	4	6	5	3	12	1.815431165
12	1	2	5	4	3	6	9	1.361573374
13	1	2	4	3	6	5	8	1.210287443
14	1	2	5	6	4	3	7	1.059001513
15	1	2	4	3	5	6	6	0.907715582
16	1	2	6	5	3	4	3	0.453857791
17	1	3	5	6	2	4	3	0.453857791
18	1	2	5	3	4	6	3	0.453857791
19	1	2	6	4	3	5	3	0.453857791
20	1	2	5	4	6	3	2	0.302571861
21	1	2	5	3	6	4	2	0.302571861
22	1	3	4	2	5	6	2	0.302571861
23	1	3	4	6	2	5	2	0.302571861
24	2	1	6	4	5	3	1	0.15128593
25	2	3	1	6	5	4	1	0.15128593
26	1	4	5	6	2	3	1	0.15128593

Continued on next page

Table E.2 – continued from previous page

No.	E	F	L	I	H	T	Count	Percentage
27	1	3	2	5	4	6	1	0.15128593
28	3	4	5	6	2	1	1	0.15128593
29	2	3	4	6	1	5	1	0.15128593
30	4	1	6	3	2	5	1	0.15128593
31	2	3	5	6	1	4	1	0.15128593
32	4	3	2	1	6	5	1	0.15128593
33	1	3	4	5	2	6	1	0.15128593
34	1	3	6	4	2	5	1	0.15128593
35	4	3	6	1	5	2	1	0.15128593
36	1	6	5	2	3	4	1	0.15128593
37	1	2	6	4	5	3	1	0.15128593
38	1	4	2	3	6	5	1	0.15128593
39	4	3	5	6	2	1	1	0.15128593
40	1	3	4	2	6	5	1	0.15128593
41	3	2	4	6	1	5	1	0.15128593
42	5	4	3	1	6	2	1	0.15128593
43	1	5	4	6	2	3	1	0.15128593
44	2	6	3	5	4	1	1	0.15128593
Total							661	100 %

E.3 Fern Images Clusters

Table E.3: Image fern cluster analysis

No.	F1	F2	F3	F4	F5	F6	F7	F8	F9	Count	Percentage
1	1	2	3	4	5	6	7	8	9	287	44.29012346
Continued on next page											

Table E.3 – continued from previous page

No.	F1	F2	F3	F4	F5	F6	F7	F8	F9	Count	Percentage
2	1	2	3	4	5	6	7	9	8	44	6.790123457
3	1	2	3	4	5	7	6	8	9	24	3.703703704
4	1	2	3	4	6	5	7	8	9	23	3.549382716
5	1	2	3	4	5	6	8	7	9	23	3.549382716
6	1	2	3	5	4	6	7	8	9	15	2.314814815
7	2	1	3	4	5	6	7	8	9	9	1.388888889
8	1	2	3	4	6	5	7	9	8	9	1.388888889
9	1	2	3	4	6	5	8	7	9	7	1.080246914
10	1	2	3	5	4	6	7	9	8	6	0.925925926
11	1	3	2	4	5	6	7	8	9	5	0.771604938
12	1	2	4	3	5	6	7	8	9	5	0.771604938
13	2	1	3	4	5	7	6	8	9	5	0.771604938
14	1	2	3	4	5	7	6	9	8	5	0.771604938
15	1	2	3	5	4	6	8	7	9	5	0.771604938
16	1	2	3	4	5	6	8	9	7	5	0.771604938
17	1	2	3	4	5	6	9	8	7	4	0.617283951
18	1	2	3	5	6	4	7	8	9	4	0.617283951
19	1	2	3	4	5	8	6	7	9	4	0.617283951
20	1	2	3	4	5	6	9	7	8	3	0.462962963
21	1	2	4	3	5	6	8	7	9	3	0.462962963
22	1	2	4	5	3	6	7	8	9	3	0.462962963
23	1	2	3	4	7	5	6	8	9	3	0.462962963
24	1	2	3	4	6	7	5	8	9	3	0.462962963
25	1	2	3	4	5	8	7	6	9	3	0.462962963
26	2	1	3	4	5	6	8	7	9	3	0.462962963
27	1	3	2	4	5	7	6	9	8	2	0.308641975
28	2	1	3	4	5	7	6	9	8	2	0.308641975
29	1	2	3	4	6	5	8	9	7	2	0.308641975
30	1	2	3	4	5	9	8	7	6	2	0.308641975

Continued on next page

Table E.3 – continued from previous page

No.	F1	F2	F3	F4	F5	F6	F7	F8	F9	Count	Percentage
31	2	1	3	4	6	5	7	8	9	2	0.308641975
32	1	3	2	4	6	5	7	8	9	2	0.308641975
33	1	2	3	4	5	7	9	8	6	2	0.308641975
34	1	3	4	5	7	8	6	2	9	2	0.308641975
35	1	2	4	5	6	7	8	9	3	2	0.308641975
36	1	2	3	4	5	7	8	6	9	2	0.308641975
37	1	2	3	4	7	6	5	9	8	2	0.308641975
38	1	3	2	4	5	6	7	9	8	2	0.308641975
39	1	2	3	4	7	6	5	8	9	2	0.308641975
40	1	2	3	5	4	9	6	7	8	1	0.154320988
41	9	8	7	5	6	4	2	1	3	1	0.154320988
42	1	2	4	3	5	7	6	9	8	1	0.154320988
43	1	3	5	6	7	8	2	9	4	1	0.154320988
44	4	3	2	1	5	6	7	8	9	1	0.154320988
45	2	1	3	4	7	5	6	8	9	1	0.154320988
46	1	2	3	4	5	8	7	9	6	1	0.154320988
47	1	5	6	7	3	4	2	8	9	1	0.154320988
48	1	2	3	5	4	8	6	7	9	1	0.154320988
49	2	1	4	3	5	6	9	8	7	1	0.154320988
50	3	1	2	4	5	7	6	9	8	1	0.154320988
51	1	3	2	8	4	6	5	7	9	1	0.154320988
52	1	2	4	3	8	7	6	9	5	1	0.154320988
53	3	9	7	8	6	1	5	4	2	1	0.154320988
54	8	1	3	4	6	2	7	5	9	1	0.154320988
55	6	1	5	2	7	9	3	8	4	1	0.154320988
56	1	2	4	5	6	7	9	8	3	1	0.154320988
57	9	1	2	3	4	5	8	6	7	1	0.154320988
58	1	2	5	3	6	4	8	7	9	1	0.154320988
59	2	1	4	3	7	5	9	6	8	1	0.154320988

Continued on next page

Table E.3 – continued from previous page

No.	F1	F2	F3	F4	F5	F6	F7	F8	F9	Count	Percentage
60	1	2	3	5	4	7	6	9	8	1	0.154320988
61	2	1	4	3	6	5	9	8	7	1	0.154320988
62	1	3	2	5	6	9	8	4	7	1	0.154320988
63	1	2	3	4	6	8	5	7	9	1	0.154320988
64	1	6	4	7	9	8	2	3	5	1	0.154320988
65	2	1	3	5	6	4	7	8	9	1	0.154320988
66	2	1	4	7	8	6	5	9	3	1	0.154320988
67	2	1	3	5	4	6	8	7	9	1	0.154320988
68	1	3	2	5	4	6	7	8	9	1	0.154320988
69	1	3	2	4	5	6	8	7	9	1	0.154320988
70	2	1	3	4	9	8	7	6	5	1	0.154320988
71	2	1	4	9	3	5	7	8	6	1	0.154320988
72	1	3	2	9	8	7	6	5	4	1	0.154320988
73	1	2	3	8	5	6	7	4	9	1	0.154320988
74	3	1	2	4	6	5	9	8	7	1	0.154320988
75	1	3	2	4	6	7	5	8	9	1	0.154320988
76	9	1	2	3	4	5	6	7	8	1	0.154320988
77	5	3	1	2	7	8	9	4	6	1	0.154320988
78	1	2	3	6	5	4	7	8	9	1	0.154320988
79	2	1	3	5	9	4	6	7	8	1	0.154320988
80	3	1	2	4	5	6	7	9	8	1	0.154320988
81	9	8	7	6	5	3	4	2	1	1	0.154320988
82	1	2	3	4	5	9	6	8	7	1	0.154320988
83	1	2	3	4	5	7	8	9	6	1	0.154320988
84	1	2	5	6	7	8	4	3	9	1	0.154320988
85	1	2	3	4	5	7	9	6	8	1	0.154320988
86	2	1	3	4	5	6	7	9	8	1	0.154320988
87	1	2	3	7	8	4	5	9	6	1	0.154320988
88	2	1	3	9	4	8	7	6	5	1	0.154320988

Continued on next page

Table E.3 – continued from previous page

No.	F1	F2	F3	F4	F5	F6	F7	F8	F9	Count	Percentage
89	1	2	3	5	7	4	6	9	8	1	0.154320988
90	1	9	2	5	4	8	7	6	3	1	0.154320988
91	1	2	3	6	4	7	5	8	9	1	0.154320988
92	2	1	3	4	5	9	6	8	7	1	0.154320988
93	3	9	7	6	8	5	1	4	2	1	0.154320988
94	1	2	3	4	9	8	6	5	7	1	0.154320988
95	1	2	5	4	6	7	9	8	3	1	0.154320988
96	2	1	6	4	3	7	5	8	9	1	0.154320988
97	1	2	4	3	5	7	6	8	9	1	0.154320988
98	2	1	3	4	8	5	6	9	7	1	0.154320988
99	1	2	3	7	6	8	4	9	5	1	0.154320988
100	2	1	4	3	5	8	6	7	9	1	0.154320988
101	1	2	4	5	6	7	8	3	9	1	0.154320988
102	1	2	3	4	5	8	6	9	7	1	0.154320988
103	5	8	2	3	1	7	9	4	6	1	0.154320988
104	2	1	3	5	6	4	7	9	8	1	0.154320988
105	1	2	3	4	9	5	8	7	6	1	0.154320988
106	1	2	3	4	6	5	9	7	8	1	0.154320988
107	5	3	2	8	1	9	6	7	4	1	0.154320988
108	2	3	1	4	5	6	9	8	7	1	0.154320988
109	1	2	3	4	5	9	7	6	8	1	0.154320988
110	1	2	5	9	6	8	7	3	4	1	0.154320988
111	1	2	3	4	7	8	6	9	5	1	0.154320988
112	1	2	3	4	5	8	9	7	6	1	0.154320988
113	2	1	3	4	5	6	8	9	7	1	0.154320988
114	1	2	4	5	6	3	7	9	8	1	0.154320988
115	1	2	3	5	4	6	9	7	8	1	0.154320988
116	1	2	3	5	4	6	8	9	7	1	0.154320988
117	1	2	4	3	8	7	6	5	9	1	0.154320988

Continued on next page

Table E.3 – continued from previous page

No.	F1	F2	F3	F4	F5	F6	F7	F8	F9	Count	Percentage
118	2	1	3	6	5	4	7	8	9	1	0.154320988
119	1	2	3	9	8	5	7	6	4	1	0.154320988
120	1	2	4	5	6	8	9	3	7	1	0.154320988
121	2	1	4	3	6	8	7	5	9	1	0.154320988
122	3	1	2	4	5	6	7	8	9	1	0.154320988
123	1	6	2	8	7	3	9	4	5	1	0.154320988
124	2	1	3	4	5	6	9	8	7	1	0.154320988
125	2	1	3	5	4	6	7	8	9	1	0.154320988
126	9	7	8	6	5	3	4	2	1	1	0.154320988
127	1	3	2	4	5	7	6	8	9	1	0.154320988
128	2	3	4	5	7	9	1	8	6	1	0.154320988
129	1	3	2	4	5	7	8	6	9	1	0.154320988
130	1	2	3	9	7	4	6	8	5	1	0.154320988
131	1	2	3	4	8	7	9	5	6	1	0.154320988
132	1	3	5	6	7	8	9	4	2	1	0.154320988
133	1	3	2	5	6	7	4	8	9	1	0.154320988
134	1	2	4	3	5	7	8	6	9	1	0.154320988
135	1	2	4	3	6	7	8	5	9	1	0.154320988
136	1	2	4	5	3	6	8	7	9	1	0.154320988
137	1	2	4	5	6	9	8	3	7	1	0.154320988
138	1	2	4	5	6	3	9	8	7	1	0.154320988
139	1	2	5	3	6	8	9	4	7	1	0.154320988
140	1	3	2	4	6	8	5	7	9	1	0.154320988
141	1	3	2	5	4	6	7	9	8	1	0.154320988
142	1	2	4	5	6	8	7	9	3	1	0.154320988
143	1	2	4	5	3	7	6	8	9	1	0.154320988
144	1	2	3	6	7	4	9	5	8	1	0.154320988
145	9	5	4	3	2	1	7	6	8	1	0.154320988
146	2	1	3	5	6	7	8	4	9	1	0.154320988

Continued on next page

Table E.3 – continued from previous page

No.	F1	F2	F3	F4	F5	F6	F7	F8	F9	Count	Percentage
147	1	2	3	4	8	5	9	7	6	1	0.154320988
148	1	3	4	5	6	7	8	2	9	1	0.154320988
149	1	2	3	8	4	5	7	9	6	1	0.154320988
150	1	2	3	4	8	7	5	6	9	1	0.154320988
151	1	3	2	4	5	8	6	9	7	1	0.154320988
152	1	2		3			4	5		0	0
153	1	2	3				4	5		0	0
154	1	2	3	6	4	8	5		7	0	0
155	1	2		3						0	0
156	1	2	3	4						0	0
157	1		2	3			4		5	0	0
158	1		2		3		4			0	0
159	1	2	3							0	0
									Total	648	100 %

E.4 Spiral Images Clusters

Table E.4: Image spiral cluster analysis

No.	S1	S2	S3	S4	S5	S6	S7	S8	S9	Count	Percentage
1	1	2	3	4	5	6	7	8	9	443	67.5304878
2	1	3	2	4	5	6	7	8	9	20	3.048780488
3	1	2	3	4	5	7	6	8	9	19	2.896341463
4	1	2	3	4	6	5	7	8	9	17	2.591463415
5	1	2	3	4	5	6	7	9	8	17	2.591463415
6	1	2	3	4	5	6	8	7	9	13	1.981707317

Continued on next page

Table E.4 – continued from previous page

No.	S1	S2	S3	S4	S5	S6	S7	S8	S9	Count	Percentage
7	1	2	3	5	4	6	7	8	9	9	1.37195122
8	1	2	4	3	5	6	7	8	9	9	1.37195122
9	1	2	3	4	5	6	8	9	7	4	0.609756098
10	1	3	2	4	5	7	6	8	9	3	0.457317073
11	1	3	2	4	6	5	7	8	9	3	0.457317073
12	1	2	3	4	6	5	8	7	9	3	0.457317073
13	1	2	3	4	5	6	9	7	8	3	0.457317073
14	1	2	3	4	6	7	8	9	5	3	0.457317073
15	1	2	4	3	5	6	7	9	8	3	0.457317073
16	1	2	3	5	6	7	4	8	9	2	0.304878049
17	1	2	3	5	4	7	6	8	9	2	0.304878049
18	2	1	3	4	5	6	7	9	8	2	0.304878049
19	1	2	3	4	5	8	6	7	9	2	0.304878049
20	1	2	3	5	4	6	7	9	8	2	0.304878049
21	1	3	2	4	5	6	8	7	9	2	0.304878049
22	1	3	2	4	6	5	8	7	9	2	0.304878049
23	1	3	4	5	6	7	8	9	2	1	0.152439024
24	1	2	3	4	7	6	8	9	5	1	0.152439024
25	1	2	3	4	5	8	9	7	6	1	0.152439024
26	1	2	3	4	6	7	8	5	9	1	0.152439024
27	1	2	3	4	7	5	6	8	9	1	0.152439024
28	1	2	4	3	5	7	6	8	9	1	0.152439024
29	1	2	3	4	5	7	8	6	9	1	0.152439024
30	1	5	7	3	9	8	6	4	2	1	0.152439024
31	5	4	7	8	6	9	3	2	1	1	0.152439024
32	1	4	5	2	6	7	3	8	9	1	0.152439024
33	1	4	2	5	3	6	7	8	9	1	0.152439024
34	2	1	3	6	4	9	8	5	7	1	0.152439024
35	8	1	9	2	6	3	7	4	5	1	0.152439024

Continued on next page

Table E.4 – continued from previous page

No.	S1	S2	S3	S4	S5	S6	S7	S8	S9	Count	Percentage
36	1	2	3	4	8	5	6	7	9	1	0.152439024
37	2	9	1	6	4	3	7	8	5	1	0.152439024
38	1	2	3	7	9	5	4	8	6	1	0.152439024
39	2	6	9	4	1	8	5	3	7	1	0.152439024
40	2	3	1	4	6	9	7	8	5	1	0.152439024
41	2	3	1	4	5	6	7	8	9	1	0.152439024
42	1	2	5	4	6	9	3	8	7	1	0.152439024
43	1	2	3	4	6	7	5	8	9	1	0.152439024
44	1	2	3	4	8	9	7	6	5	1	0.152439024
45	1	4	9	2	8	6	7	5	3	1	0.152439024
46	4	3	9	2	5	7	8	1	6	1	0.152439024
47	2	1	3	4	5	6	7	8	9	1	0.152439024
48	1	2	3	5	6	4	7	8	9	1	0.152439024
49	1	3	6	4	2	9	8	7	5	1	0.152439024
50	4	9	6	2	3	8	1	5	7	1	0.152439024
51	6	1	4	2	5	9	7	8	3	1	0.152439024
52	1	6	2	7	4	8	3	9	5	1	0.152439024
53	1	7	2	9	4	6	3	8	5	1	0.152439024
54	2	1	4	5	7	6	3	8	9	1	0.152439024
55	1	3	2	7	5	4	8	9	6	1	0.152439024
56	2	1	3	4	5	6	8	7	9	1	0.152439024
57	1	2	4	3	5	6	8	7	9	1	0.152439024
58	9	5	6	4	7	2	3	8	1	1	0.152439024
59	2	1	5	4	3	9	8	7	6	1	0.152439024
60	1	2	3	5	4	6	8	9	7	1	0.152439024
61	2	1	3	5	6	7	4	8	9	1	0.152439024
62	2	3	6	1	5	4	8	9	7	1	0.152439024
63	1	2	3	5	4	7	8	9	6	1	0.152439024
64	1	2	3	5	4	9	6	7	8	1	0.152439024

Continued on next page

Table E.4 – continued from previous page

No.	S1	S2	S3	S4	S5	S6	S7	S8	S9	Count	Percentage
65	1	4	2	5	3	6	7	9	8	1	0.152439024
66	1	3	6	7	8	9	5	4	2	1	0.152439024
67	1	2	3	4	7	9	8	5	6	1	0.152439024
68	1	2	6	4	5	9	7	8	3	1	0.152439024
69	1	6	4	8	5	9	7	2	3	1	0.152439024
70	2	1	3	4	5	7	6	8	9	1	0.152439024
71	1	2	3	5	4	9	6	8	7	1	0.152439024
72	1	2	3	4	5	7	8	9	6	1	0.152439024
73	1	2	4	3	6	5	7	8	9	1	0.152439024
74	9	5	8	7	6	1	2	3	4	1	0.152439024
75	1	3	2	4	5	6	7	9	8	1	0.152439024
76	2	3	8	5	4	1	7	6	9	1	0.152439024
77	1	6	3	2	5	7	8	4	9	1	0.152439024
78	1	6	7	8	5	4	3	9	2	1	0.152439024
79	1	2	4	3	5	6	9	7	8	1	0.152439024
80	3	1	4	2	6	5	7	8	9	1	0.152439024
81	1	3	2	5	4	6	7	8	9	1	0.152439024
82	1	3	2	5	4	6	8	7	9	1	0.152439024
83	4	1	2	3	5	8	6	7	9	1	0.152439024
84	1	2	4	3	5	7	6	9	8	1	0.152439024
85	1	3	5	2	4	6	7	8	9	1	0.152439024
86	4	1	2	3	5	7	6	8	9	1	0.152439024
87	1	2	3	4	5	7	6	9	8	1	0.152439024
88	1	2	3	4	5	6	9	8	7	1	0.152439024
89	1	6	2	7	3	8	4	9	5	1	0.152439024
90	1	2	3	4	6	7	5	9	8	1	0.152439024
91	2	1	8	7	6	4	5	3	9	1	0.152439024
92	1	8	2	3	4	5	6	7	9	1	0.152439024
93	1	2	3	5	7	4	6	8	9	1	0.152439024

Continued on next page

Table E.4 – continued from previous page

No.	S1	S2	S3	S4	S5	S6	S7	S8	S9	Count	Percentage
94	3	6	1	8	5	9	7	2	4	1	0.152439024
95	1	2	5	9	4	7	8	3	6	1	0.152439024
96	1	2	3	4	5	6				0	0
97	1	2	3	4	5	6	7	9		0	0
98										0	0
99	1	2	3	6	5	4				0	0
									Total	656	100 %

E.5 Mosaic Images Clusters

Table E.5: Image mosaic cluster analysis

No.	mosaic1	M2	M3	M4	M5	M6	M7	M8	M9	Count	Percentage
1	1	2	3	4	5	6	7	8	9	358	54.65648855
2	1	2	3	4	5	6	7	9	8	88	13.4351145
3	1	2	3	4	5	6	8	7	9	35	5.34351145
4	1	2	3	4	5	6	9	8	7	20	3.053435115
5	1	2	3	4	5	6	8	9	7	17	2.595419847
6	1	2	3	4	5	6	9	7	8	16	2.442748092
7	1	2	3	4	5	7	8	6	9	7	1.06870229
8	1	2	3	4	5	7	6	8	9	6	0.916030534
9	1	2	3	4	5	7	8	9	6	5	0.763358779
10	1	2	3	4	5	7	9	6	8	5	0.763358779
11	1	2	3	5	4	6	7	8	9	4	0.610687023
12	1	2	3	4	6	5	8	7	9	3	0.458015267
13	1	2	4	3	5	6	7	8	9	3	0.458015267

Continued on next page

Table E.5 – continued from previous page

No.	M1	M2	M3	M4	M5	M6	M7	M8	M9	Count	Percentage
14	1	2	3	4	6	5	7	8	9	3	0.458015267
15	1	2	3	4	7	5	6	8	9	2	0.305343511
16	1	2	3	4	7	9	6	8	5	2	0.305343511
17	1	2	4	5	7	8	9	3	6	2	0.305343511
18	1	2	4	3	5	6	7	9	8	2	0.305343511
19	1	2	3	4	6	5	7	9	8	2	0.305343511
20	1	2	3	5	6	7	8	4	9	2	0.305343511
21	1	2	3	4	5	7	9	8	6	2	0.305343511
22	1	2	3	4	5	8	6	9	7	2	0.305343511
23	1	2	3	4	5	9	6	7	8	2	0.305343511
24	1	2	4	6	7	8	9	3	5	1	0.152671756
25	3	9	5	6	7	1	8	2	4	1	0.152671756
26	1	2	3	4	9	8	6	5	7	1	0.152671756
27	1	2	4	8	3	5	7	9	6	1	0.152671756
28	1	2	3	4	5	7	6	9	8	1	0.152671756
29	1	2	3	5	6	7	4	8	9	1	0.152671756
30	1	3	4	5	6	7	2	9	8	1	0.152671756
31	9	5	6	7	1	2	3	4	8	1	0.152671756
32	1	3	2	9	8	4	7	5	6	1	0.152671756
33	9	8	4	6	2	7	3	1	5	1	0.152671756
34	1	3	4	5	6	2	8	7	9	1	0.152671756
35	1	2	3	4	8	5	7	9	6	1	0.152671756
36	3	8	4	9	6	2	5	7	1	1	0.152671756
37	1	2	3	4	9	5	7	6	8	1	0.152671756
38	1	2	3	6	7	5	8	9	4	1	0.152671756
39	1	2	4	3	5	9	8	6	7	1	0.152671756
40	1	2	4	9	7	6	8	3	5	1	0.152671756
41	1	2	3	8	6	5	4	7	9	1	0.152671756
42	1	2	3	4	9	6	7	8	5	1	0.152671756

Continued on next page

Table E.5 – continued from previous page

No.	M1	M2	M3	M4	M5	M6	M7	M8	M9	Count	Percentage
43	1	2	3	4	8	7	6	9	5	1	0.152671756
44	1	2	3	4	8	9	5	6	7	1	0.152671756
45	9	1	2	6	8	4	3	5	7	1	0.152671756
46	1	2	4	5	3	6	7	9	8	1	0.152671756
47	1	2	4	3	5	6	9	8	7	1	0.152671756
48	1	2	3	4	9	8	5	6	7	1	0.152671756
49	1	2	4	3	5	6	8	7	9	1	0.152671756
50	1	2	3	4	7	8	9	6	5	1	0.152671756
51	7	3	1	5	9	2	4	8	6	1	0.152671756
52	1	2	5	3	4	7	6	9	8	1	0.152671756
53	1	2	3	4	7	8	9	5	6	1	0.152671756
54	1	2	3	4	6	5	8	9	7	1	0.152671756
55	1	2	3	4	6	9	8	7	5	1	0.152671756
56	1	2	3	4	7	5	8	6	9	1	0.152671756
57	1	2	3	4	8	6	9	5	7	1	0.152671756
58	1	2	3	4	9	8	5	7	6	1	0.152671756
59	1	3	4	6	7	5	2	9	8	1	0.152671756
60	1	2	3	5	4	7	9	6	8	1	0.152671756
61	1	2	4	5	3	6	8	7	9	1	0.152671756
62	1	2	3	6	7	8	9	4	5	1	0.152671756
63	1	6	4	2	3	5	8	9	7	1	0.152671756
64	1	2	3	4	6	5	9	8	7	1	0.152671756
65	1	2	3	4	9	5	6	7	8	1	0.152671756
66	1	2	4	3	5	8	7	6	9	1	0.152671756
67	1	2	5	9	6	7	8	3	4	1	0.152671756
68	1	2	3	4	5	8	7	6	9	1	0.152671756
69	1	2	9	4	3	8	6	5	7	1	0.152671756
70	1	7	3	8	5	6	9	2	4	1	0.152671756
71	1	2	3	4	5	9	8	6	7	1	0.152671756

Continued on next page

Table E.5 – continued from previous page

No.	M1	M2	M3	M4	M5	M6	M7	M8	M9	Count	Percentage
72	1	2	3	5	6	7	8	9	4	1	0.152671756
73	1	2	3	4	6	7	8	9	5	1	0.152671756
74	1	9	3	5	2	6	7	8	4	1	0.152671756
75	1	2	3	5	6	7	9	4	8	1	0.152671756
76	1	2	3	9	8	7	6	5	4	1	0.152671756
77	1	2	3	5	7	8	9	6	4	1	0.152671756
78	1	2	5	9	8	6	7	4	3	1	0.152671756
79	1	2	3	4	5	9	8	7	6	1	0.152671756
80	1	2	3	4	7	8	6	9	5	1	0.152671756
81	1	2	3	4	6	7	8	5	9	1	0.152671756
82	1	2	3	4	6	7	5	8	9	1	0.152671756
83	1	5	6	3	2	9	4	7	8	1	0.152671756
84	1	2	3	4	7	9	5	8	6	1	0.152671756
85	1	2	5	9	3	4	6	7	8	1	0.152671756
86	1	2	5	8	4	7	3	6	9	1	0.152671756
87	1	2	3	9	7	4	5	8	6	1	0.152671756
88	1	2	4	5	6	7	8	9	3	1	0.152671756
89	1	2	3	5	4	6	7	9	8	1	0.152671756
90	1	2	3	5	6	4	7	8	9	1	0.152671756
91	2	1	4	3	6	8		7	5	0	0
92	1	2	3	5	4					0	0
93	1	2	3	4	5	6		8	7	0	0
94										0	0
95	1	2	3	4	5					0	0
									Total	655	100 %

E.6 Squares Images Clusters

Table E.6: Image squares cluster analysis

No.	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Count	percentage
1	1	6	2	7	3	8	4	9	5	273	41.67938931
2	1	2	3	4	5	6	7	8	9	116	17.70992366
3	1	9	2	8	3	7	4	6	5	31	4.732824427
4	1	2	3	4	5	6	7	9	8	25	3.816793893
5	2	6	1	7	3	8	4	9	5	16	2.442748092
6	1	3	2	5	4	7	6	9	8	16	2.442748092
7	1	6	2	7	3	8	5	9	4	10	1.526717557
8	1	6	2	8	3	7	4	9	5	9	1.374045802
9	1	6	2	7	3	9	4	8	5	8	1.221374046
10	1	7	2	6	3	8	4	9	5	7	1.06870229
11	1	2	3	4	5	7	6	9	8	7	1.06870229
12	1	3	2	4	5	6	7	8	9	6	0.916030534
13	1	2	3	5	4	7	6	9	8	5	0.763358779
14	1	2	3	4	5	6	8	7	9	4	0.610687023
15	3	6	1	7	2	8	4	9	5	4	0.610687023
16	1	6	2	7	4	8	3	9	5	4	0.610687023
17	1	6	3	7	2	8	4	9	5	3	0.458015267
18	1	3	2	4	5	6	7	9	8	3	0.458015267
19	1	4	2	5	3	6	7	9	8	3	0.458015267
20	1	2	3	7	4	8	5	9	6	3	0.458015267
21	1	8	2	9	3	7	4	6	5	3	0.458015267
22	1	7	2	8	3	6	4	9	5	3	0.458015267
23	1	7	2	8	3	9	4	6	5	3	0.458015267
24	1	6	2	8	3	7	5	9	4	2	0.305343511
25	1	3	2	4	5	7	6	9	8	2	0.305343511
26	2	3	1	4	5	6	7	8	9	2	0.305343511

Continued on next page

Table E.6 – continued from previous page

No.	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Count	percentage
27	1	3	2	4	7	8	6	9	5	1	0.152671756
28	1	3	2	4	5	8	6	9	7	1	0.152671756
29	1	3	2	4	5	7	6	8	9	1	0.152671756
30	1	7	2	8	4	9	5	3	6	1	0.152671756
31	1	5	2	6	3	7	4	8	9	1	0.152671756
32	1	5	2	6	4	7	3	9	8	1	0.152671756
33	1	7	2	6	3	9	4	8	5	1	0.152671756
34	1	5	2	9	7	8	6	3	4	1	0.152671756
35	2	6	1	8	3	9	4	7	5	1	0.152671756
36	1	3	2	4	6	5	7	9	8	1	0.152671756
37	1	6	2	8	4	7	3	9	5	1	0.152671756
38	2	6	1	7	3	8	5	9	4	1	0.152671756
39	2	7	1	6	5	8	3	9	4	1	0.152671756
40	1	3	2	4	5	9	6	7	8	1	0.152671756
41	1	6	3	8	2	7	4	9	5	1	0.152671756
42	2	3	1	5	4	6	7	8	9	1	0.152671756
43	1	3	2	6	4	8	5	9	7	1	0.152671756
44	2	3	1	5	4	7	6	9	8	1	0.152671756
45	1	8	2	6	3	7	4	9	5	1	0.152671756
46	2	6	1	7	4	9	5	8	3	1	0.152671756
47	1	9	2	7	3	6	4	8	5	1	0.152671756
48	9	6	1	3	2	4	8	5	7	1	0.152671756
49	1	7	2	8	3	9	4	5	6	1	0.152671756
50	1	3	2	8	4	5	7	9	6	1	0.152671756
51	9	6	5	3	7	2	8	1	4	1	0.152671756
52	1	2	3	4	5	8	9	6	7	1	0.152671756
53	2	8	1	7	3	9	4	6	5	1	0.152671756
54	1	3	2	5	4	7	8	9	6	1	0.152671756
55	8	9	1	4	2	5	3	7	6	1	0.152671756

Continued on next page

Table E.6 – continued from previous page

No.	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Count	percentage
56	1	2	3	4	5	8	6	9	7	1	0.152671756
57	1	3	4	5	6	8	7	2	9	1	0.152671756
58	1	2	3	4	6	7	5	9	8	1	0.152671756
59	1	4	2	3	5	6	7	8	9	1	0.152671756
60	1	2	3	4	5	7	8	9	6	1	0.152671756
61	1	3	2	5	4	6	8	7	9	1	0.152671756
62	1	2	3	4	5	7	6	8	9	1	0.152671756
63	1	2	5	4	3	6	7	8	9	1	0.152671756
64	1	2	3	4	5	7	8	6	9	1	0.152671756
65	1	4	3	2	5	6	7	8	9	1	0.152671756
66	4	7	2	6	9	5	1	3	8	1	0.152671756
67	1	7	2	9	3	8	4	6	5	1	0.152671756
68	2	6	1	8	4	7	3	9	5	1	0.152671756
69	1	5	2	8	3	9	4	7	6	1	0.152671756
70	1	7	4	3	9	2	6	8	5	1	0.152671756
71	2	3	4	5	6	7	8	9	1	1	0.152671756
72	3	6	1	7	4	8	2	9	5	1	0.152671756
73	1	6	2	9	3	8	5	7	4	1	0.152671756
74	2	3	1	7	4	5	6	9	8	1	0.152671756
75	4	6	1	8	3	7	5	9	2	1	0.152671756
76	1	3	2	5	4	8	6	9	7	1	0.152671756
77	2	9	1	6	3	7	4	8	5	1	0.152671756
78	1	9	3	8	2	6	5	7	4	1	0.152671756
79	1	9	2	6	3	8	5	7	4	1	0.152671756
80	1	6	2	9	4	7	3	8	5	1	0.152671756
81	6	9	1	7	3	5	4	8	2	1	0.152671756
82	1	2	3	5	4	8	6	9	7	1	0.152671756
83	3	6	1	7	2	9	4	8	5	1	0.152671756
84	2	6	5	8	1	7	3	9	4	1	0.152671756

Continued on next page

Table E.6 – continued from previous page

No.	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Count	percentage
85	1	6	5	7	2	8	3	9	4	1	0.152671756
86	1	9	2	8	3	6	4	7	5	1	0.152671756
87	2	7	1	6	3	8	4	9	5	1	0.152671756
88	1	4	2	5	3	7	6	9	8	1	0.152671756
89	2	4	1	5	6	7	8	3	9	1	0.152671756
90	1	6	2	7	5	8	3	9	4	1	0.152671756
91	1	2	4	3	5	9	6	8	7	1	0.152671756
92	2	8	1	6	3	7	4	9	5	1	0.152671756
93	1	3	2	5	4	7	6	8	9	1	0.152671756
94	1	9	3	8	2	7	4	6	5	1	0.152671756
95	1	6	4	7	2	8	5	9	3	1	0.152671756
96	1	4	2	7	3	8	6	9	5	1	0.152671756
97	1	7	2	6	3	8	5	9	4	1	0.152671756
98	3	8	2	7	1	6	5	9	4	1	0.152671756
99	1	3	2	5	4	9	6	8	7	1	0.152671756
100	1	6	2	8	3	9	4	7	5	1	0.152671756
101	1	2	3	4	5	6	9	8	7	1	0.152671756
102	1	5	2	6	3	8	4	9	7	1	0.152671756
103	1	2	3	4	5	8	7	6	9	1	0.152671756
104	1	7	2	6	4	8	3	9	5	1	0.152671756
105	1	2	3	5	6	7	8	4	9	1	0.152671756
106	2	9	3	6	1	8	4	7	5	1	0.152671756
107	1	3	2	7	4	8	5	9	6	1	0.152671756
108	1	2	3	4	7	9	8	6	5	1	0.152671756
109	2	3	1	4	5	8	6	9	7	1	0.152671756
110	1	6	7	9	3	5	2	4	8	1	0.152671756
111	1	6	2	9	3	7	4	8	5	1	0.152671756
112	1	2	4	3	5	6	7	8	9	1	0.152671756
113	1	4	3	2	5	6	7	9	8	1	0.152671756

Continued on next page

Table E.6 – continued from previous page

No.	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9	Count	percentage
114										0	0
115	1	2	3	4						0	0
116	1	2	3	4	5	6	7		8	0	0
									Total	655	100 %

Appendix F

Graphs of goodness test fit of displayed images for all grammars

F.1 Bottom-up tree edit distance values for all grammars

Table F.1: Bottom-up tree edit distance values for grammar 0 for pass-images 0 to 16

img.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.47368	0.62963	0.71429	0.76744	0.80392	0.83051	0.85075	0.86667	0.87952	0.89011	0.89899	0.90654	0.91304	0.91870	0.92366	0.92806
1	0.47368	0.00000	0.33333	0.48571	0.58140	0.64706	0.69492	0.73134	0.76000	0.78313	0.80220	0.81818	0.83178	0.84348	0.85366	0.86260	0.87050
2	0.62963	0.33333	0.00000	0.25714	0.39535	0.49020	0.55932	0.61194	0.65333	0.68675	0.71429	0.73737	0.75701	0.77391	0.78862	0.80153	0.81295
3	0.71429	0.48571	0.25714	0.00000	0.20930	0.33333	0.42373	0.49254	0.54667	0.59036	0.62637	0.65657	0.68224	0.70435	0.72358	0.74046	0.75540
4	0.76744	0.58140	0.39535	0.20930	0.00000	0.17647	0.28814	0.37313	0.44000	0.49398	0.53846	0.57576	0.60748	0.63478	0.65854	0.67939	0.69784
5	0.80392	0.64706	0.49020	0.33333	0.17647	0.00000	0.15254	0.25373	0.33333	0.39759	0.45055	0.49495	0.53271	0.56522	0.59350	0.61832	0.64029
6	0.83051	0.69492	0.55932	0.42373	0.28814	0.15254	0.00000	0.13433	0.22667	0.30120	0.36264	0.41414	0.45794	0.49565	0.52846	0.55725	0.58273
7	0.85075	0.73134	0.61194	0.49254	0.37313	0.25373	0.13433	0.00000	0.12000	0.20482	0.27473	0.33333	0.38318	0.42609	0.46341	0.49618	0.52518
8	0.86667	0.76000	0.65333	0.54667	0.44000	0.33333	0.22667	0.12000	0.00000	0.10843	0.18681	0.25253	0.30841	0.35652	0.39837	0.43511	0.46763
9	0.78313	0.68675	0.59036	0.49398	0.39759	0.30120	0.20482	0.10843	0.00000	0.09890	0.17172	0.23364	0.28696	0.33333	0.37405	0.41007	0.44218
10	0.71429	0.62637	0.53846	0.45055	0.36264	0.27473	0.18681	0.09890	0.00000	0.09091	0.15888	0.21739	0.26829	0.31298	0.35252	0.38776	0.41935
11	0.65657	0.57576	0.49495	0.41414	0.33333	0.25253	0.17172	0.09091	0.00000	0.08411	0.14783	0.20325	0.25191	0.29496	0.33333	0.36774	0.39877
12	0.60748	0.53271	0.45794	0.38318	0.30841	0.23364	0.15888	0.08411	0.00000	0.07826	0.13821	0.19084	0.23741	0.27891	0.31613	0.34969	0.38012
13	0.56522	0.49565	0.42609	0.35652	0.28696	0.21739	0.14783	0.07826	0.00000	0.07317	0.12977	0.17986	0.22449	0.26452	0.30061	0.33333	0.36313
14	0.52846	0.46341	0.39837	0.33333	0.26829	0.20325	0.13821	0.07317	0.00000	0.06870	0.12230	0.17007	0.21290	0.25153	0.28655	0.31844	0.34759
15	0.49618	0.43511	0.37405	0.31298	0.25191	0.19084	0.12977	0.06870	0.00000	0.06475	0.11565	0.16129	0.20245	0.23977	0.27374	0.30481	0.33333
16	0.46763	0.41007	0.35252	0.29496	0.23741	0.17986	0.12230	0.06475	0.00000	0.06122	0.10968	0.15337	0.19298	0.22905	0.26203	0.29231	0.32020

Table F.2: Bottom-up tree edit distance values for grammar 1 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.23077	0.10256	0.26923	0.14103	0.35897	0.23077	0.26923	0.14103	0.35897	0.23077	0.39744	0.26923	0.48718	0.35897	0.26923	0.14103
1	0.23077	0.00000	0.15278	0.12698	0.15942	0.17460	0.06349	0.09524	0.15942	0.17460	0.09524	0.25397	0.14286	0.33333	0.22222	0.12698	0.15942
2	0.10256	0.15278	0.00000	0.25000	0.11111	0.29167	0.15278	0.22222	0.08333	0.29167	0.15278	0.36111	0.22222	0.43056	0.29167	0.25000	0.11111
3	0.26923	0.12698	0.25000	0.00000	0.15942	0.13333	0.12698	0.10000	0.23188	0.21667	0.20635	0.18333	0.10000	0.30000	0.21667	0.06667	0.20290
4	0.14103	0.15942	0.11111	0.15942	0.00000	0.26087	0.11594	0.20290	0.08696	0.30435	0.18841	0.30435	0.15942	0.40580	0.26087	0.18841	0.05797
5	0.35897	0.17460	0.29167	0.13333	0.26087	0.00000	0.17460	0.18333	0.30435	0.11111	0.25397	0.11111	0.18333	0.20370	0.11111	0.18333	0.30435
6	0.23077	0.06349	0.15278	0.12698	0.11594	0.17460	0.00000	0.14286	0.15942	0.22222	0.09524	0.25397	0.09524	0.33333	0.17460	0.15873	0.15942
7	0.26923	0.09524	0.22222	0.10000	0.20290	0.18333	0.14286	0.00000	0.15942	0.13333	0.12698	0.18333	0.06667	0.30000	0.18333	0.10000	0.20290
8	0.14103	0.15942	0.08333	0.23188	0.08696	0.30435	0.15942	0.15942	0.00000	0.26087	0.11594	0.30435	0.15942	0.40580	0.26087	0.23188	0.08696
9	0.17460	0.29167	0.21667	0.30435	0.11111	0.22222	0.13333	0.26087	0.00000	0.17460	0.14815	0.18333	0.20370	0.07407	0.21667	0.30435	0.11111
10	0.15278	0.20635	0.18841	0.25397	0.09524	0.12698	0.11594	0.17460	0.00000	0.28571	0.12698	0.33333	0.17460	0.20635	0.18841	0.25397	0.09524
11	0.18333	0.30435	0.11111	0.25397	0.18333	0.30435	0.14815	0.28571	0.00000	0.18333	0.15686	0.14815	0.23333	0.34783	0.16667	0.30159	0.11765
12	0.15942	0.18333	0.09524	0.06667	0.15942	0.18333	0.12698	0.18333	0.00000	0.30000	0.13333	0.13333	0.20290	0.21667	0.14286	0.26667	0.10000
13	0.20370	0.33333	0.30000	0.40580	0.20370	0.33333	0.15686	0.30000	0.00000	0.20370	0.35000	0.44928	0.25926	0.38095	0.25490	0.35000	0.13333
14	0.17460	0.18333	0.26087	0.07407	0.17460	0.14815	0.13333	0.20370	0.00000	0.25000	0.30435	0.14815	0.22222	0.24074	0.21667	0.29630	0.11111
15	0.10000	0.23188	0.21667	0.20635	0.23333	0.13333	0.35000	0.25000	0.00000	0.15942	0.13333	0.12698	0.18333	0.06667	0.30000	0.18333	0.18333
16	0.08696	0.30435	0.18841	0.34783	0.20290	0.44928	0.30435	0.15942	0.00000	0.26087	0.11594	0.30435	0.15942	0.40580	0.26087	0.30435	0.15942

Table F.3: Bottom-up tree edit distance values for grammar 2 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.46667	0.61905	0.70370	0.75758	0.79487	0.84314	0.89333	0.92793	0.94969	0.96347	0.97251	0.97900	0.98422	0.98846	0.99169	0.99403
1	0.46667	0.00000	0.33333	0.48148	0.57576	0.64103	0.72549	0.81333	0.87387	0.91195	0.93607	0.95189	0.96325	0.97239	0.97980	0.98546	0.98956
2	0.61905	0.33333	0.00000	0.25926	0.39394	0.48718	0.60784	0.73333	0.81982	0.87421	0.90868	0.93127	0.94751	0.96055	0.97114	0.97923	0.98509
3	0.70370	0.48148	0.25926	0.00000	0.21212	0.33333	0.49020	0.65333	0.76577	0.83648	0.88128	0.91065	0.93176	0.94872	0.96248	0.97300	0.98061
4	0.75758	0.57576	0.39394	0.21212	0.00000	0.17949	0.37255	0.57333	0.71171	0.79874	0.85388	0.89003	0.91601	0.93688	0.95382	0.96677	0.97614
5	0.79487	0.64103	0.48718	0.33333	0.17949	0.00000	0.25490	0.49333	0.65766	0.76101	0.82648	0.86942	0.90026	0.92505	0.94517	0.96054	0.97166
6	0.84314	0.72549	0.60784	0.49020	0.37255	0.25490	0.00000	0.33333	0.54955	0.68553	0.77169	0.82818	0.86877	0.90138	0.92785	0.94808	0.96271
7	0.89333	0.81333	0.73333	0.65333	0.57333	0.49333	0.33333	0.00000	0.33333	0.53459	0.66210	0.74570	0.80577	0.85404	0.89322	0.92316	0.94482
8	0.92793	0.87387	0.81982	0.76577	0.71171	0.65766	0.54955	0.33333	0.00000	0.30818	0.49772	0.62199	0.71129	0.78304	0.84127	0.88577	0.91797
9	0.91195	0.87421	0.83648	0.79874	0.76101	0.68553	0.53459	0.30818	0.00000	0.27854	0.45704	0.58530	0.68836	0.77201	0.83593	0.88218	0.91464
10	0.90868	0.88128	0.85388	0.82648	0.77169	0.66210	0.49772	0.27854	0.00000	0.25086	0.42782	0.57002	0.68543	0.77362	0.83743	0.88223	0.91359
11	0.91065	0.89003	0.86942	0.82818	0.74570	0.62199	0.45704	0.25086	0.00000	0.23885	0.42801	0.58153	0.69886	0.78374	0.84333	0.88506	0.91498
12	0.91601	0.90026	0.86877	0.80577	0.71129	0.58530	0.42782	0.23885	0.00000	0.25049	0.45166	0.60540	0.71663	0.79471	0.84939	0.88860	0.91759
13	0.92505	0.90138	0.85404	0.78304	0.68836	0.57002	0.42801	0.25049	0.00000	0.26984	0.47456	0.62267	0.72663	0.79945	0.85166	0.89026	0.91926
14	0.92785	0.89322	0.84127	0.77201	0.68543	0.58153	0.45166	0.26984	0.00000	0.28141	0.48397	0.62615	0.72572	0.79713	0.84992	0.88958	0.91926
15	0.92316	0.88577	0.83593	0.77362	0.69886	0.60540	0.47456	0.28141	0.00000	0.28262	0.48028	0.61871	0.71797	0.79137	0.84650	0.88776	0.91822
16	0.91797	0.88218	0.83743	0.78374	0.71663	0.62267	0.48397	0.28262	0.00000	0.27607	0.46889	0.60715	0.70939	0.78618	0.84366	0.88608	0.91696

Table F.4: Bottom-up tree edit distance values for grammar 3 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.17247	0.01493	0.15423	0.23217	0.16418	0.01990	0.18408	0.01990	0.18574	0.30182	0.19900	0.28358	0.34163	0.29353	0.20398	0.31343
1	0.17247	0.00000	0.17608	0.17354	0.12887	0.17869	0.16918	0.02062	0.16918	0.26117	0.17698	0.26632	0.28694	0.25773	0.29210	0.26632	0.19416
2	0.01493	0.17608	0.00000	0.16113	0.23588	0.15449	0.01661	0.18106	0.01661	0.19601	0.30565	0.18605	0.29070	0.34551	0.28405	0.20100	0.31063
3	0.15423	0.17354	0.16113	0.00000	0.12593	0.01667	0.15410	0.17708	0.15410	0.14444	0.20926	0.15370	0.16296	0.24815	0.17778	0.15370	0.22222
4	0.23217	0.12887	0.23588	0.12593	0.00000	0.12987	0.22948	0.13021	0.22948	0.20833	0.14205	0.21402	0.22538	0.16098	0.23106	0.21402	0.15152
5	0.16418	0.17869	0.15449	0.01667	0.12987	0.00000	0.15410	0.17535	0.15410	0.15399	0.21336	0.14471	0.17625	0.25232	0.16327	0.15213	0.21892
6	0.01990	0.16918	0.01661	0.15410	0.22948	0.15410	0.00000	0.17420	0.00000	0.18928	0.29983	0.18928	0.28476	0.34003	0.28476	0.18760	0.30486
7	0.18408	0.02062	0.18106	0.17708	0.13021	0.17535	0.17420	0.00000	0.17420	0.26563	0.18229	0.26389	0.28646	0.26042	0.28472	0.26389	0.17882
8	0.01990	0.16918	0.01661	0.15410	0.22948	0.15410	0.00000	0.17420	0.00000	0.18928	0.29983	0.18928	0.28476	0.34003	0.28476	0.18760	0.30486
9	0.26117	0.19601	0.14444	0.20833	0.15399	0.18928	0.26563	0.18928	0.00000	0.14859	0.01807	0.12651	0.19679	0.13855	0.02410	0.16265	0.02410
10	0.30565	0.20926	0.14205	0.21336	0.29983	0.18229	0.29983	0.14859	0.00000	0.15292	0.14700	0.10559	0.15321	0.14431	0.02484	0.14431	0.27178
11	0.15370	0.21402	0.14471	0.18928	0.26389	0.18928	0.01807	0.15292	0.00000	0.13481	0.20121	0.12676	0.02012	0.15895	0.02012	0.15679	0.24281
12	0.22538	0.17625	0.28476	0.28646	0.28476	0.12651	0.14700	0.13481	0.00000	0.10965	0.01974	0.12602	0.15094	0.12602	0.25610	0.26619	0.25610
13	0.25232	0.34003	0.26042	0.34003	0.19679	0.10559	0.20121	0.10965	0.00000	0.11429	0.19309	0.10692	0.19309	0.31359	0.23381	0.31359	0.21135
14	0.28476	0.28472	0.28476	0.13855	0.15321	0.12676	0.01974	0.11429	0.00000	0.12602	0.14885	0.12602	0.25610	0.26439	0.25610	0.13503	0.20319
15	0.26389	0.18760	0.02410	0.14431	0.02012	0.12602	0.19309	0.12602	0.00000	0.15041	0.00000	0.16202	0.24281	0.16202	0.14090	0.19124	0.14090
16	0.30486	0.16265	0.02484	0.15895	0.15094	0.10692	0.14885	0.15041	0.00000	0.15041	0.27700	0.15647	0.27700	0.18200	0.13147	0.18200	0.27700

Table F.5: Bottom-up tree edit distance values for grammar 4 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.80952	0.84000	0.92453	0.87879	0.92982	0.93846	0.96040	0.93846	0.94521	0.93846	0.94203	0.96190	0.94203	0.96460	0.96694	0.97576
1	0.80952	0.00000	0.52000	0.62264	0.87879	0.64912	0.81538	0.80198	0.75385	0.83562	0.81538	0.88406	0.80952	0.76812	0.82301	0.90083	0.87879
2	0.84000	0.52000	0.00000	0.77358	0.75758	0.64912	0.63077	0.88119	0.75385	0.78082	0.63077	0.88406	0.80952	0.71014	0.82301	0.80165	0.92727
3	0.92453	0.62264	0.77358	0.00000	0.92453	0.22807	0.69231	0.48515	0.50769	0.83562	0.69231	0.88406	0.50476	0.59420	0.57522	0.83471	0.68485
4	0.87879	0.87879	0.75758	0.92453	0.00000	0.85965	0.81538	0.96040	0.50769	0.61644	0.87692	0.59420	0.92381	0.53623	0.89381	0.90083	0.97576
5	0.92982	0.64912	0.64912	0.22807	0.85965	0.00000	0.56923	0.56436	0.44615	0.78082	0.56923	0.82609	0.50476	0.47826	0.50442	0.76860	0.73333
6	0.93846	0.81538	0.63077	0.69231	0.81538	0.56923	0.00000	0.72277	0.69231	0.72603	0.50769	0.82609	0.65714	0.65217	0.57522	0.47107	0.83030
7	0.96040	0.80198	0.88119	0.48515	0.96040	0.56436	0.72277	0.00000	0.68317	0.88119	0.80198	0.92079	0.12381	0.72277	0.39823	0.76860	0.39394
8	0.93846	0.75385	0.75385	0.50769	0.50769	0.44615	0.69231	0.68317	0.00000	0.50685	0.75385	0.53623	0.65714	0.18841	0.64602	0.83471	0.80606
9	0.83562	0.78082	0.83562	0.61644	0.78082	0.72603	0.88119	0.50685	0.00000	0.78082	0.50685	0.84762	0.50685	0.82301	0.73554	0.92727	0.68142
10	0.63077	0.69231	0.87692	0.56923	0.50769	0.80198	0.75385	0.78082	0.00000	0.82609	0.73333	0.65217	0.75221	0.73554	0.87879	0.78761	0.80165
11	0.88406	0.59420	0.82609	0.82609	0.92079	0.53623	0.50685	0.82609	0.00000	0.88571	0.47826	0.85841	0.83471	0.95152	0.71681	0.66942	0.85841
12	0.92381	0.50476	0.65714	0.12381	0.65714	0.84762	0.73333	0.88571	0.00000	0.65714	0.32743	0.70248	0.44242	0.29204	0.50413	0.36283	0.52137
13	0.47826	0.65217	0.72277	0.18841	0.50685	0.65217	0.47826	0.65714	0.00000	0.64602	0.80165	0.83030	0.50442	0.57025	0.64602	0.52137	0.72093
14	0.57522	0.39823	0.64602	0.82301	0.75221	0.85841	0.32743	0.64602	0.00000	0.53719	0.53939	0.43363	0.53719	0.43363	0.58974	0.78295	0.56589
15	0.76860	0.83471	0.73554	0.73554	0.83471	0.70248	0.80165	0.53719	0.00000	0.78182	0.76860	0.66942	0.66942	0.76860	0.68992	0.37984	0.72093
16	0.80606	0.92727	0.87879	0.95152	0.44242	0.83030	0.53939	0.78182	0.00000	0.53939	0.66061	0.61212	0.68485	0.87879	0.78182	0.92727	0.90303

Table F.6: Bottom-up tree edit distance values for grammar 5 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.41509	0.57864	0.67071	0.72976	0.77086	0.80110	0.82429	0.84264	0.85752	0.86983	0.88018	0.88900	0.89662	0.90325	0.90909	0.91426
1	0.41509	0.00000	0.29903	0.45220	0.55044	0.61880	0.66911	0.70769	0.73822	0.76297	0.78344	0.80066	0.81535	0.82801	0.83905	0.84876	0.85737
2	0.57864	0.29903	0.00000	0.23369	0.37111	0.46674	0.53712	0.59109	0.63379	0.66842	0.69706	0.72115	0.74169	0.75941	0.77486	0.78844	0.80047
3	0.67071	0.45220	0.23369	0.00000	0.19178	0.31468	0.40513	0.47449	0.52937	0.57387	0.61068	0.64163	0.66803	0.69081	0.71066	0.72811	0.74358
4	0.72976	0.55044	0.37111	0.19178	0.00000	0.16262	0.27314	0.35789	0.42495	0.47932	0.52430	0.56212	0.59437	0.62220	0.64646	0.66778	0.68669
5	0.77086	0.61880	0.46674	0.31468	0.16262	0.00000	0.14115	0.24130	0.32052	0.38477	0.43791	0.48261	0.52072	0.55360	0.58226	0.60746	0.62979
6	0.80110	0.66911	0.53712	0.40513	0.27314	0.14115	0.00000	0.12470	0.21610	0.29022	0.35153	0.40309	0.44706	0.48499	0.51806	0.54713	0.57290
7	0.82429	0.70769	0.59109	0.47449	0.35789	0.24130	0.12470	0.00000	0.11168	0.19567	0.26515	0.32358	0.37340	0.41639	0.45386	0.48680	0.51600
8	0.84264	0.73822	0.63379	0.52937	0.42495	0.32052	0.21610	0.11168	0.00000	0.10112	0.17876	0.24406	0.29974	0.34778	0.38966	0.42648	0.45911
9	0.76297	0.66842	0.57387	0.47932	0.38477	0.29022	0.19567	0.10112	0.00000	0.09238	0.16455	0.22609	0.27918	0.32546	0.36615	0.40221	0.43439
10	0.69706	0.61068	0.52430	0.43791	0.35153	0.26515	0.17876	0.09238	0.00000	0.08504	0.15243	0.21058	0.26126	0.30582	0.34532	0.38056	0.41220
11	0.64163	0.56212	0.48261	0.40309	0.32358	0.24406	0.16455	0.08504	0.00000	0.07877	0.14197	0.19706	0.24550	0.28842	0.32673	0.36112	0.39217
12	0.59437	0.52072	0.44706	0.37340	0.29974	0.22609	0.15243	0.07877	0.00000	0.07337	0.13286	0.18517	0.23153	0.27290	0.31004	0.34357	0.37399
13	0.55360	0.48499	0.41639	0.34778	0.27918	0.21058	0.14197	0.07337	0.00000	0.06866	0.12484	0.17463	0.21907	0.25896	0.29497	0.32765	0.35743
14	0.51806	0.45386	0.38966	0.32546	0.26126	0.19706	0.13286	0.06866	0.00000	0.06452	0.11774	0.16523	0.20788	0.24637	0.28130	0.31313	0.34227
15	0.48680	0.42648	0.36615	0.30582	0.24550	0.18517	0.12484	0.06452	0.00000	0.06085	0.11140	0.15679	0.19777	0.23495	0.26884	0.29985	0.32834
16	0.45911	0.40221	0.34532	0.28842	0.23153	0.17463	0.11774	0.06085	0.00000	0.05757	0.10571	0.14917	0.18861	0.22455	0.25744	0.28765	0.31550

Table F.7: Bottom-up tree edit distance values for grammar 6 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.61538	0.76190	0.83871	0.88372	0.91228	0.93151	0.94505	0.95495	0.96241	0.96815	0.97268	0.97630	0.97925	0.98168	0.98371	0.98542
1	0.61538	0.00000	0.47619	0.64516	0.74419	0.80702	0.84932	0.87912	0.90090	0.91729	0.92994	0.93989	0.94787	0.95436	0.95971	0.96417	0.96793
2	0.76190	0.47619	0.00000	0.38710	0.55814	0.66667	0.73973	0.79121	0.82883	0.85714	0.87898	0.89617	0.90995	0.92116	0.93040	0.93811	0.94461
3	0.83871	0.64516	0.38710	0.00000	0.32558	0.49123	0.60274	0.68132	0.73874	0.78195	0.81529	0.84153	0.86256	0.87967	0.89377	0.90554	0.91545
4	0.88372	0.74419	0.55814	0.32558	0.00000	0.28070	0.43836	0.54945	0.63063	0.69173	0.73885	0.77596	0.80569	0.82988	0.84982	0.86645	0.88047
5	0.91228	0.80702	0.66667	0.49123	0.28070	0.00000	0.24658	0.39560	0.50450	0.58647	0.64968	0.69945	0.73934	0.77178	0.79853	0.82085	0.83965
6	0.93151	0.84932	0.73973	0.60274	0.43836	0.24658	0.00000	0.21978	0.36036	0.46617	0.54777	0.61202	0.66351	0.70539	0.73993	0.76873	0.79300
7	0.94505	0.87912	0.79121	0.68132	0.54945	0.39560	0.21978	0.00000	0.19820	0.33083	0.43312	0.51366	0.57820	0.63071	0.67399	0.71010	0.74052
8	0.95495	0.90090	0.82883	0.73874	0.63063	0.50450	0.36036	0.19820	0.00000	0.18045	0.30573	0.40437	0.48341	0.54772	0.60073	0.64495	0.68222
9	0.91729	0.85714	0.78195	0.69173	0.58647	0.46617	0.33083	0.18045	0.00000	0.16561	0.28415	0.37915	0.45643	0.52015	0.57329	0.61808	0.65617
10	0.87898	0.81529	0.73885	0.64968	0.54777	0.43312	0.30573	0.16561	0.00000	0.15301	0.26540	0.35685	0.43223	0.49511	0.54810	0.59318	0.63183
11	0.84153	0.77596	0.69945	0.61202	0.51366	0.40437	0.28415	0.15301	0.00000	0.14218	0.24896	0.33700	0.41042	0.47230	0.52493	0.57007	0.60907
12	0.80569	0.73934	0.66351	0.57820	0.48341	0.37915	0.26540	0.14218	0.00000	0.13278	0.23443	0.31922	0.39067	0.45144	0.50356	0.54860	0.58777
13	0.77178	0.70539	0.63071	0.54772	0.45643	0.35685	0.24896	0.13278	0.00000	0.12454	0.22150	0.30321	0.37270	0.43230	0.48380	0.52860	0.56781
14	0.73993	0.67399	0.60073	0.52015	0.43223	0.33700	0.23443	0.12454	0.00000	0.11726	0.20991	0.28871	0.35629	0.41469	0.46548	0.50995	0.54908
15	0.71010	0.64495	0.57329	0.49511	0.41042	0.31922	0.22150	0.11726	0.00000	0.11079	0.19948	0.27553	0.34125	0.39842	0.44846	0.49251	0.53149
16	0.68222	0.61808	0.54810	0.47230	0.39067	0.30321	0.20991	0.11079	0.00000	0.10499	0.19002	0.26350	0.32742	0.38336	0.43261	0.47619	0.51494

Table F.8: Bottom-up tree edit distance values for grammar 7 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.53846	0.71429	0.80645	0.86047	0.89474	0.91781	0.93407	0.94595	0.95489	0.96178	0.96721	0.97156	0.97510	0.97802	0.98046	0.98251
1	0.53846	0.00000	0.42857	0.61290	0.72093	0.78947	0.83562	0.86813	0.89189	0.90977	0.92357	0.93443	0.94313	0.95021	0.95604	0.96091	0.96501
2	0.71429	0.42857	0.00000	0.35484	0.53488	0.64912	0.72603	0.78022	0.81982	0.84962	0.87261	0.89071	0.90521	0.91701	0.92674	0.93485	0.94169
3	0.80645	0.61290	0.35484	0.00000	0.30233	0.47368	0.58904	0.67033	0.72973	0.77444	0.80892	0.83607	0.85782	0.87552	0.89011	0.90228	0.91254
4	0.86047	0.72093	0.53488	0.30233	0.00000	0.26316	0.42466	0.53846	0.62162	0.68421	0.73248	0.77049	0.80095	0.82573	0.84615	0.86319	0.87755
5	0.89474	0.78947	0.64912	0.47368	0.26316	0.00000	0.23288	0.38462	0.49550	0.57895	0.64331	0.69399	0.73460	0.76763	0.79487	0.81759	0.83673
6	0.91781	0.83562	0.72603	0.58904	0.42466	0.23288	0.00000	0.20879	0.35135	0.45865	0.54140	0.60656	0.65877	0.70124	0.73626	0.76547	0.79009
7	0.93407	0.86813	0.78022	0.67033	0.53846	0.38462	0.20879	0.00000	0.18919	0.32331	0.42675	0.50820	0.57346	0.62656	0.67033	0.70684	0.73761
8	0.94595	0.89189	0.81982	0.72973	0.62162	0.49550	0.35135	0.18919	0.00000	0.17293	0.29936	0.39891	0.47867	0.54357	0.59707	0.64169	0.67930
9	0.90977	0.84962	0.77444	0.68421	0.57895	0.45865	0.32331	0.17293	0.00000	0.15924	0.27869	0.37441	0.45228	0.51648	0.57003	0.61516	0.65354
10	0.87261	0.80892	0.73248	0.64331	0.54140	0.42675	0.29936	0.15924	0.00000	0.14754	0.26066	0.35270	0.42857	0.49186	0.54519	0.59055	0.62945
11	0.83607	0.77049	0.69399	0.60656	0.50820	0.39891	0.27869	0.14754	0.00000	0.13744	0.24481	0.33333	0.40717	0.46939	0.52231	0.56770	0.60691
12	0.80095	0.73460	0.65877	0.57346	0.47867	0.37441	0.26066	0.13744	0.00000	0.12863	0.23077	0.31596	0.38776	0.44882	0.50119	0.54644	0.58580
13	0.76763	0.70124	0.62656	0.54357	0.45228	0.35270	0.24481	0.12863	0.00000	0.12088	0.21824	0.30029	0.37008	0.42993	0.48164	0.52663	0.56600
14	0.73626	0.67033	0.59707	0.51648	0.42857	0.33333	0.23077	0.12088	0.00000	0.11401	0.20700	0.28609	0.35392	0.41253	0.46351	0.50814	0.54742
15	0.70684	0.64169	0.57003	0.49186	0.40717	0.31596	0.21824	0.11401	0.00000	0.10787	0.19685	0.27316	0.33909	0.39645	0.44665	0.49085	0.52995
16	0.67930	0.61516	0.54519	0.46939	0.38776	0.30029	0.20700	0.10787	0.00000	0.10236	0.18765	0.26134	0.32544	0.38156	0.43095	0.47465	0.51351

Table F.9: Bottom-up tree edit distance values for grammar 8 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.47368	0.62963	0.71429	0.76744	0.80392	0.83051	0.85075	0.86667	0.87952	0.89011	0.89899	0.90654	0.91304	0.91870	0.92366	0.92806
1	0.47368	0.00000	0.33333	0.48571	0.58140	0.64706	0.69492	0.73134	0.76000	0.78313	0.80220	0.81818	0.83178	0.84348	0.85366	0.86260	0.87050
2	0.62963	0.33333	0.00000	0.25714	0.39535	0.49020	0.55932	0.61194	0.65333	0.68675	0.71429	0.73737	0.75701	0.77391	0.78862	0.80153	0.81295
3	0.71429	0.48571	0.25714	0.00000	0.20930	0.33333	0.42373	0.49254	0.54667	0.59036	0.62637	0.65657	0.68224	0.70435	0.72358	0.74046	0.75540
4	0.76744	0.58140	0.39535	0.20930	0.00000	0.17647	0.28814	0.37313	0.44000	0.49398	0.53846	0.57576	0.60748	0.63478	0.65854	0.67939	0.69784
5	0.80392	0.64706	0.49020	0.33333	0.17647	0.00000	0.15254	0.25373	0.33333	0.39759	0.45055	0.49495	0.53271	0.56522	0.59350	0.61832	0.64029
6	0.83051	0.69492	0.55932	0.42373	0.28814	0.15254	0.00000	0.13433	0.22667	0.30120	0.36264	0.41414	0.45794	0.49565	0.52846	0.55725	0.58273
7	0.85075	0.73134	0.61194	0.49254	0.37313	0.25373	0.13433	0.00000	0.12000	0.20482	0.27473	0.33333	0.38318	0.42609	0.46341	0.49618	0.52518
8	0.86667	0.76000	0.65333	0.54667	0.44000	0.33333	0.22667	0.12000	0.00000	0.10843	0.18681	0.25253	0.30841	0.35652	0.39837	0.43511	0.46763
9	0.78313	0.68675	0.59036	0.49398	0.39759	0.30120	0.20482	0.10843	0.00000	0.09890	0.17172	0.23364	0.28696	0.33333	0.37405	0.41007	0.44218
10	0.71429	0.62637	0.53846	0.45055	0.36264	0.27473	0.18681	0.09890	0.00000	0.09091	0.15888	0.21739	0.26829	0.31298	0.35252	0.38776	0.41935
11	0.65657	0.57576	0.49495	0.41414	0.33333	0.25253	0.17172	0.09091	0.00000	0.08411	0.14783	0.20325	0.25191	0.29496	0.33333	0.36774	0.39877
12	0.60748	0.53271	0.45794	0.38318	0.30841	0.23364	0.15888	0.08411	0.00000	0.07826	0.13821	0.19084	0.23741	0.27891	0.31613	0.34969	0.38012
13	0.56522	0.49565	0.42609	0.35652	0.28696	0.21739	0.14783	0.07826	0.00000	0.07317	0.12977	0.17986	0.22449	0.26452	0.30061	0.33333	0.36313
14	0.52846	0.46341	0.39837	0.33333	0.26829	0.20325	0.13821	0.07317	0.00000	0.06870	0.12230	0.17007	0.21290	0.25153	0.28655	0.31844	0.34759
15	0.49618	0.43511	0.37405	0.31298	0.25191	0.19084	0.12977	0.06870	0.00000	0.06475	0.11565	0.16129	0.20245	0.23977	0.27374	0.30481	0.33333
16	0.46763	0.41007	0.35252	0.29496	0.23741	0.17986	0.12230	0.06475	0.00000	0.06122	0.10968	0.15337	0.19298	0.22905	0.26203	0.29231	0.32020

Table F.10: Bottom-up tree edit distance values for grammar 9 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.59300	0.70292	0.76609	0.80711	0.83589	0.85719	0.87360	0.88663	0.89722	0.90600	0.91340	0.91972	0.92518	0.92995	0.93414	0.93786
1	0.59308	0.00000	0.43293	0.55350	0.63179	0.68672	0.72739	0.75871	0.78358	0.80380	0.82056	0.83469	0.84675	0.85717	0.86627	0.87428	0.88138
2	0.70280	0.43261	0.00000	0.34059	0.45625	0.53739	0.59746	0.64373	0.68045	0.71031	0.73507	0.75593	0.77374	0.78913	0.80256	0.81438	0.82487
3	0.76616	0.55357	0.34099	0.00000	0.28121	0.38842	0.46781	0.52895	0.57749	0.61697	0.64969	0.67727	0.70082	0.72116	0.73892	0.75455	0.76841
4	0.80716	0.63186	0.45655	0.28125	0.00000	0.23930	0.33804	0.41409	0.47446	0.52356	0.56426	0.59856	0.62786	0.65316	0.67525	0.69469	0.71194
5	0.83585	0.68663	0.53741	0.38818	0.23896	0.00000	0.20793	0.29896	0.37122	0.42998	0.47869	0.51974	0.55479	0.58507	0.61150	0.63476	0.65539
6	0.85721	0.72739	0.59758	0.46777	0.33796	0.20815	0.00000	0.18423	0.26830	0.33666	0.39334	0.44110	0.48188	0.51712	0.54787	0.57494	0.59895
7	0.87359	0.75868	0.64377	0.52885	0.41394	0.29902	0.18411	0.00000	0.16513	0.24314	0.30782	0.36231	0.40885	0.44906	0.48415	0.51503	0.54243
8	0.88661	0.78353	0.68045	0.57737	0.47429	0.37121	0.26813	0.16505	0.00000	0.14962	0.22230	0.28353	0.33582	0.38100	0.42042	0.45513	0.48591
9	0.80377	0.71033	0.61689	0.52344	0.43000	0.33656	0.24311	0.14967	0.00000	0.13688	0.20483	0.26287	0.31301	0.35676	0.39527	0.42944	0.45994
10	0.73513	0.64968	0.56424	0.47880	0.39335	0.30791	0.22247	0.13703	0.00000	0.12624	0.19000	0.24509	0.29316	0.33548	0.37301	0.40653	0.43665
11	0.67723	0.59850	0.51978	0.44105	0.36233	0.28360	0.20488	0.12615	0.00000	0.11695	0.17701	0.22942	0.27555	0.31648	0.35303	0.38586	0.41553
12	0.62783	0.55486	0.48188	0.40891	0.33593	0.26296	0.18998	0.11701	0.00000	0.10905	0.16579	0.21573	0.26003	0.29959	0.33514	0.36726	0.39641
13	0.58507	0.51705	0.44903	0.38101	0.31298	0.24496	0.17694	0.10892	0.00000	0.10198	0.15575	0.20345	0.24604	0.28431	0.31888	0.35027	0.37889
14	0.54783	0.48414	0.42046	0.35677	0.29309	0.22940	0.16571	0.10203	0.00000	0.09592	0.14699	0.19260	0.23358	0.27060	0.30421	0.33486	0.36292
15	0.51506	0.45520	0.39533	0.33546	0.27559	0.21572	0.15585	0.09598	0.00000	0.09056	0.13919	0.18287	0.22234	0.25817	0.29085	0.32077	0.34826
16	0.48607	0.42959	0.37312	0.31664	0.26017	0.20369	0.14721	0.09074	0.00000	0.08589	0.13227	0.17418	0.21222	0.24692	0.27868	0.30788	0.33480

Table F.11: Bottom-up tree edit distance values for grammar 10 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.32244	0.43804	0.48495	0.53463	0.56250	0.58524	0.60983	0.65895	0.64263	0.69223	0.69336	0.72589	0.74036	0.76960	0.76295	0.75974
1	0.32255	0.00000	0.34700	0.37650	0.49688	0.51429	0.51053	0.60474	0.62450	0.65795	0.67417	0.66274	0.69012	0.70352	0.72352	0.73972	0.73360
2	0.42490	0.34125	0.00000	0.36496	0.45151	0.48335	0.50490	0.52848	0.57199	0.59736	0.60496	0.64122	0.66962	0.68249	0.70108	0.70145	0.72890
3	0.45608	0.36149	0.36909	0.00000	0.42222	0.45455	0.47943	0.53497	0.57677	0.55852	0.60315	0.63099	0.66872	0.66875	0.68862	0.68409	0.71377
4	0.52481	0.47984	0.39767	0.41938	0.00000	0.43750	0.48039	0.54870	0.53190	0.60200	0.57746	0.63329	0.66979	0.68642	0.72511	0.70115	0.73220
5	0.54915	0.52707	0.47721	0.47863	0.46581	0.00000	0.47932	0.52948	0.55436	0.57839	0.59791	0.62329	0.67068	0.66399	0.67177	0.69630	0.70302
6	0.60144	0.55236	0.53141	0.53992	0.50785	0.49215	0.00000	0.55744	0.57865	0.62539	0.60109	0.64226	0.64929	0.67422	0.70990	0.70978	0.71158
7	0.61890	0.59451	0.56829	0.53841	0.56098	0.50305	0.51829	0.00000	0.57183	0.55461	0.62307	0.61221	0.64203	0.65848	0.67807	0.70015	0.72570
8	0.64471	0.61768	0.60135	0.59234	0.57432	0.55405	0.54223	0.51633	0.00000	0.55503	0.60129	0.61020	0.61613	0.65084	0.71064	0.72176	0.72594
9	0.63841	0.60544	0.61381	0.55939	0.55835	0.59393	0.58399	0.56829	0.00000	0.58239	0.63451	0.65108	0.65530	0.69518	0.72435	0.71817	0.71353
10	0.63214	0.63019	0.63410	0.59502	0.61553	0.57596	0.58818	0.59404	0.00000	0.59705	0.62342	0.63368	0.64492	0.67784	0.72810	0.70760	0.73710
11	0.61571	0.64458	0.61524	0.61903	0.60388	0.59915	0.60767	0.58874	0.00000	0.61935	0.63667	0.66438	0.67334	0.70433	0.70625	0.73885	0.72515
12	0.64617	0.63644	0.62937	0.64042	0.57983	0.62494	0.66873	0.64573	0.00000	0.63621	0.67937	0.69119	0.71245	0.74458	0.71323	0.72190	0.74907
13	0.64124	0.66170	0.66510	0.65530	0.68683	0.65616	0.63272	0.62804	0.00000	0.63681	0.66615	0.68593	0.71099	0.72196	0.71596	0.72907	0.75151
14	0.65620	0.66103	0.65660	0.65660	0.65902	0.65821	0.66506	0.64976	0.00000	0.66245	0.69546	0.70670	0.71916	0.75540	0.74690	0.73045	0.74273
15	0.70851	0.70851	0.70618	0.68791	0.70851	0.70152	0.70773	0.71318	0.00000	0.73200	0.72880	0.74170	0.74589	0.73017	0.78298	0.76966	0.78594
16	0.68481	0.69290	0.68518	0.68003	0.69548	0.67562	0.69070	0.68591	0.00000	0.67921	0.70570	0.73394	0.72548	0.74028	0.74404	0.76082	0.75815

Table F.12: Bottom-up tree edit distance values for grammar 11 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	1.00000	0.23077	1.00000	0.37500	1.00000	0.47368	1.00000	0.54545	1.00000	0.60000	1.00000	0.64286	1.00000	0.67742	1.00000	0.70588
1	1.00000	0.00000	1.00000	0.20000	1.00000	0.33333	1.00000	0.42857	1.00000	0.50000	1.00000	0.55556	1.00000	0.60000	1.00000	0.63636	1.00000
2	0.23077	1.00000	0.00000	1.00000	0.18750	1.00000	0.31579	1.00000	0.40909	1.00000	0.48000	1.00000	0.53571	1.00000	0.58065	1.00000	0.61765
3	1.00000	0.20000	1.00000	0.00000	1.00000	0.16667	1.00000	0.28571	1.00000	0.37500	1.00000	0.44444	1.00000	0.50000	1.00000	0.54545	1.00000
4	0.37500	1.00000	0.18750	1.00000	0.00000	1.00000	0.15789	1.00000	0.27273	1.00000	0.36000	1.00000	0.42857	1.00000	0.48387	1.00000	0.52941
5	1.00000	0.33333	1.00000	0.16667	1.00000	0.00000	1.00000	0.14286	1.00000	0.25000	1.00000	0.33333	1.00000	0.40000	1.00000	0.45455	1.00000
6	0.47368	1.00000	0.31579	1.00000	0.15789	1.00000	0.00000	1.00000	0.13636	1.00000	0.24000	1.00000	0.32143	1.00000	0.38710	1.00000	0.44118
7	1.00000	0.42857	1.00000	0.28571	1.00000	0.14286	1.00000	0.00000	1.00000	0.12500	1.00000	0.22222	1.00000	0.30000	1.00000	0.36364	1.00000
8	0.54545	1.00000	0.40909	1.00000	0.27273	1.00000	0.13636	1.00000	0.00000	1.00000	0.12000	1.00000	0.21429	1.00000	0.29032	1.00000	0.35294
9	0.50000	1.00000	0.37500	1.00000	0.25000	1.00000	0.12500	1.00000	0.00000	1.00000	0.11111	1.00000	0.20000	1.00000	0.27273	1.00000	0.33333
10	0.48000	1.00000	0.36000	1.00000	0.24000	1.00000	0.12000	1.00000	0.00000	1.00000	0.10714	1.00000	0.19355	1.00000	0.26471	1.00000	0.32432
11	0.44444	1.00000	0.33333	1.00000	0.22222	1.00000	0.11111	1.00000	0.00000	1.00000	0.10000	1.00000	0.18182	1.00000	0.25000	1.00000	0.30769
12	0.42857	1.00000	0.32143	1.00000	0.21429	1.00000	0.10714	1.00000	0.00000	1.00000	0.09677	1.00000	0.17647	1.00000	0.24324	1.00000	0.30000
13	0.40000	1.00000	0.30000	1.00000	0.20000	1.00000	0.10000	1.00000	0.00000	1.00000	0.09091	1.00000	0.16667	1.00000	0.23077	1.00000	0.28571
14	0.38710	1.00000	0.29032	1.00000	0.19355	1.00000	0.09677	1.00000	0.00000	1.00000	0.08824	1.00000	0.16216	1.00000	0.22500	1.00000	0.27907
15	0.36364	1.00000	0.27273	1.00000	0.18182	1.00000	0.09091	1.00000	0.00000	1.00000	0.08333	1.00000	0.15385	1.00000	0.21429	1.00000	0.26667
16	0.35294	1.00000	0.26471	1.00000	0.17647	1.00000	0.08824	1.00000	0.00000	1.00000	0.08108	1.00000	0.15000	1.00000	0.20930	1.00000	0.26087

Table F.13: Bottom-up tree edit distance values for grammar 12 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.16842	0.23158	0.09901	0.05263	0.21053	0.27368	0.04211	0.11579	0.25263	0.33684	0.08421	0.15789	0.29474	0.37895	0.06316	0.11579
1	0.16842	0.00000	0.13095	0.20792	0.16304	0.05952	0.17857	0.17895	0.12791	0.13095	0.25000	0.17391	0.14286	0.17857	0.29762	0.17895	0.12791
2	0.23158	0.13095	0.00000	0.30693	0.22826	0.12346	0.06667	0.26316	0.19767	0.05333	0.14667	0.26087	0.19277	0.10667	0.20000	0.28421	0.19767
3	0.09901	0.20792	0.30693	0.00000	0.10891	0.21782	0.31683	0.09901	0.19802	0.30693	0.40594	0.10891	0.20792	0.31683	0.41584	0.08911	0.18812
4	0.05263	0.16304	0.22826	0.10891	0.00000	0.17391	0.23913	0.08421	0.10870	0.25000	0.33696	0.04348	0.11957	0.26087	0.34783	0.09474	0.09783
5	0.21053	0.05952	0.12346	0.21782	0.17391	0.00000	0.13580	0.22105	0.17442	0.12346	0.24691	0.18478	0.13253	0.13580	0.25926	0.21053	0.16279
6	0.27368	0.17857	0.06667	0.31683	0.23913	0.13580	0.00000	0.30526	0.24419	0.10667	0.13889	0.27174	0.20482	0.05556	0.15278	0.31579	0.23256
7	0.04211	0.17895	0.26316	0.09901	0.08421	0.22105	0.30526	0.00000	0.11579	0.23158	0.33684	0.05263	0.15789	0.27368	0.37895	0.06316	0.14737
8	0.11579	0.12791	0.19767	0.19802	0.10870	0.17442	0.24419	0.11579	0.00000	0.18605	0.25581	0.10870	0.05814	0.23256	0.30233	0.16842	0.06977
9	0.13095	0.05333	0.30693	0.25000	0.12346	0.10667	0.23158	0.18605	0.00000	0.14667	0.22826	0.18072	0.06667	0.20000	0.28421	0.22093	0.08000
10	0.14667	0.40594	0.33696	0.24691	0.13889	0.33684	0.25581	0.14667	0.00000	0.33696	0.25301	0.13889	0.07576	0.38947	0.31395	0.21333	0.09091
11	0.10891	0.04348	0.18478	0.27174	0.05263	0.10870	0.22826	0.33696	0.00000	0.11957	0.23913	0.34783	0.09474	0.13043	0.27174	0.35870	0.06522
12	0.11957	0.13253	0.20482	0.15789	0.05814	0.18072	0.25301	0.11957	0.00000	0.19277	0.26506	0.20000	0.10465	0.22892	0.30120	0.17391	0.07229
13	0.13580	0.05556	0.27368	0.23256	0.06667	0.13889	0.23913	0.19277	0.00000	0.15278	0.31579	0.25581	0.12000	0.16667	0.29348	0.22892	0.08333
14	0.15278	0.37895	0.30233	0.20000	0.07576	0.34783	0.26506	0.15278	0.00000	0.42105	0.34884	0.25333	0.13636	0.40217	0.32530	0.22222	0.09524
15	0.06316	0.16842	0.28421	0.38947	0.09474	0.20000	0.31579	0.42105	0.00000	0.11579	0.23158	0.33684	0.05263	0.15789	0.27368	0.37895	0.11579
16	0.06977	0.22093	0.31395	0.13043	0.10465	0.25581	0.34884	0.11579	0.00000	0.18605	0.25581	0.10870	0.05814	0.23256	0.30233	0.04651	0.12791

Table F.14: Bottom-up tree edit distance values for grammar 13 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.50000	0.70000	0.80000	0.85714	0.89286	0.91667	0.93333	0.94545	0.95455	0.96154	0.96703	0.97143	0.97500	0.97794	0.98039	0.98246
1	0.50000	0.00000	0.40000	0.60000	0.71429	0.78571	0.83333	0.86667	0.89091	0.90909	0.92308	0.93407	0.94286	0.95000	0.95588	0.96078	0.96491
2	0.70000	0.40000	0.00000	0.33333	0.52381	0.64286	0.72222	0.77778	0.81818	0.84848	0.87179	0.89011	0.90476	0.91667	0.92647	0.93464	0.94152
3	0.80000	0.60000	0.33333	0.00000	0.28571	0.46429	0.58333	0.66667	0.72727	0.77273	0.80769	0.83516	0.85714	0.87500	0.88971	0.90196	0.91228
4	0.85714	0.71429	0.52381	0.28571	0.00000	0.25000	0.41667	0.53333	0.61818	0.68182	0.73077	0.76923	0.80000	0.82500	0.84559	0.86275	0.87719
5	0.89286	0.78571	0.64286	0.46429	0.25000	0.00000	0.22222	0.37778	0.49091	0.57576	0.64103	0.69231	0.73333	0.76667	0.79412	0.81699	0.83626
6	0.91667	0.83333	0.72222	0.58333	0.41667	0.22222	0.00000	0.20000	0.34545	0.45455	0.53846	0.60440	0.65714	0.70000	0.73529	0.76471	0.78947
7	0.93333	0.86667	0.77778	0.66667	0.53333	0.37778	0.20000	0.00000	0.18182	0.31818	0.42308	0.50549	0.57143	0.62500	0.66912	0.70588	0.73684
8	0.94545	0.89091	0.81818	0.72727	0.61818	0.49091	0.34545	0.18182	0.00000	0.16667	0.29487	0.39560	0.47619	0.54167	0.59559	0.64052	0.67836
9	0.90909	0.84848	0.77273	0.68182	0.57576	0.45455	0.31818	0.16667	0.00000	0.15385	0.27473	0.37143	0.45000	0.51471	0.56863	0.61404	0.65263
10	0.87179	0.80769	0.73077	0.64103	0.53846	0.42308	0.29487	0.15385	0.00000	0.14286	0.25714	0.35000	0.42647	0.49020	0.54386	0.58947	0.62857
11	0.83516	0.76923	0.69231	0.60440	0.50549	0.39560	0.27473	0.14286	0.00000	0.13333	0.24167	0.33088	0.40523	0.46784	0.52105	0.56667	0.60606
12	0.80000	0.73333	0.65714	0.57143	0.47619	0.37143	0.25714	0.13333	0.00000	0.12500	0.22794	0.31373	0.38596	0.44737	0.50000	0.54545	0.58498
13	0.76667	0.70000	0.62500	0.54167	0.45000	0.35000	0.24167	0.12500	0.00000	0.11765	0.21569	0.29825	0.36842	0.42857	0.48052	0.52569	0.56522
14	0.73529	0.66912	0.59559	0.51471	0.42647	0.33088	0.22794	0.11765	0.00000	0.11111	0.20468	0.28421	0.35238	0.41126	0.46245	0.50725	0.54667
15	0.70588	0.64052	0.56863	0.49020	0.40523	0.31373	0.21569	0.11111	0.00000	0.10526	0.19474	0.27143	0.33766	0.39526	0.44565	0.49000	0.52923
16	0.67836	0.61404	0.54386	0.46784	0.38596	0.29825	0.20468	0.10526	0.00000	0.10000	0.18571	0.25974	0.32411	0.38043	0.43000	0.47385	0.51282

Table F.15: Bottom-up tree edit distance values for grammar 14 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.28571	0.46429	0.58333	0.66667	0.72727	0.77273	0.80769	0.83516	0.85714	0.87500	0.88971	0.90196	0.91228	0.92105	0.92857	0.93506
1	0.28571	0.00000	0.25000	0.41667	0.53333	0.61818	0.68182	0.73077	0.76923	0.80000	0.82500	0.84559	0.86275	0.87719	0.88947	0.90000	0.90909
2	0.46429	0.25000	0.00000	0.22222	0.37778	0.49091	0.57576	0.64103	0.69231	0.73333	0.76667	0.79412	0.81699	0.83626	0.85263	0.86667	0.87879
3	0.58333	0.41667	0.22222	0.00000	0.20000	0.34545	0.45455	0.53846	0.60440	0.65714	0.70000	0.73529	0.76471	0.78947	0.81053	0.82857	0.84416
4	0.66667	0.53333	0.37778	0.20000	0.00000	0.18182	0.31818	0.42308	0.50549	0.57143	0.62500	0.66912	0.70588	0.73684	0.76316	0.78571	0.80519
5	0.72727	0.61818	0.49091	0.34545	0.18182	0.00000	0.16667	0.29487	0.39560	0.47619	0.54167	0.59559	0.64052	0.67836	0.71053	0.73810	0.76190
6	0.77273	0.68182	0.57576	0.45455	0.31818	0.16667	0.00000	0.15385	0.27473	0.37143	0.45000	0.51471	0.56863	0.61404	0.65263	0.68571	0.71429
7	0.80769	0.73077	0.64103	0.53846	0.42308	0.29487	0.15385	0.00000	0.14286	0.25714	0.35000	0.42647	0.49020	0.54386	0.58947	0.62857	0.66234
8	0.83516	0.76923	0.69231	0.60440	0.50549	0.39560	0.27473	0.14286	0.00000	0.13333	0.24167	0.33088	0.40523	0.46784	0.52105	0.56667	0.60606
9	0.80000	0.73333	0.65714	0.57143	0.47619	0.37143	0.25714	0.13333	0.00000	0.12500	0.22794	0.31373	0.38596	0.44737	0.50000	0.54545	0.58498
10	0.76667	0.70000	0.62500	0.54167	0.45000	0.35000	0.24167	0.12500	0.00000	0.11765	0.21569	0.29825	0.36842	0.42857	0.48052	0.52569	0.56522
11	0.73529	0.66912	0.59559	0.51471	0.42647	0.33088	0.22794	0.11765	0.00000	0.11111	0.20468	0.28421	0.35238	0.41126	0.46245	0.50725	0.54667
12	0.70588	0.64052	0.56863	0.49020	0.40523	0.31373	0.21569	0.11111	0.00000	0.10526	0.19474	0.27143	0.33766	0.39526	0.44565	0.49000	0.52923
13	0.67836	0.61404	0.54386	0.46784	0.38596	0.29825	0.20468	0.10526	0.00000	0.10000	0.18571	0.25974	0.32411	0.38043	0.43000	0.47385	0.51282
14	0.65263	0.58947	0.52105	0.44737	0.36842	0.28421	0.19474	0.10000	0.00000	0.09524	0.17749	0.24901	0.31159	0.36667	0.41538	0.45869	0.49735
15	0.62857	0.56667	0.50000	0.42857	0.35238	0.27143	0.18571	0.09524	0.00000	0.09091	0.16996	0.23913	0.30000	0.35385	0.40171	0.44444	0.48276
16	0.60606	0.54545	0.48052	0.41126	0.33766	0.25974	0.17749	0.09091	0.00000	0.08696	0.16304	0.23000	0.28923	0.34188	0.38889	0.43103	0.46897

Table F.16: Bottom-up tree edit distance values for grammar 15 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.32258	0.51163	0.63158	0.71233	0.76923	0.81081	0.84444	0.87425	0.90141	0.92527	0.94488	0.96000	0.97111	0.97906	0.98468	0.98865
1	0.32258	0.00000	0.27907	0.45614	0.57534	0.65934	0.72072	0.77037	0.81437	0.85446	0.88968	0.91864	0.94095	0.95736	0.96909	0.97739	0.98325
2	0.51163	0.27907	0.00000	0.24561	0.41096	0.52747	0.61261	0.68148	0.74251	0.79812	0.84698	0.88714	0.91810	0.94085	0.95713	0.96864	0.97677
3	0.63158	0.45614	0.24561	0.00000	0.21918	0.37363	0.48649	0.57778	0.65868	0.73239	0.79715	0.85039	0.89143	0.92160	0.94317	0.95842	0.96921
4	0.71233	0.57534	0.41096	0.21918	0.00000	0.19780	0.34234	0.45926	0.56287	0.65728	0.74021	0.80840	0.86095	0.89959	0.92722	0.94675	0.96056
5	0.76923	0.65934	0.52747	0.37363	0.19780	0.00000	0.18018	0.32593	0.45509	0.57277	0.67616	0.76115	0.82667	0.87483	0.90927	0.93363	0.95084
6	0.81081	0.72072	0.61261	0.48649	0.34234	0.18018	0.00000	0.17778	0.33533	0.47887	0.60498	0.70866	0.78857	0.84732	0.88933	0.91904	0.94003
7	0.84444	0.77037	0.68148	0.57778	0.45926	0.32593	0.17778	0.00000	0.19162	0.36620	0.51957	0.64567	0.74286	0.81431	0.86540	0.90153	0.92707
8	0.87425	0.81437	0.74251	0.65868	0.56287	0.45509	0.33533	0.19162	0.00000	0.21596	0.40569	0.56168	0.68190	0.77029	0.83350	0.87819	0.90978
9	0.85446	0.79812	0.73239	0.65728	0.57277	0.47887	0.36620	0.21596	0.00000	0.24199	0.44094	0.59429	0.70702	0.78764	0.84464	0.88493	0.91366
10	0.84698	0.79715	0.74021	0.67616	0.60498	0.51957	0.40569	0.24199	0.00000	0.26247	0.46476	0.61348	0.71984	0.79504	0.84819	0.88610	0.91357
11	0.85039	0.80840	0.76115	0.70866	0.64567	0.56168	0.44094	0.26247	0.00000	0.27429	0.47593	0.62014	0.72210	0.79417	0.84556	0.88281	0.91033
12	0.86095	0.82667	0.78857	0.74286	0.68190	0.59429	0.46476	0.27429	0.00000	0.27785	0.47657	0.61707	0.71637	0.78719	0.83851	0.87644	0.90505
13	0.87483	0.84732	0.81431	0.77029	0.70702	0.61348	0.47593	0.27785	0.00000	0.27517	0.46973	0.60724	0.70531	0.77638	0.82890	0.86851	0.89890
14	0.88933	0.86540	0.83350	0.78764	0.71984	0.62014	0.47657	0.27517	0.00000	0.26842	0.45813	0.59343	0.69148	0.76394	0.81859	0.86052	0.89306
15	0.90153	0.87819	0.84464	0.79504	0.72210	0.61707	0.46973	0.26842	0.00000	0.25932	0.44426	0.57828	0.67734	0.75203	0.80935	0.85382	0.88849
16	0.90978	0.88493	0.84819	0.79417	0.71637	0.60724	0.45813	0.25932	0.00000	0.24970	0.43064	0.56437	0.66522	0.74259	0.80264	0.84945	0.88585

Table F.17: Bottom-up tree edit distance values for grammar 16 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.90602	0.90139	0.87692	0.88427	0.87716	0.86912	0.83059	0.83715	0.84245	0.83059	0.80055	0.78116	0.78451	0.75849	0.64179	0.87716
1	0.90602	0.00000	0.07195	0.16593	0.20117	0.28341	0.34214	0.43612	0.47137	0.37739	0.43612	0.53010	0.56535	0.57709	0.63583	0.72981	0.24816
2	0.90139	0.07195	0.00000	0.12481	0.16179	0.23575	0.29738	0.39599	0.43297	0.34669	0.39599	0.50693	0.54391	0.55624	0.60555	0.71649	0.21109
3	0.87692	0.16593	0.12481	0.00000	0.07009	0.16581	0.23419	0.32991	0.38462	0.26154	0.32991	0.43932	0.48034	0.50769	0.57607	0.67179	0.12479
4	0.88427	0.20117	0.16179	0.07009	0.00000	0.11754	0.18987	0.30561	0.34901	0.23327	0.30561	0.42134	0.46474	0.47920	0.55154	0.66727	0.07414
5	0.87716	0.28341	0.23575	0.16581	0.11754	0.00000	0.09405	0.21689	0.26296	0.18618	0.24760	0.38580	0.43186	0.44722	0.50864	0.64683	0.07869
6	0.86912	0.34214	0.29738	0.23419	0.18987	0.09405	0.00000	0.16564	0.21472	0.14928	0.19836	0.34560	0.39468	0.42740	0.47648	0.62372	0.14012
7	0.83059	0.43612	0.39599	0.32991	0.30561	0.21689	0.16564	0.00000	0.09647	0.14223	0.05882	0.22824	0.28471	0.34118	0.39765	0.54824	0.26296
8	0.83715	0.47137	0.43297	0.38462	0.34901	0.26296	0.21472	0.09647	0.00000	0.21225	0.13412	0.18575	0.24682	0.28753	0.34860	0.53181	0.30902
9	0.37739	0.34669	0.26154	0.23327	0.18618	0.14928	0.14223	0.21225	0.00000	0.10722	0.24726	0.29978	0.36980	0.45733	0.57987	0.18618	0.28228
10	0.39599	0.32991	0.30561	0.24760	0.19836	0.05882	0.13412	0.10722	0.00000	0.19059	0.24706	0.34118	0.39765	0.54824	0.26296	0.20941	0.32235
11	0.43932	0.42134	0.38580	0.34560	0.22824	0.18575	0.24726	0.19059	0.00000	0.11357	0.24654	0.31302	0.46814	0.38580	0.13573	0.20222	0.49030
12	0.46474	0.43186	0.39468	0.28471	0.24682	0.29978	0.24706	0.11357	0.00000	0.17325	0.24620	0.41641	0.43186	0.18006	0.12462	0.44073	0.53799
13	0.44722	0.42740	0.34118	0.28753	0.36980	0.34118	0.24654	0.17325	0.00000	0.16498	0.38047	0.44722	0.24654	0.13805	0.35354	0.48822	0.70370
14	0.47648	0.39765	0.34860	0.45733	0.39765	0.31302	0.24620	0.16498	0.00000	0.30566	0.52399	0.29086	0.19192	0.30566	0.39623	0.66792	0.90912
15	0.54824	0.53181	0.57987	0.54824	0.46814	0.41641	0.38047	0.30566	0.00000	0.64683	0.49030	0.35354	0.08458	0.24378	0.52239	0.93258	0.93178
16	0.30902	0.18618	0.26296	0.38580	0.43186	0.44722	0.52399	0.64683	0.00000	0.37044	0.49328	0.63148	0.70825	0.83109	0.81239	0.81016	0.80555

Table F.18: Bottom-up tree edit distance values for grammar 17 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.36778	0.36778	0.46505	0.85410	0.85410	0.53799	0.85410	0.85410	0.75758	0.80140	0.79634	0.74518	0.77393	0.76358	0.71020	0.74822
1	0.36778	0.00000	0.24579	0.56902	0.59596	0.73064	0.67677	0.59596	0.78451	0.83444	0.77036	0.80907	0.81894	0.75205	0.77884	0.79409	0.72386
2	0.36778	0.24579	0.00000	0.60142	0.77224	0.62989	0.62989	0.74377	0.62989	0.84627	0.82622	0.77725	0.83236	0.80310	0.75596	0.79409	0.76447
3	0.46505	0.56902	0.60142	0.00000	0.71084	0.71084	0.35743	0.74297	0.80723	0.83444	0.88829	0.88544	0.79212	0.86873	0.86273	0.77884	0.85381
4	0.85410	0.59596	0.77224	0.71084	0.00000	0.52239	0.76119	0.36318	0.68159	0.93496	0.86346	0.91090	0.91953	0.81768	0.88561	0.91611	0.81320
5	0.85410	0.73064	0.62989	0.71084	0.52239	0.00000	0.68362	0.50282	0.32203	0.94678	0.92552	0.87908	0.93965	0.91249	0.83222	0.91611	0.88629
6	0.53799	0.67677	0.62989	0.35743	0.76119	0.68362	0.00000	0.68362	0.68362	0.88766	0.92552	0.91726	0.86588	0.91249	0.90086	0.83222	0.89442
7	0.85410	0.59596	0.74377	0.74297	0.36318	0.50282	0.68362	0.00000	0.50345	0.95270	0.90690	0.92999	0.94635	0.88332	0.91611	0.93136	0.85381
8	0.85410	0.78451	0.62989	0.80723	0.68159	0.32203	0.68362	0.50345	0.00000	0.96452	0.95035	0.91090	0.95977	0.94166	0.88561	0.93899	0.91878
9	0.83444	0.84627	0.83444	0.93496	0.94678	0.88766	0.95270	0.96452	0.00000	0.19586	0.19586	0.24316	0.43237	0.43237	0.27864	0.43237	0.43237
10	0.82622	0.88829	0.86346	0.92552	0.92552	0.90690	0.95035	0.19586	0.00000	0.13111	0.28006	0.29247	0.35454	0.32971	0.29247	0.37936	0.42281
11	0.88544	0.91090	0.87908	0.91726	0.92999	0.91090	0.19586	0.13111	0.00000	0.28719	0.36356	0.29992	0.29992	0.35084	0.29992	0.43357	0.52267
12	0.91953	0.93965	0.86588	0.94635	0.95977	0.24316	0.28006	0.28719	0.00000	0.31601	0.31601	0.16848	0.32942	0.35624	0.24895	0.50377	0.50377
13	0.91249	0.91249	0.88332	0.94166	0.43237	0.29247	0.36356	0.31601	0.00000	0.21240	0.29991	0.15406	0.27074	0.38742	0.27074	0.41659	0.63537
14	0.90086	0.91611	0.88561	0.43237	0.35454	0.29992	0.31601	0.21240	0.00000	0.25262	0.19161	0.13060	0.35939	0.38990	0.28313	0.66444	0.81697
15	0.93136	0.93899	0.27864	0.32971	0.29992	0.16848	0.29991	0.25262	0.00000	0.25262	0.25262	0.23737	0.48141	0.46616	0.58055	0.90848	0.89323
16	0.91878	0.43237	0.29247	0.35084	0.32942	0.15406	0.19161	0.25262	0.00000	0.17157	0.35497	0.30152	0.36650	0.65888	0.74010	0.82132	0.70761

Table F.19: Bottom-up tree edit distance values for grammar 18 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.34247	0.21176	0.34343	0.51765	0.36986	0.46575	0.58904	0.30928	0.31429	0.50515	0.42735	0.71366	0.80751	0.45882	0.80597	0.84772
1	0.34247	0.00000	0.36471	0.57576	0.40000	0.27692	0.52055	0.49275	0.39175	0.54286	0.34021	0.53846	0.81498	0.76056	0.44706	0.82587	0.82234
2	0.21176	0.36471	0.00000	0.38384	0.44706	0.43529	0.52941	0.62353	0.14433	0.36190	0.44330	0.28205	0.73128	0.77934	0.43529	0.80100	0.83756
3	0.34343	0.57576	0.38384	0.00000	0.29293	0.56566	0.31313	0.40404	0.38384	0.07619	0.25253	0.20513	0.56828	0.67136	0.24242	0.66169	0.70051
4	0.51765	0.40000	0.44706	0.29293	0.00000	0.49412	0.21176	0.24706	0.51546	0.30476	0.14433	0.32479	0.69163	0.60563	0.11765	0.66667	0.67513
5	0.36986	0.27692	0.43529	0.56566	0.49412	0.00000	0.46575	0.49275	0.47423	0.56190	0.51546	0.58974	0.81057	0.79812	0.38824	0.80597	0.82234
6	0.46575	0.52055	0.52941	0.31313	0.21176	0.46575	0.00000	0.15068	0.58763	0.32381	0.30928	0.38462	0.70044	0.68545	0.16471	0.64179	0.68528
7	0.58904	0.49275	0.62353	0.40404	0.24706	0.49275	0.15068	0.00000	0.63918	0.40952	0.30928	0.45299	0.74009	0.69953	0.21176	0.69154	0.65482
8	0.30928	0.39175	0.14433	0.38384	0.51546	0.47423	0.58763	0.63918	0.00000	0.36190	0.39175	0.29060	0.73128	0.77934	0.47423	0.80100	0.82234
9	0.54286	0.36190	0.07619	0.30476	0.56190	0.32381	0.40952	0.36190	0.00000	0.23810	0.15385	0.57269	0.65728	0.25714	0.64677	0.68528	0.62857
10	0.44330	0.25253	0.14433	0.51546	0.30928	0.30928	0.39175	0.23810	0.00000	0.26496	0.67401	0.61033	0.18557	0.66667	0.65990	0.63918	0.37113
11	0.20513	0.32479	0.58974	0.38462	0.45299	0.29060	0.15385	0.26496	0.00000	0.59031	0.62911	0.31624	0.64179	0.67513	0.65812	0.44444	0.52991
12	0.69163	0.81057	0.70044	0.74009	0.73128	0.57269	0.67401	0.59031	0.00000	0.12775	0.66960	0.13656	0.17621	0.84141	0.70925	0.77093	0.70044
13	0.79812	0.68545	0.69953	0.77934	0.65728	0.61033	0.62911	0.12775	0.00000	0.64789	0.08451	0.09859	0.83568	0.71362	0.73709	0.68545	0.11268
14	0.16471	0.21176	0.47423	0.25714	0.18557	0.31624	0.66960	0.64789	0.00000	0.64677	0.65990	0.54118	0.23529	0.35294	0.16471	0.66667	0.70270
15	0.69154	0.80100	0.64677	0.66667	0.64179	0.13656	0.08451	0.64677	0.00000	0.05473	0.80100	0.67662	0.72637	0.64179	0.03980	0.08955	0.75124
16	0.82234	0.68528	0.65990	0.67513	0.17621	0.09859	0.65990	0.05473	0.00000	0.84772	0.71574	0.72081	0.68528	0.06599	0.07107	0.75127	0.10152

Table F.20: Bottom-up tree edit distance values for grammar 19 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.44444	0.55556	0.77778	0.44444	0.77778	0.44444	0.44444	0.55556	0.58333	0.66667	0.66667	0.58333	0.83333	0.58333	0.58333	0.66667
1	0.44444	0.00000	0.55556	0.77778	0.44444	0.77778	0.44444	0.00000	0.55556	0.33333	0.50000	0.66667	0.41667	0.83333	0.58333	0.33333	0.66667
2	0.55556	0.55556	0.00000	0.44444	0.33333	0.44444	0.33333	0.55556	0.00000	0.66667	0.33333	0.41667	0.50000	0.58333	0.50000	0.50000	0.33333
3	0.77778	0.77778	0.44444	0.00000	0.55556	0.00000	0.55556	0.77778	0.44444	0.83333	0.58333	0.33333	0.66667	0.33333	0.50000	0.66667	0.41667
4	0.44444	0.44444	0.33333	0.55556	0.00000	0.55556	0.00000	0.44444	0.33333	0.58333	0.50000	0.50000	0.33333	0.66667	0.33333	0.41667	0.50000
5	0.77778	0.77778	0.44444	0.00000	0.55556	0.00000	0.55556	0.77778	0.44444	0.83333	0.58333	0.33333	0.66667	0.33333	0.50000	0.66667	0.41667
6	0.44444	0.44444	0.33333	0.55556	0.00000	0.55556	0.00000	0.44444	0.33333	0.58333	0.50000	0.50000	0.33333	0.66667	0.33333	0.41667	0.50000
7	0.44444	0.00000	0.55556	0.77778	0.44444	0.77778	0.44444	0.00000	0.55556	0.33333	0.50000	0.66667	0.41667	0.83333	0.58333	0.33333	0.66667
8	0.55556	0.55556	0.00000	0.44444	0.33333	0.44444	0.33333	0.55556	0.00000	0.66667	0.33333	0.41667	0.50000	0.58333	0.50000	0.50000	0.33333
9	0.33333	0.66667	0.83333	0.58333	0.83333	0.58333	0.33333	0.66667	0.00000	0.50000	0.66667	0.41667	0.83333	0.58333	0.33333	0.66667	0.83333
10	0.33333	0.58333	0.50000	0.58333	0.50000	0.50000	0.33333	0.50000	0.00000	0.41667	0.33333	0.58333	0.50000	0.33333	0.33333	0.58333	0.50000
11	0.33333	0.50000	0.33333	0.50000	0.66667	0.41667	0.66667	0.41667	0.00000	0.50000	0.33333	0.33333	0.50000	0.25000	0.33333	0.33333	0.50000
12	0.33333	0.66667	0.33333	0.41667	0.50000	0.41667	0.33333	0.50000	0.00000	0.66667	0.33333	0.25000	0.50000	0.66667	0.33333	0.25000	0.50000
13	0.33333	0.66667	0.83333	0.58333	0.83333	0.58333	0.33333	0.66667	0.00000	0.50000	0.66667	0.41667	0.00000	0.50000	0.66667	0.41667	0.83333
14	0.33333	0.58333	0.50000	0.58333	0.50000	0.33333	0.33333	0.50000	0.00000	0.41667	0.33333	0.50000	0.00000	0.41667	0.33333	0.58333	0.50000
15	0.33333	0.50000	0.33333	0.33333	0.50000	0.25000	0.66667	0.41667	0.00000	0.50000	0.66667	0.41667	0.00000	0.50000	0.33333	0.33333	0.50000
16	0.33333	0.66667	0.33333	0.25000	0.50000	0.41667	0.33333	0.50000	0.00000	0.41667	0.33333	0.50000	0.00000	0.66667	0.33333	0.25000	0.50000

Table F.21: Bottom-up tree edit distance values for grammar 20 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.11852	0.15278	0.12097	0.04032	0.11364	0.14894	0.16935	0.08871	0.08730	0.12593	0.20161	0.12097	0.09677	0.12121	0.25000	0.16935
1	0.11852	0.00000	0.07639	0.19259	0.14815	0.03704	0.07092	0.23704	0.17778	0.08148	0.02963	0.26667	0.20741	0.11111	0.05926	0.31111	0.26667
2	0.15278	0.07639	0.00000	0.25000	0.18056	0.10417	0.03472	0.29167	0.22222	0.14583	0.07639	0.31944	0.25000	0.17361	0.10417	0.36111	0.29167
3	0.12097	0.19259	0.25000	0.00000	0.09091	0.16667	0.22695	0.08929	0.06957	0.16667	0.21481	0.09821	0.03571	0.13821	0.18939	0.16964	0.11607
4	0.04032	0.14815	0.18056	0.09091	0.00000	0.12121	0.15603	0.16529	0.08264	0.11905	0.15556	0.17355	0.09091	0.08943	0.12879	0.23967	0.15702
5	0.11364	0.03704	0.10417	0.16667	0.12121	0.00000	0.07801	0.23485	0.17424	0.07576	0.05926	0.24242	0.18182	0.08333	0.03030	0.30303	0.25758
6	0.14894	0.07092	0.03472	0.22695	0.15603	0.07801	0.00000	0.29078	0.21986	0.14184	0.07092	0.29787	0.22695	0.14894	0.07801	0.35461	0.28369
7	0.16935	0.23704	0.29167	0.08929	0.16529	0.23485	0.29078	0.00000	0.09565	0.17460	0.23704	0.04717	0.08929	0.17073	0.23485	0.15094	0.09434
8	0.08871	0.17778	0.22222	0.06957	0.08264	0.17424	0.21986	0.09565	0.00000	0.12698	0.16296	0.13043	0.04348	0.12195	0.15909	0.20870	0.13913
9	0.08148	0.14583	0.16667	0.11905	0.07576	0.14184	0.17460	0.12698	0.00000	0.08148	0.20635	0.15873	0.03968	0.07576	0.29365	0.24603	0.12698
10	0.07639	0.21481	0.15556	0.05926	0.07092	0.23704	0.16296	0.08148	0.00000	0.26667	0.19259	0.11111	0.03704	0.33333	0.27407	0.17778	0.11852
11	0.09821	0.17355	0.24242	0.29787	0.04717	0.13043	0.20635	0.26667	0.00000	0.09821	0.17886	0.24242	0.13592	0.11429	0.17241	0.24000	0.15534
12	0.09091	0.18182	0.22695	0.08929	0.04348	0.15873	0.19259	0.09821	0.00000	0.13008	0.16667	0.19643	0.12500	0.10345	0.16000	0.21429	0.14286
13	0.08333	0.14894	0.17073	0.12195	0.03968	0.11111	0.17886	0.13008	0.00000	0.08333	0.28455	0.23577	0.11382	0.09600	0.30081	0.25203	0.13008
14	0.07801	0.23485	0.15909	0.07576	0.03704	0.24242	0.16667	0.08333	0.00000	0.32576	0.26515	0.16667	0.10606	0.34091	0.28030	0.18182	0.12121
15	0.15094	0.20870	0.29365	0.33333	0.13592	0.19643	0.28455	0.32576	0.00000	0.10476	0.18966	0.25600	0.05208	0.09804	0.18584	0.25410	0.11458
16	0.13913	0.24603	0.27407	0.11429	0.12500	0.23577	0.26515	0.10476	0.00000	0.13793	0.17600	0.14286	0.04762	0.13274	0.17213	0.20000	0.10476

Table F.22: Bottom-up tree edit distance values for grammar 21 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.00568	0.03295	0.00568	0.03295	0.03636	0.06364	0.00682	0.03295	0.03750	0.06477	0.03750	0.06477	0.06818	0.09545	0.00341	0.03295
1	0.00568	0.00000	0.03295	0.00568	0.03636	0.03295	0.06364	0.00341	0.03409	0.03295	0.06477	0.03409	0.06477	0.06477	0.09545	0.00682	0.03750
2	0.03295	0.03295	0.00000	0.03636	0.00586	0.00586	0.03400	0.03409	0.00352	0.00703	0.03400	0.00703	0.03517	0.03869	0.06682	0.03409	0.00703
3	0.00568	0.00568	0.03636	0.00000	0.03295	0.03295	0.06364	0.00682	0.03750	0.03750	0.06818	0.03295	0.06477	0.06477	0.09545	0.00682	0.03750
4	0.03295	0.03636	0.00586	0.03295	0.00000	0.00586	0.03400	0.03750	0.00703	0.01055	0.03869	0.00703	0.03400	0.03869	0.06682	0.03409	0.00703
5	0.03636	0.03295	0.00586	0.03295	0.00586	0.00000	0.03400	0.03409	0.00703	0.00703	0.03869	0.00352	0.03517	0.03400	0.06682	0.03750	0.01055
6	0.06364	0.06364	0.03400	0.06364	0.03400	0.03400	0.00000	0.06477	0.03517	0.03869	0.00726	0.03517	0.00363	0.00726	0.03511	0.06477	0.03869
7	0.00682	0.00341	0.03409	0.00682	0.03750	0.03409	0.06477	0.00000	0.03295	0.03295	0.06364	0.03295	0.06364	0.06364	0.09432	0.00568	0.03750
8	0.03295	0.03409	0.00352	0.03750	0.00703	0.00703	0.03517	0.03295	0.00000	0.00586	0.03400	0.00586	0.03400	0.03751	0.06565	0.03409	0.00586
9	0.03295	0.00703	0.03750	0.01055	0.00703	0.03869	0.03295	0.00586	0.00000	0.03400	0.00586	0.03751	0.03400	0.06565	0.03750	0.01055	0.00586
10	0.03400	0.06818	0.03869	0.03869	0.00726	0.06364	0.03400	0.03400	0.00000	0.03751	0.00605	0.00605	0.03511	0.06477	0.03869	0.03517	0.00605
11	0.03295	0.00703	0.00352	0.03517	0.03295	0.00586	0.00586	0.03751	0.00000	0.03400	0.03400	0.06565	0.03750	0.01055	0.00703	0.03869	0.00586
12	0.03400	0.03517	0.00363	0.06364	0.03400	0.03751	0.00605	0.03400	0.00000	0.00605	0.03511	0.06477	0.03869	0.03517	0.00726	0.03517	0.00605
13	0.03400	0.00726	0.06364	0.03751	0.03400	0.00605	0.03400	0.00605	0.00000	0.03511	0.06818	0.04220	0.03869	0.01090	0.03869	0.01090	0.00605
14	0.03511	0.09432	0.06565	0.06565	0.03511	0.06565	0.03511	0.03511	0.00000	0.09545	0.07034	0.06682	0.03995	0.06682	0.03995	0.03632	0.00626
15	0.00568	0.03409	0.03750	0.06477	0.03750	0.06477	0.06818	0.09545	0.00000	0.03295	0.03295	0.06364	0.03295	0.06364	0.06364	0.09432	0.03295
16	0.00586	0.01055	0.03869	0.01055	0.03869	0.04220	0.07034	0.03295	0.00000	0.00586	0.03400	0.00586	0.03400	0.03751	0.06565	0.00703	0.03400

Table F.23: Bottom-up tree edit distance values for grammar 22 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.00980	0.02941	0.00980	0.02941	0.02941	0.04902	0.01471	0.02941	0.03431	0.05392	0.03431	0.05392	0.05392	0.07353	0.01471	0.02941
1	0.00980	0.00000	0.02941	0.00980	0.02941	0.02941	0.04902	0.01471	0.03431	0.02941	0.05392	0.03431	0.05392	0.05392	0.07353	0.01471	0.03431
2	0.02941	0.02941	0.00000	0.02941	0.00980	0.00980	0.02941	0.03431	0.01471	0.01471	0.02941	0.01471	0.03431	0.03431	0.05392	0.03431	0.01471
3	0.00980	0.00980	0.02941	0.00000	0.02941	0.02941	0.04902	0.01471	0.03431	0.03431	0.05392	0.02941	0.05392	0.05392	0.07353	0.01471	0.03431
4	0.02941	0.02941	0.00980	0.02941	0.00000	0.00980	0.02941	0.03431	0.01471	0.01471	0.03431	0.01471	0.02941	0.03431	0.05392	0.03431	0.01471
5	0.02941	0.02941	0.00980	0.02941	0.00980	0.00000	0.02941	0.03431	0.01471	0.01471	0.03431	0.01471	0.03431	0.02941	0.05392	0.03431	0.01471
6	0.04902	0.04902	0.02941	0.04902	0.02941	0.02941	0.00000	0.05392	0.03431	0.03431	0.01471	0.03431	0.01471	0.01471	0.02941	0.05392	0.03431
7	0.01471	0.01471	0.03431	0.01471	0.03431	0.03431	0.05392	0.00000	0.02941	0.02941	0.04902	0.02941	0.04902	0.04902	0.06863	0.00980	0.03431
8	0.02941	0.03431	0.01471	0.03431	0.01471	0.01471	0.03431	0.02941	0.00000	0.00980	0.02941	0.00980	0.02941	0.02941	0.04902	0.03431	0.00980
9	0.02941	0.01471	0.03431	0.01471	0.01471	0.03431	0.02941	0.00980	0.00000	0.02941	0.00980	0.02941	0.02941	0.04902	0.03431	0.01471	0.00980
10	0.02941	0.05392	0.03431	0.03431	0.01471	0.04902	0.02941	0.02941	0.00000	0.02941	0.00980	0.00980	0.02941	0.05392	0.03431	0.03431	0.00980
11	0.02941	0.01471	0.01471	0.03431	0.02941	0.00980	0.00980	0.02941	0.00000	0.02941	0.02941	0.04902	0.03431	0.01471	0.01471	0.03431	0.00980
12	0.02941	0.03431	0.01471	0.04902	0.02941	0.02941	0.00980	0.02941	0.00000	0.00980	0.02941	0.05392	0.03431	0.03431	0.01471	0.03431	0.00980
13	0.02941	0.01471	0.04902	0.02941	0.02941	0.00980	0.02941	0.00980	0.00000	0.02941	0.05392	0.03431	0.03431	0.01471	0.03431	0.01471	0.00980
14	0.02941	0.06863	0.04902	0.04902	0.02941	0.04902	0.02941	0.02941	0.00000	0.07353	0.05392	0.05392	0.03431	0.05392	0.03431	0.03431	0.00980
15	0.00980	0.03431	0.03431	0.05392	0.03431	0.05392	0.05392	0.07353	0.00000	0.02941	0.02941	0.04902	0.02941	0.04902	0.04902	0.06863	0.02941
16	0.00980	0.01471	0.03431	0.01471	0.03431	0.03431	0.05392	0.02941	0.00000	0.00980	0.02941	0.00980	0.02941	0.02941	0.04902	0.01471	0.02941

Table F.24: Bottom-up tree edit distance values for grammar 23 for pass-images 0 to 16

No.	**Image generation sequence**																
	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
0	0.00000	0.17143	0.22727	0.05714	0.29897	0.26136	0.41739	0.17143	0.22727	0.16456	0.36792	0.25000	0.41739	0.38679	0.50376	0.27848	0.35849
1	0.17143	0.00000	0.32955	0.17143	0.41237	0.32955	0.49565	0.06557	0.35227	0.25316	0.45283	0.35227	0.51304	0.45283	0.57143	0.34177	0.46226
2	0.22727	0.32955	0.00000	0.26136	0.12371	0.04545	0.25217	0.35227	0.04545	0.13636	0.18868	0.06818	0.26957	0.20755	0.36090	0.22727	0.25472
3	0.05714	0.17143	0.26136	0.00000	0.29897	0.22727	0.41739	0.18571	0.26136	0.17722	0.39623	0.22727	0.41739	0.36792	0.50376	0.27848	0.38679
4	0.29897	0.41237	0.12371	0.29897	0.00000	0.12371	0.17391	0.41237	0.13402	0.23711	0.13208	0.12371	0.17391	0.12264	0.29323	0.26804	0.20755
5	0.26136	0.32955	0.04545	0.22727	0.12371	0.00000	0.25217	0.36364	0.07955	0.14773	0.21698	0.04545	0.26957	0.18868	0.36090	0.22727	0.28302
6	0.41739	0.49565	0.25217	0.41739	0.17391	0.25217	0.00000	0.51304	0.27826	0.34783	0.11304	0.26957	0.03478	0.10435	0.15038	0.39130	0.17391
7	0.17143	0.06557	0.35227	0.18571	0.41237	0.36364	0.51304	0.00000	0.32955	0.25316	0.45283	0.32955	0.49565	0.45283	0.57143	0.34177	0.46226
8	0.22727	0.35227	0.04545	0.26136	0.13402	0.07955	0.27826	0.32955	0.00000	0.13636	0.18868	0.04545	0.25217	0.21698	0.36090	0.22727	0.25472
9	0.25316	0.13636	0.17722	0.23711	0.14773	0.34783	0.25316	0.13636	0.00000	0.27358	0.13636	0.34783	0.27358	0.42857	0.20253	0.32075	0.27835
10	0.18868	0.39623	0.13208	0.21698	0.11304	0.45283	0.18868	0.27358	0.00000	0.21698	0.10435	0.03774	0.21805	0.33962	0.15094	0.18868	0.21774
11	0.22727	0.12371	0.04545	0.26957	0.32955	0.04545	0.13636	0.21698	0.00000	0.25217	0.18868	0.36090	0.22727	0.27358	0.24742	0.34677	0.25472
12	0.17391	0.26957	0.03478	0.49565	0.25217	0.34783	0.10435	0.25217	0.00000	0.10435	0.15038	0.39130	0.17391	0.24348	0.19355	0.17391	0.20301
13	0.18868	0.10435	0.45283	0.21698	0.27358	0.03774	0.18868	0.10435	0.00000	0.21805	0.33962	0.16981	0.18868	0.23387	0.15094	0.25564	0.21774
14	0.15038	0.57143	0.36090	0.42857	0.21805	0.36090	0.15038	0.21805	0.00000	0.48120	0.27068	0.33835	0.15038	0.27068	0.12030	0.15038	0.17881
15	0.34177	0.22727	0.20253	0.33962	0.22727	0.39130	0.33962	0.48120	0.00000	0.27358	0.20619	0.38710	0.27358	0.42857	0.38710	0.50331	0.20619
16	0.25472	0.32075	0.15094	0.27358	0.17391	0.16981	0.27068	0.27358	0.00000	0.11321	0.16129	0.03774	0.21805	0.18548	0.31788	0.11321	0.16129

F.2 Similarity threshold and bottom-up tree edit distance values for grammars

F.2. SIMILARITY THRESHOLD AND BOTTOM-UP TREE EDIT DISTANCE VALUES FOR GRAMMARS

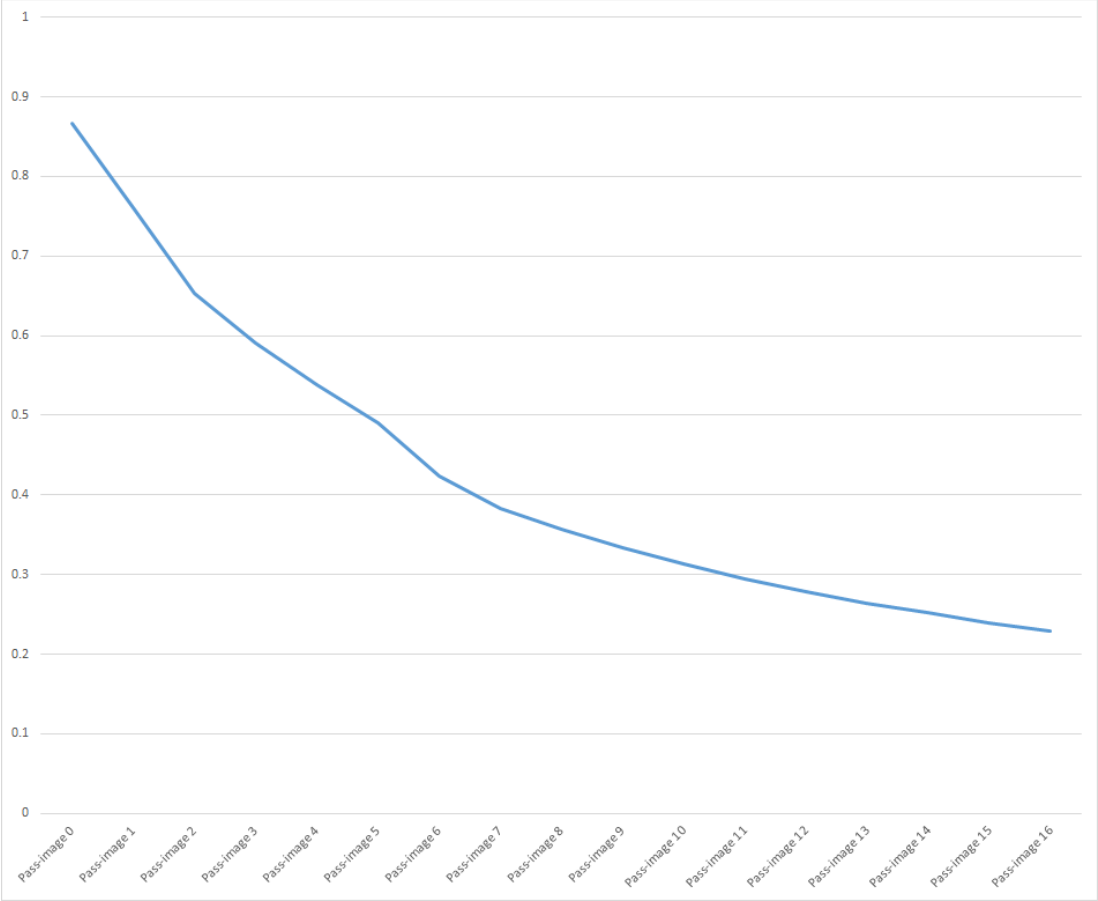


Figure F.1: Similarity threshold values for grammar 0

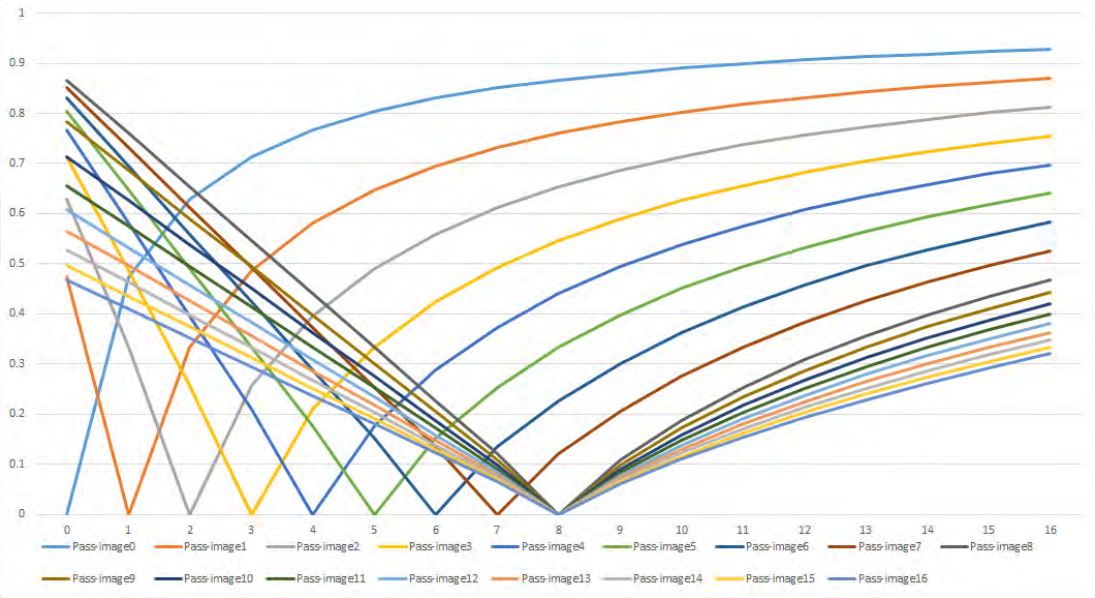


Figure F.2: Bottom-up tree edit distance values for grammar 0 for pass-images 0 to 16

F.2. SIMILARITY THRESHOLD AND BOTTOM-UP TREE EDIT DISTANCE VALUES FOR GRAMMARS

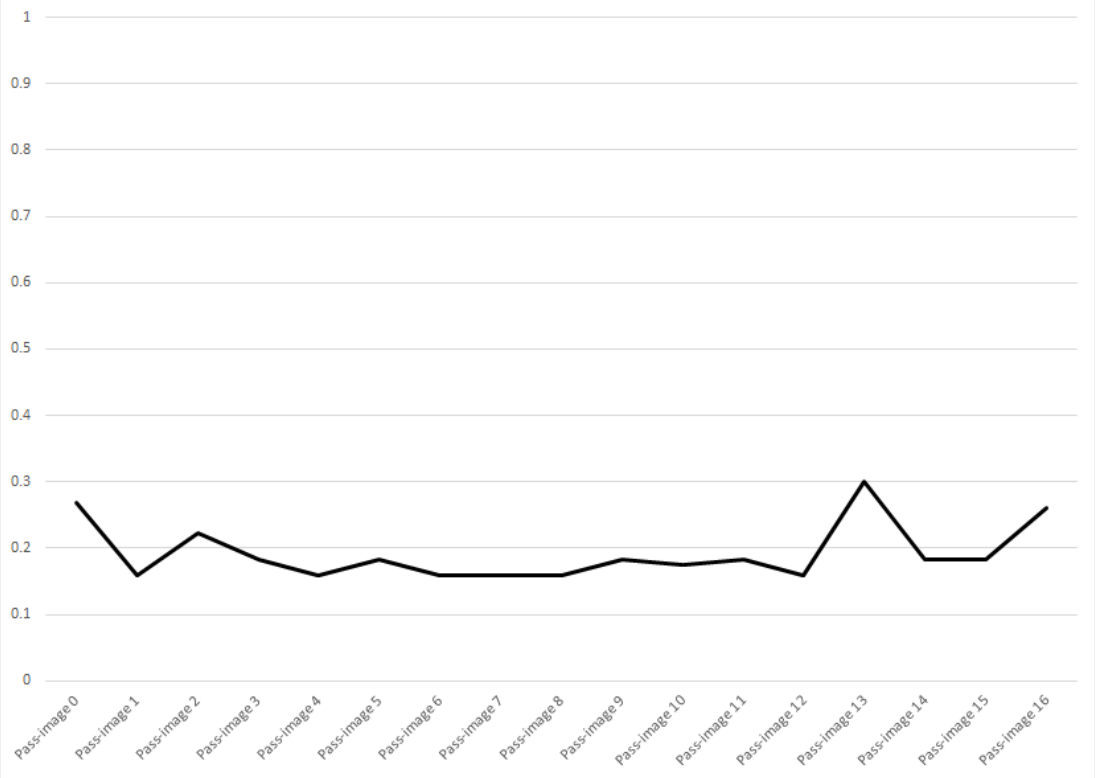


Figure F.3: Similarity threshold values for grammar 1

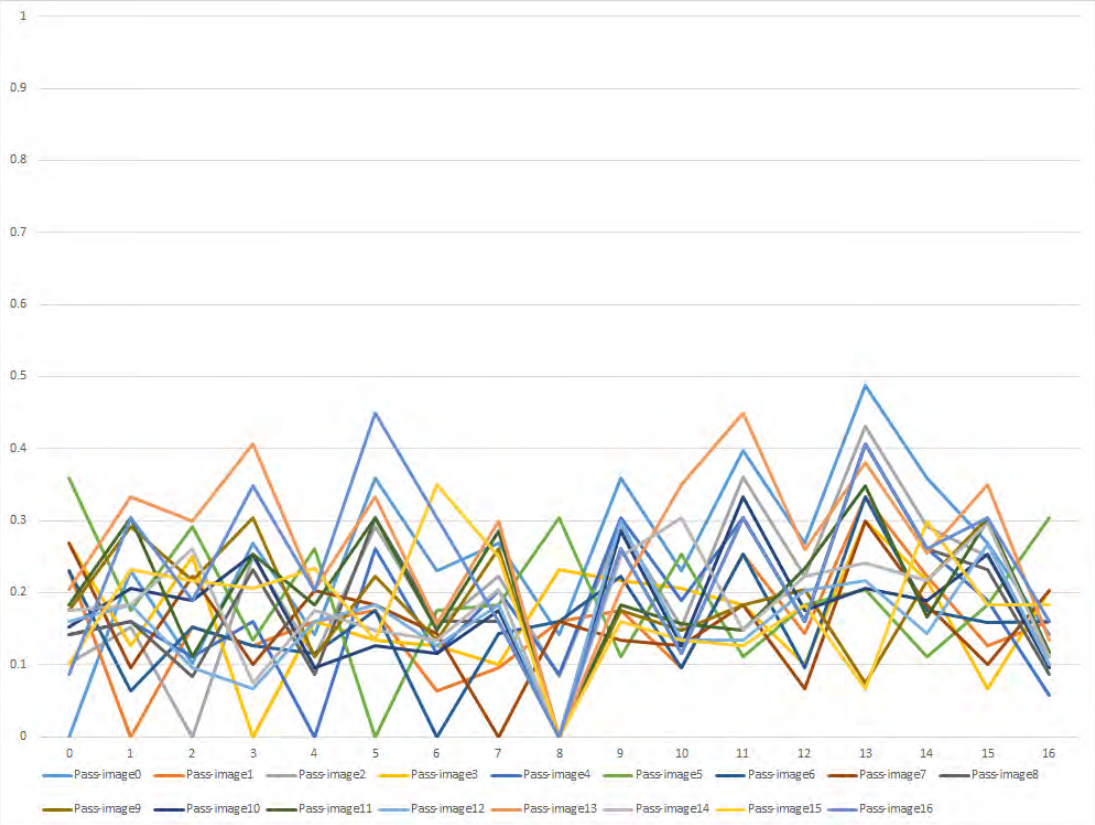


Figure F.4: Bottom-up tree edit distance values for grammar 1 for pass-images 0 to 16

F.2. SIMILARITY THRESHOLD AND BOTTOM-UP TREE EDIT DISTANCE VALUES FOR GRAMMARS

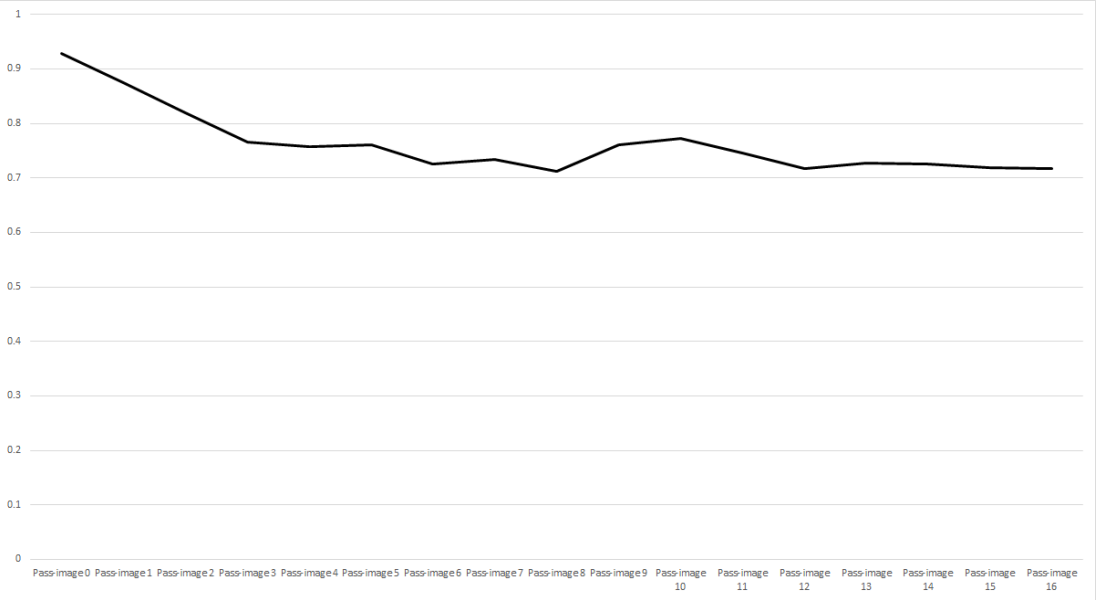


Figure F.5: Similarity threshold values for grammar 2

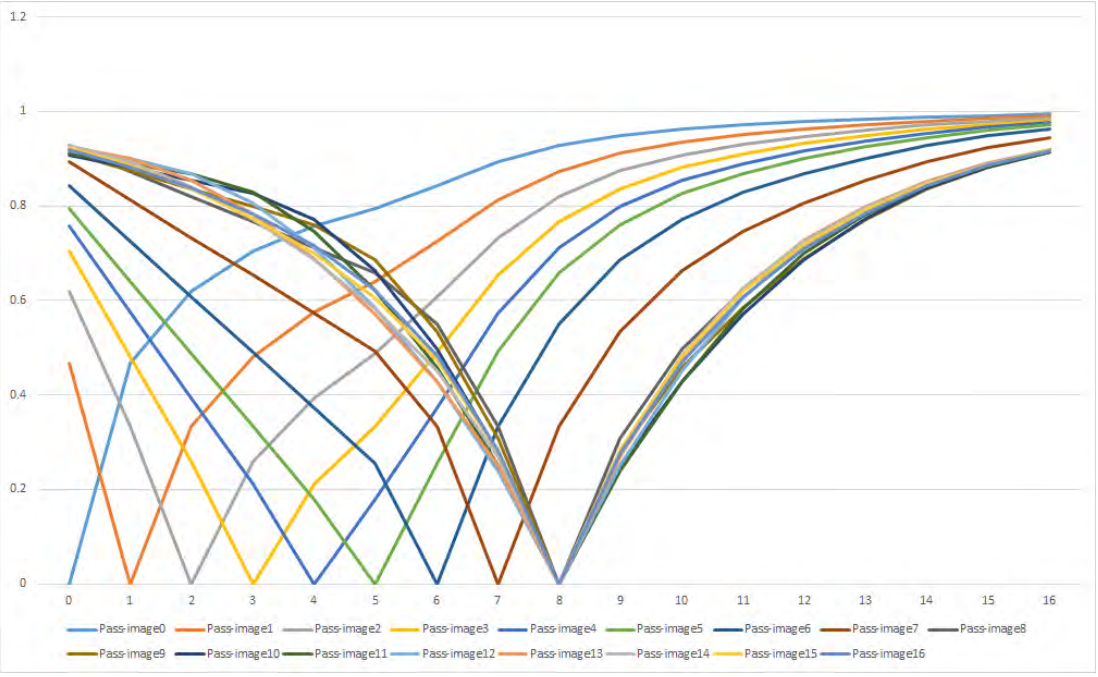


Figure F.6: Bottom-up tree edit distance values for grammar 2 for pass-images 0 to 16

F.2. SIMILARITY THRESHOLD AND BOTTOM-UP TREE EDIT DISTANCE VALUES FOR GRAMMARS

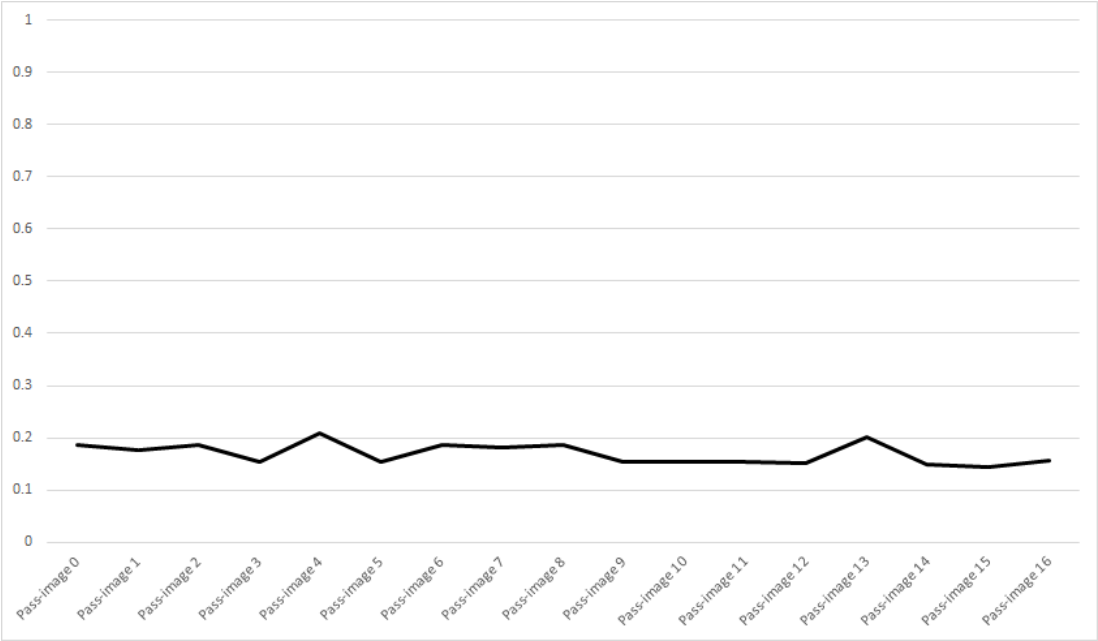


Figure F.7: Similarity threshold values for grammar 3

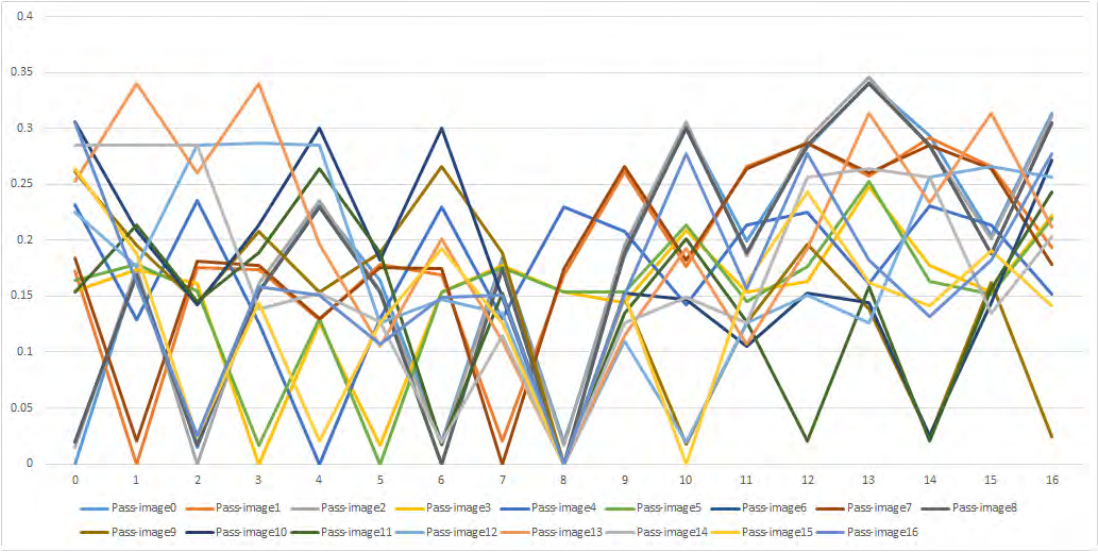


Figure F.8: Bottom-up tree edit distance values for grammar 3 for pass-images 0 to 16

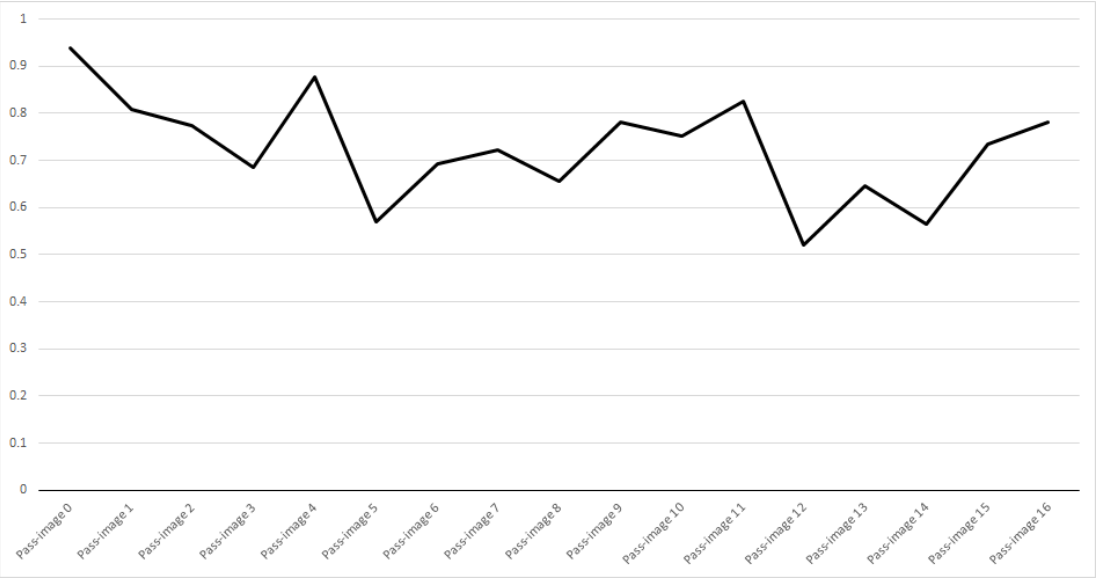


Figure F.9: Similarity threshold values for grammar 4

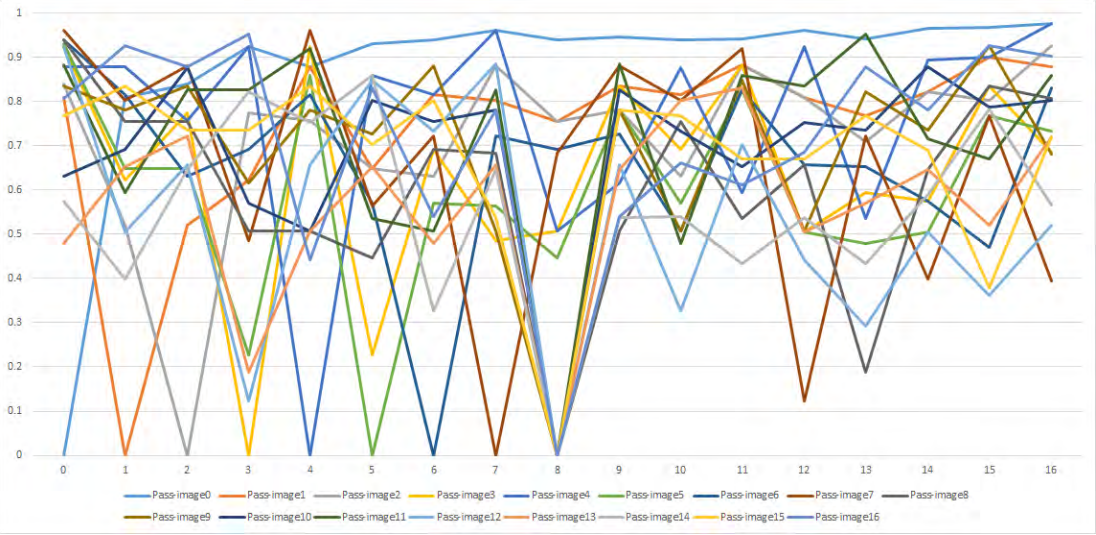


Figure F.10: Bottom-up tree edit distance values for grammar 4 for pass-images 0 to 16

F.2. SIMILARITY THRESHOLD AND BOTTOM-UP TREE EDIT DISTANCE VALUES FOR GRAMMARS

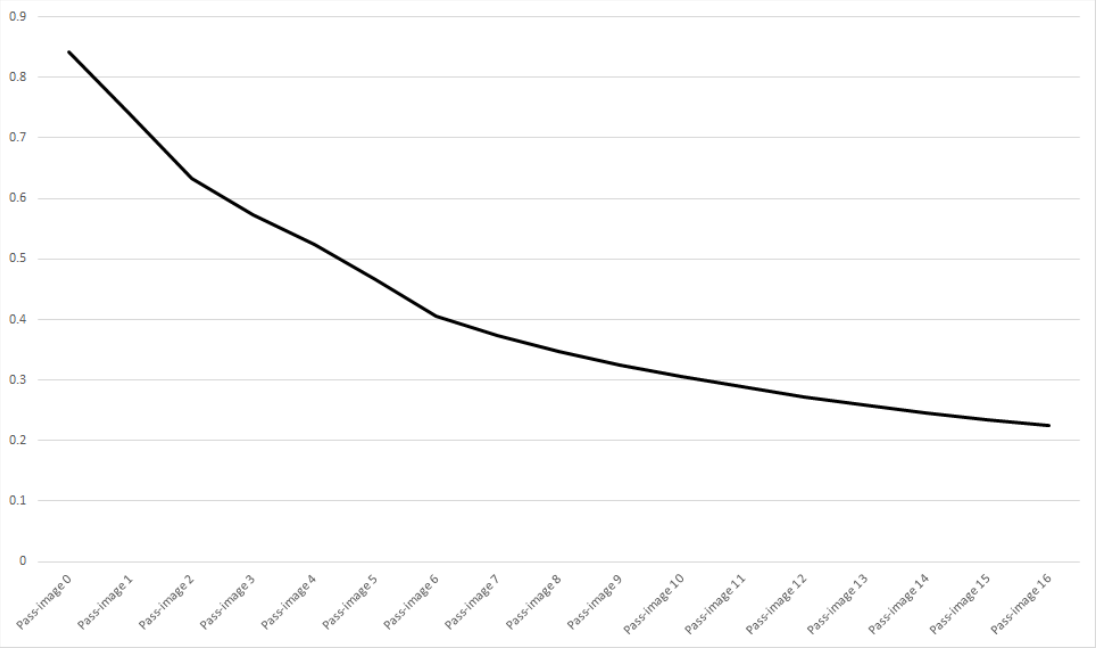


Figure F.11: Similarity threshold values for grammar 5

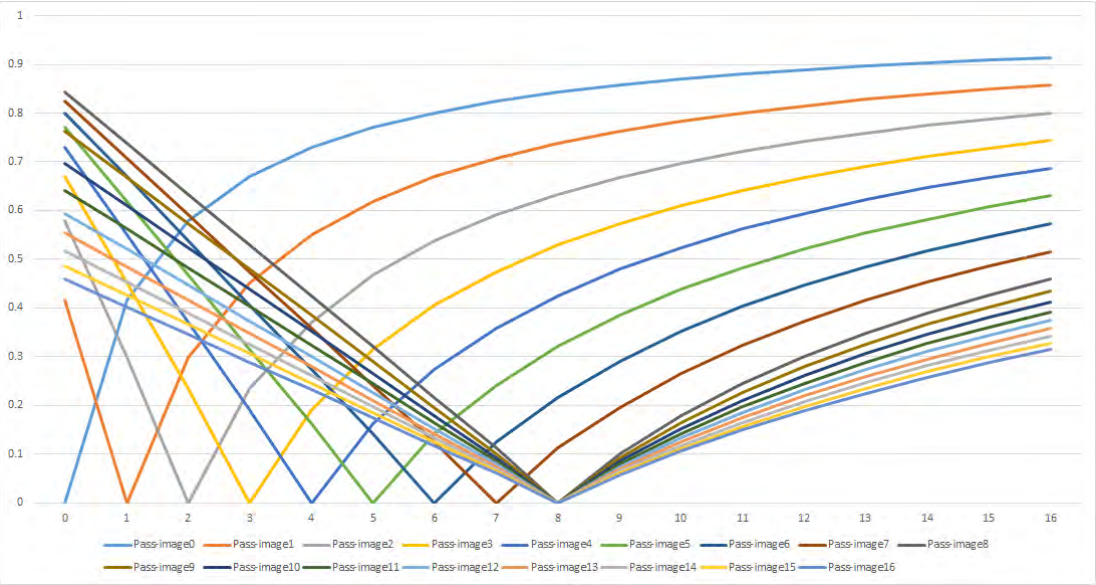


Figure F.12: Bottom-up tree edit distance values for grammar 5 for pass-images 0 to 16

F.2. SIMILARITY THRESHOLD AND BOTTOM-UP TREE EDIT DISTANCE VALUES FOR GRAMMARS

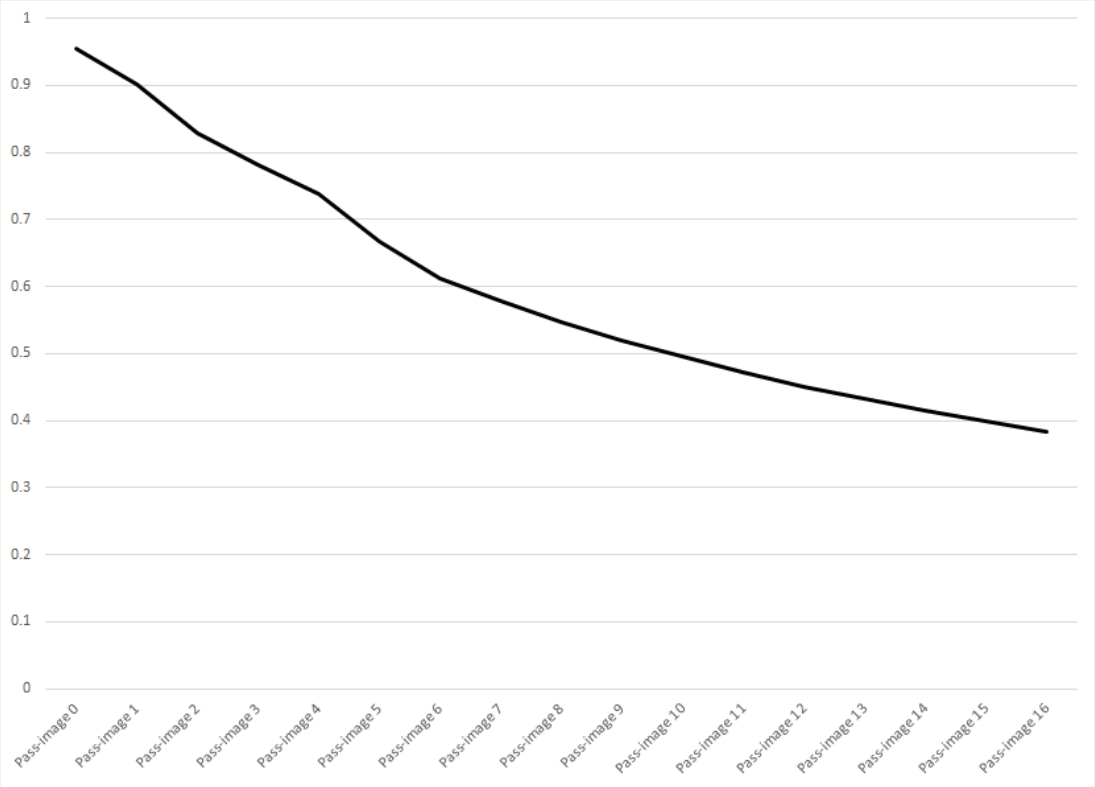


Figure F.13: Similarity threshold values for grammar 6

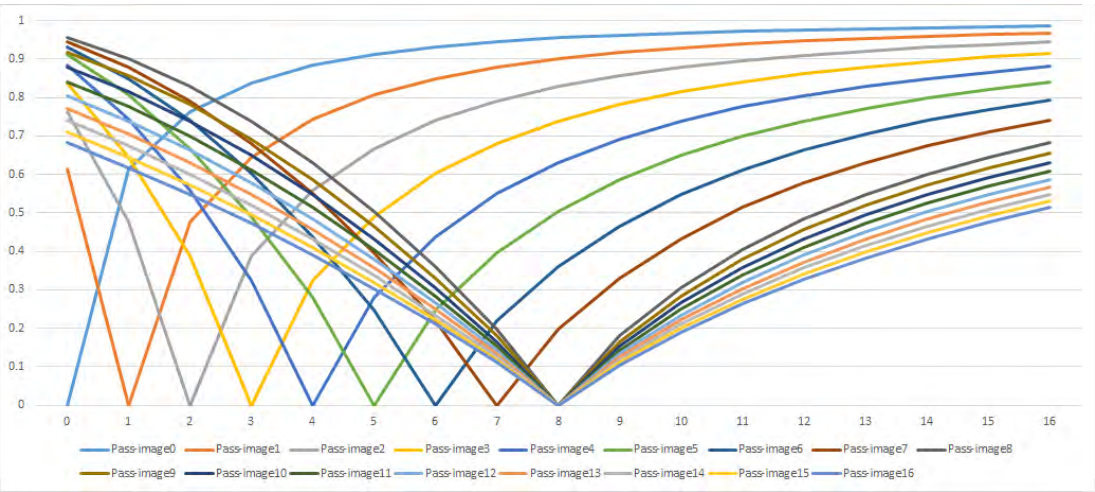


Figure F.14: Bottom-up tree edit distance values for grammar 6 for pass-images 0 to 16

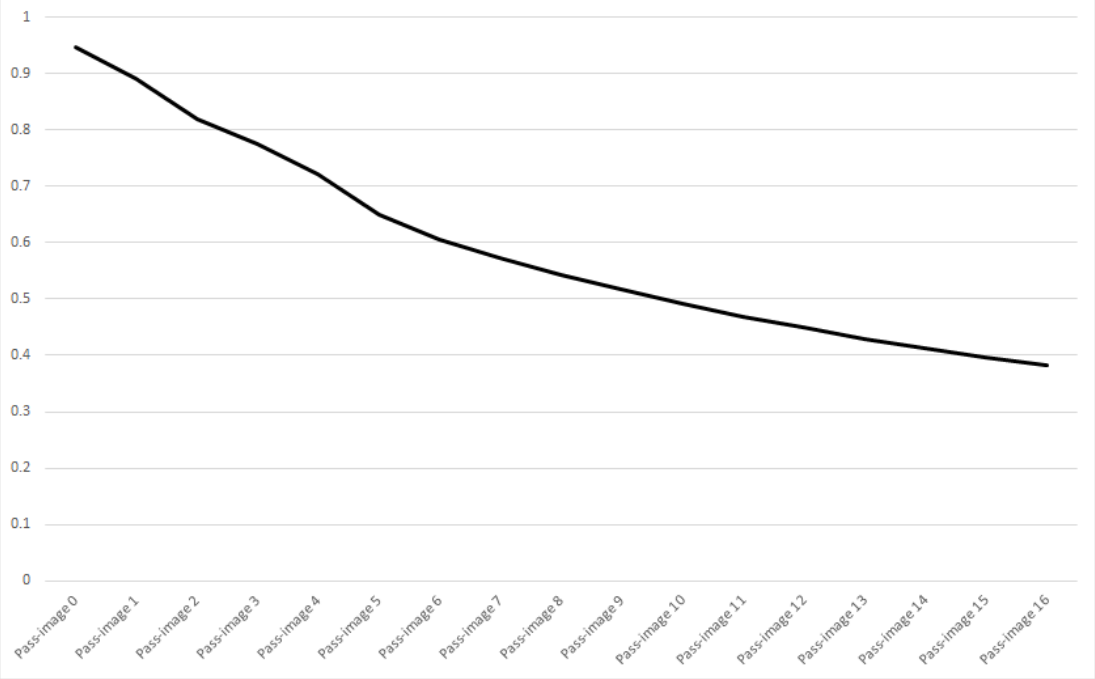


Figure F.15: Similarity threshold values for grammar 7

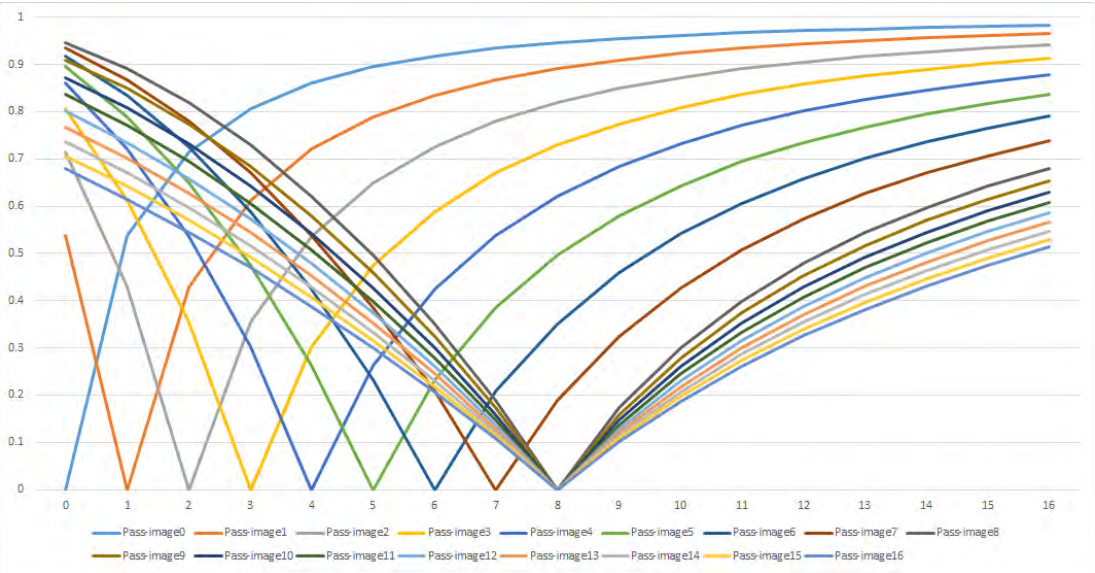


Figure F.16: Bottom-up tree edit distance values for grammar 7 for pass-images 0 to 16

F.2. SIMILARITY THRESHOLD AND BOTTOM-UP TREE EDIT DISTANCE VALUES FOR GRAMMARS

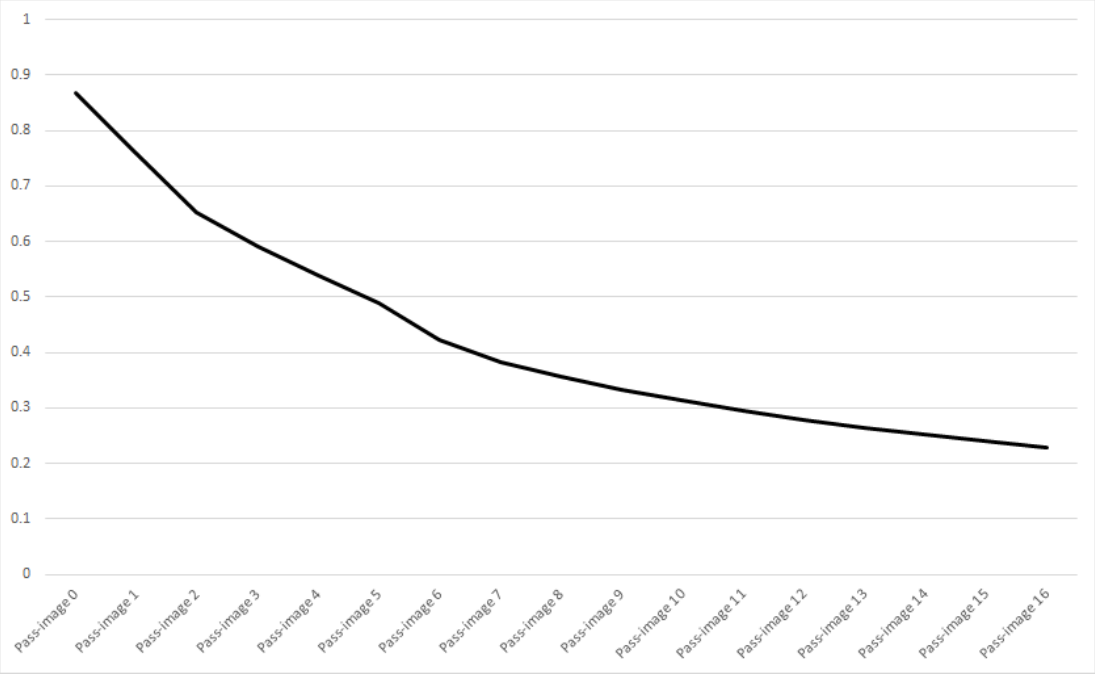


Figure F.17: Similarity threshold values for grammar 8

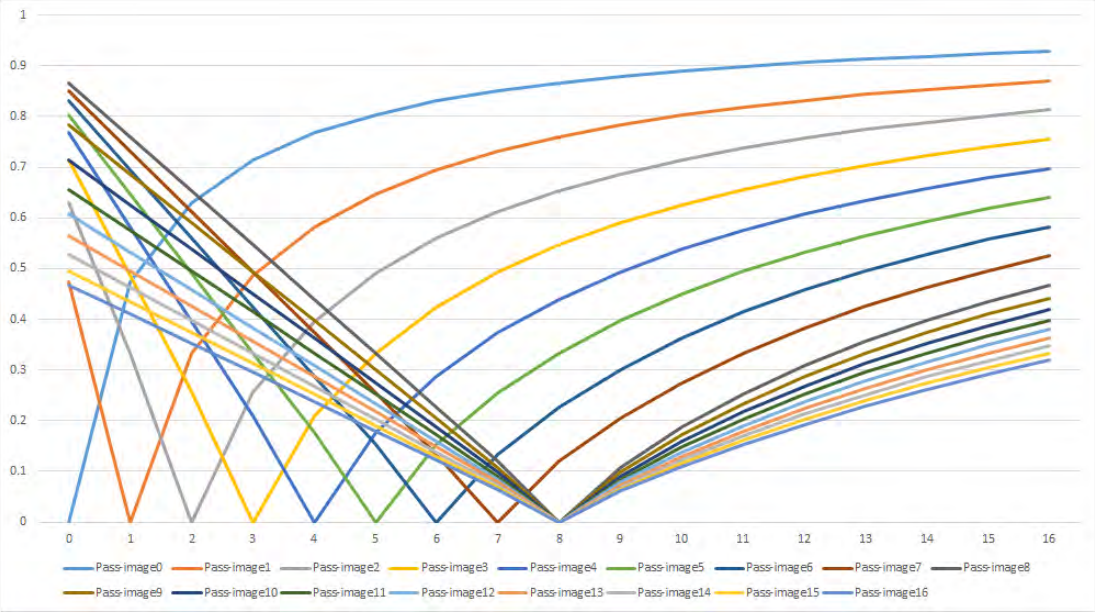


Figure F.18: Bottom-up tree edit distance values for grammar 8 for pass-images 0 to 16

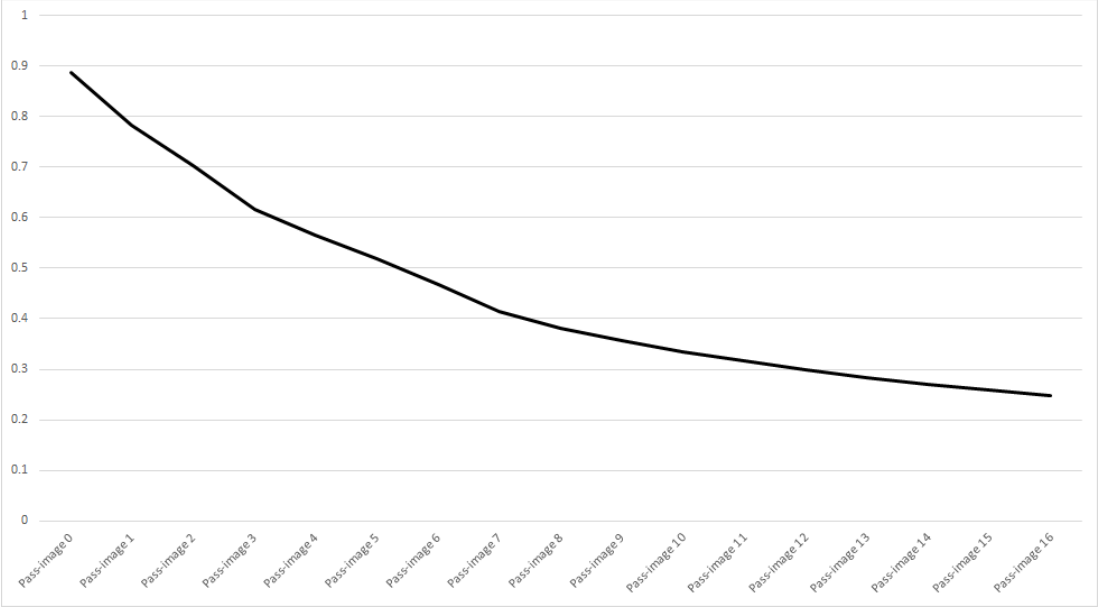


Figure F.19: Similarity threshold values for grammar 9

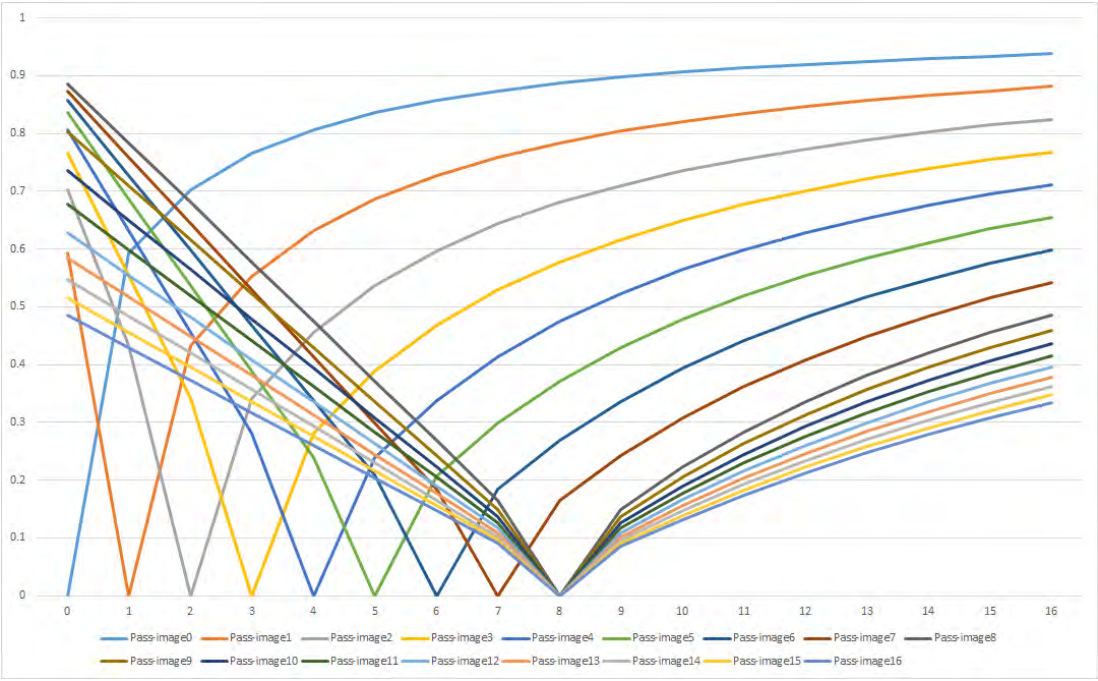


Figure F.20: Bottom-up tree edit distance values for grammar 9 for pass-images 0 to 16

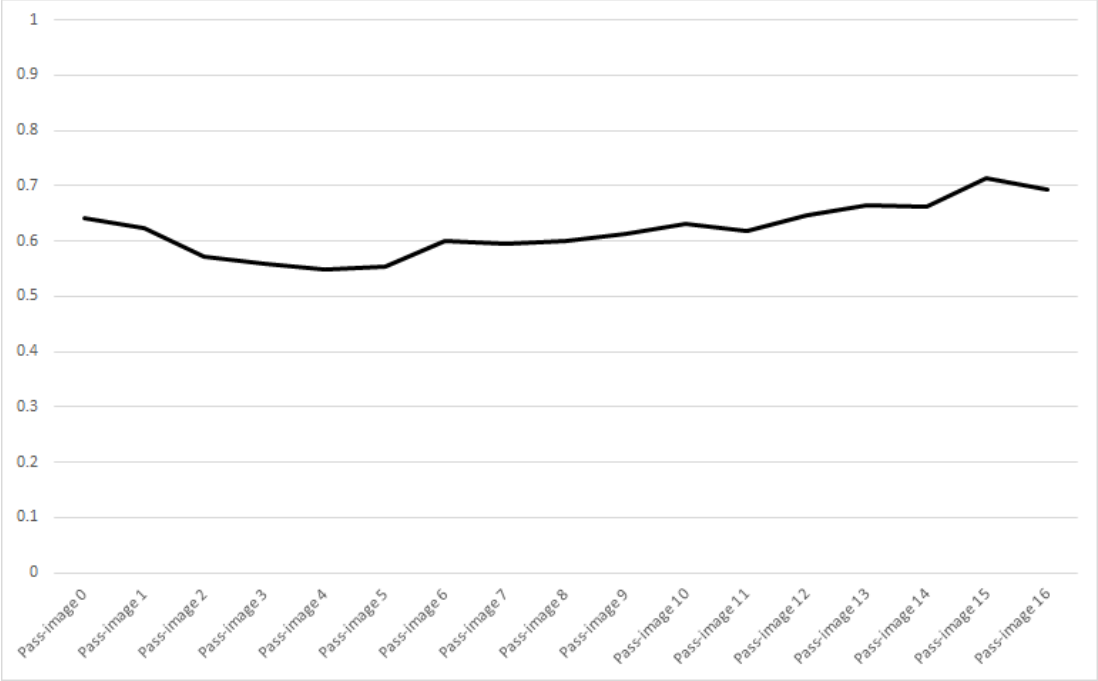


Figure F.21: Similarity threshold values for grammar 10

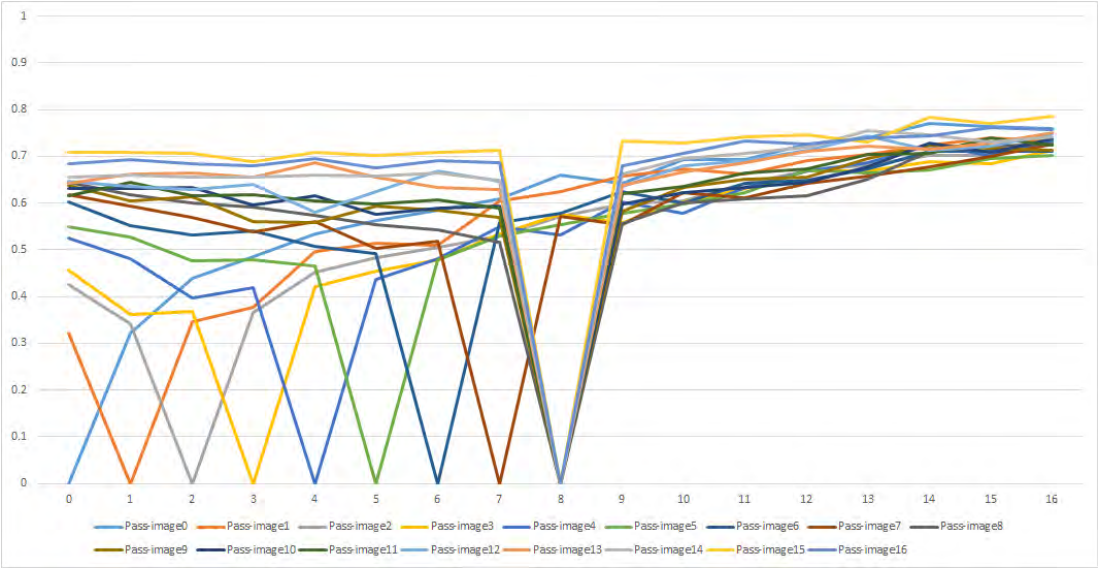


Figure F.22: Bottom-up tree edit distance values for grammar 10 for pass-images 0 to 16

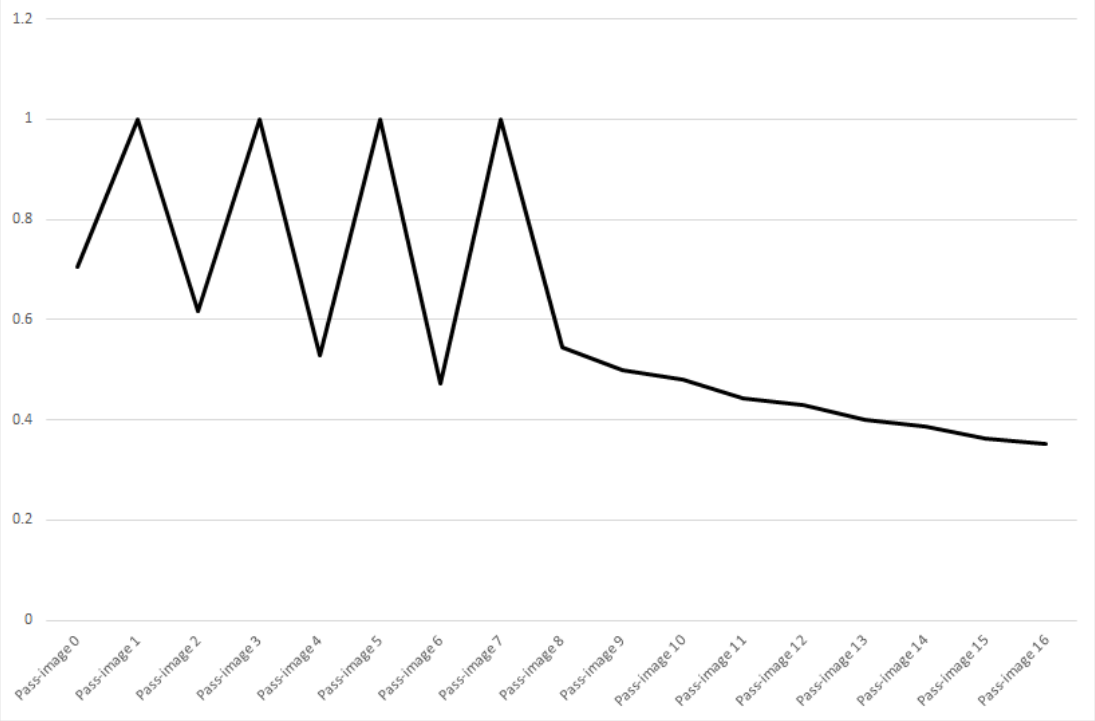


Figure F.23: Similarity threshold values for grammar 11

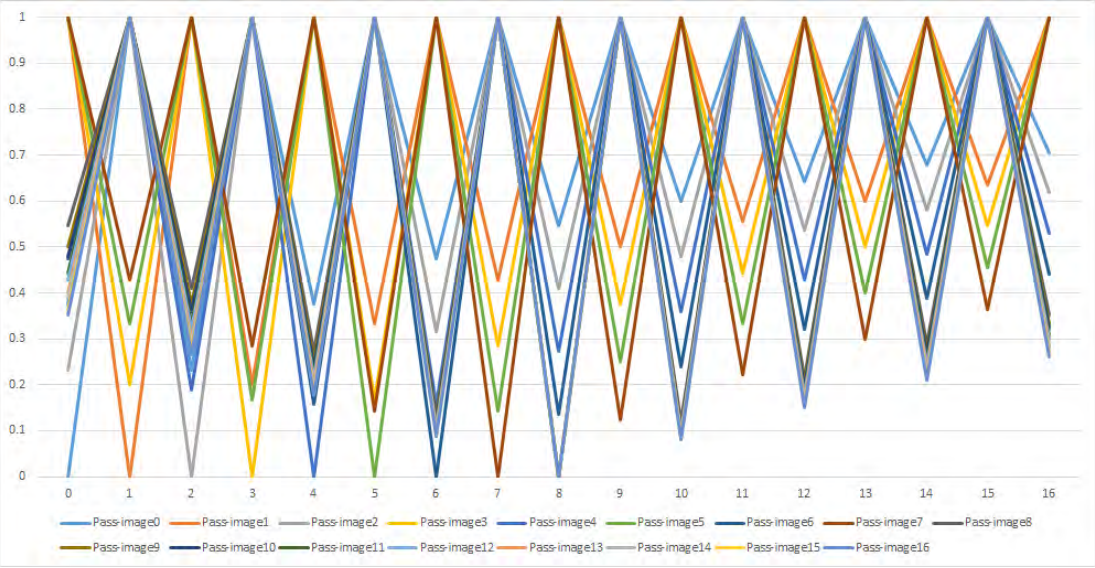


Figure F.24: Bottom-up tree edit distance values for grammar 11 for pass-images 0 to 16

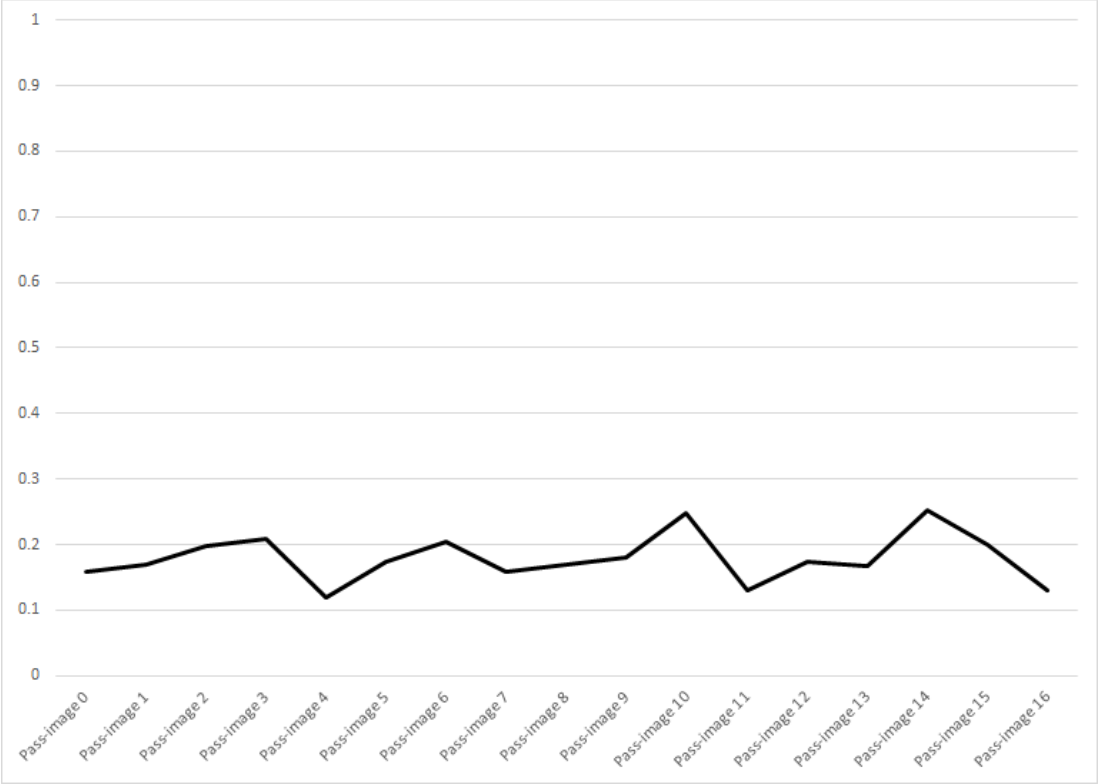


Figure F.25: Similarity threshold values for grammar 12

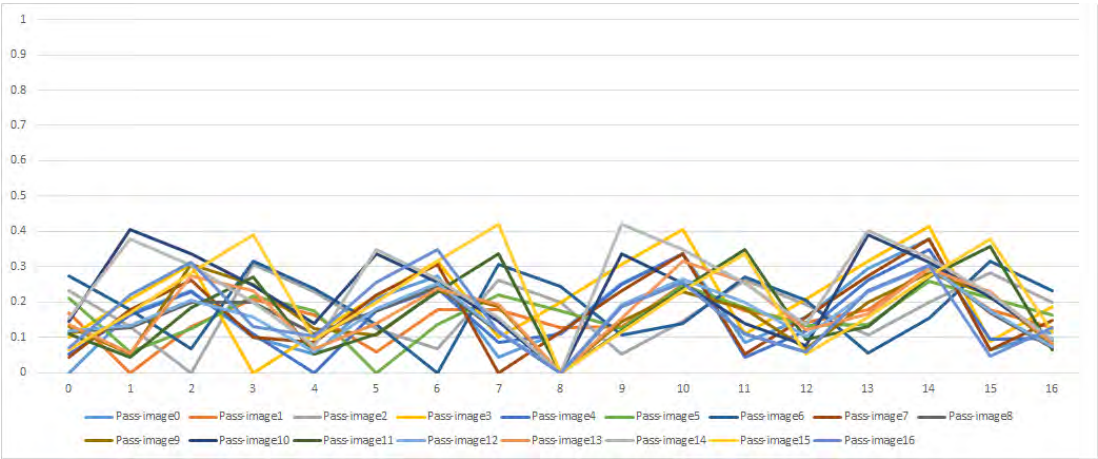


Figure F.26: Bottom-up tree edit distance values for grammar 12 for pass-images 0 to 16

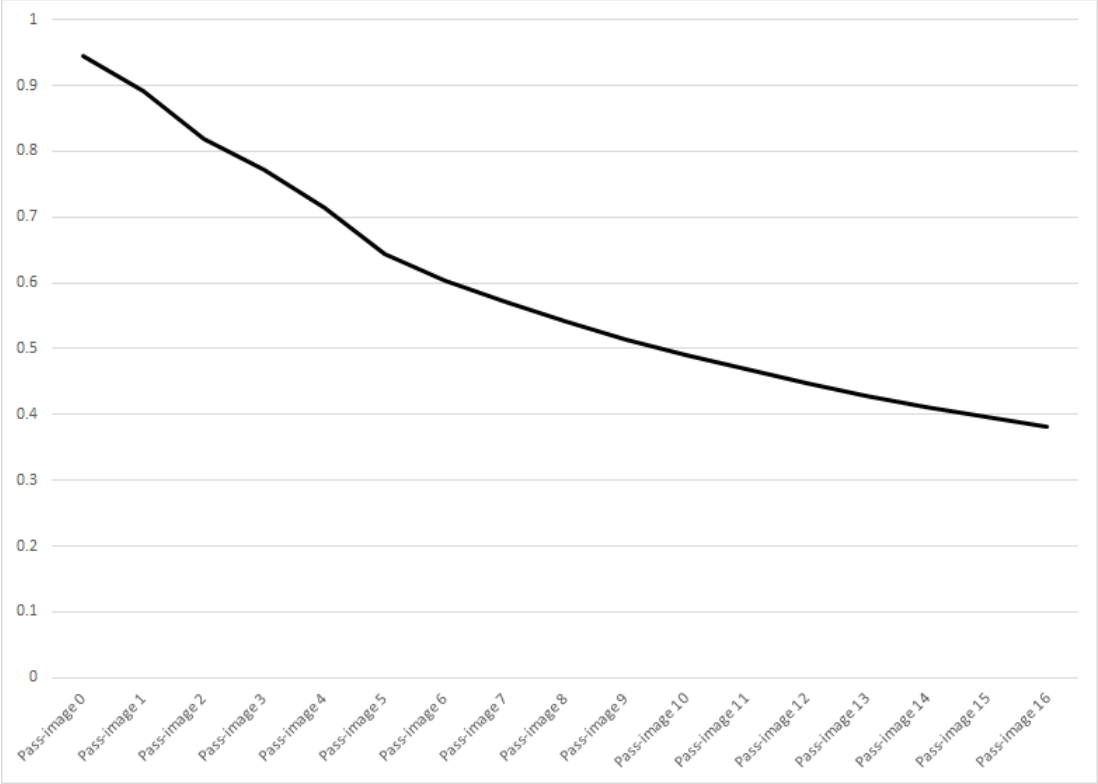


Figure F.27: Similarity threshold values for grammar 13

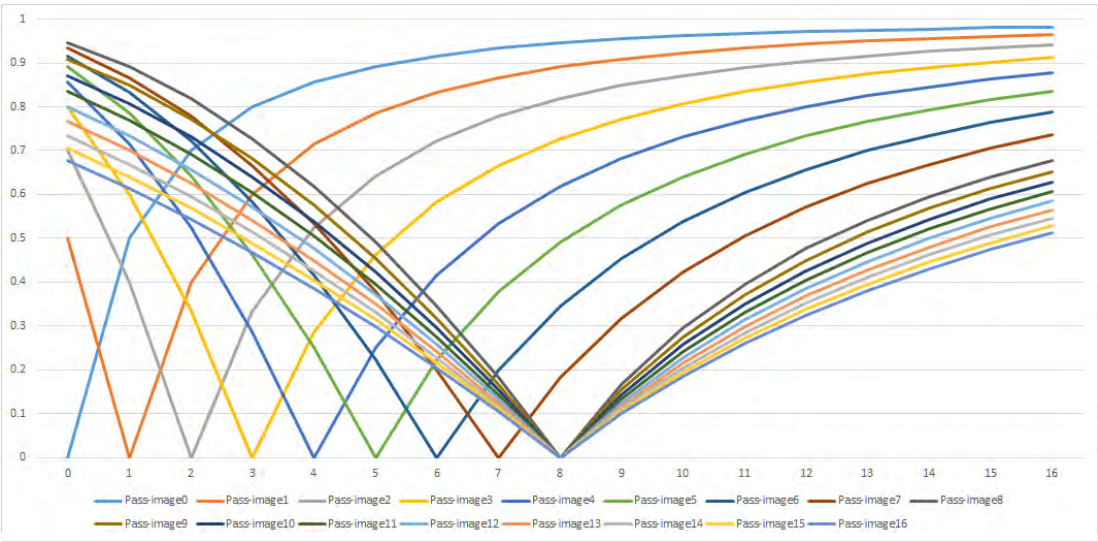


Figure F.28: Bottom-up tree edit distance values for grammar 13 for pass-images 0 to 16

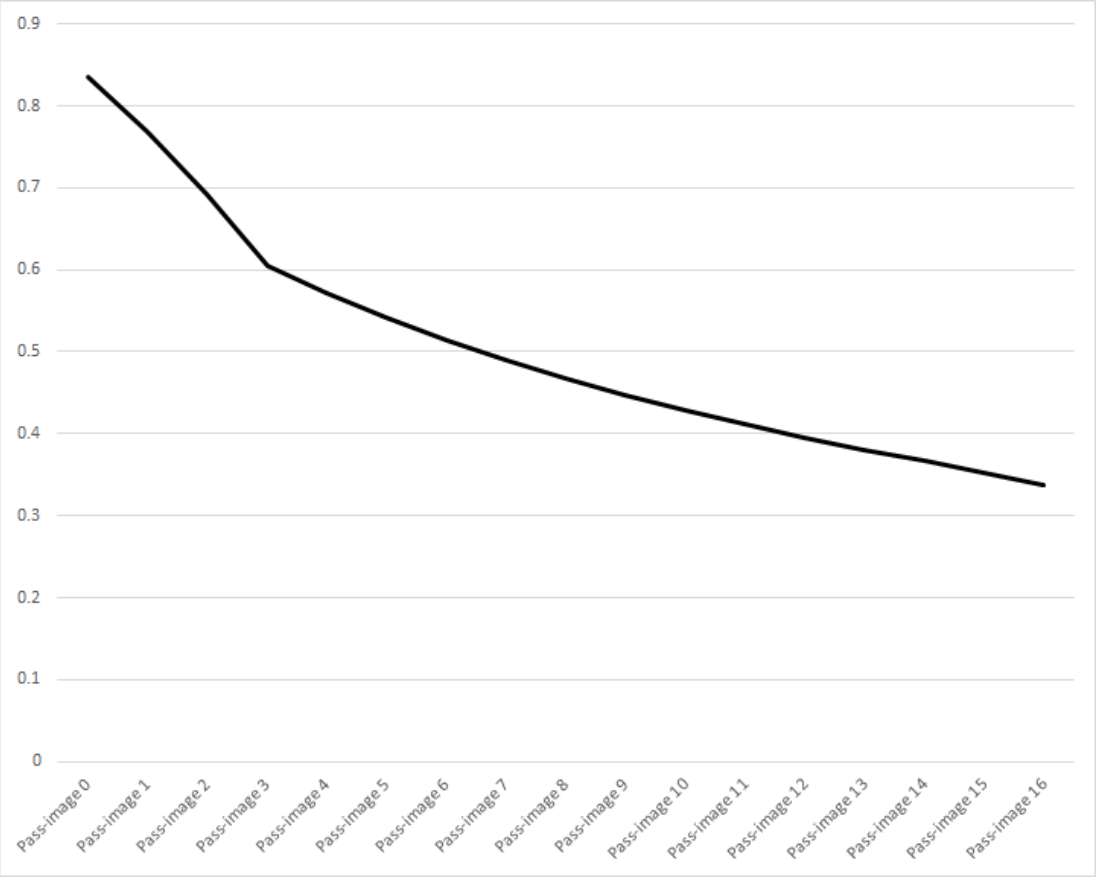


Figure F.29: Similarity threshold values for grammar 14

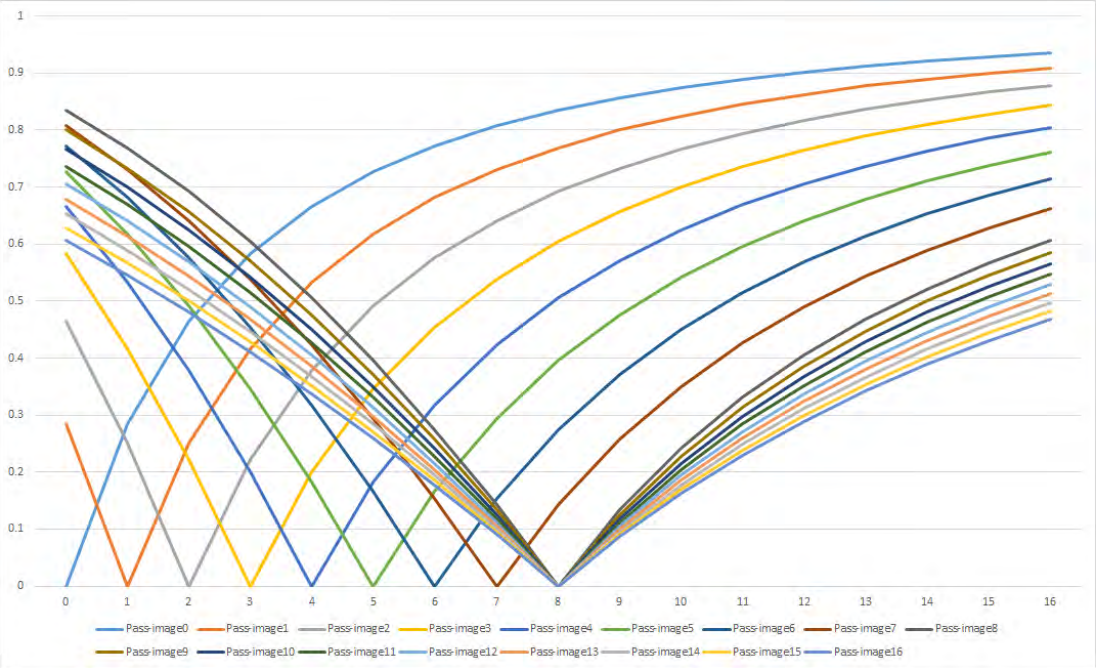


Figure F.30: Bottom-up tree edit distance values for grammar 14 for pass-images 0 to 16

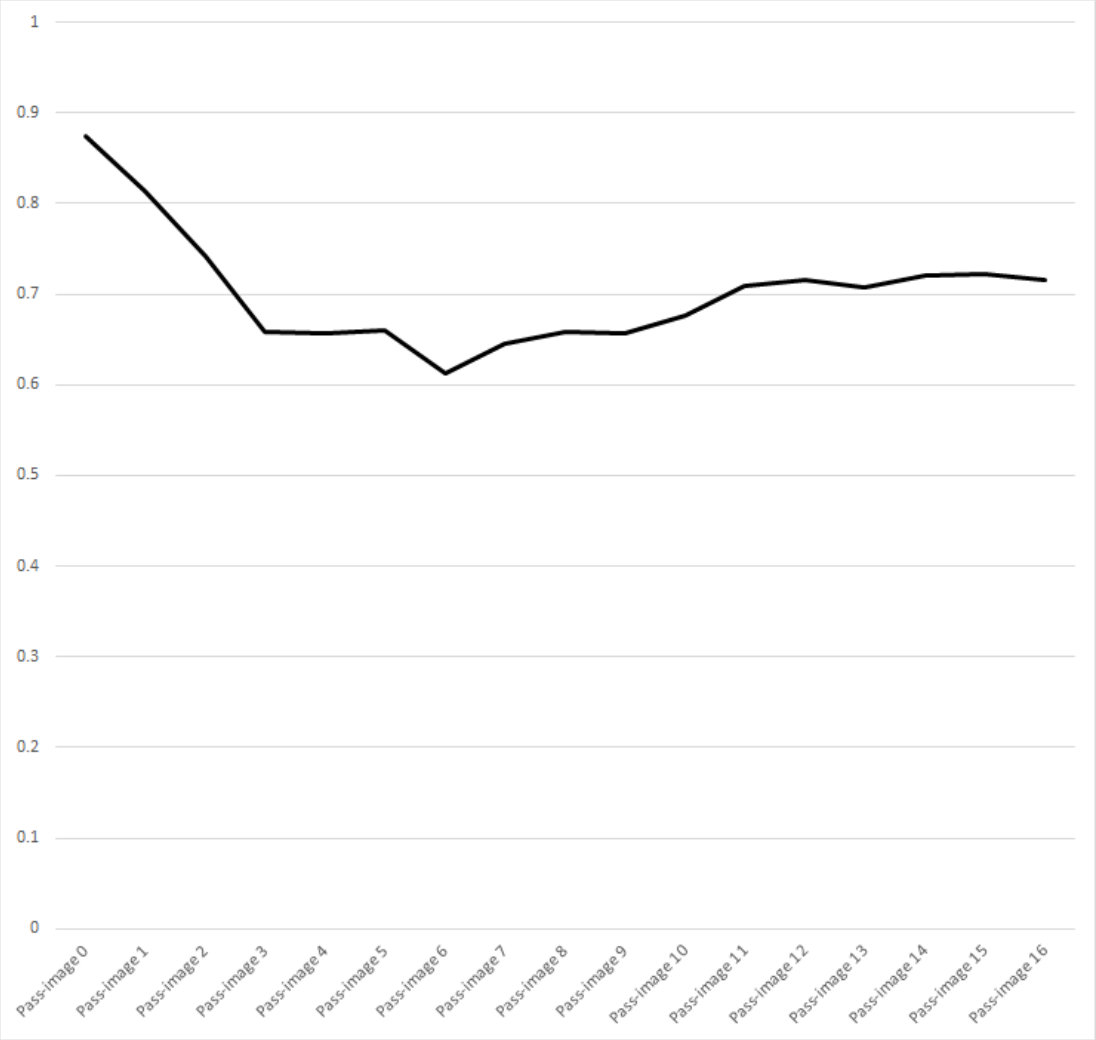


Figure F.31: Similarity threshold values for grammar 15

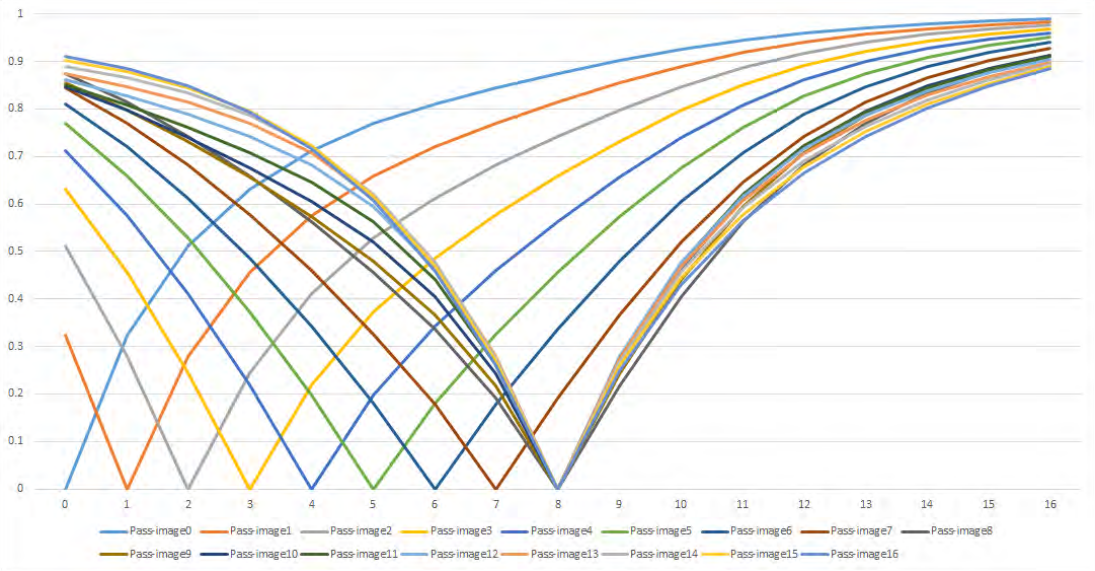


Figure F.32: Bottom-up tree edit distance values for grammar 15 for pass-images 0 to 16

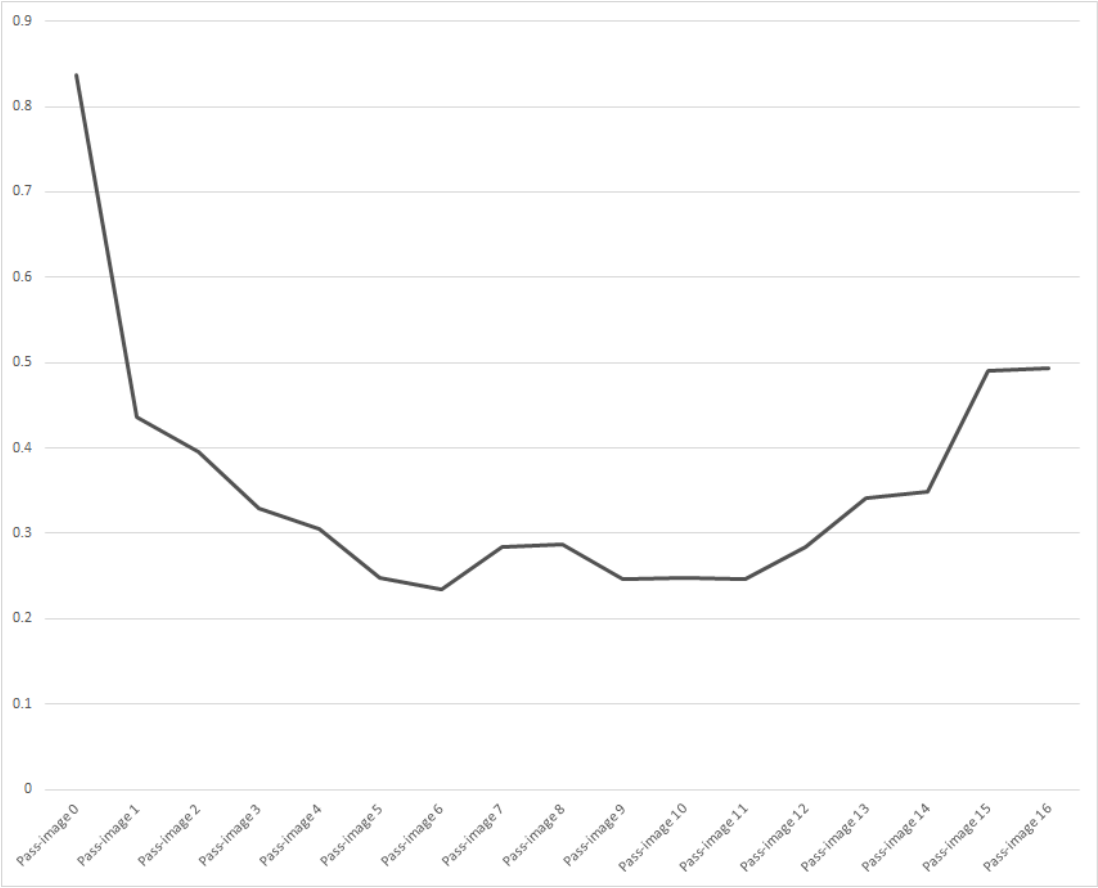


Figure F.33: Similarity threshold values for grammar 16

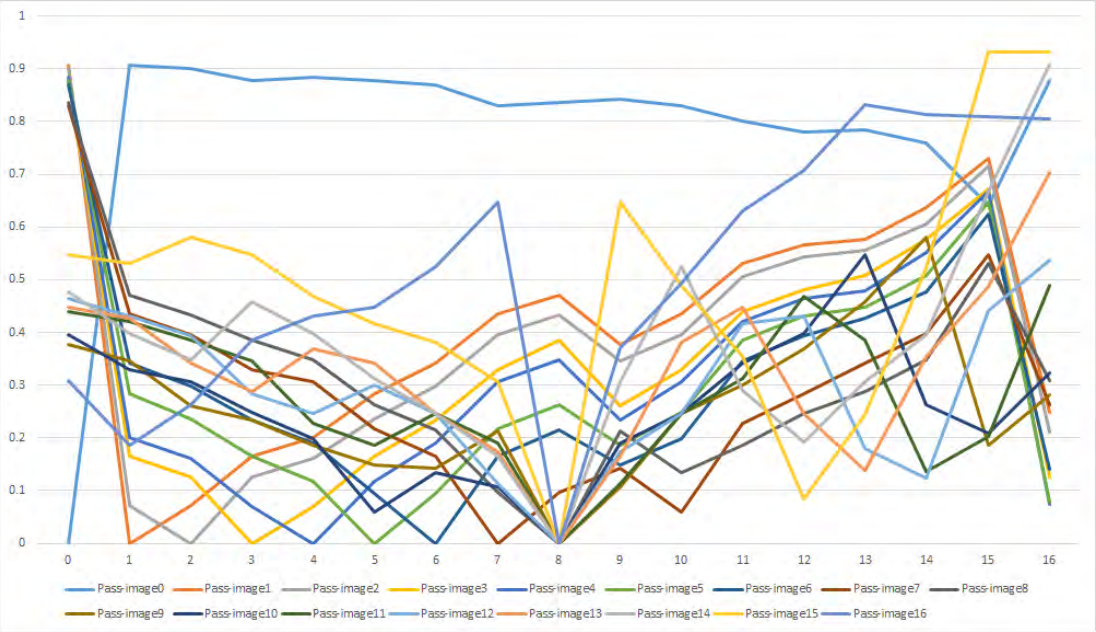


Figure F.34: Bottom-up tree edit distance values for grammar 16 for pass-images 0 to 16

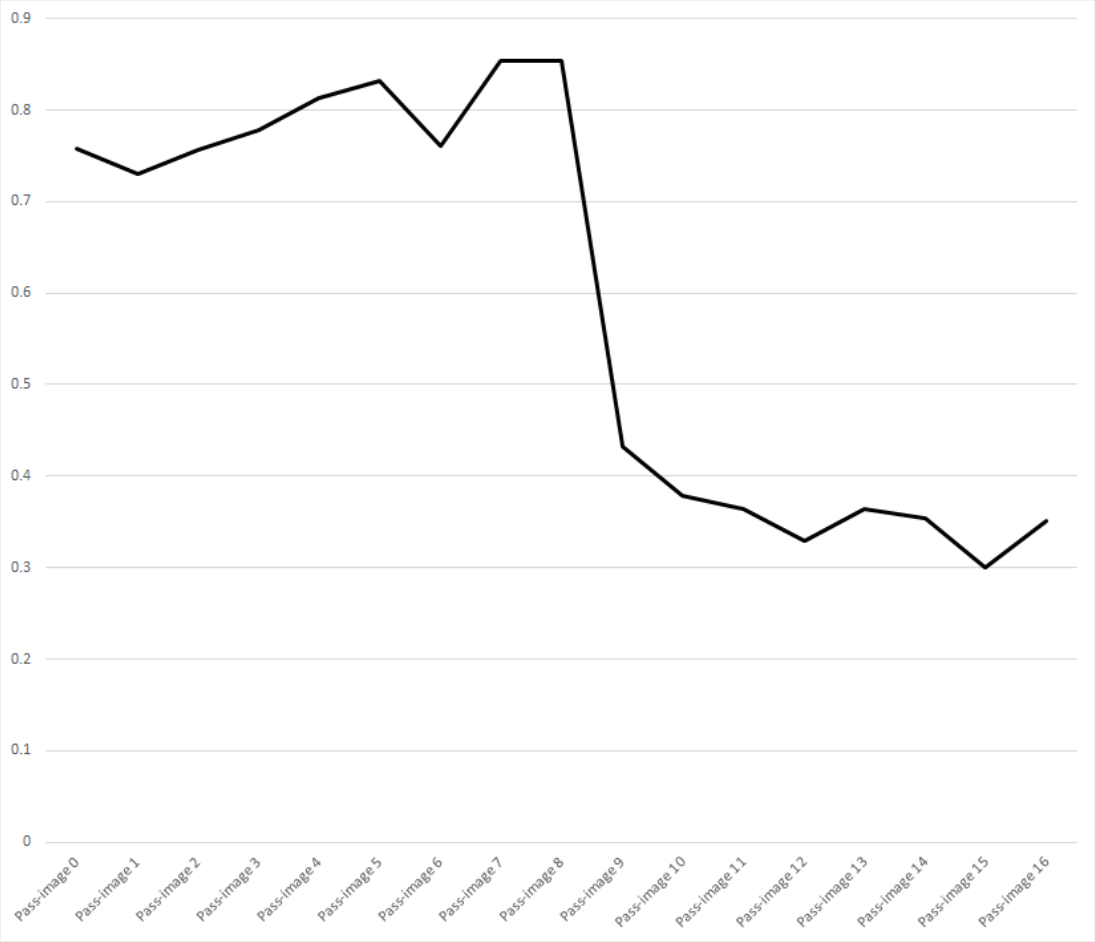


Figure F.35: Similarity threshold values for grammar 17

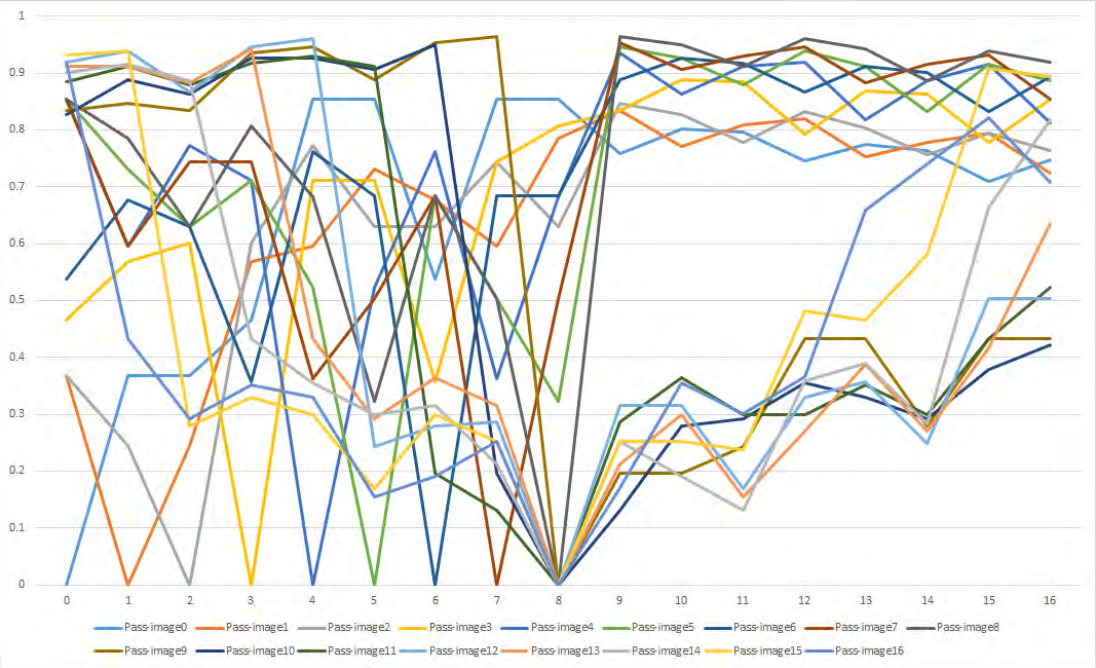


Figure F.36: Bottom-up tree edit distance values for grammar 17 for pass-images 0 to 16

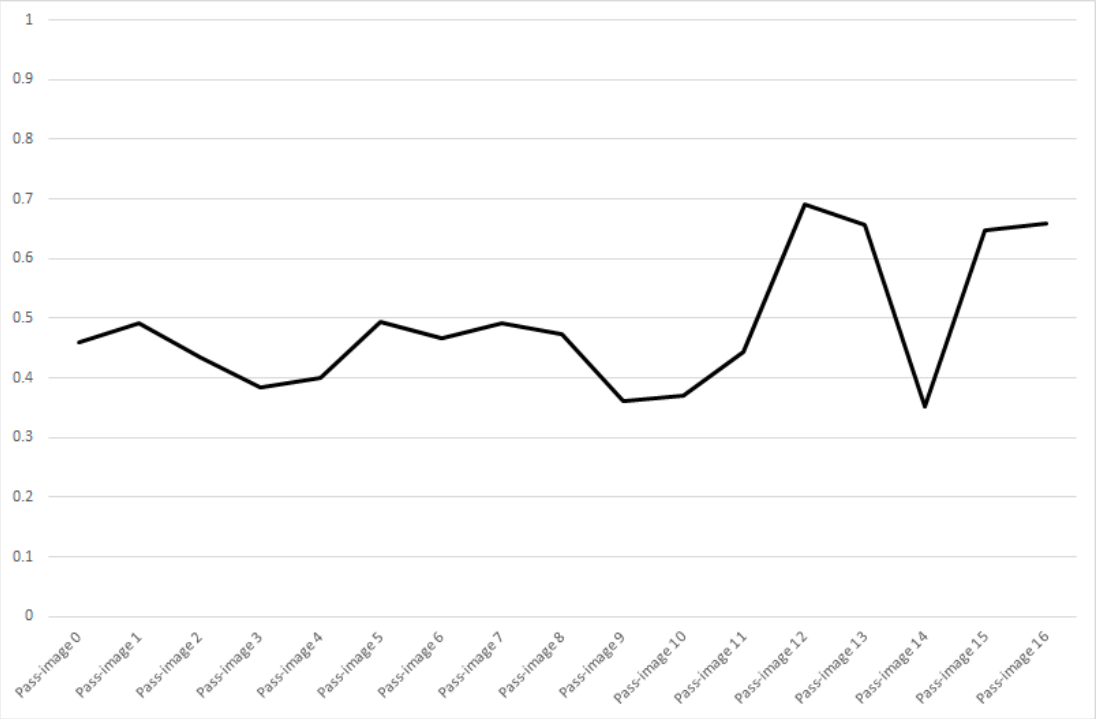


Figure F.37: Similarity threshold values for grammar 18

F.3 Goodness of fit graphs for all grammars

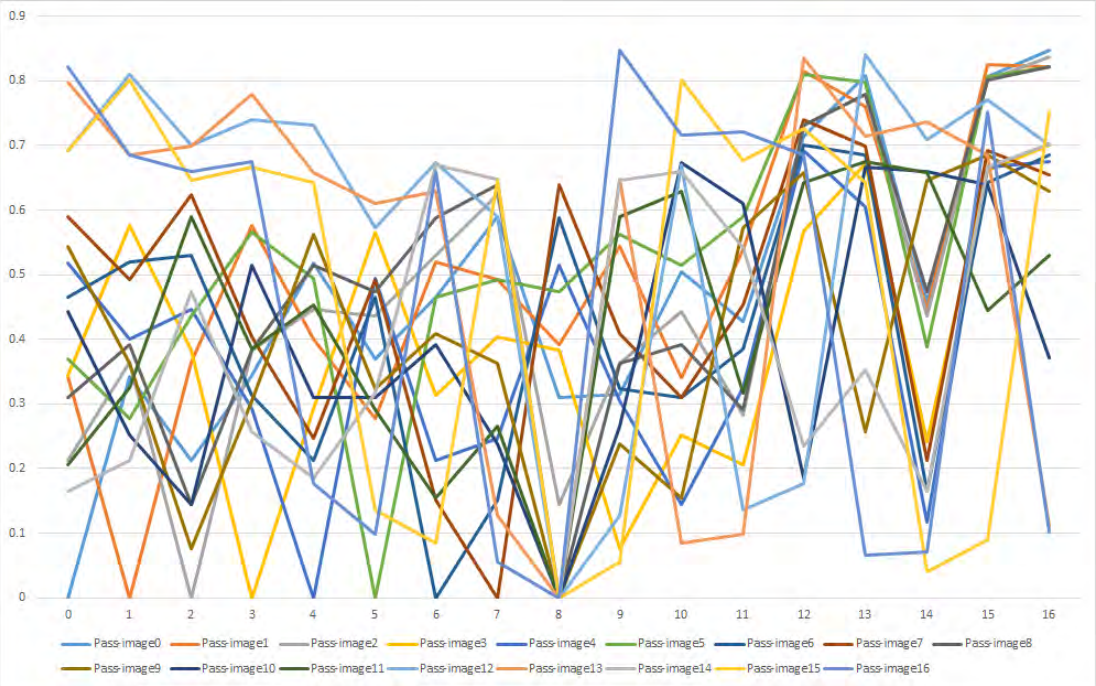


Figure F.38: Bottom-up tree edit distance values for grammar 18 for pass-images 0 to 16

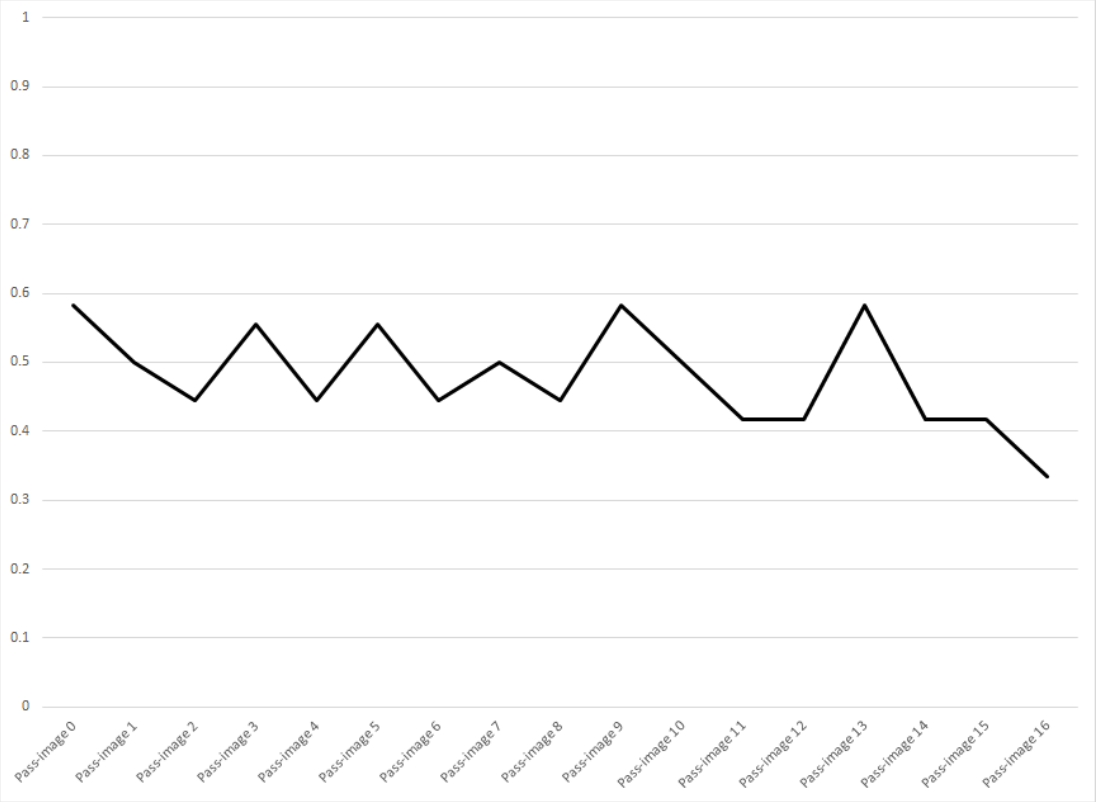


Figure F.39: Similarity threshold values for grammar 19

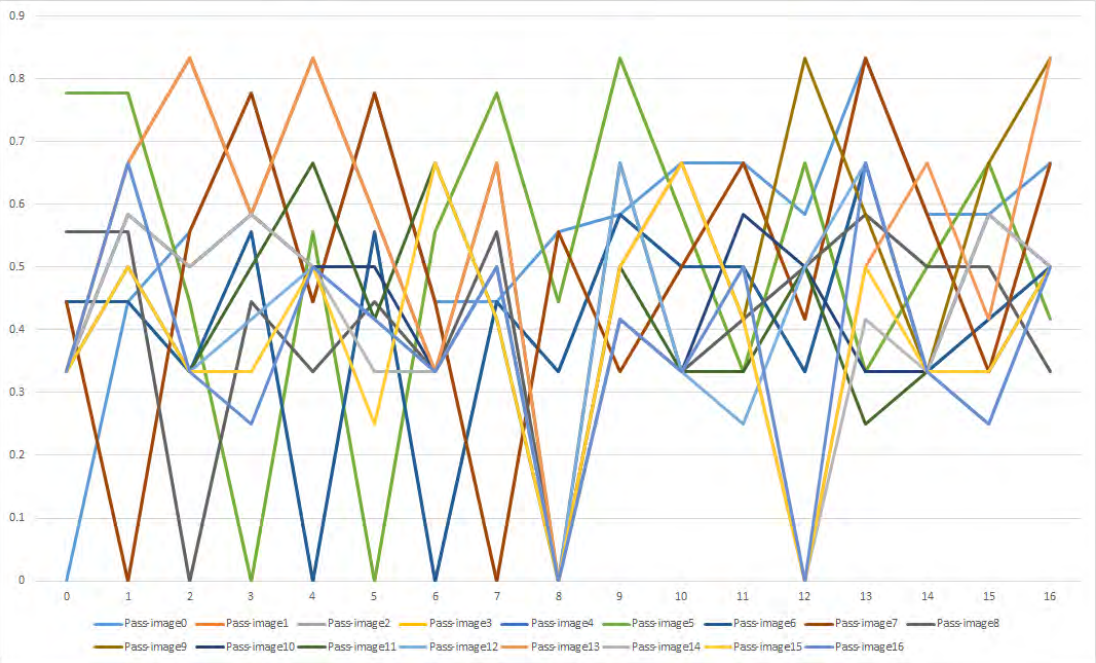


Figure F.40: Bottom-up tree edit distance values for grammar 19 for pass-images 0 to 16

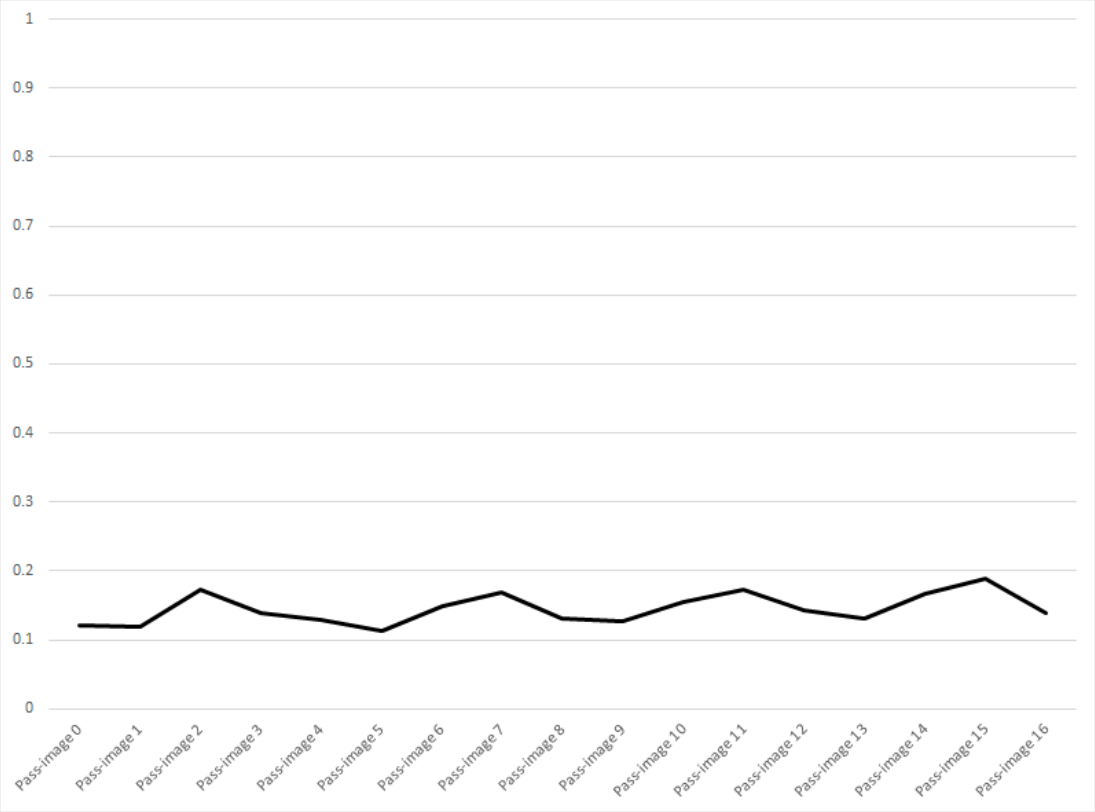


Figure F.41: Similarity threshold values for grammar 20

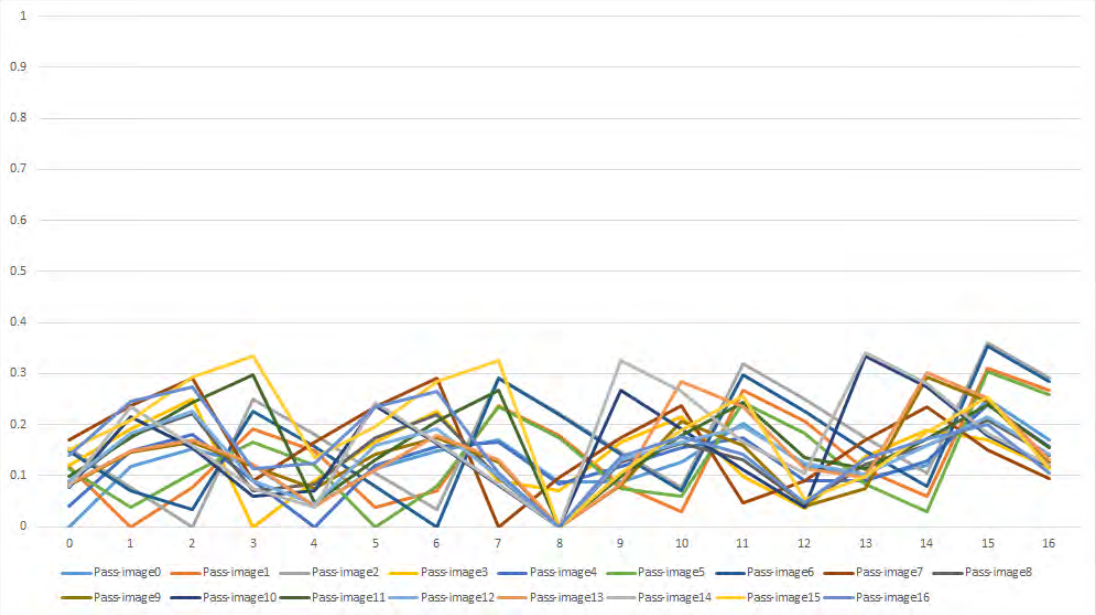


Figure F.42: Bottom-up tree edit distance values for grammar 20 for pass-images 0 to 16

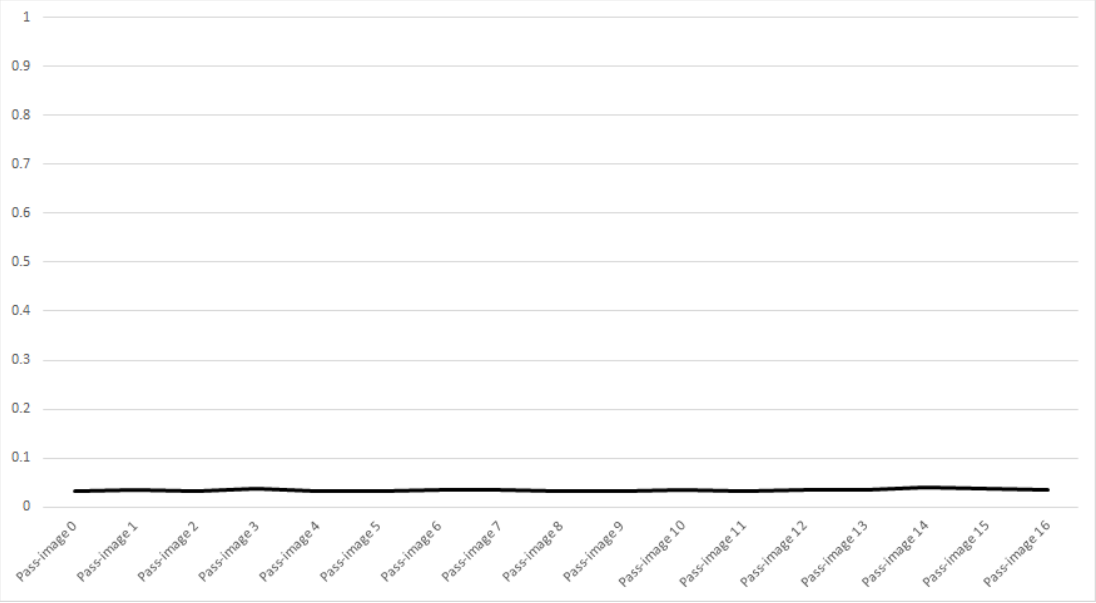


Figure F.43: Similarity threshold values for grammar 21

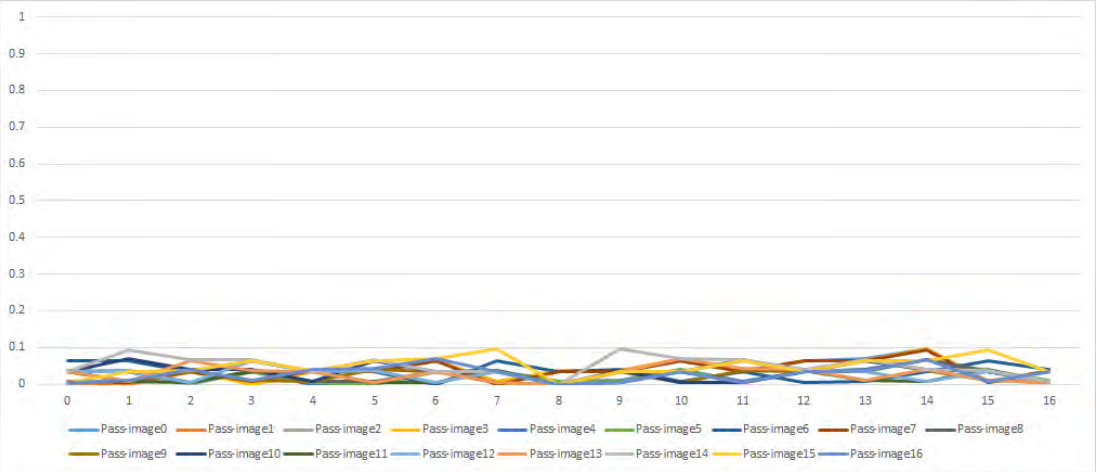


Figure F.44: Bottom-up tree edit distance values for grammar 21 for pass-images 0 to 16

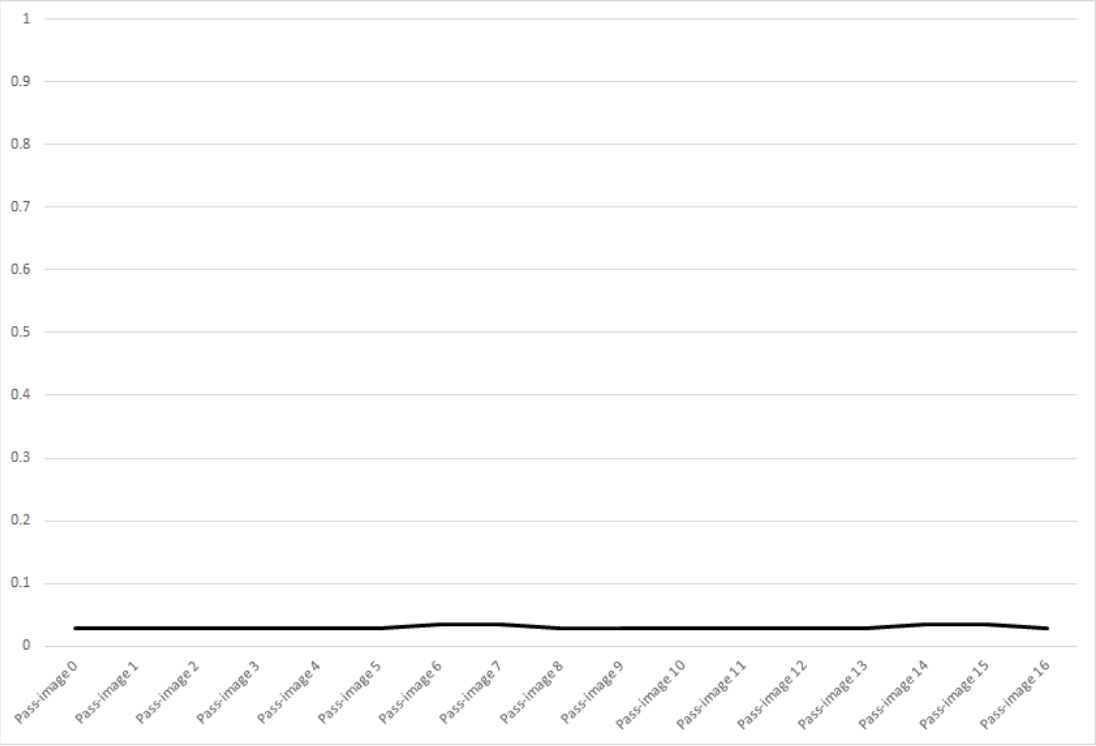


Figure F.45: Similarity threshold values for grammar 22

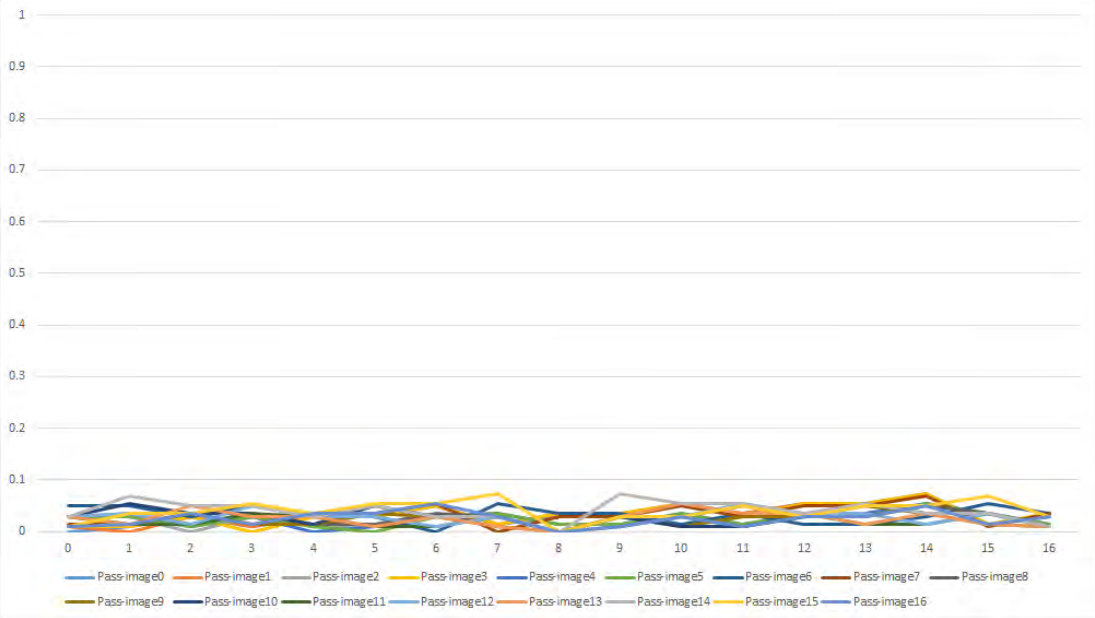


Figure F.46: Bottom-up tree edit distance values for grammar 22 for pass-images 0 to 16

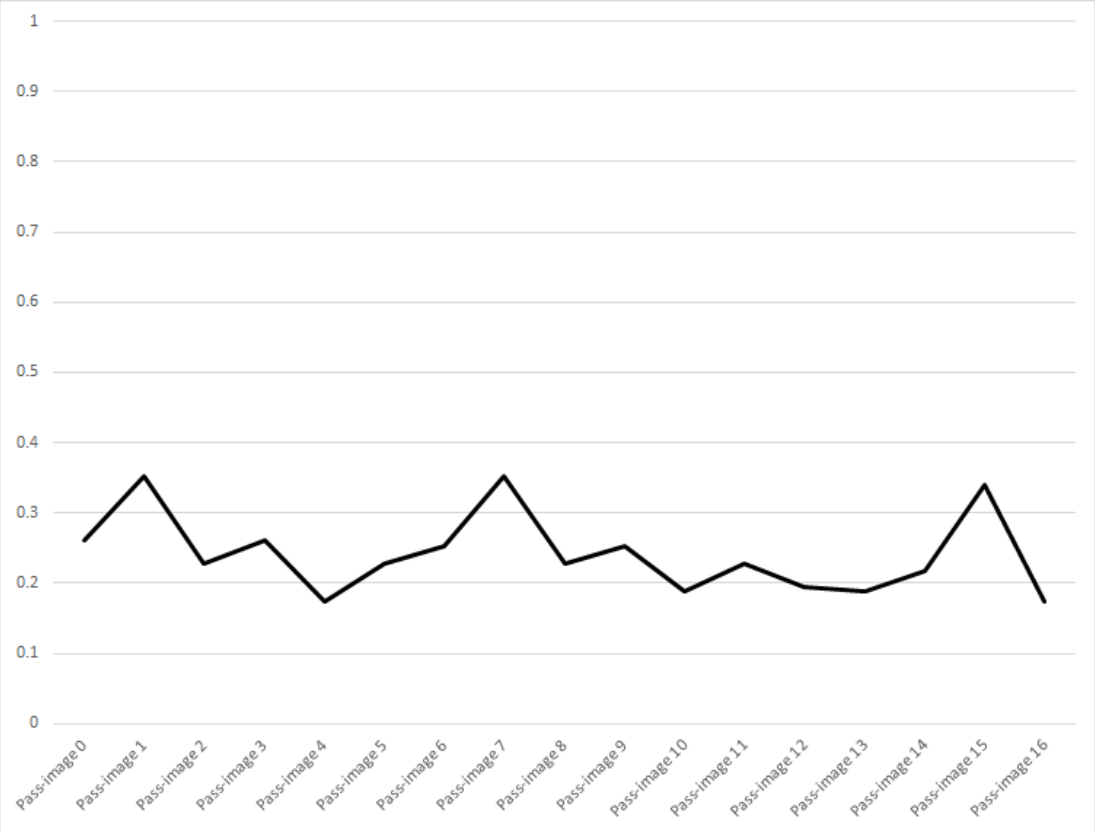


Figure F.47: Similarity threshold values for grammar 23

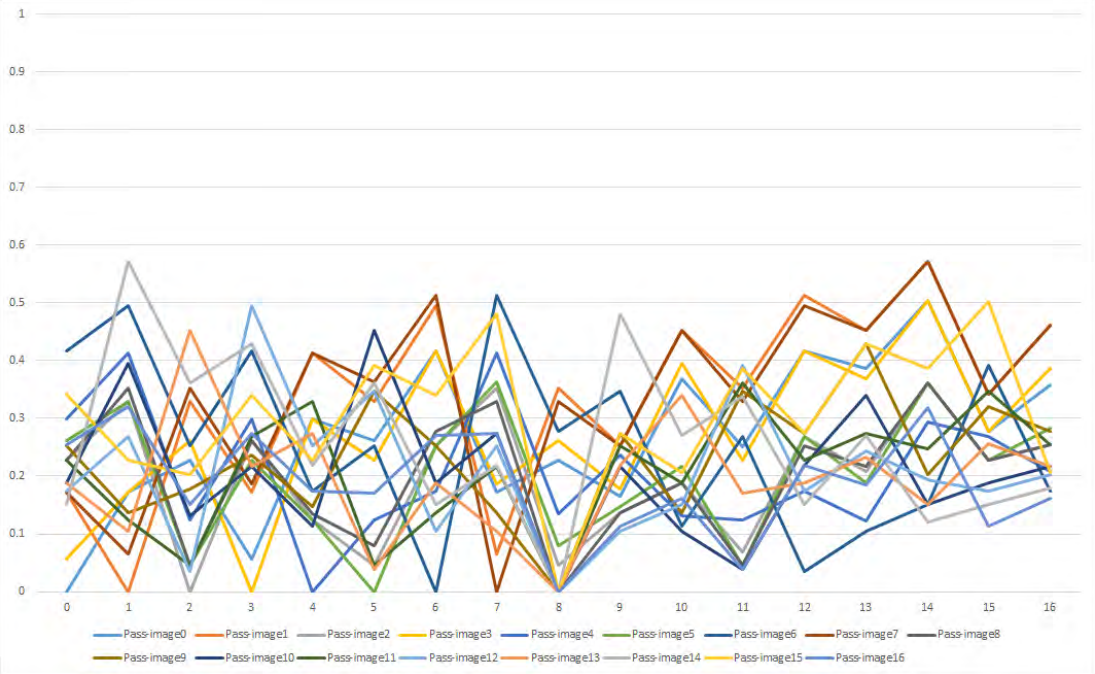


Figure F.48: Bottom-up tree edit distance values for grammar 23 for pass-images 0 to 16

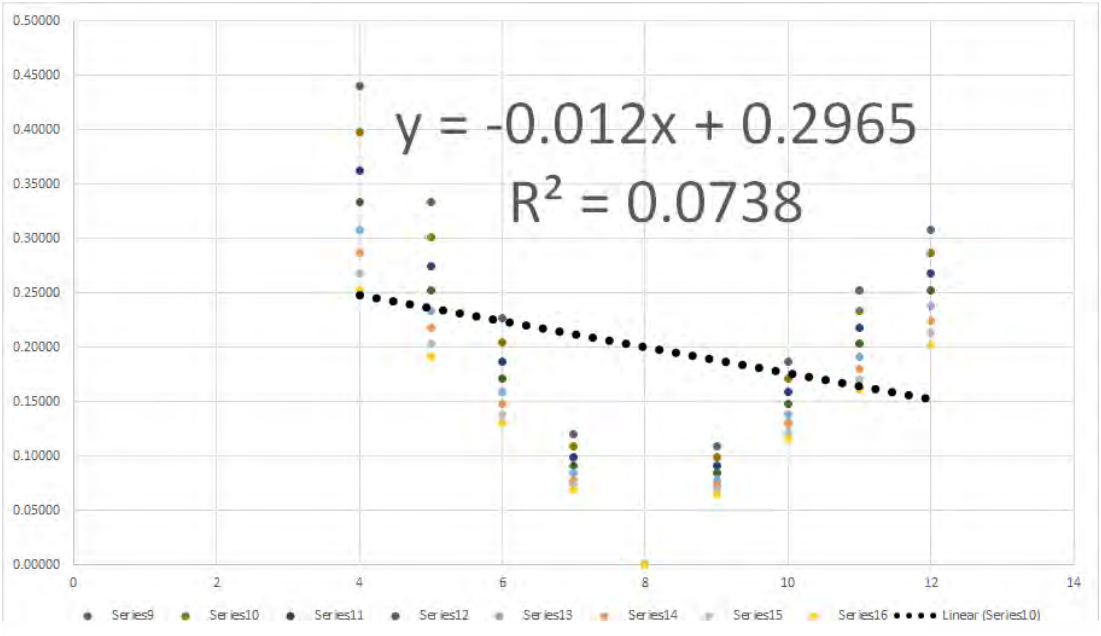


Figure F.49: Goodness of linear fit graph for displayed images for grammar 0 for pass-images 9 to 16

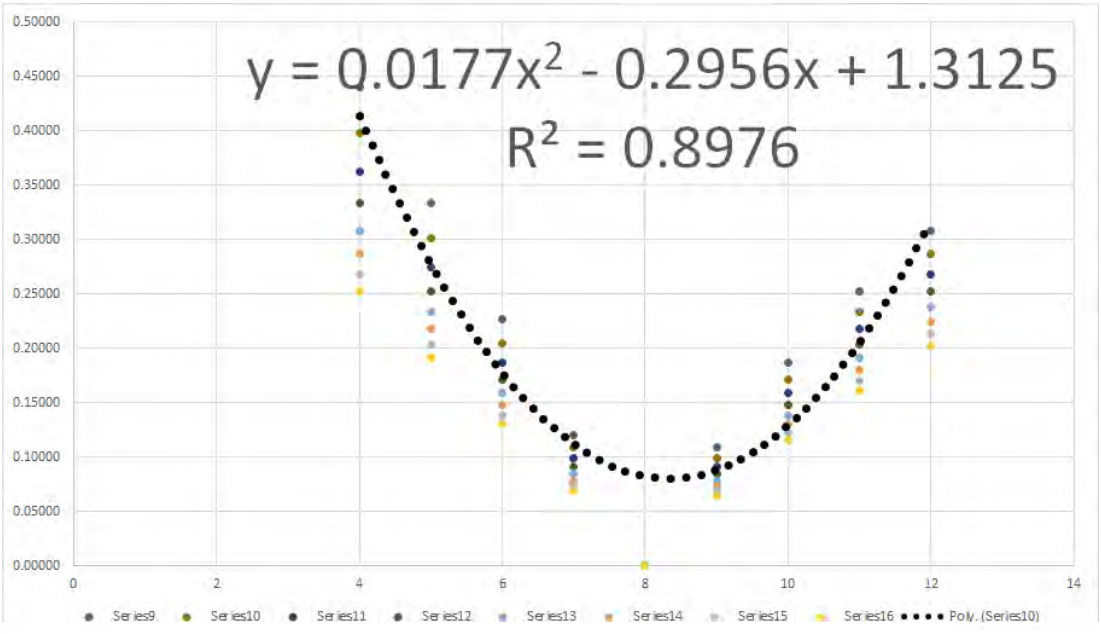


Figure F.50: Goodness of quadratic fit graph for displayed images for grammar 0 for pass-images 9 to 16

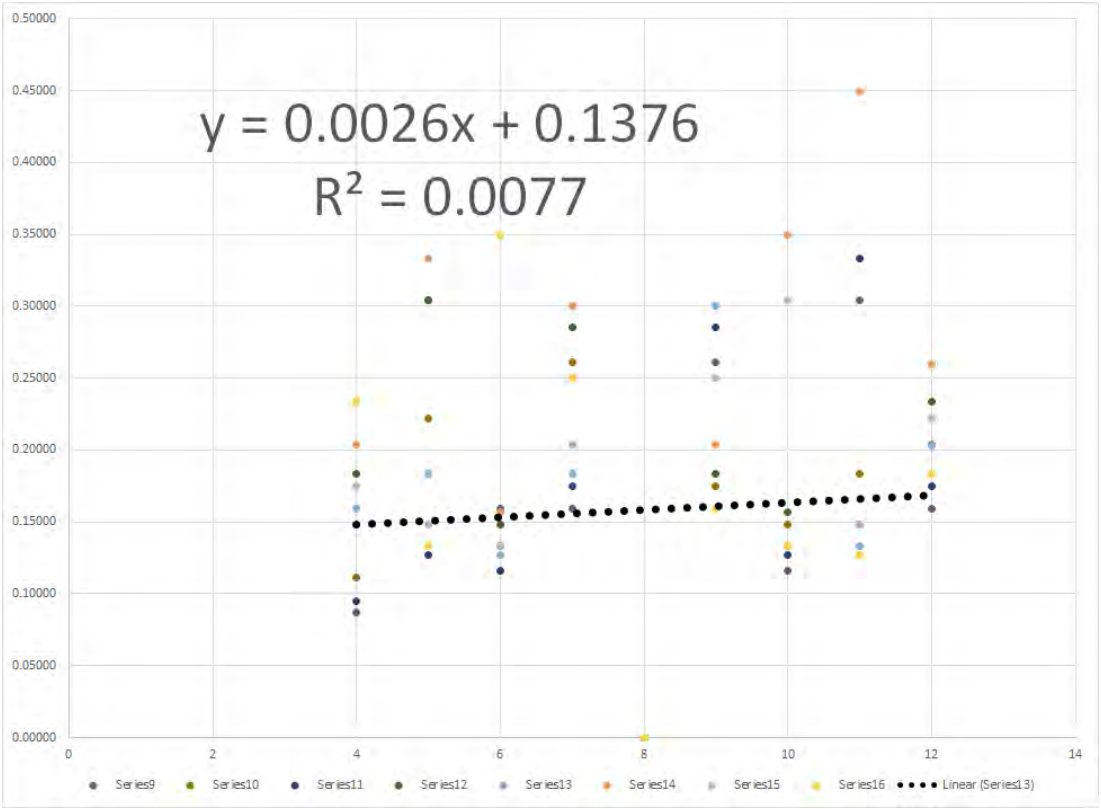


Figure F.51: Goodness of linear fit graph for displayed images for grammar 1 for pass-images 9 to 16

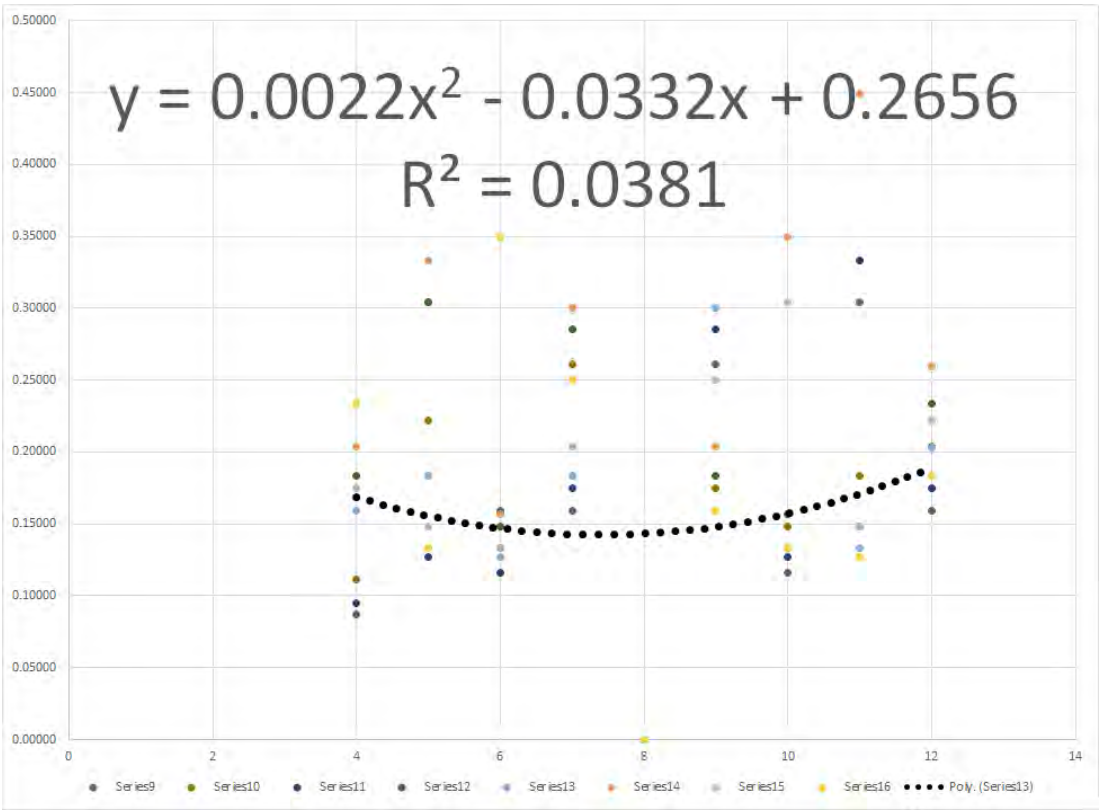


Figure F.52: Goodness of quadratic fit graph for displayed images for grammar 1 for pass-images 9 to 16

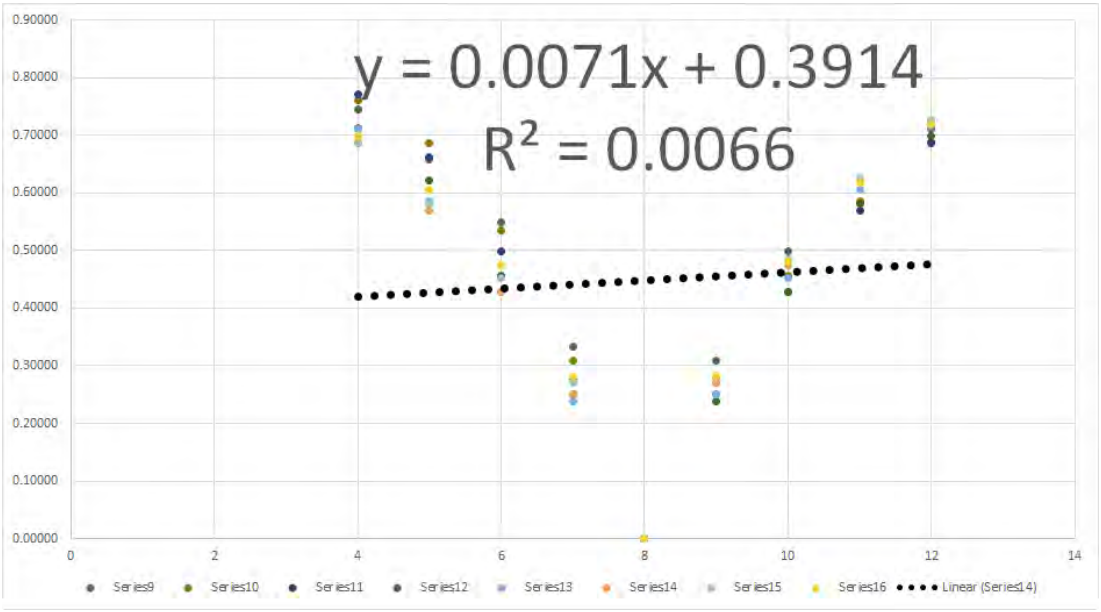


Figure F.53: Goodness of linear fit graph for displayed images for grammar 2 for pass-images 9 to 16

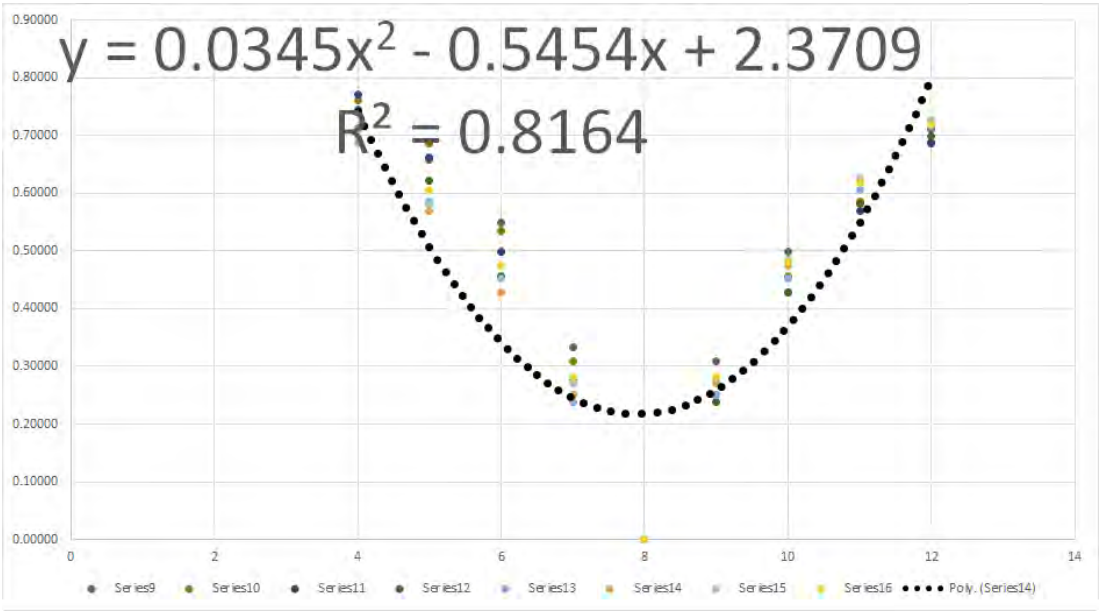


Figure F.54: Goodness of quadratic fit graph for displayed images for grammar 2 for pass-images 9 to 16

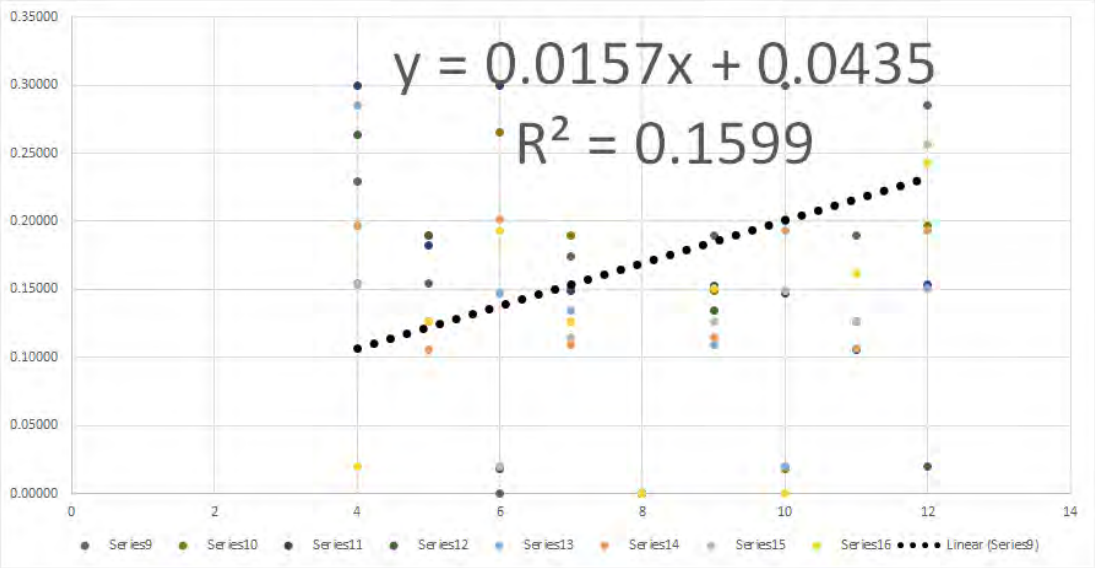


Figure F.55: Goodness of linear fit graph for displayed images for grammar 3 for pass-images 9 to 16

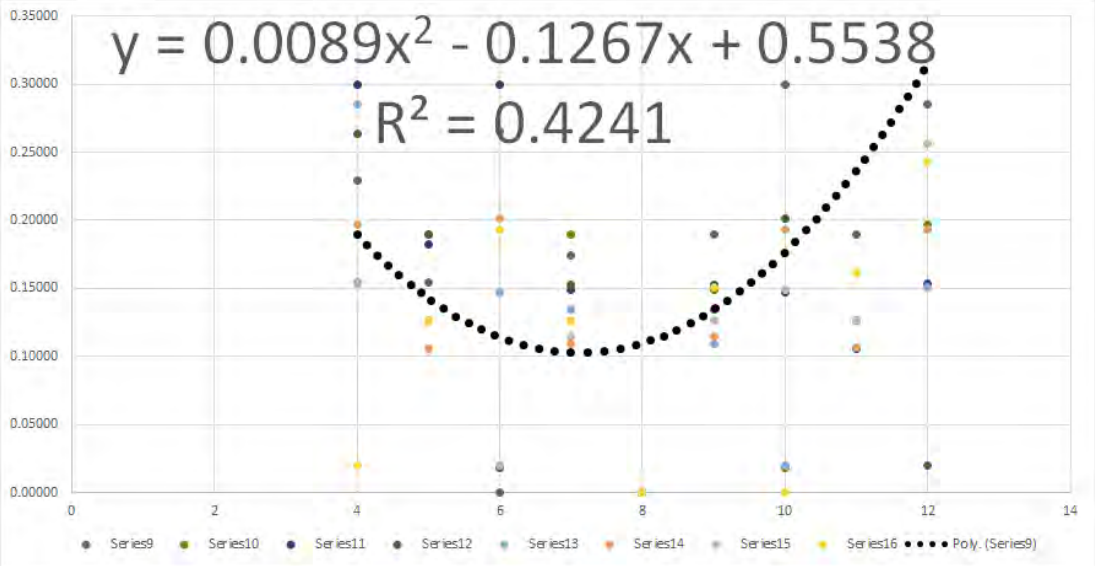


Figure F.56: Goodness of quadratic fit graph for displayed images for grammar 3 for pass-images 9 to 16

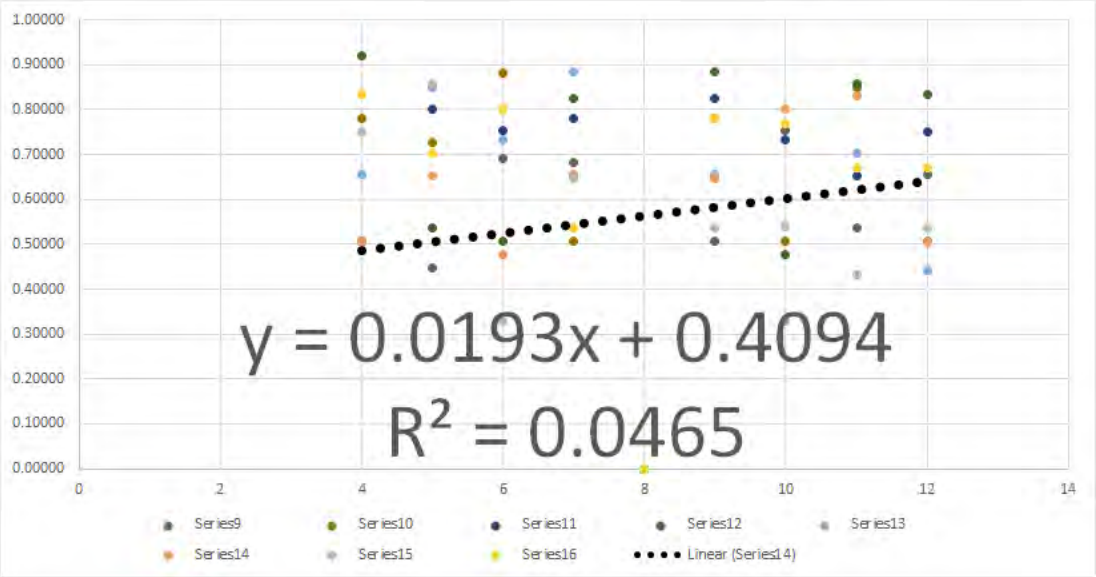


Figure F.57: Goodness of linear fit graph for displayed images for grammar 4 for pass-images 9 to 16

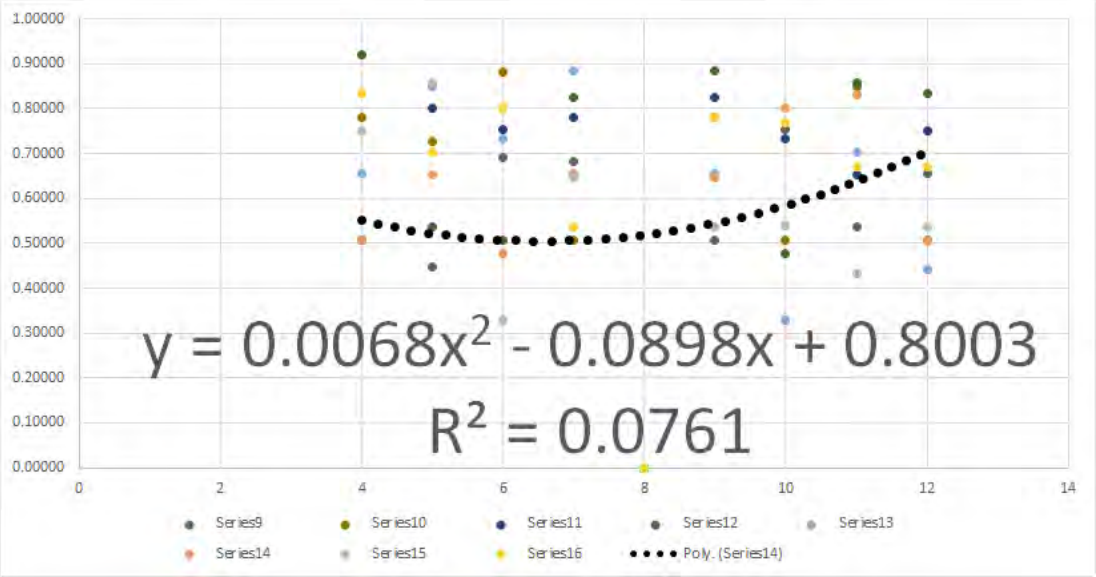


Figure F.58: Goodness of quadratic fit graph for displayed images for grammar 4 for pass-images 9 to 16

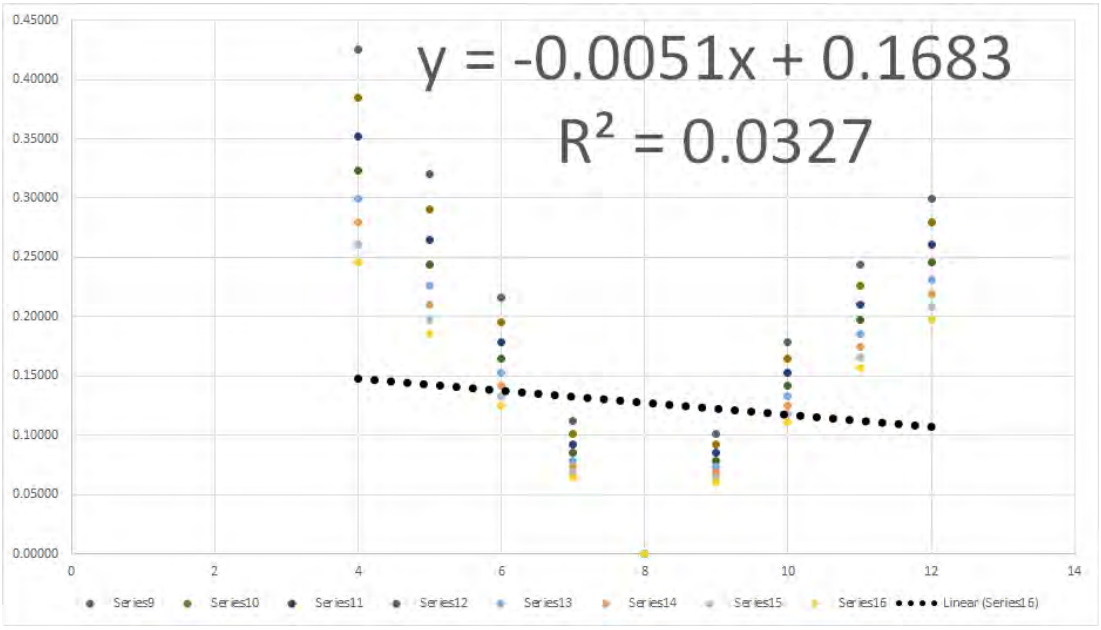


Figure F.59: Goodness of linear fit graph for displayed images for grammar 5 for pass-images 9 to 16

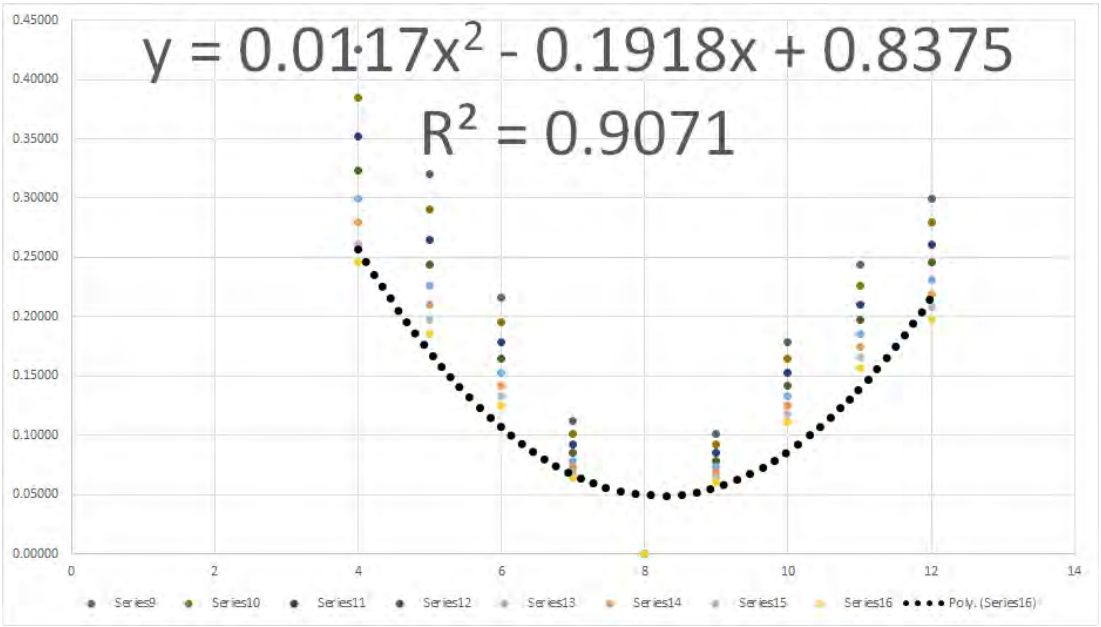


Figure F.60: Goodness of quadratic fit graph for displayed images for grammar 5 for pass-images 9 to 16

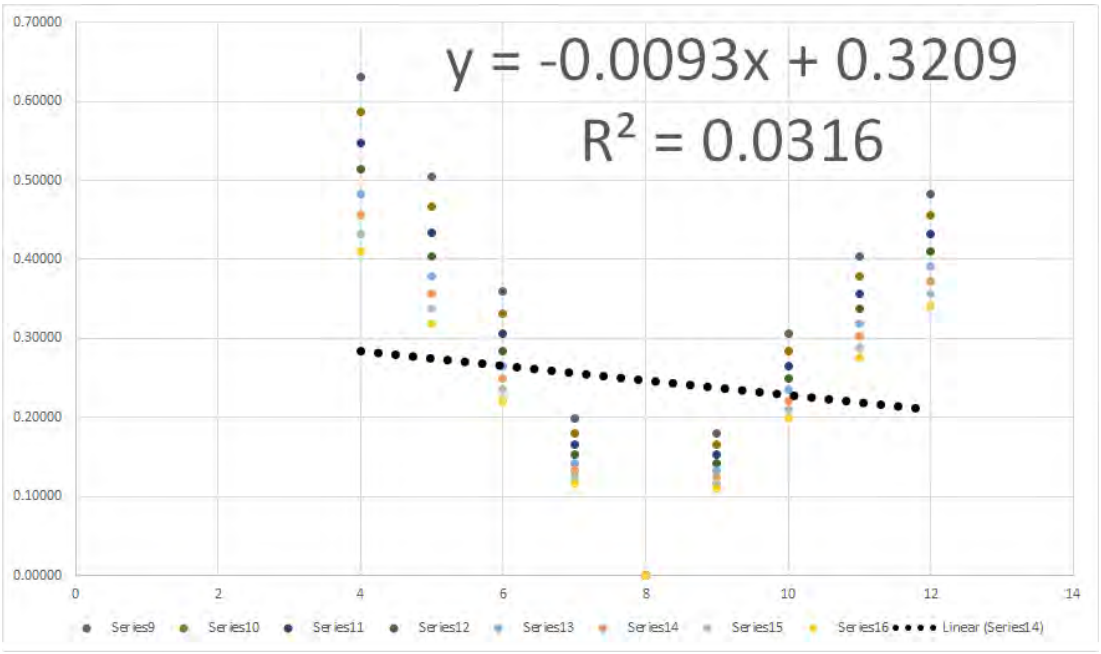


Figure F.61: Goodness of linear fit graph for displayed images for grammar 6 for pass-images 9 to 16

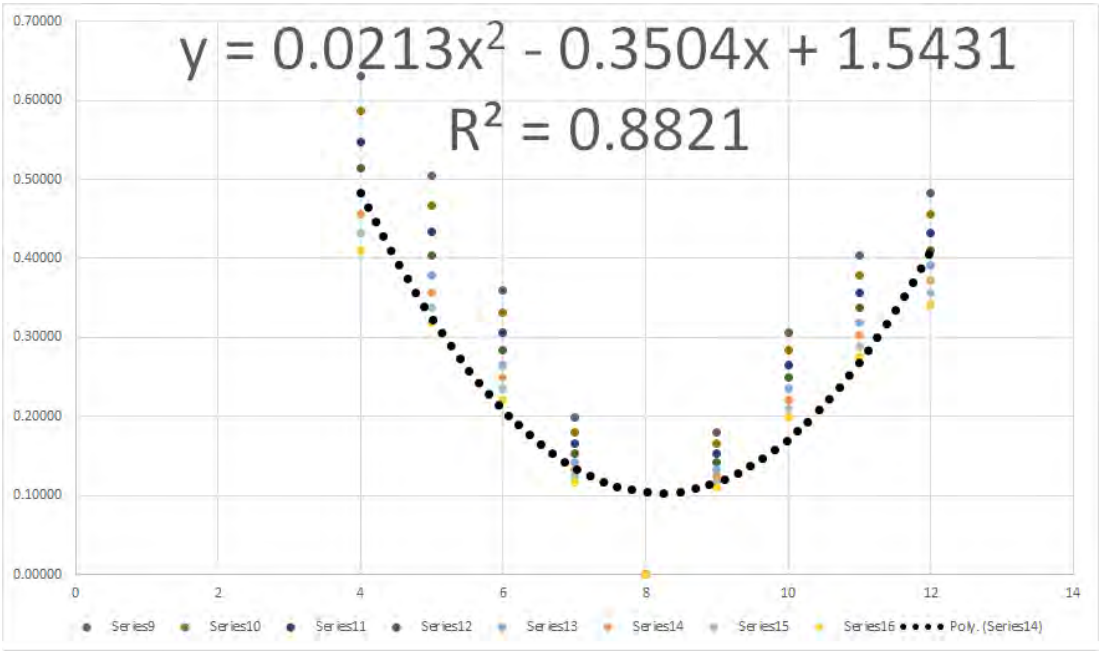


Figure F.62: Goodness of quadratic fit graph for displayed images for grammar 6 for pass-images 9 to 16

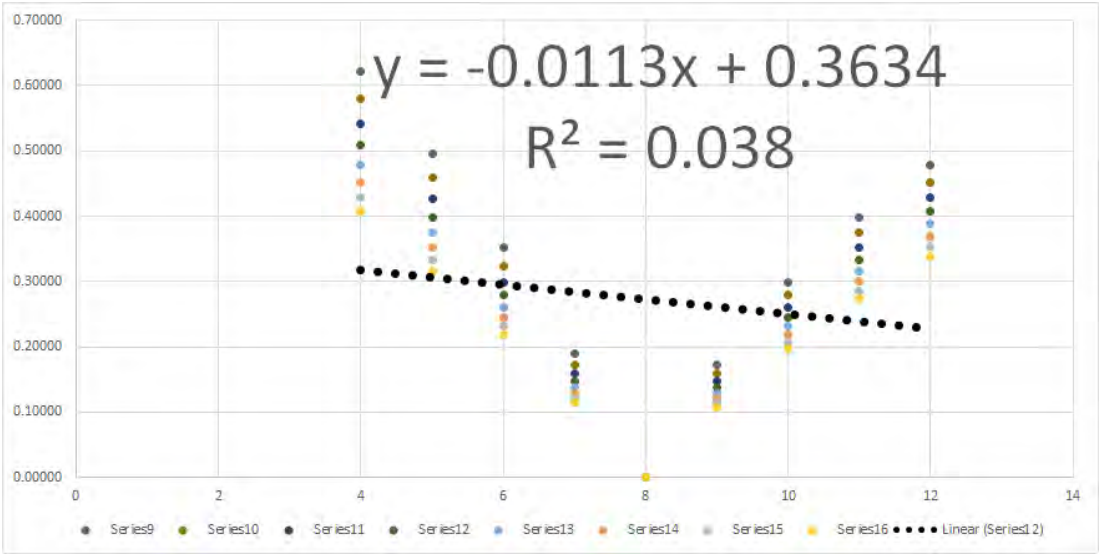


Figure F.63: Goodness of linear fit graph for displayed images for grammar 7 for pass-images 9 to 16

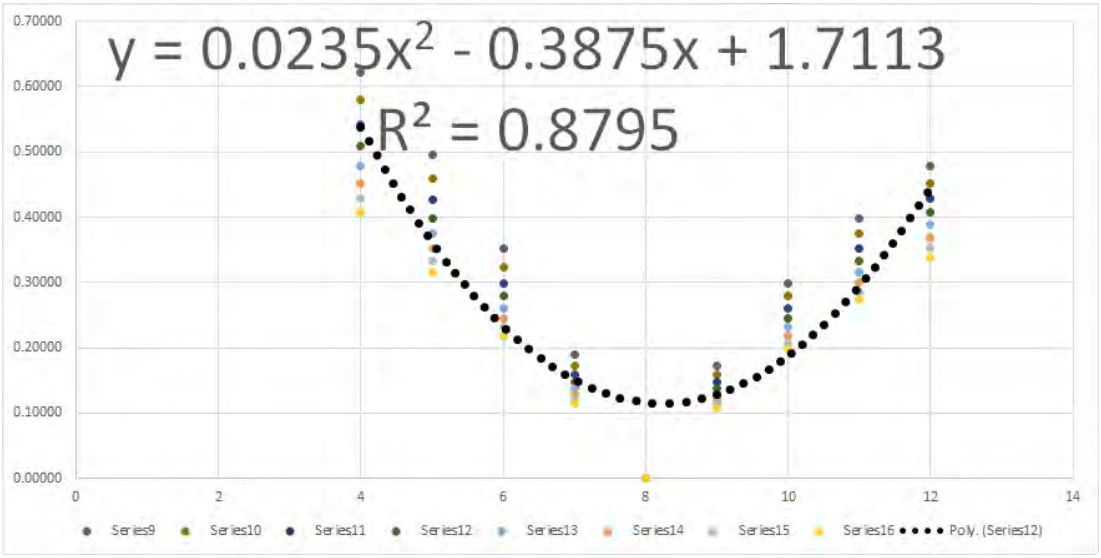


Figure F.64: Goodness of quadratic fit graph for displayed images for grammar 7 for pass-images 9 to 16

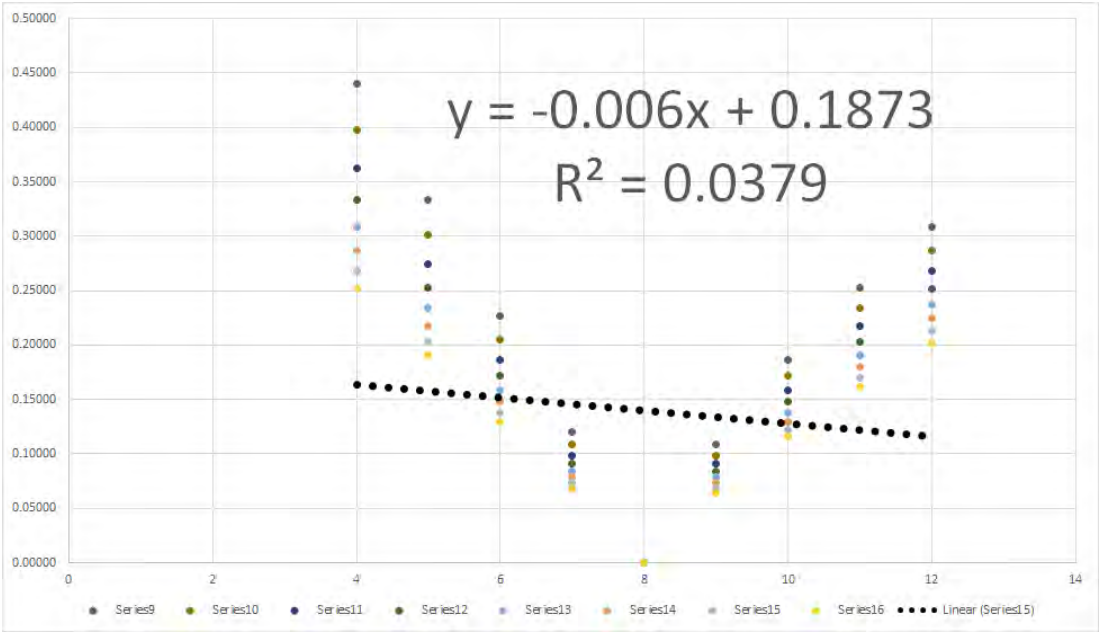


Figure F.65: Goodness of linear fit graph for displayed images for grammar 8 for pass-images 9 to 16

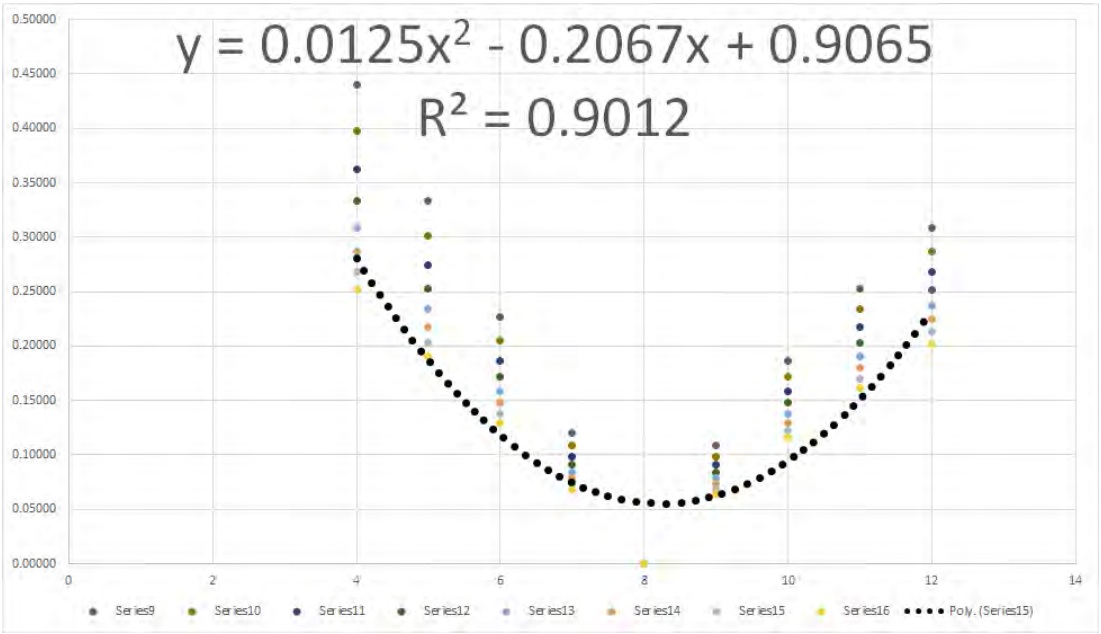


Figure F.66: Goodness of quadratic fit graph for displayed images for grammar 8 for pass-images 9 to 16

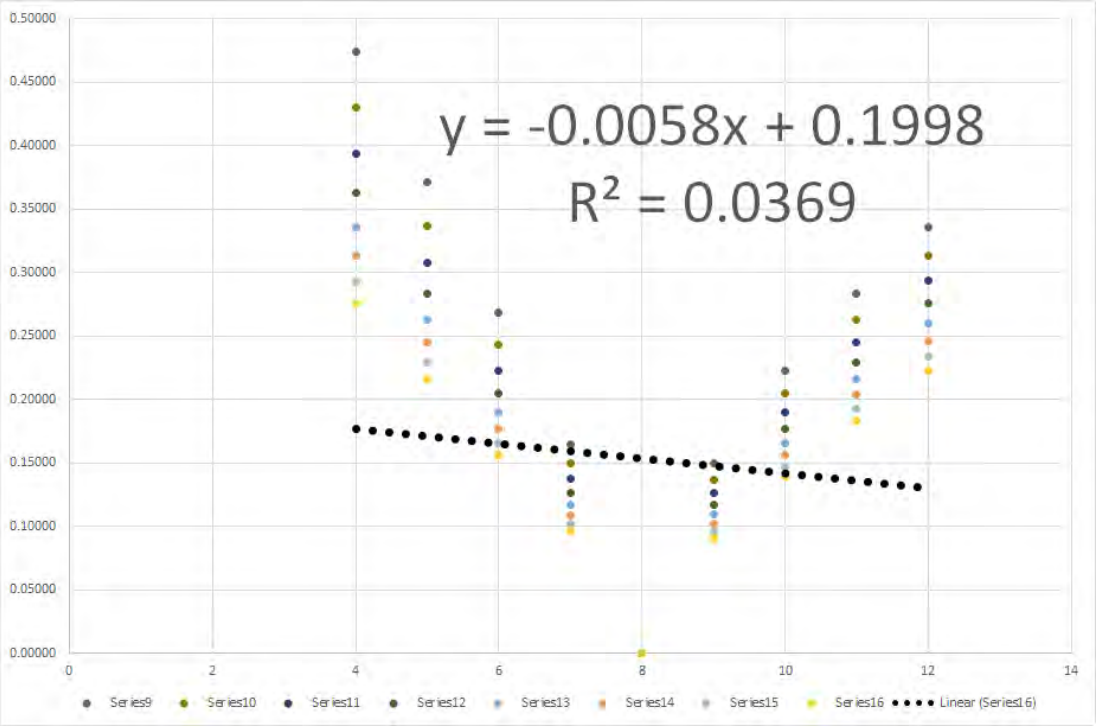


Figure F.67: Goodness of linear fit graph for displayed images for grammar 9 for pass-images 9 to 16

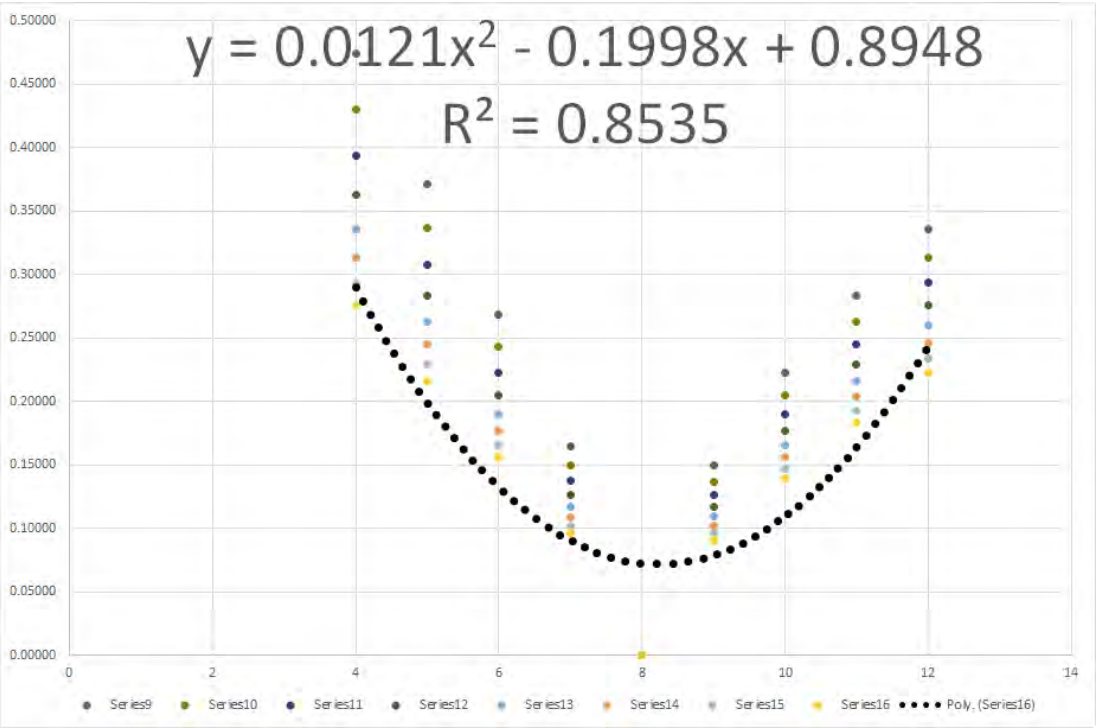


Figure F.68: Goodness of quadratic fit graph for displayed images for grammar 9 for pass-images 9 to 16

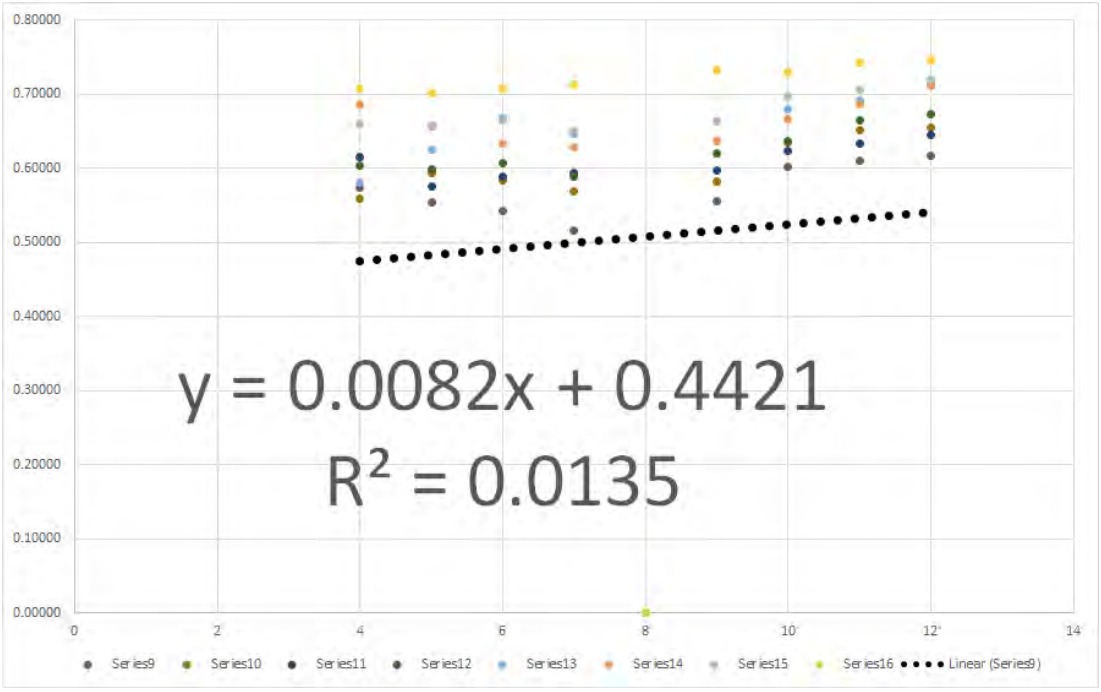


Figure F.69: Goodness of linear fit graph for displayed images for grammar 10 for pass-images 9 to 16

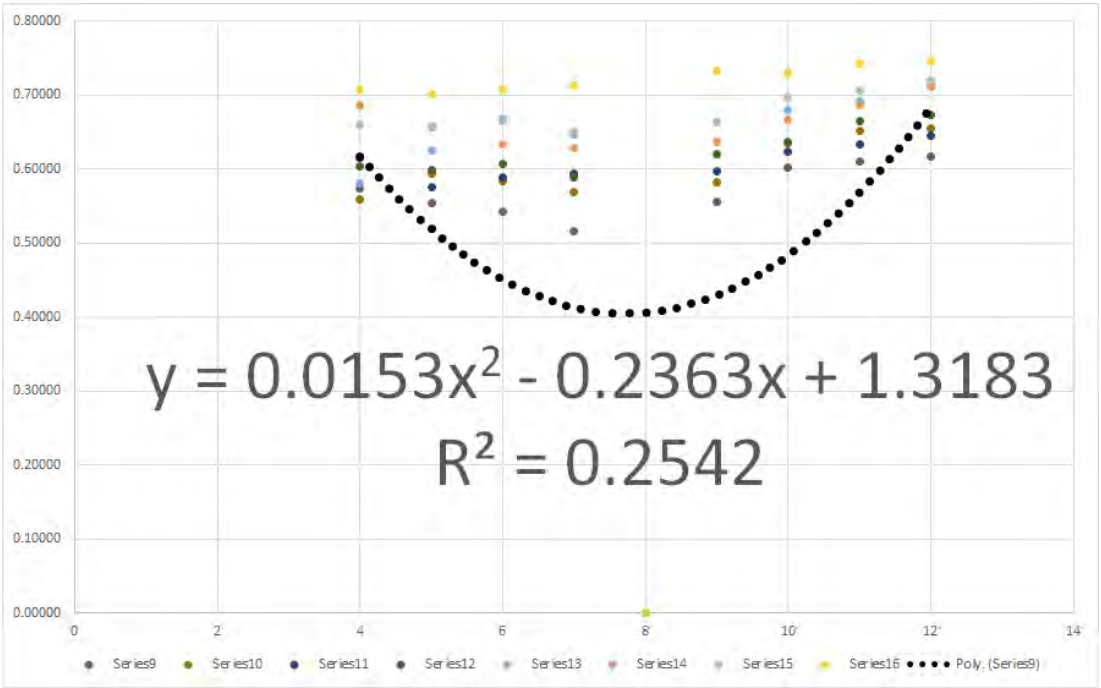


Figure F.70: Goodness of quadratic fit graph for displayed images for grammar 10 for pass-images 9 to 16

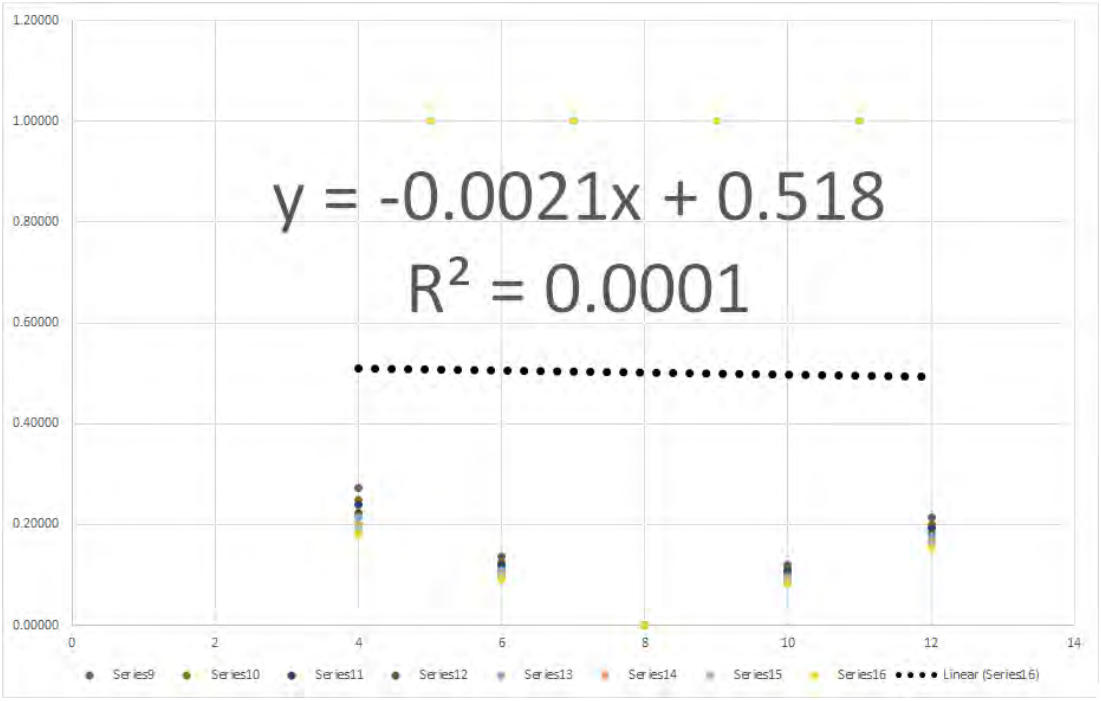


Figure F.71: Goodness of linear fit graph for displayed images for grammar 11 for pass-images 9 to 16

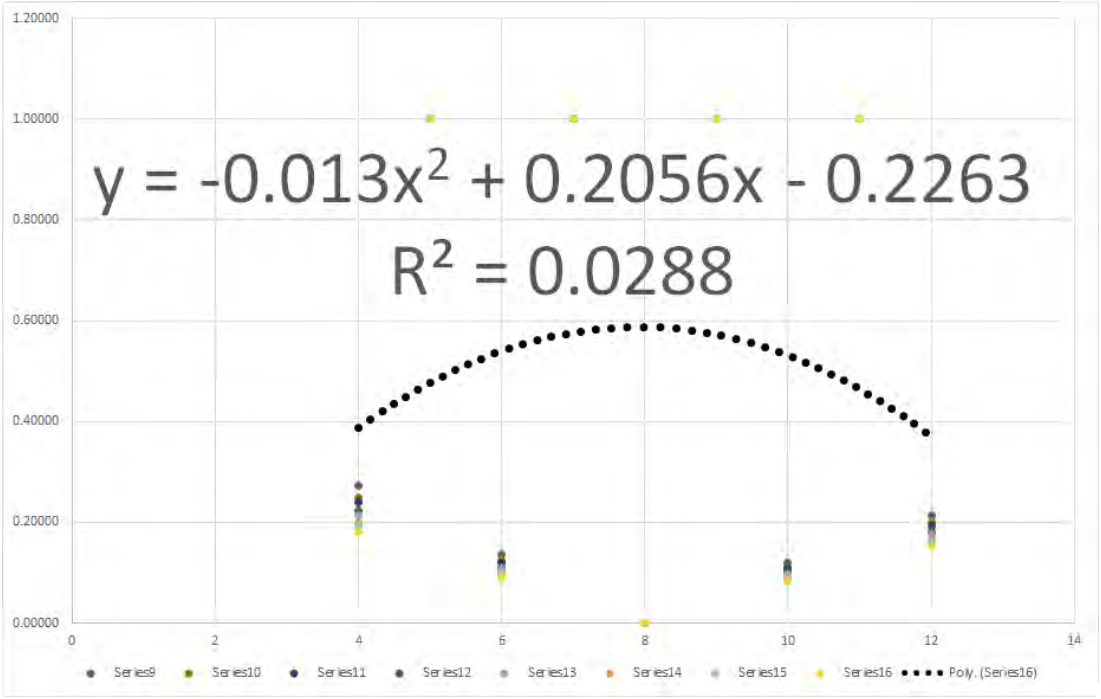


Figure F.72: Goodness of quadratic fit graph for displayed images for grammar 11 for pass-images 9 to 16

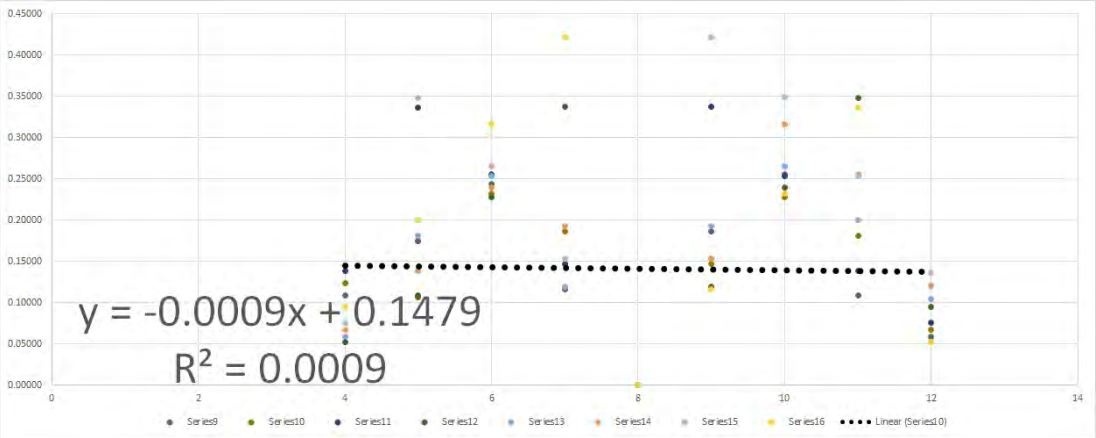


Figure F.73: Goodness of linear fit graph for displayed images for grammar12 for pass-images 9 to 16

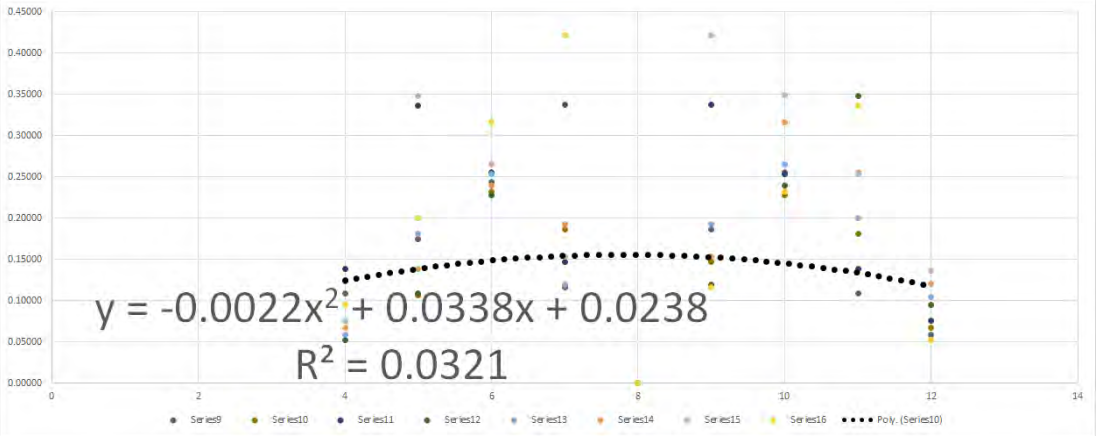


Figure F.74: Goodness of quadratic fit graph for displayed images for grammar 12 for pass-images 9 to 16

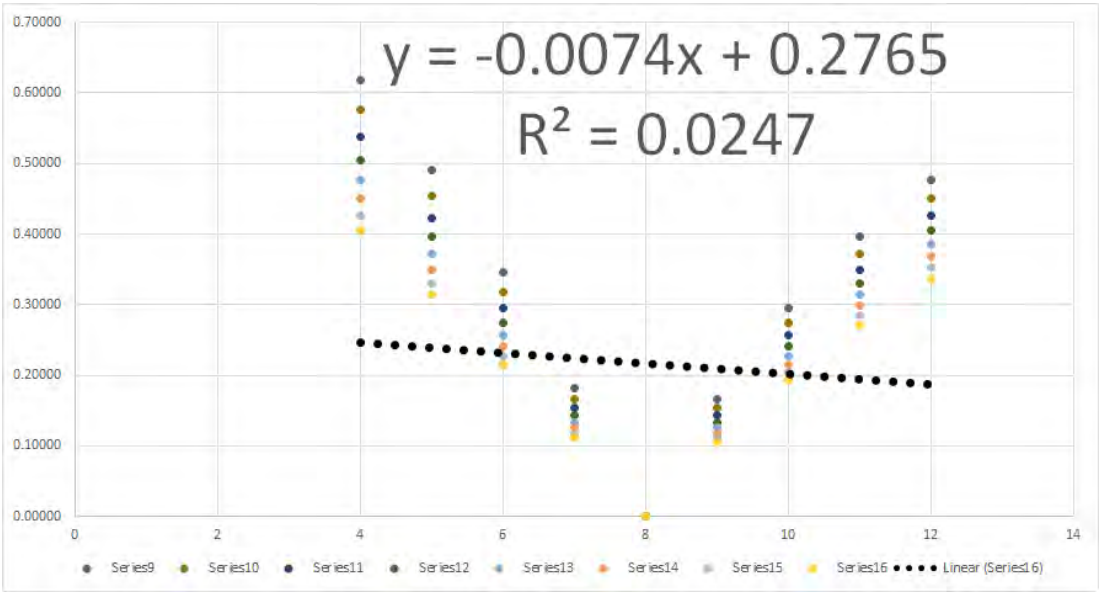


Figure F.75: Goodness of linear fit graph for displayed images for grammar 13 for pass-images 9 to 16

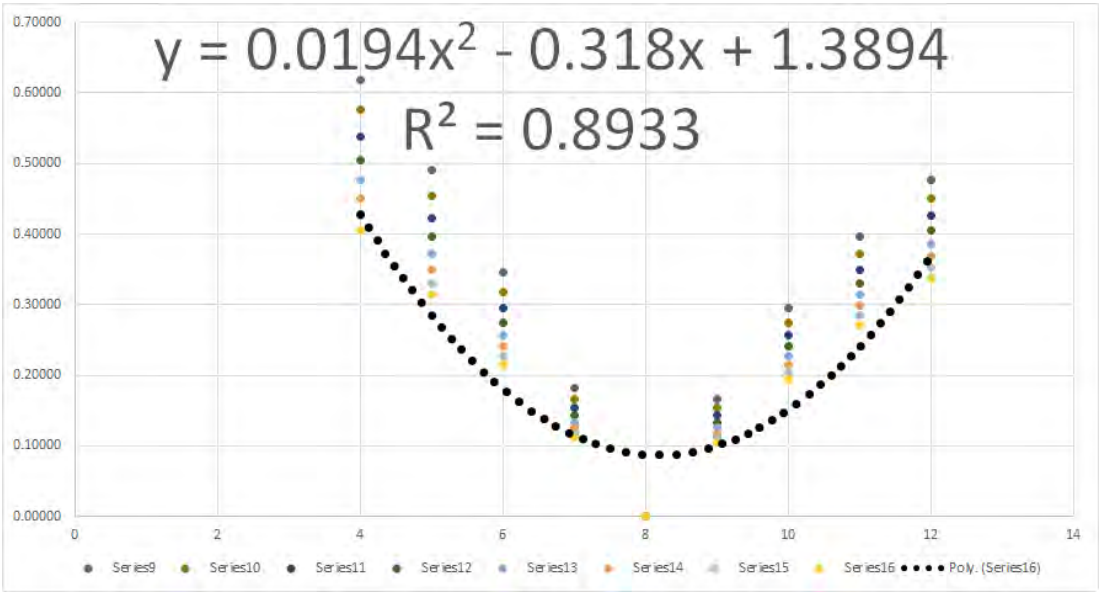


Figure F.76: Goodness of quadratic fit graph for displayed images for grammar 13 for pass-images 9 to 16

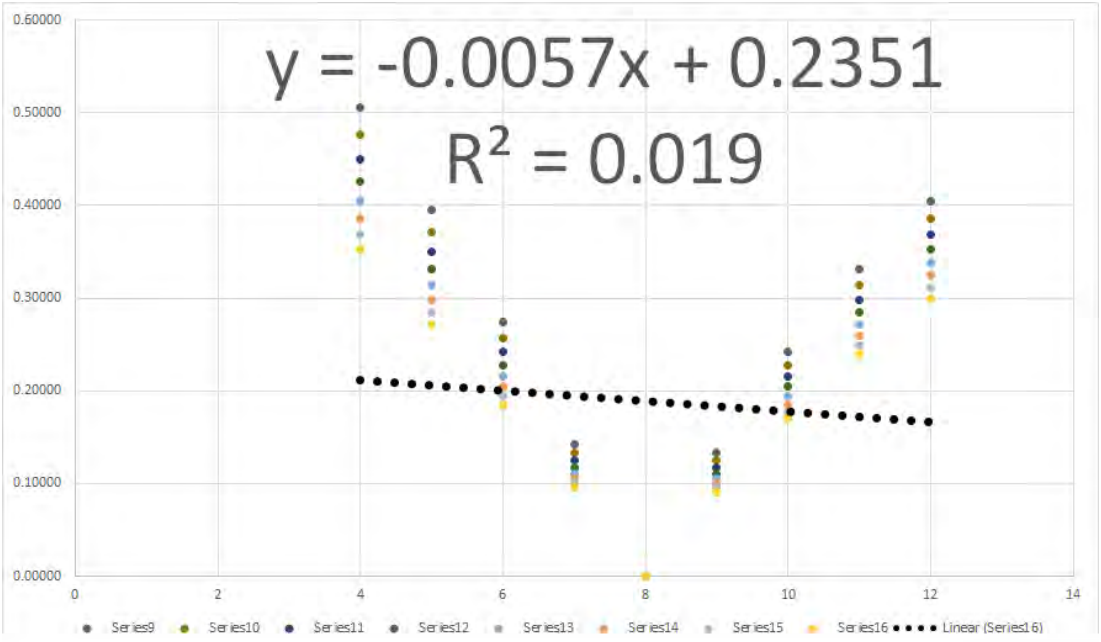


Figure F.77: Goodness of linear fit graph for displayed images for grammar 14 for pass-images 9 to 16

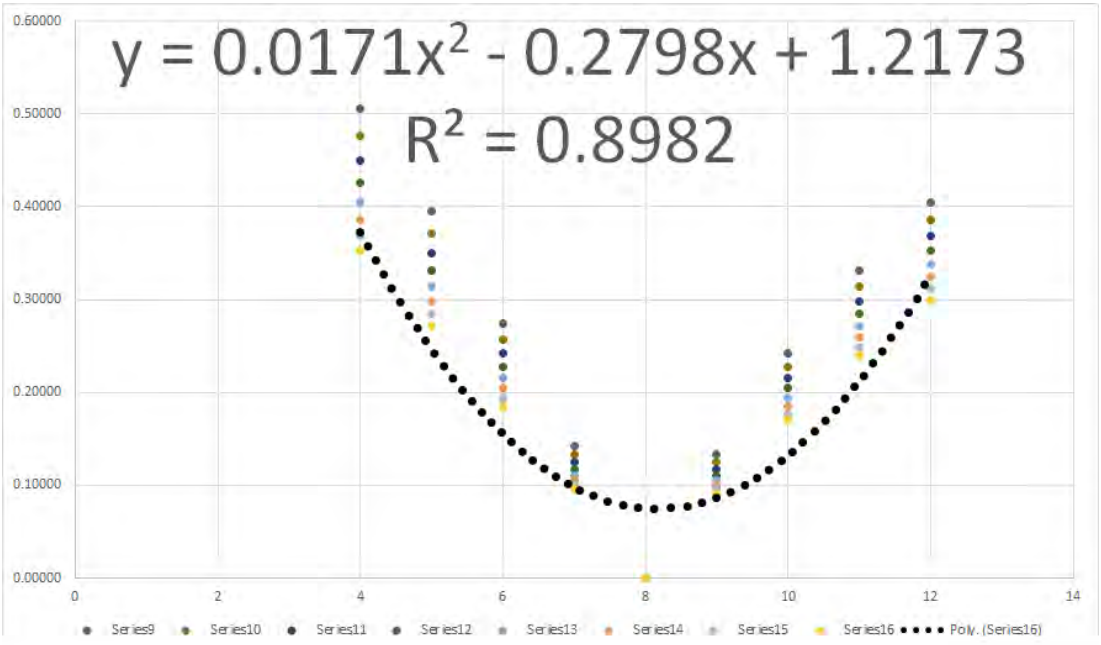


Figure F.78: Goodness of quadratic fit graph for displayed images for grammar 14 for pass-images 9 to 16

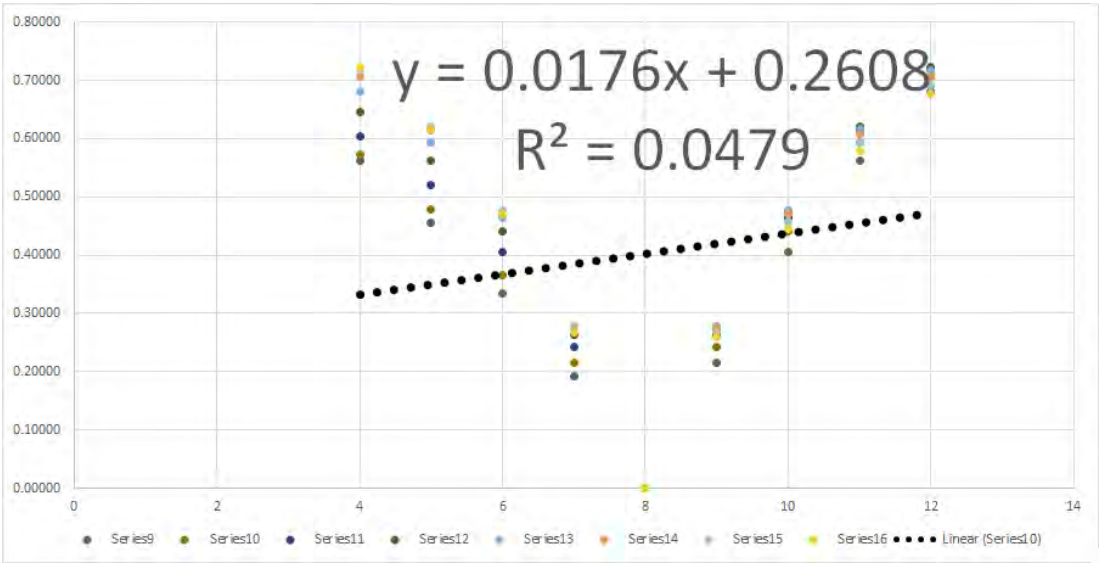


Figure F.79: Goodness of linear fit graph for displayed images for grammar 15 for pass-images 9 to 16

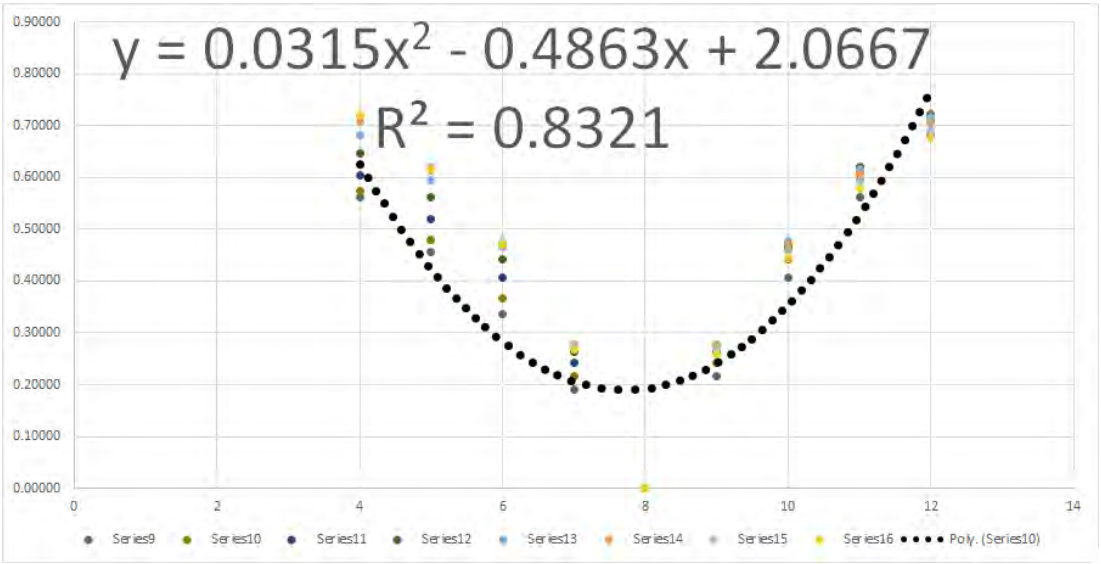


Figure F.80: Goodness of quadratic fit graph for displayed images for grammar 15 for pass-images 9 to 16

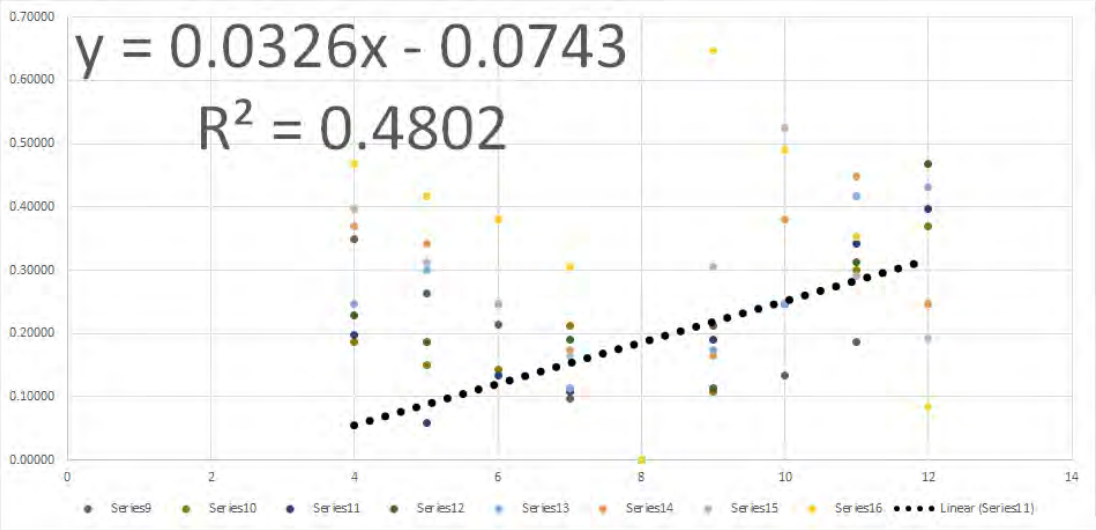


Figure F.81: Goodness of linear fit graph for displayed images for grammar 16 for pass-images 9 to 16

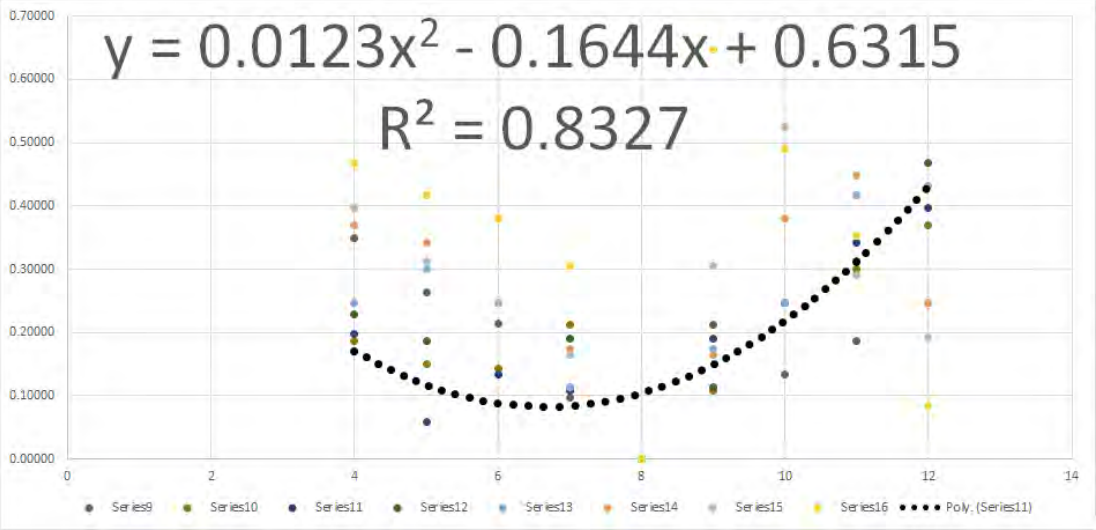


Figure F.82: Goodness of quadratic fit graph for displayed images for grammar 16 for pass-images 9 to 16

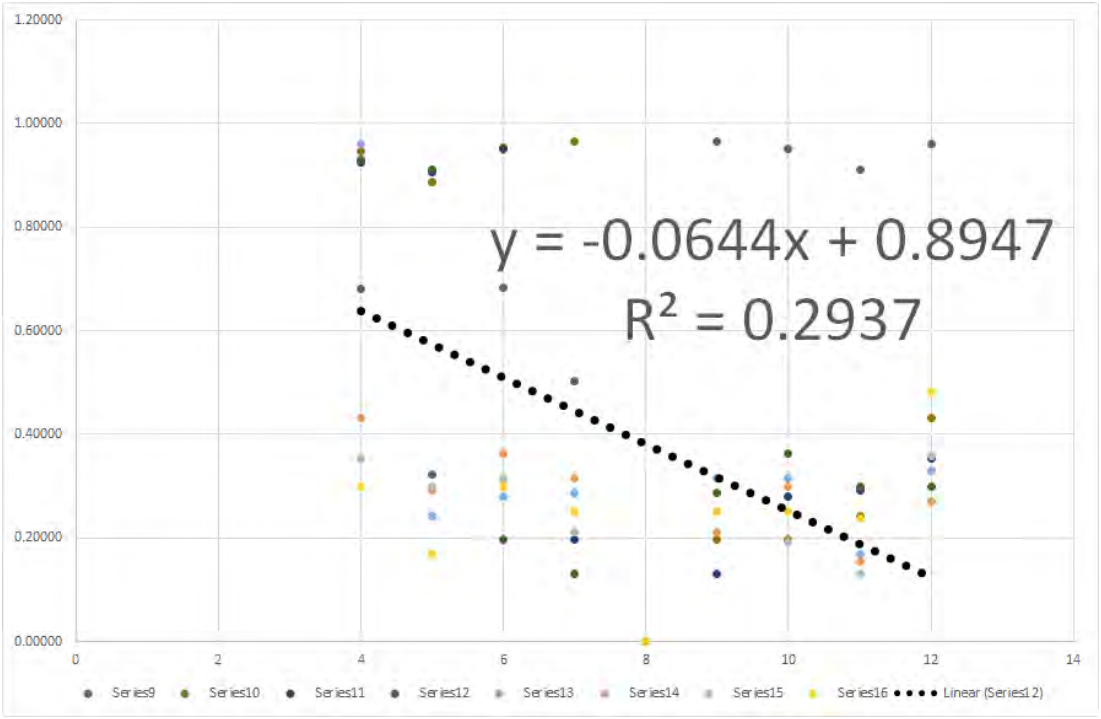


Figure F.83: Goodness of linear fit graph for displayed images for grammar 17 for pass-images 9 to 16

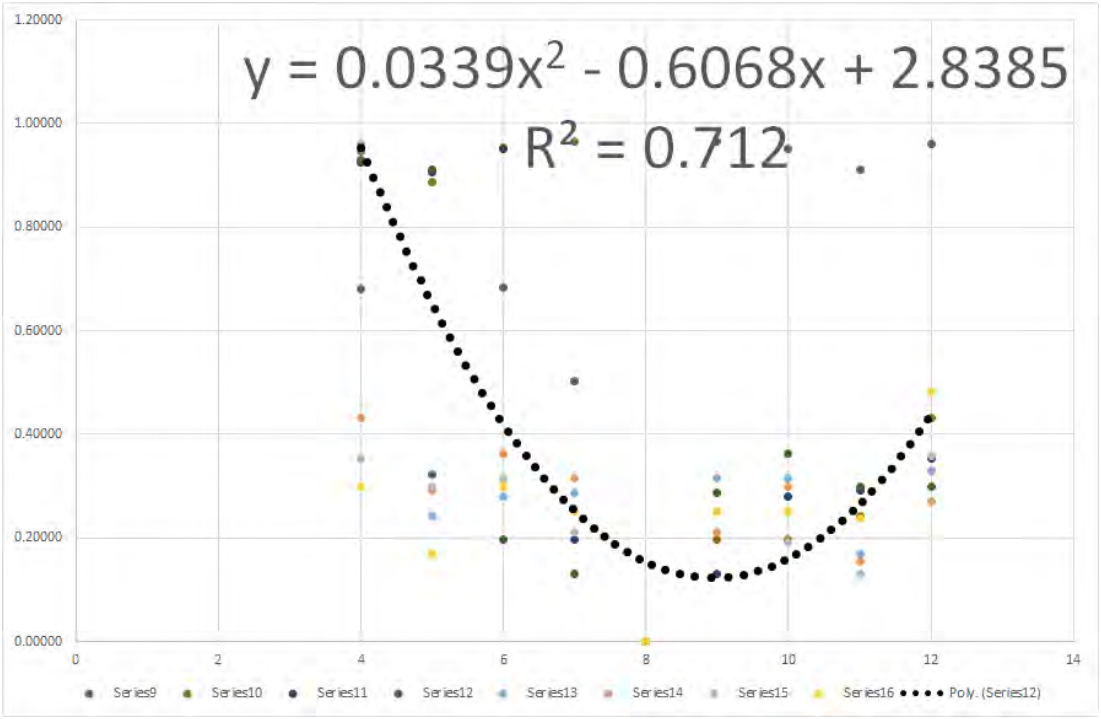


Figure F.84: Goodness of quadratic fit graph for displayed images for grammar 17 for pass-images 9 to 16

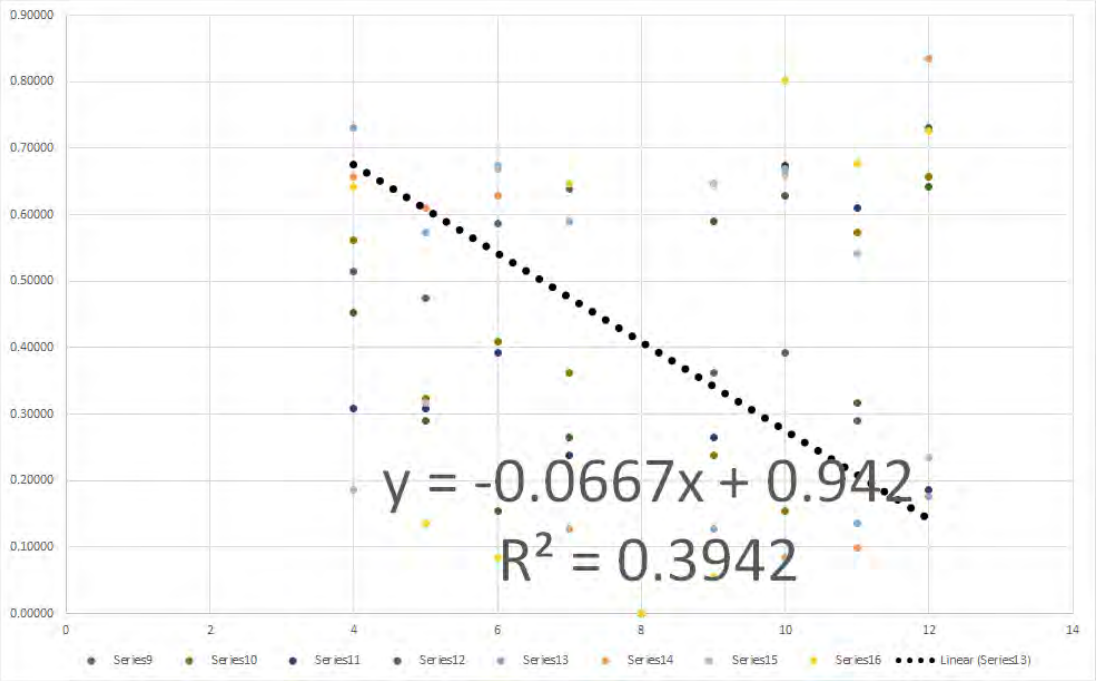


Figure F.85: Goodness of linear fit graph for displayed images for grammar 18 for pass-images 9 to 16

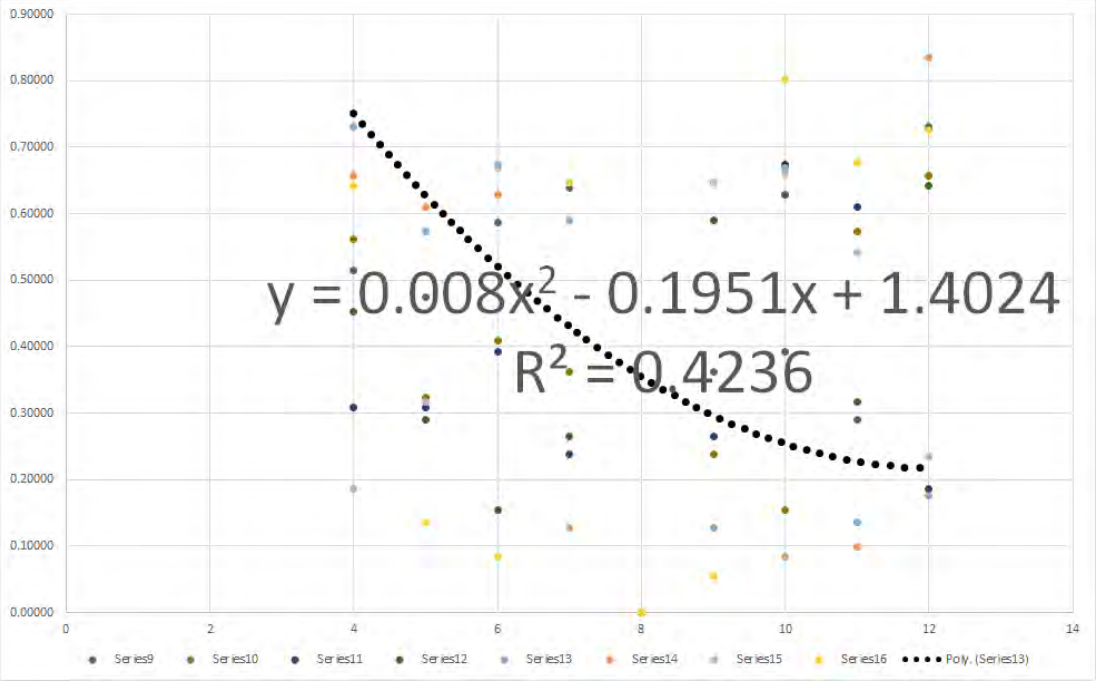


Figure F.86: Goodness of quadratic fit graph for displayed images for grammar 18 for pass-images 9 to 16

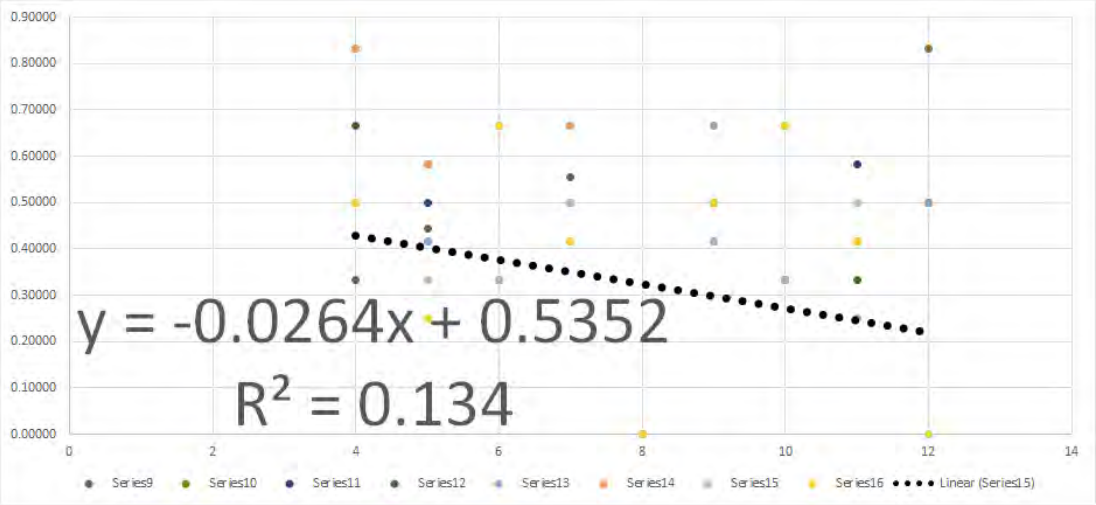


Figure F.87: Goodness of linear fit graph for displayed images for grammar 19 for pass-images 9 to 16

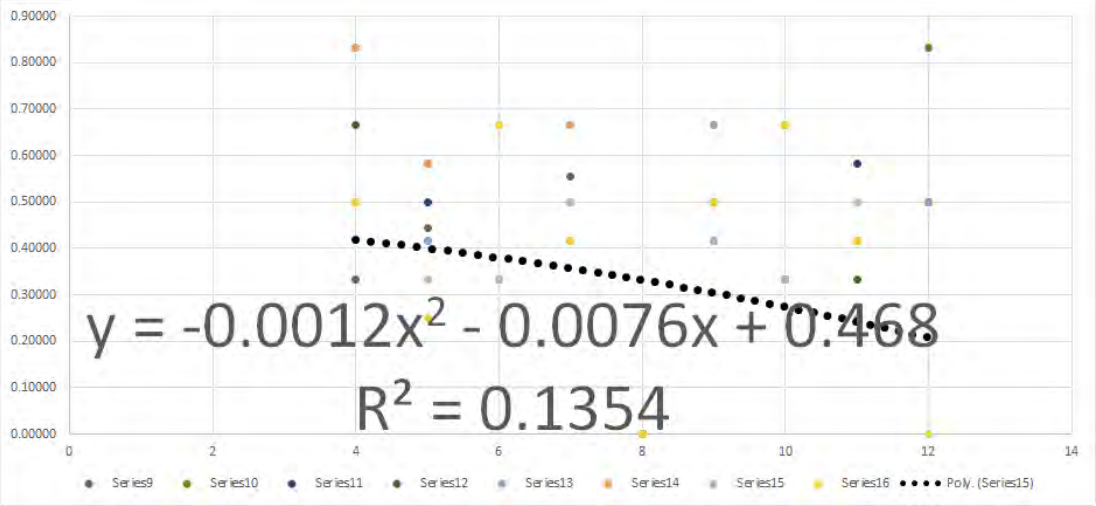


Figure F.88: Goodness of quadratic fit graph for displayed images for grammar 19 for pass-images 9 to 16

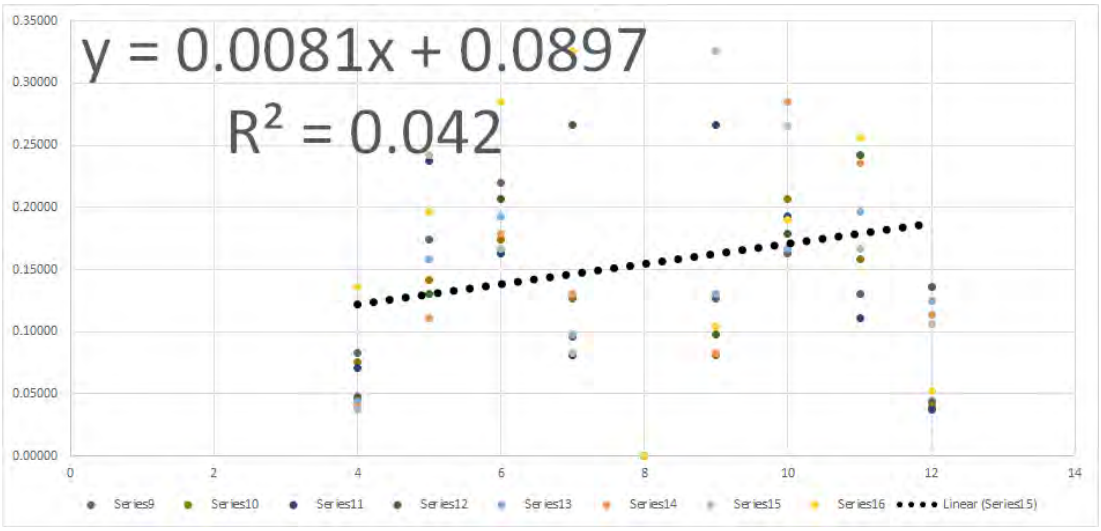


Figure F.89: Goodness of linear fit graph for displayed images for grammar 20 for pass-images 9 to 16

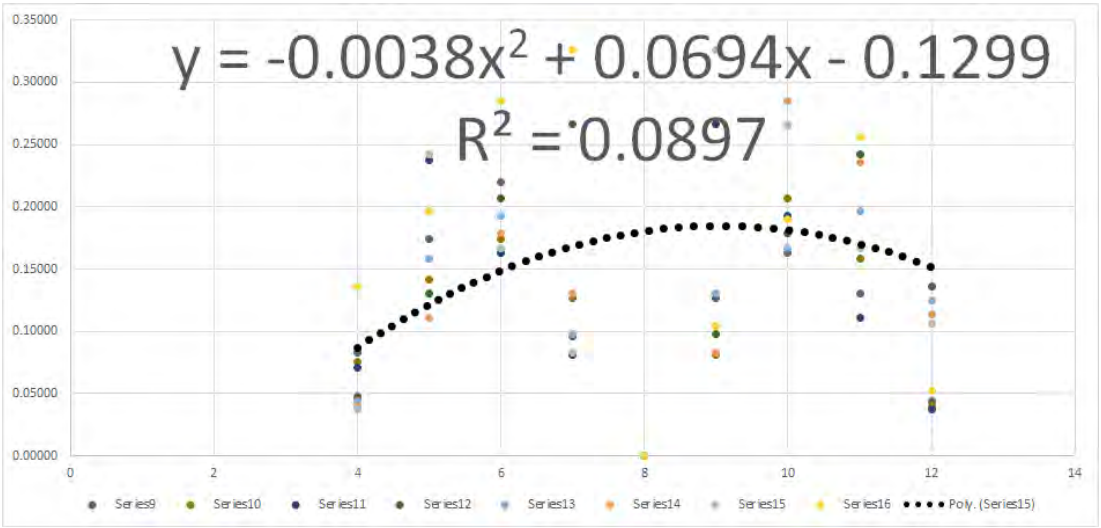


Figure F.90: Goodness of quadratic fit graph for displayed images for grammar 20 for pass-images 9 to 16

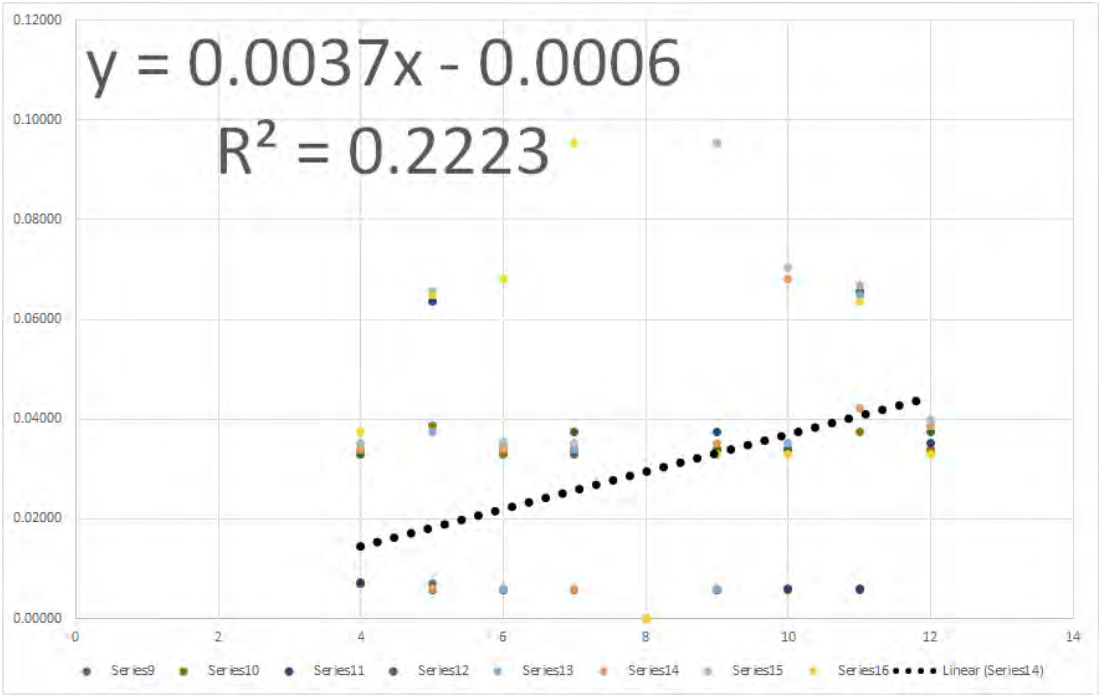


Figure F.91: Goodness of linear fit graph for displayed images for grammar 21 for pass-images 9 to 16

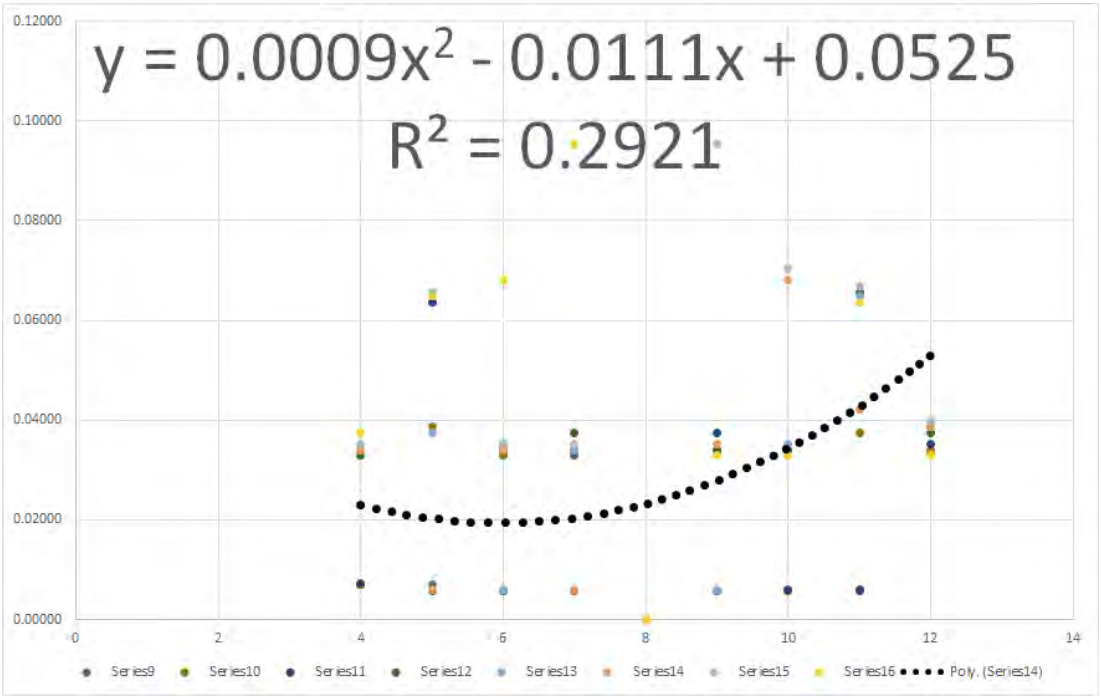


Figure F.92: Goodness of quadratic fit graph for displayed images for grammar 21 for pass-images 9 to 16

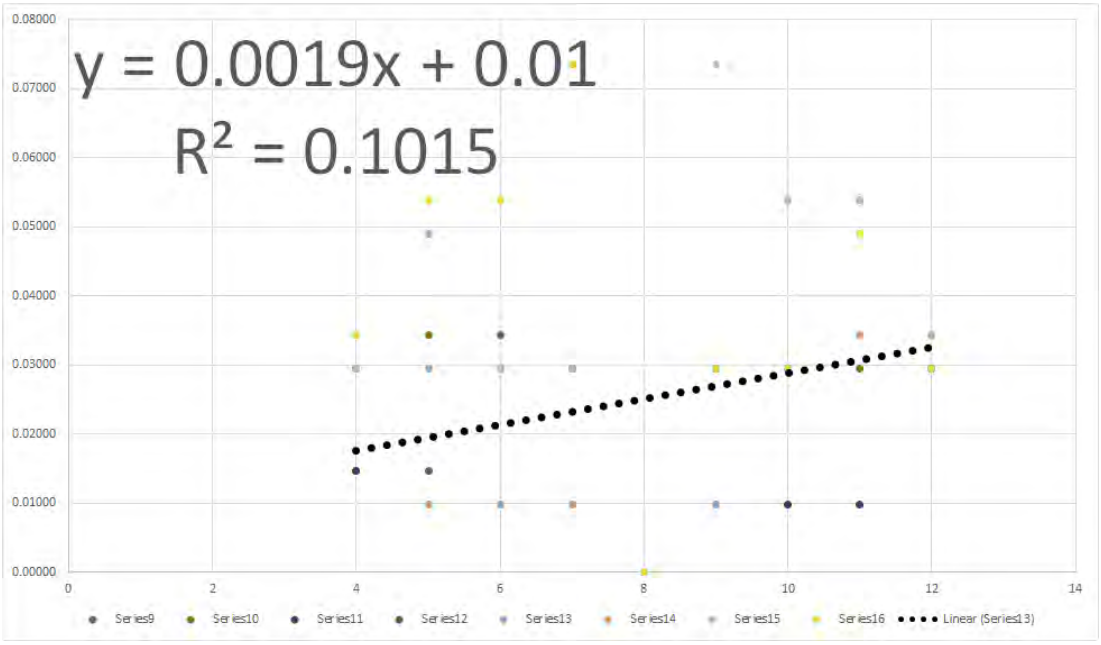


Figure F.93: Goodness of linear fit graph for displayed images for grammar 22 for pass-images 9 to 16

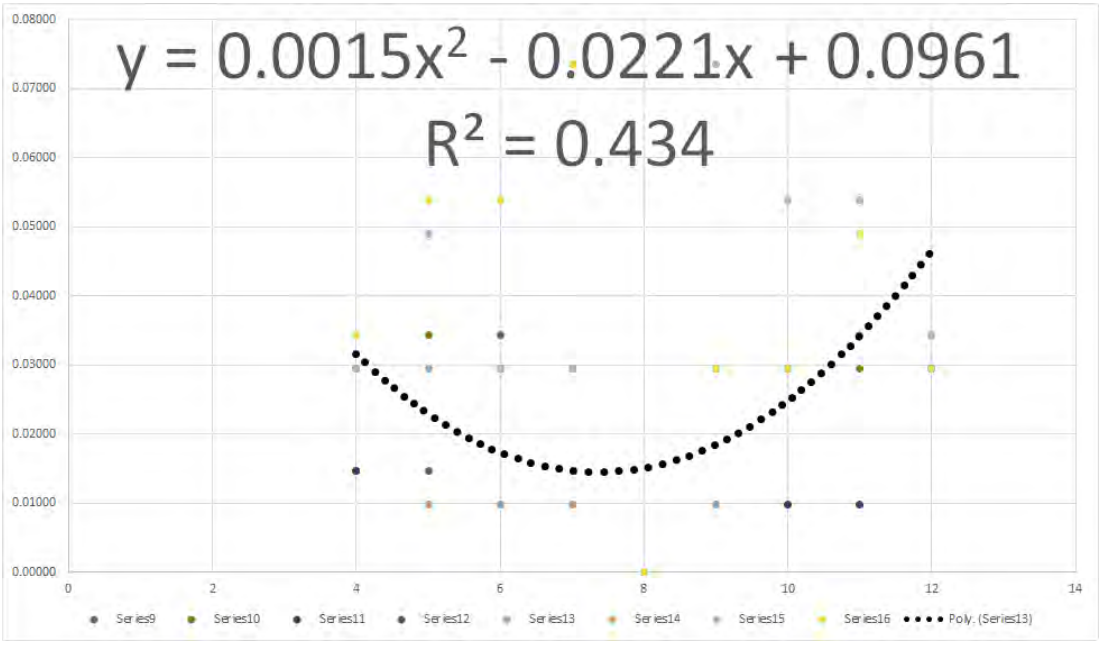


Figure F.94: Goodness of quadratic fit graph for displayed images for grammar 22 for pass-images 9 to 16

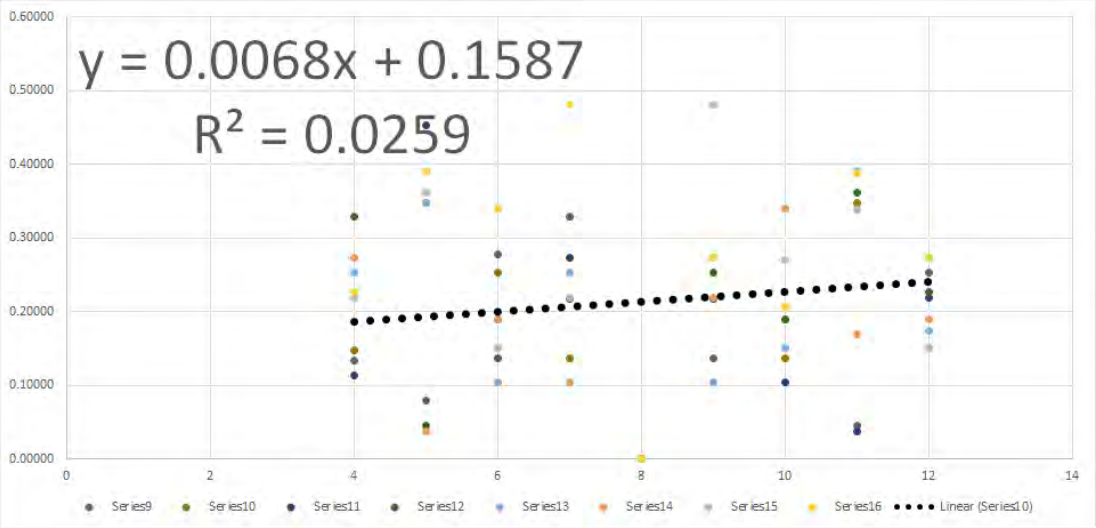


Figure F.95: Goodness of linear fit graph for displayed images for grammar23 for pass-images 9 to 16

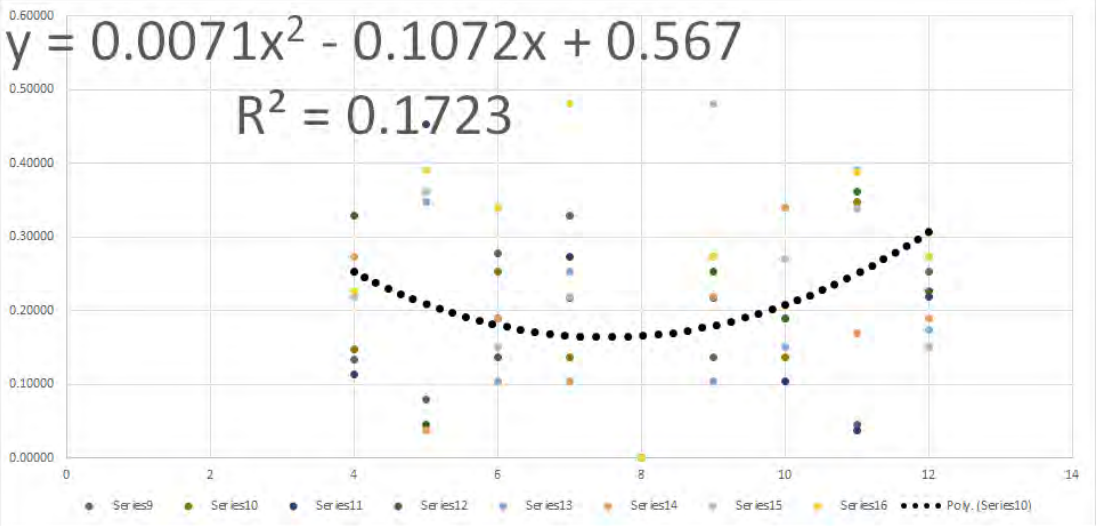


Figure F.96: Goodness of quadratic fit graph for displayed images for grammar 23 for pass-images 9 to 16

F.4 Table of goodness of fit values for all grammars

Table F.25: Goodness of fit values for all grammars

Quadratic goodness of fit test					Linear goodness of fit test		
grammar	a	b	c	R^2	m	c	R^2
0	0.0177	-0.2956	1.3125	0.8976	-0.0120	0.2965	0.0738
1	0.0022	-0.0332	0.2656	0.0381	0.0026	0.1376	0.0077
2	0.0345	-0.5454	2.3709	0.8164	0.0071	0.3914	0.0066
3	0.0089	-0.1267	0.5538	0.4241	0.0157	0.0435	0.0599
4	0.0068	-0.0898	0.8003	0.0761	0.0193	0.4094	0.0465
5	0.0117	-0.1918	0.8375	0.9071	-0.0051	0.0168	0.0327
6	0.0213	-0.3504	1.5431	0.8821	-0.0093	0.3209	0.0316
7	0.0235	-0.3875	1.7113	0.8795	-0.0113	0.3634	0.0380
8	0.0125	-0.2067	0.9065	0.9012	-0.0060	0.1873	0.0379
9	0.0121	-0.1998	0.8948	0.8535	-0.0058	0.1998	0.0369
10	0.0153	-0.2363	1.3183	0.2542	0.0082	0.4421	0.0135
11	-0.0130	0.2056	-0.2263	0.0288	-0.0021	0.5180	0.0001
12	-0.0022	0.0338	0.0238	0.0321	-0.0009	0.1479	0.0009
13	0.0194	-0.3180	1.3894	0.8933	-0.0074	0.2765	0.0247
14	0.0171	-0.2798	1.2173	0.8982	-0.0057	0.2351	0.0190
15	0.0315	-0.4863	2.0667	0.8321	0.0176	0.2608	0.0479
16	0.0123	-0.1644	0.6315	0.8327	0.0326	-0.0743	0.4802
17	0.0339	-0.6668	2.8385	0.7120	-0.0644	0.8947	0.2937
18	0.0080	-0.1951	1.4024	0.4236	-0.0667	0.9420	0.3942
19	-0.0012	-0.0076	0.4680	0.1354	-0.0264	0.5352	0.1340
20	-0.0038	0.0694	-0.1299	0.0897	0.0081	0.0897	0.0420
21	0.0009	-0.0111	0.0525	0.2921	0.0037	-0.0006	0.2223
22	0.0015	-0.0221	0.0961	0.4340	0.0019	0.0100	0.0102
23	0.0071	-0.1072	0.5670	0.1723	0.0068	0.1587	0.0259