

https://dspace.org/

Towards automated three-dimensional tracking of nephrons through stacked histological image sets

Bhikha, Charita

2016-03-17
http://hdl.handle.net/10539/20124

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

Towards Automated Three-Dimensional Tracking of Nephrons through Stacked Histological Image Sets

A dissertation submitted to the Faculty of Engineering and the Built Environment, University of Witwatersrand for the degree of Master of Science in Engineering.

Charita Bhikha

August, 2015

DECLARATION

I declare that this research proposal is my own unaided work. It is being submitted to the Degree of Master of Science in Engineering to the University of the Witwatersrand, Johannesburg. It has not been submitted before for any degree or examination to any other University.

……..

Signature

…….... day of …… year …….

ABSTRACT

The three-dimensional microarchitecture of the mammalian kidney is of keen interest in the fields of cell biology and biomedical engineering as it plays a crucial role in renal function. This study presents a novel approach to the automatic tracking of individual nephrons through three-dimensional histological image sets of mouse and rat kidneys. The image database forms part of a previous study carried out at the University of Aarhus, Denmark. The previous study involved manually tracking a few hundred nephrons through the image sets in order to explore the renal microarchitecture, the results of which forms the gold standard for this study. The purpose of the current research is to develop methods which contribute towards creating an automated, intelligent system as a standard tool for such image sets. This would reduce the excessive time and human effort previously required for the tracking task, enabling a larger sample of nephrons to be tracked. It would also be desirable, in future, to explore the renal microstructure of various species and diseased specimens.

The developed algorithm is robust, able to isolate closely packed nephrons and track their convoluted paths despite a number of non-ideal conditions such as local image distortions, artefacts and connective tissue interference. The system consists of initial image pre-processing steps such as background removal, adaptive histogram equalisation and image segmentation. A feature extraction stage achieves data abstraction and information concentration by extracting shape

descriptors, radial shape profiles and key coordinates for each nephron cross-section. A custom graph-based tracking algorithm is implemented to track the nephrons using the extracted coordinates. A rule-base and machine learning algorithms including an Artificial Neural Network and Support Vector Machine are used to evaluate the shape features and other information to validate the algorithm's results through each of its iterations.

The validation steps prove to be highly effective in rejecting incorrect tracking moves, with the rule-base having greater than 90% accuracy and the Artificial Neural Network and Support Vector Machine both producing 93% classification accuracies. Comparison of a selection of automatically and manually tracked nephrons yielded results of 95% accuracy and 98% tracking extent for the proximal convoluted tubule, proximal straight tubule and ascending thick limb of the loop of Henle. The ascending and descending thin limbs of the loop of Henle pose a challenge, having low accuracy and low tracking extent due to the low resolution, narrow diameter and high density of cross-sections in the inner medulla. Limited manual intervention is proposed as a solution to these limitations, enabling full nephron paths to be obtained with an average of 17 manual corrections per mouse nephron and 58 manual corrections per rat nephron.

The developed semi-automatic system saves a considerable amount of time and effort in comparison with the manual task. Furthermore, the developed methodology forms a foundation for future development towards a fully automated tracking system for nephrons.

ACKNOWLEDGEMENTS

I would like to thank my supervisors Robyn Letts, Prof. David Rubin and Adam Pantanowitz, for their invaluable support, advice, feedback, and constant interest and motivation.

I would also like to thank all members of the Biomedical Engineering Research Group for their inspiring discussions and stimulating ideas.

Thank you to all of my friends and family for providing support, enthusiasm and comfort throughout the course of my studies.

Finally, many thanks to the team at the Departments of Cell Biology, Connective Tissue Biology, and Neurobiology, Institute of Anatomy, University of Aarhus, Aarhus, Denmark, for providing the data which forms the core of this project.

CONTENTS

Declaration i

Abstract ii

Acknowledgement iv

List of Figures ix

List of Tables xii

List of Symbols xiii

List of Abbreviations xiv

1. Introduction **1**

2. Background **3**

2.1 An Overview of Renal Histology 3

2.2 Existing Solutions 5

2.2.1. Nephron Tracking and Three-Dimensional Reconstruction 5

2.2.2. Glomeruli Detection 6

2.2.3. Automated Tracking of other Biological Structures 7

2.3 The Nephron Tracking Problem 8

2.4 Graph Theory 8

2.5 Machine Learning 9

2.5.1. An Overview of Basic Machine Learning Principles 9

2.5.2. Application to Medical Imaging 11

2.5.3. Application to the Nephron Tracking Problem 12

3. Project Framework **14**

3.1 Research Question 14

3.2 Rationale 14

3.3 Objectives 16

3.4 Assumptions 16
3.5 Success Criteria 16

4. Analysis of the Problem Domain 17

4.1 The Image Sets Acquired from the University of Aarhus 17
4.2 An Ideal Solution 21
4.3 The Complexities of the Problem 21

5. System Overview 24

6. Image Processing 26

6.1 Image Registration 26
6.2 Image Processing Procedure 29
6.2.1. Conversion to Grayscale 29
6.2.2. Background Removal 29
6.2.3. Histogram Equalisation 31
6.2.4. Thresholding 32
6.2.5. Removal of Unwanted Cross-Sections 33
6.3 Image Segmentation 34
6.4 Automatic Parameter Variation 35

7. Feature Extraction 37

7.1 Node Allocation 37
7.2 Shape Measurements 39
7.3 Data Structures 44
7.4 Glomeruli Detection 45

8. Tracking Algorithm 48

8.1 Local Image Registration 50
8.2 Graph-based Tracking 52
8.3 Edge Formation 53
8.4 Skipping Images 54
8.5 Validation Steps 54
8.6 Region Control 56

8.7 Reconstruction 58
8.8 Manual Intervention 58

9. Machine Learning Validation 60

9.1 Feature Selection 60
9.2 Training Set Formation 61
9.3 Training 62
9.4 Reinforced Learning 63
9.5 Feature Analysis 64
9.6 Optimisation 66

10. Results 68

10.1 Pre-Tracking Stages 68
10.2 Measuring Similarity between Paths 70
10.3 Possible Outcomes 72
10.4 Tracking Results 74
10.5 Efficacy of Validation Steps 82
10.6 Machine Learning Classification 83
10.7 Monitoring Runtime Output 87
10.8 Processing Times 87

11. Analysis & Discussion 90

11.1 Performance per Area of the Nephron 91
11.2 Effect of Image Properties on Performance 93

12. Recommendations & Future Work 96

12.1 Recommendations for Future Image Sets 96
12.2 Future Work 99

13. Conclusion 100

References 102

Appendices

Appendix A: Longitudinal reconstructions

Appendix B: Additional Results

Appendix C: Nephron Tracking Spreadsheet

Appendix D: Performance Data

Appendix E: A Review of the Path Comparison Method

Appendix F: Additional Feature Analysis

Appendix G: Proof of Ethics Clearance

Appendix H: MATLAB Code

Appendix I: Research Article published in the journal *Computational and Mathematical Methods in Medicine*

List of Figures

Figure No. Page

2.1 Basic anatomy of the nephron 4

2.2 Viewing a nephron's path as a walk through nodes in 3D space. 9

2.3 The generalised process for machine learning algorithms 10

4.1 Examples of images in the cortex and the medulla 18

4.2 Labelled structures in sections through the cortex and inner medulla . 20

4.3 Examples of interfering physical artefacts in the image sets 22

5.1 A high level overview of the nephron tracking system 25

6.1 The image pre-processing outputs at each stage 26

6.2 An example of local non-rigid distortion 27

6.3 The issue posed by the four-polygon alignment method 28

6.4 The procedure for background removal 30

6.5 Visual effect of local and global histogram equalisation 32

6.6 An example of connective tissue cross-sections in the cortex 33

6.7 Parameter variation using custom sigmoid functions 36

6.8 Features of the datasets having an inherent sigmoidal characteristic... 36

7.1 K-means clustering of nephrons resulting in Voronoi cells 39

7.2 Processing of the shape profile data 42

7.3 The shape profiles relative to nodes on a cross-section 42

7.4 Comparison of shape profiles of a pair of nodes forming a move 43

7.5 The features extracted per cross-section in each image 44

7.6 A result of the glomeruli detection method 47

8.1 An activity diagram of the tracking algorithm 48

8.2 An example of a manually tracked rat nephron is shown 49

8.3 The efficacy of additional translational image alignment 51

8.4 An example of an area which cannot be aligned, introducing error..... 51

8.5 The concept of vertical and horizontal tracking 53

8.6 Examples of moves blocked by the distance validation rule..... 55

8.7 Examples of moves blocked by the bidirectional validation rule 55

8.8 Examples of moves blocked by the skipping validation rule..... 56

8.9 Formation of a region signal from the output of the region classifier.. 57

8.10 A graph of the relationships between error and automaticity 59

9.1 Labelling of the training examples 62

9.2 A schematic showing the method employed for reinforced learning... 63

9.3 Results of the RELIEFF feature selection method 65

9.4 Results of Principal Component Analysis 65

9.5 Measurement of the convergence of training accuracy 67

10.1 The variety of cases which could occur during pre-processing 69

10.2 The different cases which could occur during tracking 72

10.3 Examples of incorrect linkage to multiple structures 73

10.4 A histogram of the number of manual corrections required 77

10.5 Examples of premature termination during tracking 79

10.6 An example of an image after automatically tracking a rat PCT 80

10.7 A manually vs. automatically tracked mouse nephron..... 80

10.8 A manually vs. semi-automatically tracked mouse nephron 81

10.9 A manually vs. semi-automatically tracked rat nephron 81

10.10 Examples of true and false positives and negatives of the ANN 86

10.11 An example of an output log during the tracking of a nephron 88

10.12 A pie chart of the distribution of execution time among routines..... 89

11.1 The implication of a chosen slice thickness on tracking..... 93

11.2 Measurements of the changes in morphology for three image sets 94

12.1 An example of a mouse slide at a much higher resolution 97

12.2 Examples of a longitudinal and transverse slice through the kidney ... 98

List of Tables

Table No. Page

4.1 Characteristics of the average mouse and rat dataset 18

8.1 Different modes of tracking are created at transitions 57

9.1 The intermediate and final output classes of the learning functions 61

10.1 The segmentation accuracy of samples from 4 datasets 69

10.2 Test results on a chosen set of 16 mouse nephrons 75

10.3 Test results on a chosen set of 11 rat nephrons 75

10.4 The accuracies and invalid move rejection rate of the validations 83

10.5 The confusion matrix and accuracies of the ANN and SVM 84

10.6 The confusion matrix of the final classification of the test set 85

10.7 Various performance indicators for the ANN and SVM 85

10.8 The distribution of time among the main components of the code 88

10.9 The times taken to process cortical and medullary images 89

11.1 A summary of the implications and effects of artefacts 91

11.2 A high-level ceiling analysis of the system 95

List of Symbols

Vectors are indicated by variables in bold.

z	Image number
f_i	Shape factor i where i={1,…,6}
K	Number of clusters or centroids or nodes per segment
f	Nephron number
	Number of observations
m_V	Number of examples
	Number of elements in a vector V
n	Number of features
i_z	Identity number of a single nephron cross section in image z
r	Residual
	Radius in shape profile
X	Input for machine learning algorithm
Y	Output for machine learning algorithm
$\boldsymbol{\Psi}$	Automatically tracked path
$\boldsymbol{\Upsilon}$	Manually Tracked Path
α	Accuracy
β	Extent
θ	Polynomial coefficients in machine learning
	Angle in shape profile
δ	Angle increment for shape profile
Iz	Image z
C	General constant
	Set of centroids
T_{bgrnd}	Threshold for background removal

List of Abbreviations

3D	Three-dimensional
2D	Two-dimensional
PCT	Proximal Convoluted Tubule
PST	Proximal Straight Tubule
DTL	Descending Thin Limb
LH	Loop of Henle
ATL	Ascending Thin Limb
TAL	Thick Ascending Limb
DCT	Distal Convoluted Tubule
ICT	Interstitial connective tissue
BV	Blood vessels
ANN	Artificial Neural Network
SVM	Support Vector Machine
ML	Machine Learning

CHAPTER 1

Introduction

The kidney performs the vital bodily functions of water and solute exchange, blood pressure regulation and urine concentration through the functional unit of the nephron. Approximately one million nephrons intricately populate each human kidney, producing distinct regions with differing functionalities [**1**] [**2**]. The spatial distribution of nephrons within the kidney forms its microarchitecture.

The microarchitecture of the kidney has recently been the focus of a number of studies [**3**] [**4**] [**5**]. In particular, the functional implications of the renal microstructure on the underlying physiological mechanisms involved are of great interest [**6**] [**7**] [**8**]. Nephrons are the target for many drugs which regulate blood pressure and solute concentrations and hence important bodily functions [**2**]. A deeper understanding of its anatomy may lead to a better understanding of physiological function and disease, which may be beneficial to drug development, disease diagnosis and treatment.

A deeper characterisation of the microarchitecture also enables further development of models and simulations that accurately describe the functionality of the nephron and the kidney. This is a fundamental step towards the development of an artificial kidney or dialysis device. For researchers studying and modelling kidney function, some of the most useful statistics are the ratio of long-looped nephrons to short-looped nephrons and the change of this ratio across different individuals and species, the distribution in lengths within these categories, and the relative lengths of different parts of the nephron [**7**] [**9**].

A previous study carried out by the Department of Biomedicine at the University of Aarhus, Denmark, involved manually tracking the paths taken by a few hundred nephrons through histological image sets of mouse and rat kidneys, and thereafter performing an in-depth analysis of the findings [**9**] [**10**] . The manual tracking task required an exhaustive amount of time and effort per dataset, which

posed a limit on the amount of data that could be acquired. This created the need for an automatic tracking tool which could be used as a standard tool on multiple datasets. This would allow the renal characterisation of multiple species as well as diseased specimens. Since the microstructure of nephrons can vary in the same kidney, it is important to obtain large samples when taking measurements such as nephron lengths, in order to render the findings more statistically accurate and representative of a variety of kidney specimens.

The image database [**11**] has been made available for use through collaboration between the University of Aarhus, Denmark, and the University of the Witwatersrand, Johannesburg. This study attempts to aid and improve the process of modelling the renal microstructure by creating an automatic software tool to track nephrons through the image sets. The manually tracked nephrons form the gold standard comparison for this study.

Various potential methodologies have been investigated and tested. The system developed in this dissertation comprises three main stages; image pre-processing, feature extraction and nephron tracking. Machine learning algorithms have been employed to accurately guide the tracking algorithm. The final system is semi-automated, occasionally requiring user input for tracking in the inner medulla where the small size, dense nephron cross-sections prove to be difficult to track automatically.

Chapter 2 introduces basic concepts of the kidney on both macroscopic and microscopic scales in order to highlight details that are relevant to the problem. An overview of existing methodologies in related fields is also discussed. The aims, objectives, rationale and scope of the study are presented in Chapter 3. Particular characteristics of the images which make the tracking task complex and introduce a number of non-ideal factors are explored in Chapter 4. A brief overview of the system is included in Chapter 5, and the developed methodology consisting of the stages of image pre-processing, feature extraction and tracking is detailed in Chapters 6, 7 and 8, respectively. Chapter 9 is dedicated to the machine learning aspects of the system. Results are presented in Chapter 10, followed by a detailed analysis and discussion in Chapter 11. Final conclusions are drawn and recommendations made in Chapter 12.

CHAPTER 2

Background

This chapter serves to provide background knowledge on concepts and fields relevant to this study, and to explore existing solutions, methodologies and applications.

2.1 An Overview of Renal Histology

A basic understanding of the anatomy and histology of the kidney is required in order to correctly model the problem, identify structures in the images and interpret results of the study in light of their biological implications.

The kidneys are a pair of bean-shaped organs lying posteriorly in the abdominal cavity [**12**]. From a high-level perspective, one of the main functions of the kidneys is to take in unfiltered blood, and produce urine and filtered blood as outputs. This filtering and reabsorption function is performed by the kidney's functional unit called the nephron. Approximately 1 million nephrons populate each human kidney [**13**].

A nephron is a long, tortuous, unbranched tubular structure, varying in diameter along its length [**1**]. Its length is broken up into seven parts, namely the proximal convoluted tubule (PCT), proximal straight tubule (PST), descending thin limb (DTL), ascending thin limb, (ATL), thick ascending limb (TAL) and distal convoluted tubule (DCT) [**1**]. The nephrons are arranged such that the PCT, PST, TAL and DCT occur in the outer part of the kidney called the cortex, while the DTL and ATL form loops of Henle in the inner region called the medulla [**1**], as illustrated in Figure 2.1. Water and various solutes are exchanged between the filtrate and the blood along the length of the nephron [**14**].

A glomerulus and Bowman's capsule (making up a renal corpuscle) occurs at the start of each PCT; this is the site at which blood is filtered to form the renal

filtrate which fills the nephron tubule lumen. The renal corpuscle has a vascular pole at which the glomerulus meets blood vessels (afferent and efferent arterioles) and a urinary pole where the Bowman's capsule fuses with the nephron tubule [**1**]. The glomeruli are clearly visible in the image sets.

At its distal end, each DCT joins a collecting duct which is a common structure collecting the filtrate from a family of nephrons [**1**]. This is the only site at which branching will be seen in the nephron network [**1**]. The collecting ducts drain into the minor and major calyces of the kidney, which then empty into the ureters and subsequently the bladder.

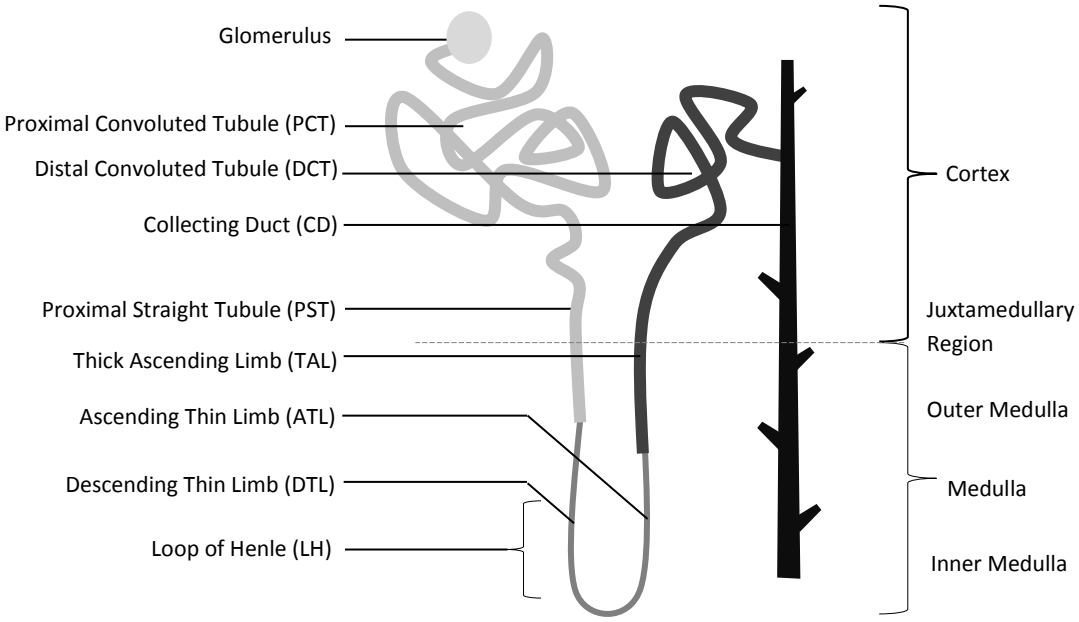


Figure 2.1: Basic anatomy of the nephron. Adapted from **[**1**]**.

Toluidine blue is the dye used in preparation of the image sets. It is a basic stain commonly used in renal pathology [**15**]. It has a high affinity for acidic tissues, producing a bluish purple stain [**16**]. It also increases the sharpness of histological images [**16**]. In a typical Haematoxylin and Eosin (H&E) stained kidney specimen, the various parts of the nephrons can be distinguished by the number of nuclei, diameter, thickness of the wall and types of cells making up the tubule [**1**]. The given images stained with toluidine blue results in the diameter and wall thickness being the only differentiating features.

The nephrons are in close contact with the renal blood supply in order to perform the filtering and solute exchange functions [**1**]. The arteries, veins and capillary networks are seen in the image sets, having varying sizes and are more irregularly shaped compared to the nephrons. However, many blood vessels, especially those emerging to and from the glomeruli, are very similar in appearance to the nephrons and may be confused.

The presence of loops of Henle in the inner medulla and the convolutions in the cortex are high-level examples of structure influencing renal function [**14**]. Looking closer, there are cortical nephrons with short loops and juxtamedullary nephrons with long loops. These have differing filtering rates [**10**]. Deeper characterisation of the renal microarchitecture may reveal additional structural aspects which have important functional implications.

2.2 Existing Solutions

2.2.1 Nephron Tracking and Three-Dimensional Reconstruction

The spatial distribution of nephrons has been explored in previous studies although all instances of tracking were performed manually and therefore the resulting statistics were based on a limited number of nephrons. The mouse or rat kidney is commonly used as it is small enough to fit on microscopic slides while adequately representing the structure of mammalian kidneys.

One of the previous studies carried out at the University of Aarhus, Denmark (from which the image sets were obtained) involved reconstructing 151 complete nephrons from the manually tracked data of a mouse kidney [**10**]. The tracking was done on 30 families of nephrons, where a family refers to all nephrons emptying into a common collecting duct [**10**]. The glomeruli were used as starting points. A number of statistics were calculated and the spatial interrelations of each part of the nephrons were thoroughly discussed, revealing some important features of the kidney [**10**]. A later study involved manually tracking 56 nephrons of a rat kidney and taking a variety of measurements such as the lengths of different parts of the nephron and glomerular volumes [**9**]. Computer-aided 3D reconstruction was also carried out for visualisation purposes.

In a different set of studies by Pannabecker and Dantzler [**4**] [**5**], the 3D architecture of the rat kidney was investigated. Various cross-sections of rat nephrons were physically labelled using differential staining/immunocytochemistry techniques. Immunofluorescence allowed visual differentiation between parts of the nephron by means of distinct fluorescence during microscopy. The digitised images were used to manually track the TDL and TAL near the papillary tip. 3D reconstruction involved creating a mesh of three-dimensional cylinder-like objects which were created for each individual nephron cross-section in each image. Existing imaging software called Amira visualisation was used. Although immunofluorescence aided the tracking process, the tracking procedure was not automated in any obvious manner.

Both these sets of studies involved manually tracking nephron cross-sections in different areas of interest in the kidney. The tracking processes were computer-aided in the sense that the software provided a user-interface; the tracking was not automated or predictive and no machine learning was used.

2.2.2 Glomeruli Detection

The glomeruli need to be detected as they serve as good starting points for tracking. Automated glomerulus detection is an important step during computer-aided diagnosis of kidney disease during a biopsy [**17**]. The change in size and shape of glomeruli is an indicator of the degree of damage in the kidney [**17**]. The biggest challenge for accurate detection is the fact that the surrounding contours are not continuous [**18**] and that other surrounding tissue produce strong noise levels [**17**]. The shape and size of the glomeruli also vary.

A set of papers [**19**] [**20**] document using a log edge detector and wavelet transform to produce a low resolution image with enhanced glomeruli edges. Spline curve fitting is applied through a genetic algorithm to obtain an accurate closed curve around the glomeruli. Another study [**17**] has shown that the watershed algorithm can produce a more accurate closed glomerulus edge.

These methods often require a starting seed and are not suitable for purely automated glomeruli detection. The images in this study differ widely from those used in other studies (usually H&E images). In contrast to images in previous

studies, the nephrons produce stronger edges than glomeruli. Also, accurate closed curves around the glomeruli are not necessarily needed, merely indicate coordinates. A custom glomerulus detection method is therefore devised for this study.

2.2.3 Automated Tracking of other Biological Structures

It is important to note the difference between automatic tracking and automatic segmentation. Automatic segmentation is the isolation of independent structures in images, such as the separation of organs in CT and MRI images [**21**] [**22**], or the differentiation between tissue types in histological images, mostly for purposes of visualisation or further processing. The segmentation can be pixel (2D) or voxel (3D) based. Commonly employed techniques for segmentation include edge detectors [**23**], histogram-based methods, the watershed transform, region growing [**21**], morphological operations and active contour modelling [**22**].

In contrast, automatic tracking utilises segmentation results to create an abstract computational reconstruction of the structure for purposes of accurate measurement. Currently, there exists no method for the automatic tracking of nephrons through serial slices. However, methods for the automatic tracking of other biological structures do currently exist, although these are for one or a few objects in a single image.

A common example is the tracking of blood vessels in retinal images [**23**]. One study [**24**] makes use of a Kalman filter as the basis for tracking, using the position and orientation of vessel fragments as states. Gradient information and expected vessel structure are used to estimate the next state during tracking. System noise is also taken into account. A number of verification or correctness checks specific to the problem are used to improve results [**24**]. Another study [**25**] uses correlations with rotated templates to track vessels iteratively in local pixel areas, in order to avoid image-wide operations which are generally slow.

The portal and hepatic venous trees of the liver has also been automatically tracked. One approach uses Laplacian-based contraction to obtain a skeleton of the vessel system [**26**], which is then broken up into nodes. Tracking consists of

using orientation and diameter consistency metrics to model continuity between nodes. Maximisation of the continuity function provides the best candidate.

2.3 The Nephron Tracking Problem

The methods from the aforementioned applications cannot be directly applied to the current nephron tracking problem due to a number of factors. The nephrons are sectioned transversely, enabling one to track individual nephron cross-sections from image to image. In contrast, retinal images and CT images of the hepatic venous tree capture a single longitudinal view of the entire structure in question. Another crucial difference is the vast number of independent nephrons needing tracking versus one or a few structures in other applications. Moreover, the tortuosity of the nephrons poses a major challenge. The vast amount of data (700-3000 high resolution sections through the kidney per dataset) also poses a limitation on how the data is to be processed in an efficient manner.

Although existing methodologies cannot be used directly and completely to fulfil the requirements of the automated nephron tracking problem, several of the methods have been adopted and combined in the current approach. This includes graph-based tracking, various metrics to indicate confidence per iteration and a set of validation rules to eliminate error. In addition to this analytic heuristic technique, the high modelling capability of machine learning is employed for path validation. Machine learning is highly appropriate for such a problem as it can automatically model the complex system with high accuracy through training. The machine learning component is discussed in greater detail in Section 2.5.

2.4 Graph Theory

The primary structure of the designed tracking algorithm in this study adopts basic concepts used in graph theory as described by [**39**] and is summarised below.

A graph (G) consists of a collection of *nodes* (V) interconnected through *edges* (E). In general, a node is an object which possesses certain attributes. An edge connects two nodes, establishing a relationship between them, i.e. G = (V, E) and E = (V_1, V_2). An edge can be undirected or directed where the edge points from a *parent* node to a *child* node. Each node has a potential to have 0-1 parent/s and 0-

n children. In terms of nephron tracking, each individual nephron cross-section can be seen as a node. The ‘nodes’ are then progressively linked, or tracked, to form a list of parent-child pairs.

A *walk* is a sequence of nodes and edges as shown in Figure 2.2. Given a set of directed edges, a walk can be reconstructed through inference of the parent-child pairs. The resultant nephron path can be seen as a bidirectional *walk* in 3D space through the nodes making up a nephron, starting at some initial seed and ideally ending at the glomerulus and collecting duct.

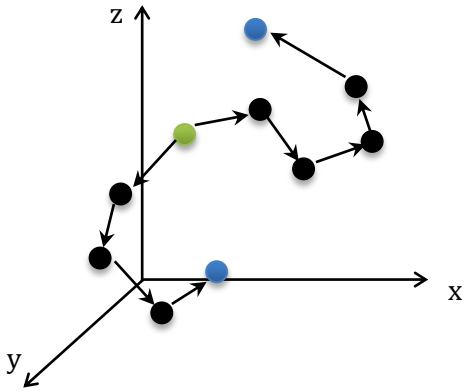


Figure 2.2: A nephron’s path can be seen as a walk through a set of nodes in 3D space. The walk occurs in two directions from a starting seed (green) towards endpoints (blue) which should be a glomerulus and collecting duct.

2.5 Machine Learning

2.5.1 An Overview of Basic Machine Learning Principles

A machine learning algorithm forms a hypothesis, or a prediction function, based on experience through given inputs and outputs [**27**], i.e. a training set {**X**,**Y**}. Once a learning algorithm has been trained, it can be used to predict new unseen examples. The process is summarised in Figure 2.3.

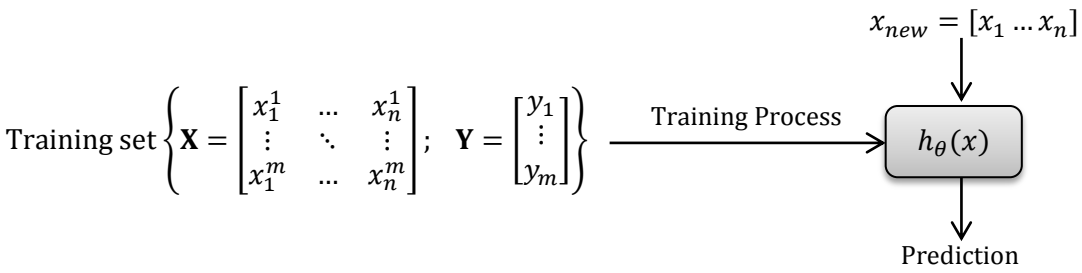


Figure 2.3: The general process followed when using machine learning algorithms. $h_\theta(x)$ is the prediction function.

The weights (**θ**) of the generalised polynomial function $h_\theta(x)$ as in equation (2.1) are adjusted with each example, such that some cost/error objective function as in equation (2.2) is minimised [**27**]. This is done through methods such as gradient descent and back-propagation [**27**], and is termed 'learning'. Popular learning algorithms include Logistic Regression, Decision Trees, Bayesian Classifiers and many more [**28**].

$$h_\theta(x) = \text{sigmoid}(\theta_1 x_1 + \theta_2 x_2 + \cdots + \theta_n x_n) = \text{sigmoid}(\boldsymbol{\theta}^T \mathbf{X}) \quad (2.1)$$

$$\boldsymbol{\theta} = \arg\min_\theta \frac{1}{m} \sum_{i=1}^{m} y^i \log\left(h_\theta(x^i)\right) + \left(1 - y^i\right) \log\left(1 - h_\theta(x^i)\right) \dots + \frac{\lambda}{2m} \sum_{j=1}^{n} \theta_j^2 \quad (2.2)$$

where x^i and y^i are the input features and output of the *i*th example, respectively. *m* is the number of examples and *n* is the number of features.

Learning can be supervised, where the correct outputs are provided [**29**], or unsupervised, where intrinsic patterns are sought for within the given data [**29**]. A supervised problem may be of a regression type, where there is a continuous valued output, or a classification type, where the output is a discrete label [**27**]. Randomisation and normalisation (feature scaling) of the input is essential for good results during training [**28**]. Once the machine learning algorithm is well-trained, it can be used to classify new, unseen examples.

An underfit hypothesis is one that is too simple or of a low order [**28**]. It has high bias and cannot even represent the training set well. An overfit hypothesis is one that has too high an order. It works very well for the training set but cannot

accurately predict new examples [**28**]. It is said to have high variance as it captures noise and outliers [**28**]. The regularisation parameter of a machine learning algorithm controls the level of generalisation of the hypothesis and can be adjusted to address under- or over- fitting [**28**]. Additional features or polynomial features can also solve a high bias problem, while decreasing features and adding more training examples can resolve overfitting. In addition to the training set, a validation and test set is also used during training to prevent bias towards the training set.

Additional theory on machine learning can be found at [**27**] and [**28**].

2.5.2 Application to Medical Imaging

Artificial intelligence, or machine learning, has found application in the medical imaging field. It is particularly advantageous because biological structures cannot usually be described with high accuracy through simple predictive equations. Large modelling capacity combined with flexible input and output choices make these algorithms highly desirable.

Feature-based machine learning (FML) involves computing features of objects in the images which are then used as inputs to the machine learning algorithm. The output is typically not in the image space but rather a classification or numerical value [**30**]. One such application involved using a multi-layer perceptron neural network to classify breast lesions as either malignant, fibroadenoma, fibrocystic disease or benign [**31**]. Features such as cellularity, cohesiveness, clump thickness and uniformity were computed from images of fine needle aspirate smears [**31**].

Pixel, or voxel, based machine learning (PML) uses image pixels as direct inputs, or features. PML can automatically infer features and hence reduces error and data loss that occurs through feature extraction [**30**]. The output can be a classification or a processed image containing, for example, a detected coordinate, boundary curve or enhanced object [**30**]. For example, a feed-forward neural network has been used to aid detection of boundaries during automatic segmentation of the colon in CT images [**32**]. Using a processed binary image as an input, the network is able to extract fluid filled regions of the colon [**32**]. The training time and

computational power required for PML is very large due to the high dimensionality produced by image inputs.

2.5.3 Application to the Nephron Tracking Problem

The nephron tracking problem has a large number of inputs (either raw or processed images, or features such as shape, colour, position, size) and complex unknown functions. A non-linear, high dimensional machine learning algorithm is able to model these functions through supervised learning on the datasets.

For this study, two supervised classifiers are chosen for performance comparison. These are an Artificial Neural Network (ANN) and a Support Vector Machine (SVM), which are the most popular and powerful non-linear machine learning algorithms [**28**]. Both ANNs and SVMs are capable of modelling complex systems with high accuracy through supervised learning.

An ANN is a biologically inspired non-linear machine learning algorithm. It consists of multiple calculating units called neurons, each of which outputs a weighted sum of its inputs [**27**]. The neurons are arranged into multiple interconnected layers. Propagation of the input through the layers results in an intricate interrelation of the inputs dependant on the weighting factors at each neuron [**27**]. ANNs are capable of representing highly complex hypotheses, as the input features are progressively mapped into more complex features by the deeper layers [**27**]. It mimics the notion of the brain's plasticity, using 'one algorithm for all learning' [**28**].

An SVM is one of the most powerful machine learning algorithms available [**28**]. It is sometimes cleaner than logistic regression and ANN for complex hypotheses [**28**]. It is also known as a large margin classifier as it maximises the distance of the boundary from the examples. It uses computed features called landmarks, which can be computed with or without a variety of kernels.

For each of the algorithms, regularisation, the chosen features, the number of examples and the degree of the hypothesis need to be carefully chosen to optimise performance. It has been shown that most algorithms in the same class seem to

perform equally well, provided that there are a large number of training examples (>10000) [**28**].

The large amount of data available (3 sets of mouse kidneys of ≈1000 images/set and 3 sets of rat kidneys of ≈4000images/set) means that sufficient training can occur in order to optimise a highly complex hypothesis function. Some sets can be used for training and others for independent testing. Multiple datasets will result in an algorithm that is not an over-fit to one particular set of images.

Feature-based learning is chosen as image-based outputs are not required (nephron detections are more easily obtained through other methods due to the homogenous, easily identifiable nature of the cross-sections). The approach is to use the machine learning classification as a validation step post-tracking. This also reduces the dimensionality of the problem.

ANNs have low transparency – the optimised weightings cannot be easily interpreted to infer a model of the system. SVMs are slightly more transparent as the landmarks and margins can be interpreted [**28**]. However, transparency is not an issue, as this problem does not require an understanding of the underlying mathematical model. The algorithm simply relies on an output of tracking accuracy for purposes of path validation.

CHAPTER 3

Project Framework

3.1 Research Question

The work presented in this study forms part of a larger research goal, which aims to aid the process of exploring the spatial microstructure of the kidney in order to advance research findings in the fields of renal physiology and pathology. It also aims to verify the existing conclusions drawn from the small sample of nephrons in the previous study, on a larger more representative sample set.

In terms of this study, the research aims to:

- Determine how 3D structures, or representations, of individual nephrons can be automatically extracted from serial slices of the kidney.
- Develop methodologies towards an automated nephron tracking system.
- Determine how effectively and accurately an automated approach to tracking can be compared to the manual method.
- Quantify how much manual intervention is necessary in the automatic approach to obtain the paths of entire nephrons.

Once tracked, the results can be processed to extract useful statistics or reconstruct a 3D representation of the renal microstructure.

3.2 Rationale

Why does software need to be developed?

The manual tracking problem requires an exhaustive amount of effort per dataset. Each mouse and rat dataset has on average 1000 and 3000 images, respectively. Manually tracking one long-looped mouse nephron requires tracking about 1800 nephron cross-sections. This poses a limit on the amount of data that can be acquired (the number of nephrons and kidneys analysed). This creates the need for an automatic tracking algorithm which could be used as a standard tool on

multiple datasets, requiring little to no human effort apart from operation and occasional human intervention in the tracking process, although this should be minimised. Larger sets of results are needed in order for the extracted characteristics to be statistically representative of all kidney specimens.

Methods for 3D reconstruction from 2D images have been widely established, such as in 3D magnetic resonance (MRI) and computed tomography (CT) scans. However, these tools are not suitable to this problem due to a number of reasons:

- The problem is not limited to visualisation, but requires accurate measurements to be made per nephron, which requires accurate tracking of each individual nephron's cross-sections through the images.
- Existing tools are adapted to isolating only a few objects with relatively simple shapes/contours, e.g. the gross structure of the liver or heart, whereas the kidney has thousands of densely packed, intertwined nephrons, each of which takes a tortuous path in 3D space.
- MRI/CT image sets are not typically as large in volume (hundreds to thousands of high resolution images for the kidney data sets). This poses a challenge in terms of memory.
- Due to the large number of intertwined nephrons surrounded by interstitial tissue, generic algorithms could very easily incorrectly link nephrons or misjudge the correct path.

Existing software packages could perhaps be used on the results of the tracking algorithm rather than the raw images for purposes of visualisation. This research focuses on the development of the methods required for automated tracking rather than research on aspects of nephrology. Developing these methods is an essential step towards fully automated nephron tracking.

The resources that were required for this research project include the image sets and software development tools both of which were readily available. In particular, MATLAB Version R2012a [**33**], the Image Processing Toolbox, Neural Network Toolbox and Statistics Toolbox were used.

3.3 Objectives

The designed system needs to be:

- Automated to a high degree: Minimal effort must be required for setup and calibration, and user input must be minimised during tracking.
- Robust: It is able to track the convolutions in the tortuous path of nephrons and is capable of handling a wide range of cases, accommodating variability in the input data.
- Intelligent: The system makes use of modern techniques and makes informed decisions through computed models rather than depending on hard-coded rules.
- Practical: The code is reasonably efficient and user interaction is made easy.

3.4 Assumptions

- For purposes of verification, the assumption is made that the manually tracked data is absolutely correct. The accuracy of the algorithm will be measured against this gold standard. Visual inspection can also be used to verify results on nephrons which have not been previously tracked.
- The algorithm is only expected to work for datasets with reasonably clear data, which follows the constraints outlined in Section 12.1.2.
- A few parameters can be adjusted at the start of automated tracking in order to optimise the code for a particular dataset, i.e. calibrate the system to the input.

3.5 Success Criteria

The solution will be deemed successful if:

- The algorithm is able to track large portions of the paths of the manually tracked nephrons in an automatic manner.
- Complete nephron paths can be obtained using limited manual intervention.
- The algorithm works with a variety of datasets, with a minimal number of parameters needing to be adjusted.
- The algorithm has high specificity and sensitivity.
- The results can be used to provide a visual representation of the spatial distribution of the nephrons.
- The developed methodologies contribute to future work in this field.

CHAPTER 4

Analysis of the Problem Domain

In order to construct a working solution to the problem, the available data must first be analysed to identify requirements, constraints and limitations posed by the images.

4.1 The Image Sets Acquired from the University of Aarhus

The image sets acquired from the University of Aarhus, Denmark consist of images from three mouse kidney specimens and three rat kidney specimens. According to their previous studies [**9**] [**10**], tissue blocks were cut from each of the six kidneys perpendicular to the longitudinal axis extending from the cortical capsule to the papillary tip [**10**]. The tissue blocks were then fixed with glutaraldehyde, post-fixed with OsO4, stained én bloc with uranyl acetate, and embedded in flat molds in Epon [**9**] [**10**]. The blocks were then sliced transversely into consecutive sections using a microtome equipped with a Diatome histoknife [**9**]. Each slice was then stained with toluidine blue [**9**], digitised using a microscope and digital camera and labelled sequentially. A custom software interface was used for the manual tracking and labelling task; which is discussed in detail in [**9**] [**10**].

The animal experiments were carried out in accordance with the animal care license provided by the Danish National Animal Experiments Inspectorate [**9**] [**34**] (ethics clearance number 2004/561-818). Due to the work being purely computational, additional ethics clearance was not required on part of the University of Witwatersrand.

Some noteworthy characteristics of the datasets are tabulated in Table 4.1.

Table 4.1: Characteristics of the average mouse and rat dataset **[**9**] [**10**] [**11**]**

	Mouse data	**Rat data**
Isotropic scale factor (x-y)	1.16 μm per pixel	1.53 μm per pixel
Slice thickness	2.5 μm x 0.5=5 μm (every 2nd slice present)	2.5 μm
Average no. of images	984	4392
Resolution	2500 x 1675 pixels	2750 x 2500 pixels

A nephron cross-section is defined as a cross-section through a single nephron at one location in an image. As one proceeds through an image set, it can be seen that the microstructure or morphology changes drastically from the cortex to the medulla. The Appendix contains a reconstructed view of the entire specimen in a longitudinal plane in order to illustrate the regions and the changes between them. Figure 4.1 displays examples of images in the cortex and medulla [**11**].

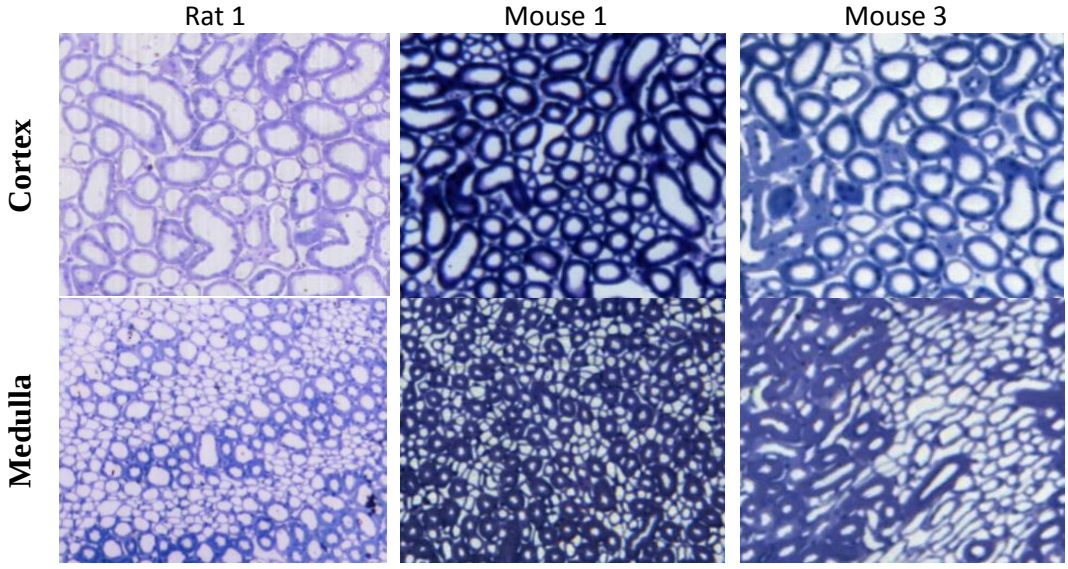


Figure 4.1: Examples of images in the cortex (left) versus the medulla (right) **[**11**]** are shown at equal magnification. A change in nephron characteristics, particularly wall intensity, tubule density and decreasing diameter can be seen.

Histological Variations

Nephrons belonging to the same collecting duct family have their loops running together in the medulla [**10**]. Cortical nephrons have shorter loops of Henle while juxtamedullary nephrons extend deeper into the medulla, have longer loops of

Henle and larger glomeruli [**1**] [**10**]. Different cross-sections of the nephron may stain with different intensities as the cell composition varies. The DTL in particular has very thin, lightly stained walls as can be seen in images of the medulla in Figures 4.1 and 4.2.

Cortex

The renal cortex is composed primarily of the PCT, DCT and glomeruli, which are relatively large in diameter (Glomeruli: 150-240μm, PCT: 40-50μm, DCT: 20-50μm [**2**]) as seen in Figure 4.2. Large blood vessels (arcuate arteries and veins) and smaller capillaries are also present. While most nephron cross-sections appear circular, there are many elliptical and elongated cross-sections in the cortex due to the turning and winding of the PCT and DCT. The PCT is longer, larger in diameter and more convoluted than the DCT [**10**], making up the majority of cross-sections in the cortex. The PCT also has a fuzzier border. The glomerulus, PCT and DCT related to the same nephron are found in the same vicinity in the cortex [**10**]. The glomeruli are randomly dispersed throughout the cortex and are easily distinguishable by eye on the microscopic images as large circles containing a ball of convoluted blood vessels. From observation, the TAL and initial DCT cross-sections are much smaller in diameter than PCT and distal DCT cross-sections and are dispersed in between these larger tubules.

Medulla

The outer medulla contains a mixture of large PCT and DCT cross-sections as well as small DTL and ATL cross-sections. Deeper in the outer medulla, the PST with an outer diameter of about 60μm, suddenly narrows to about 10-15μm and continues as the DTL into the inner medulla [**1**] [**2**].

The inner medulla primarily consists of the thin limbs of the loop of Henle. These are seen as densely packed circular structures. The descending limb has a much thinner wall than the ascending limb. All cross-sections are circular except for small elongated cross-sections at the bends of the loop of Henle. From observation, the surrounding capillary networks called the vasa recta are difficult to distinguish from nephron cross-sections as they are very similar in appearance.

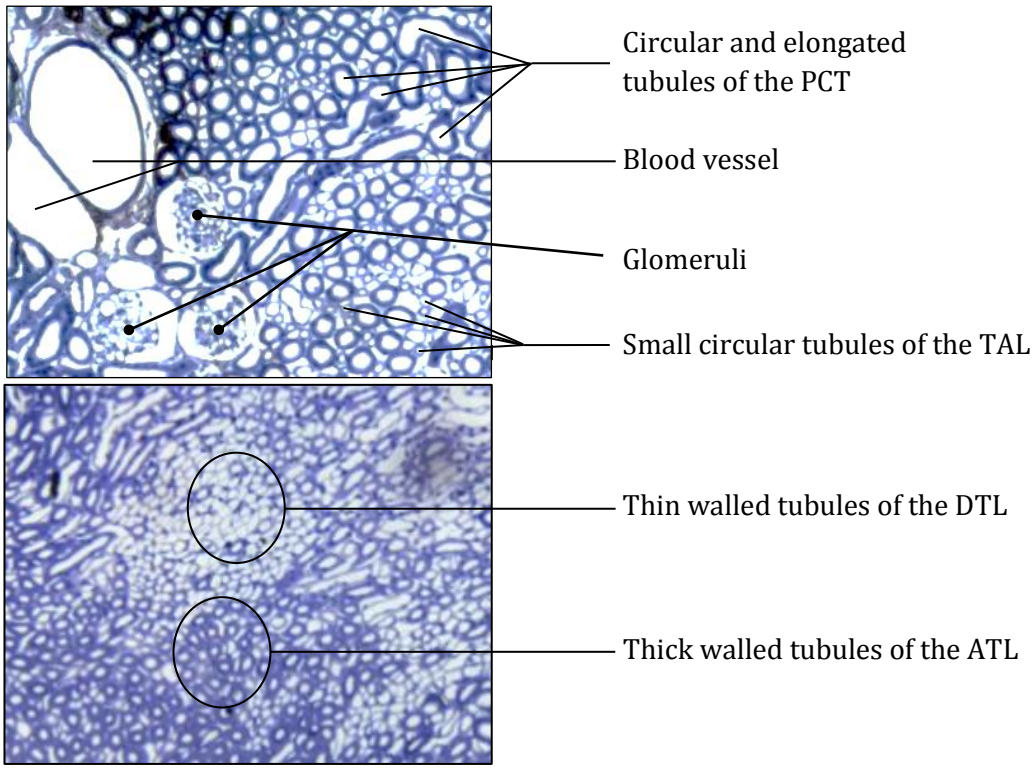


Figure 4.2: A section of an image through the cortex (top) and inner medulla (bottom) showing numerous structures **[**11**]**.

The varying structure from the cortex to medulla means that processing parameters will have to change progressively through the image set in order to accommodate the varying intensities, sizes of objects and the amount of unwanted objects such as blood vessels, the interstitial connective tissue, the background, and artefacts.

Analysis of the images from the medulla poses a greater challenge compared with the cortex because the cross-sections are much smaller and concentrated, making it more difficult to isolate them accurately. Even though tracking in this area would be more prone to error (as the probability of mistakenly jumping onto the wrong cross-section is higher), the fact that the paths are mostly straight and unidirectional in this region can be used as a criterion for error checking. Other known information can also be used for guidance or error checking, e.g. slices 1-300 may consist primarily of the cortex, or a diameter of 5-10 pixels indicates a thin limb of the loop of Henle in the medulla.

4.2 An Ideal Solution

An ideal solution would consist of aligning the images and producing binary images using the required conditioning steps. A 3D segmentation algorithm such as region-growing, Watershed segmentation or the flood-fill algorithm could then be applied to the entire 3D volume, ideally isolating a particular nephron given a starting seed. Each isolated volume could then be independently analysed.

However, the data presents many complexities which do not make such a solution viable. Image misalignment, local distortions and missing data (or tissue) between adjacent images produces a definite discontinuity from image to image. This, in combination with interference from connective tissue cross-sections and other non-ideal factors result in multiple nephrons being linked using these techniques. Since there is not continuity between adjacent images (in contrast to the x-y image planes), linkage of the nephron cross-sections merely by pixel connectivity is not reliable and is error prone as it requires only a few pixels to be incorrectly connected from different nephrons. This is especially true for the inner medulla where tubule density is high. Such a solution would also not be capable of intelligently handling distorted images and artefacts. Additionally, these algorithms require the whole volume to be actively processed, which is difficult to carry out as it requires a massive amount of physical memory on the order of 25GB.

4.3 The Complexities of the Problem

Broadly speaking, the complications are firstly due to features of the specimens themselves, and secondly due to the large amount of data per dataset. The designed system must be able to counteract these complexities while accurately tracking the path of each nephron through the 3D image space.

4.3.1 Artefacts

Physical artefacts are structures, or processes, which contaminate or distort the original tissue, causing reduced visibility or complete obscuration of the tissue. They are induced during tissue preparation. An image artefact is an anomaly caused during the image capturing process.

Large physical artefacts seen in the images include tissue cuts, folds and external matter, which affect all the nephron cross-sections in the vicinity. Some artefacts only affect single nephron cross-sections, such as the presence of external matter in the lumen. A number of examples are displayed in Figure 4.3. Artefacts hinder tracking if they occur in a number of successive images. This is typically where user-input is then required.

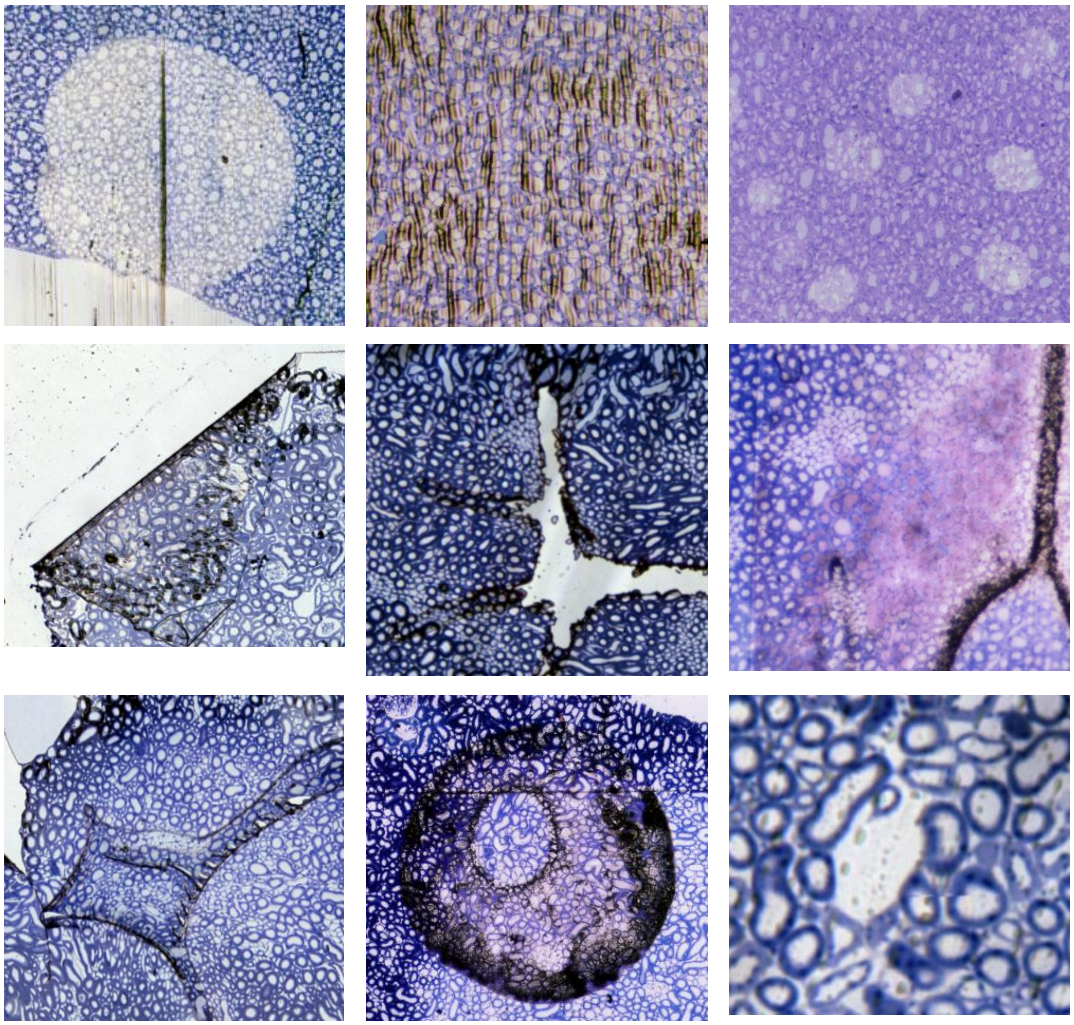


Figure 4.3: Examples of interfering physical artefacts in the image sets **[**11**]**. These include cuts, folds, external matter, blurring effects, bright spots and occluding matter in the lumens.

Some images also have areas of sharp non-uniform intensities, particularly large bright spots which could be a result of both non-ideal tissue preparation and image capturing. These cause incorrect merging of cross-sections or elimination of a large number of nephron cross-sections during pre-processing. These images

cannot simply be excluded as the frequency of images with artefacts is too high (one in every 5-10 images). Also, the artefacts do not affect all of the nephron cross-sections in the image and the defective images are useful for the most part.

The presence of these vastly different artefacts require each image to be evaluated and processed individually during tracking so that the artefact can be bypassed automatically or by the user. This is another reason why a generic three-dimensional tracking algorithm such as flood-fill cannot be used.

Another anomaly is the misalignment between images. This is due to local tissue distortions (a physical artefact causing non-rigid deformation) as well as capturing slides which were not aligned (an image artefact causing translation and rotation). This is discussed in more detail in Sections 6.1 and 8.1. In addition to the nephrons, interstitial connective tissue and blood vessels are present. Although these are not artefacts, they do cause interference during tracking. Blood vessels link the glomeruli of multiple nephrons, while connective tissue causes the incorrect linking of multiple nephrons during tracking.

4.3.2 Memory

Each image set occupies about 700MB and 2GB for the mice and rat datasets, respectively when stored in a compressed form (JPEG images). In order to be processed in MATLAB (or any software), the images must be decompressed into a matrix form, where each matrix entry is a pixel value occupying at least four bytes. This then equates to a decompressed size of about $\frac{1675\times2500\times900\times4}{1024^3}$ = 14GB for a mouse dataset and $\frac{2500\times2750\times2500\times4}{1024^3}$ = 64GB for a rat dataset.

This implies that it is not possible to process the whole volume at once considering typical physical memory limitations of 8-16GB. Rather, smaller batches of images should be processed in a more intelligent, controlled manner, as is required for the complex nephron path tracking problem.

CHAPTER 5

System Overview

The task of manually tracking nephrons through an image stack is a seemingly trivial one for a human being. However, transferring the vision, interpretation and decision making abilities of the human operator into software is a very complex task. Obtaining results that are as accurate as manual tracking results is even more difficult. In order to attempt to do so, the system developed in this study uses a combination of techniques from the domains of computer vision, feature computation, graph theory and machine learning.

Although the purpose of this system is not to make an "end-diagnosis", from a methodological perspective, the problem fits the generic architecture of a Computer Aided Diagnosis (CAD) system [**35**]. CAD systems assist medical practitioners in interpreting microscope, x-ray, MRI and ultrasound images by automatically marking, measuring or detecting certain regions of interest [**29**]. These systems use a combination of image processing and artificial intelligence techniques. The architecture of a CAD system can be generalised as [**29**] [**35**]:

1. *Image Pre-processing*: Involves steps such as image registration, noise reduction, edge enhancement and intensity equalisation in order to increase quality or amplify visibility of features [**29**].
2. *Definition of Regions of Interest*: Separating or detecting the objects of interest using methods such as image segmentation or contour matching [**29**].
3. *Feature Extraction and Selection*: Computing features by measuring characteristics such as size, shape and colour [**29**] [**36**].
4. *Classification*: Involves pattern recognition through supervised classifiers such as a Decision Tree, Artificial Neural Network or Bayesian Network classifier, or unsupervised methods such as clustering.

A number of factors affect the accuracy of these systems, for example image quality, noise and complexity of the target objects [11]. Figure 5.1 describes the architecture of the designed nephron tracking system.

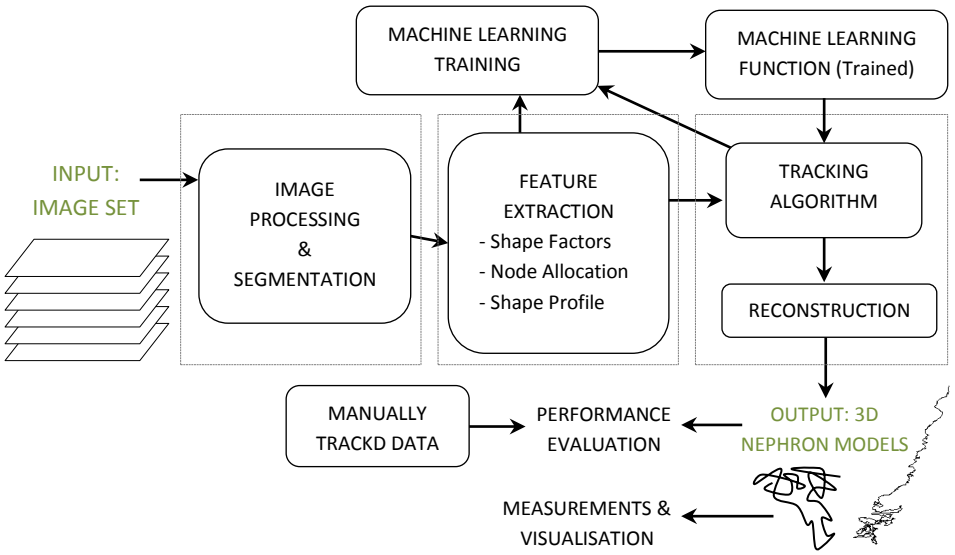


Figure 5.1: A high level overview of the nephron tracking system, showing the main sub-systems and the flow of information between them.

The system is implemented in MATLAB [**33**] as a series of independent modules where structures of information are progressively passed on from one stage to the next. This framework is related to an object-orientated approach in that the major functions are decomposed into independent, reusable blocks. The development of the system is incremental, involving continuous reiteration through the three main stages to achieve optimal performance.

There are a number of parameters in each stage which need to be calibrated to each image set. These are discussed in their relevant sections. In order to easily do so, a single *settings* file must be initialised prior to execution, which contains all of the information needed to automatically adjust and vary the parameters involved.

CHAPTER 6

Image Processing

Computer vision aims to mimic the capabilities of human vision by processing, analysing and transforming raw images into a form that can be more easily and accurately interpreted by a machine [**36**]. It forms a crucial component of many automated processes in the real world [**37**] including the current nephron tracking task.

The image processing steps prepare the images for subsequent stages by creating uniformity among all nephron cross-sections and counteracting non-ideal factors described in Section 4.3. The images are processed such that required features (nephron cross-sections) are enhanced while unwanted features (such as interstitial connective tissue (ICT) cross-sections, large blood vessels, background pixels and large artefacts) are filtered out or reduced. The final product of image pre-processing is a binary image of the lumens of the nephrons as shown in Figure 6.1.

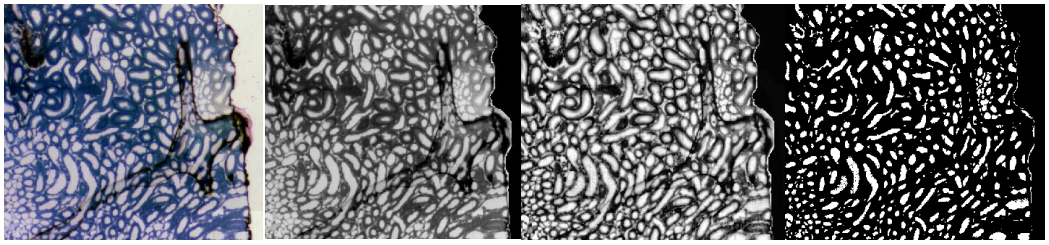


Figure 6.1: Each colour image is processed into a binary image containing nephron cross-sections of all sizes. Each raw image **[**11**]** undergoes conversion to grayscale, background removal, histogram equalisation and binarisation.

6.1 Image Registration

Image alignment was carried out on the datasets [**11**] during the previous study in order to ease the manual tracking process [**9**] [**10**]. The procedure involved iteratively estimating the translational and rotational offsets between adjacent images and applying the rigid transformation using custom software written in C

[**9**]. This alignment was apparently not sufficient due to the local distortions induced during the sectioning process [**9**]. The distortions have the effect of pinching, compressing or stretching local regions of tissue. The rat image sets were then further aligned using five manually placed landmarks which divided each image into four polygons, each of which then underwent a non-rigid transformation [**9**].

These processes have resulted in the images being sufficiently aligned from a global perspective. However, local distortions in the mouse datasets were still not fully compensated for, especially since only every second slice of the dataset [**11**] is present. This is shown in Figure 6.2, where one local area can be aligned while a nearby area is misaligned.

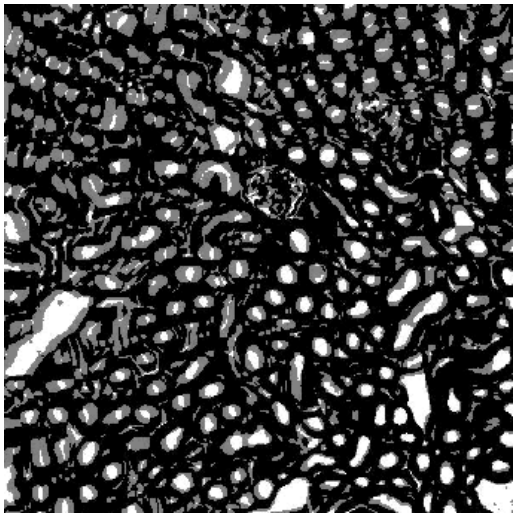


Figure 6.2: A pair of superimposed adjacent sub-images (binarised) from a mouse dataset is shown (derived from **[**11**]**). The bottom right area is well-aligned while the top areas are misaligned. This cannot be corrected using a translation and rotation only as the misalignment is due to localised stretching/compression.

Misalignment due to local distortions in the rat datasets was minimal as they were compensated for by the four-quadrant alignment method. However, this had resulted in nephron cross-sections incorrectly merging at the junctions of the four polygons, as shown in Figure 6.3. This resulted in multiple nephrons being linked during tracking. The nephrons around these junctions were therefore excluded from the study as the merge cannot be reversed.

The images were not further aligned during the pre-processing stage, although further alignment is performed during the tracking process as the local distortions require the areas around each cross-section to be handled locally and independently. This local alignment is discussed in Section 8.1 as part of the tracking system. Advanced non-linear image registration techniques such as RANSAC [**37**] were not applied as:

- Cropped local regions can be aligned using simpler methods. Non-linear alignment usually makes use of six more parameters in addition to the two employed (x and y translation), which increasing the order of the process.
- Large cumulative transforms over the image set must be avoided as they may over-morph the images.
- A small amount of the misalignment is due to the progressive change in morphology and not only due to induced distortions. A non-linear registration would counteract this change in morphology, which is undesirable as the characteristic nature of the nephrons must remain unchanged.

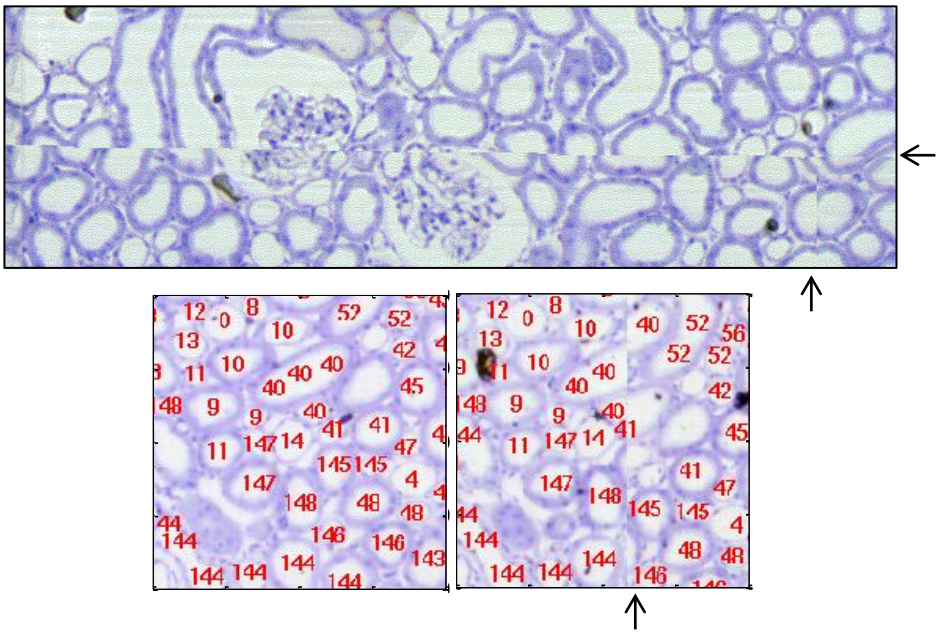


Figure 6.3: The arrows indicate the junctions of the polygons created during the four-polygon alignment method, which results in the merging of cross-sections from different nephrons. In the labelled image (below) the merge between cross-sections from nephrons 40 and 41 can be seen **[**11**]**.

6.2 Image Processing Procedure

Each nephron tubule consists of a lumen enclosed by the tubule wall, which differs in thickness depending on its location, i.e. the PCT, PST and DCT have thick walls while the DTL, ATL and TAL have very thin walls. It would be ideal to extract both the wall and lumen of each tubule but this is a difficult task due to the walls of adjacent tubules touching one another. One potential method which could be applied is spline curve fitting using a genetic algorithm, which has been used to isolate different types of tissue in histological images [**19**]. However, the vast number of single nephron cross-sections per image that would need separation is too large (≈ 8000 per image in the cortex to ≈ 36000 per image in the medulla) and the problem becomes unnecessarily complex for current purposes.

It was decided that the lumen of a nephron cross-section alone contains sufficient amount of information to represent the original structure in the colour image, i.e. location, size and shape of the nephron cross-section is provided by the lumen alone. The lumens are also more easily and accurately isolated juxtaposed to the walls of the nephrons and are thus chosen as the objects to be isolated. Each image undergoes the following procedures:

6.2.1. Conversion to Grayscale

The staining used on the specimens (toluidine blue [**10**]) results in all structures being monochrome. The colour information is thus discarded by conversion to a grayscale image by retaining the value component (or luminance) of the hue-saturation-value (HSV) image. The colour information could however be useful (e.g. if a more differentiating stain is used in future image sets) and this would require the pre-processing stage to be modified accordingly.

6.2.2. Background Removal

The tissue slice is isolated by removing the white background space. First, the image is thresholded at the image's average intensity value plus some constant C.

$$T_{\mathrm{bgrnd}}{}^{z} = \mathrm{mean}(I_z) + C \qquad (6.1)$$

This is chosen instead of a constant value only as each image differs in intensity, some by a large amount. Furthermore, this value results in a sharp contrast between the background (BG) and the tissue. The C value must be chosen to suite each image set. For example, the images in one rat dataset have a very large bright tissue centre. A C value that is too low causes the nephron cross-sections to merge into one large binary element when binarised. The large component could then be mistaken for the background. Another mouse dataset has a darker background with lots of matter, and a C value that is too high results in large chunks of the background not being removed. This value must be chosen once-off during system calibration by a trial-and-error approach.

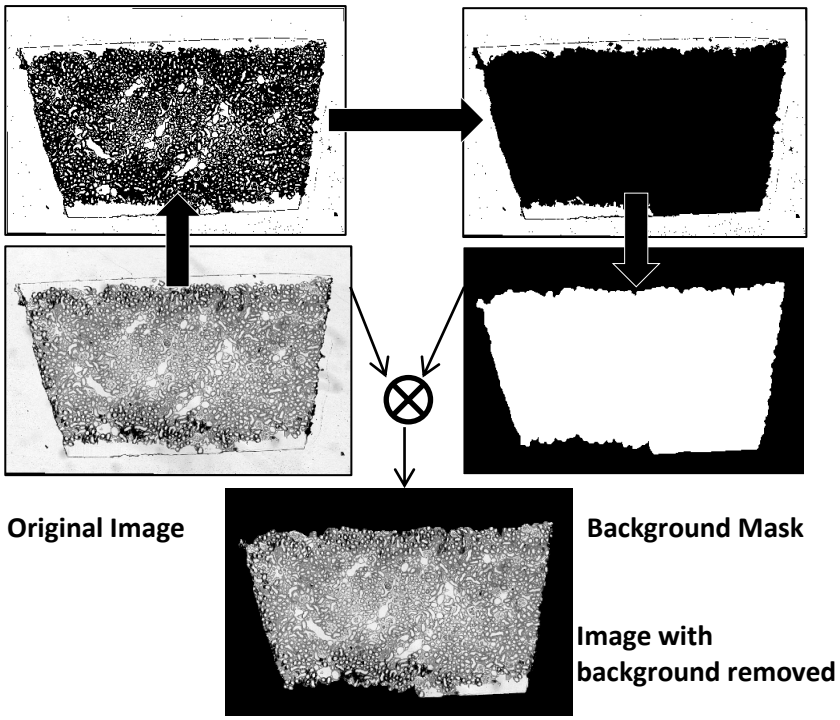


Figure 6.4: The procedure for background removal is shown. The raw image **[**11**]** is binarised. The background mask is formed by morphological closing and inversion of the largest components in the binary image. Finally the mask is multiplied with the image.

The binary image is segmented (using simple 8-neighbour connectivity), thereafter obtaining the largest cross-sections which then form a background mask. The mask first undergoes morphological image closing using a 20x20 circular kernel in order to remove small objects occurring in the background. The

mask is then inverted and applied to the original image by multiplication. These steps are shown in Figure 6.4. Background removal must occur prior to (and without any) image equalisation so as not to amplify the intensity or texture of matter occurring in the background.

6.2.3. Histogram Equalisation

Histogram equalisation involves normalising the histogram of an image such that all intensity values are equally distributed among the pixels in the image. It is the most crucial image processing step and is required in order to counteract uneven intensities on both a local and global scale as illustrated in Figure 6.5.

1. 'Globally': Uneven intensity may occur across large areas in the image, such as random large bright spots or a brighter centre as a result of the image acquisition process. Furthermore, some images are irregularly bright or dark in comparison to the rest of the image set. Global equalisation is achieved by using a large equalisation window [**36**], about a tenth of the size of the image.

2. 'Locally': Uneven intensities may occur in small local areas (especially in the inner medulla) as a result of narrow diameter nephron cross-sections having a much lighter wall. Local equalisation is applied by using a much smaller equalisation window of about 5 times the size of the average nephron cross-section in the image.

The sizes of the equalisation windows must be suited to each image per dataset. Images of the cortex require a large local equalisation window (≈ 40 pixels), as the nephron cross-sections are larger than in the medulla. Too small a window results in 'hyper-equalisation', where large white areas (such as large nephrons) acquire a rough, broken texture. The nephron cross-sections in the inner medulla are much smaller and have very thin, light tubule walls, and are thus much more dependent on good equalisation. This requires a smaller local equalisation window (≈ 20 pixels). A window that is too large will not adequately equalise the intensity of these nephron walls, resulting in the cross-sections being removed or merged when binarisation occurs.

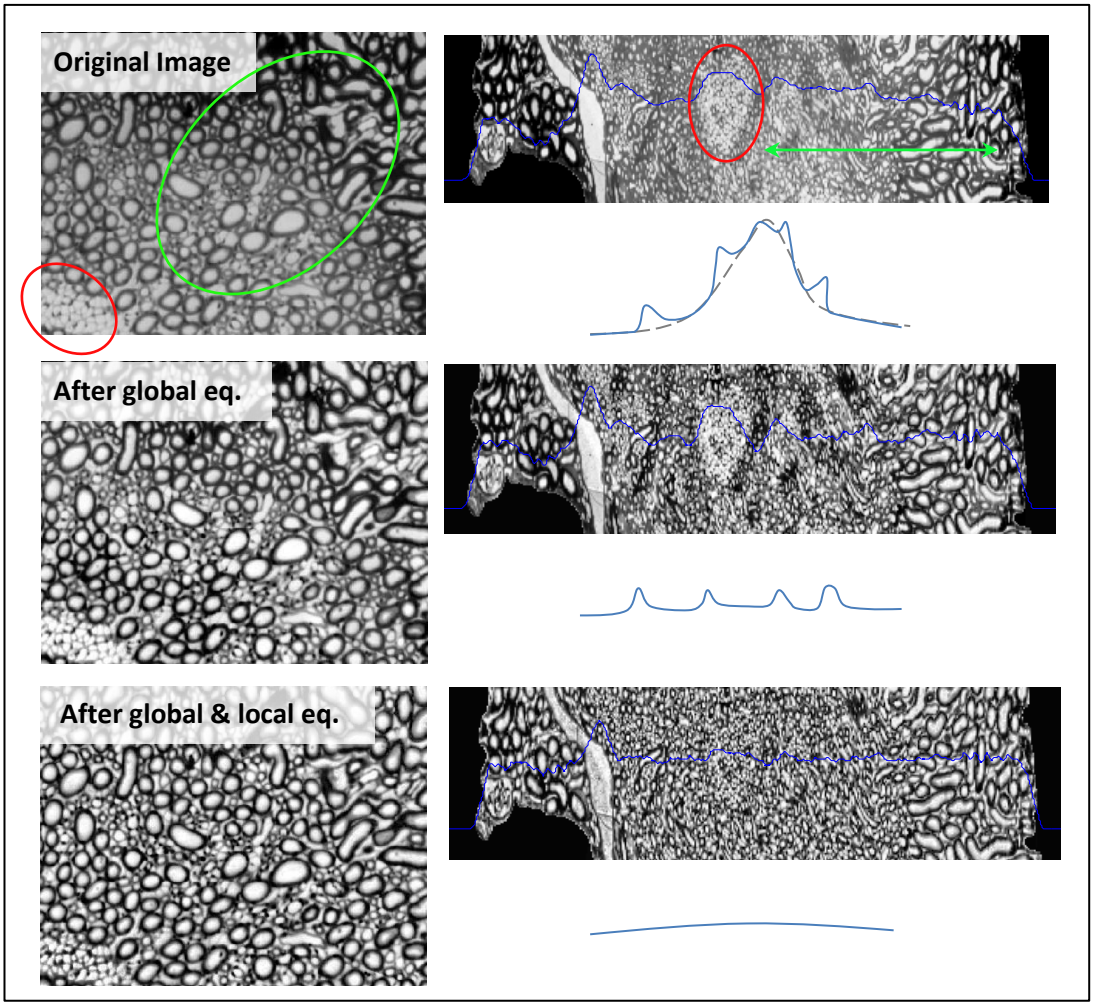


Figure 6.5: Top: There are large regions of uneven intensity (shown in green) which require equalisation with a large window. There are also much smaller, local uneven intensities due to thin walled cross-sections. If not equalised locally, these groups of nephrons will merge into large binary cross-sections and would not be able to be differentiated. Middle: After global equalisation, the image intensity is uniform over large areas as seen in the real and conceptual histograms. Bottom: After local equalisation, uniform intensity is achieved across small areas as well. Images adapted from [**11**].

6.2.4. Thresholding

The image is thresholded at a constant value T to create a binary image. Adaptive thresholding is not used as uniformity was achieved through the equalisation steps. The threshold value is chosen such that it does not allow independent lumens to merge while also not letting small nephron cross-sections disappear. It

also varies through the image set such that the inner medulla images have a slightly higher value to prevent the dense, thin-walled cross-sections from merging, while the cortex has a lower value to prevent segmentation of large nephron cross-sections.

6.2.5. Removal of Unwanted Cross-Sections

Unwanted cross-sections include those of the blood vessels and connective tissue. Connective tissue cross-sections appear between nephron cross-sections in the cortex. They are characteristically irregularly shaped, fragmented and typically small and thin, as shown in Figure 6.6. It is difficult to remove these cross-sections without also removing some nephron cross-sections which are similar in appearance (particularly cross-sections of the TAL and DCT). It is important not to remove these nephron cross-sections as the TAL and DCT will otherwise not be able to be tracked.

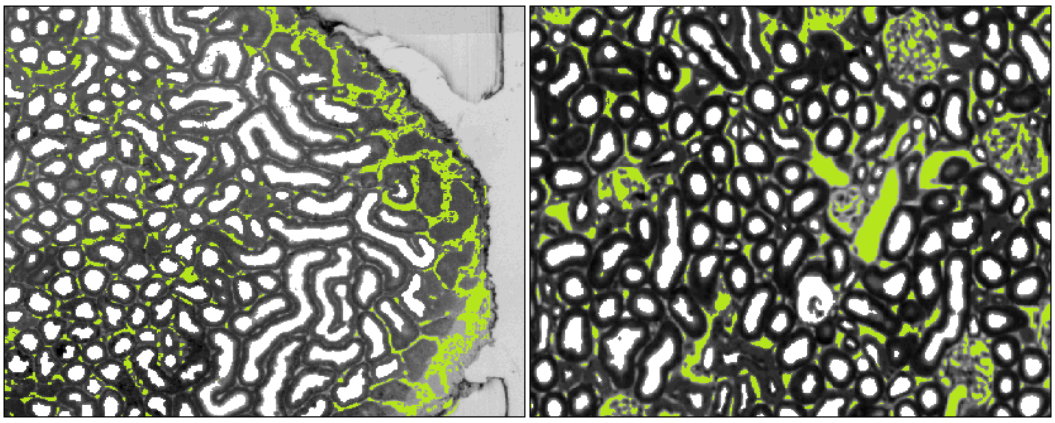


Figure 6.6: Connective tissue cross-sections have been manually marked in green in images of the cortex from a rat (left) and a mouse (right) image set (adapted from **[**11**]**). The irregularly shape connective tissue cross-sections are distinct, although there are some nephron binary cross-sections which have similar characteristics.

Size-based component exclusion: Binary components that are very small (<10 pixels) and very large (> 100 000 pixels) can be confidently identified not to be nephron cross-sections and are removed. Large components are normally large blood vessels while small components are small connective tissue cross-sections or noise from the equalisation.

Morphological erode/dilate cycles: Performing morphological eroding *p* times, followed by a dilation *p* times results in the elimination of larger connective tissue cross-sections. While this works well, it is applied very sparingly as it also results in the removal of small nephron cross-sections. This cannot be done in the inner medulla where the cross-sections are only a few pixels wide.

Shape-based clustering: Clustering the cross-sections based on their shape factors is another possible way to eliminate unwanted cross-sections.

Morphological operations and clustering invariably results in the loss of some nephron cross-sections as well (usually elongated or C-shaped cross-sections). This is highly undesirable, as the tracking process depends on each cross-section along the nephron's path to be present, especially elongated ones. It is therefore decided to include all possible data (including unwanted ICT) rather than to have missing data.

Obtaining the final binary image is one of the most important tasks, as the accuracy of following stages depends on how well cross-sections are isolated from one another. A compromise must be made between the number of connective tissue cross-sections present and the number of small nephron cross-sections which do not get eliminated.

Further pre-processing involves the removal of highly distorted images and replacing them with the image above or below. An average of 4 images per dataset had to be been manually replaced. However, an automatic method can be devised if a larger number of images are defective, for example by analysing the mean intensity of each image in the image set.

6.3 Image Segmentation

Connected component segmentation [**36**] (using a 4-connected neighbourhood) is used to segment the image into independent nephron cross-sections. Although this is the simplest segmentation technique, it produces satisfactory results because a good binary image was obtained in the previous steps. A labelled binary image is formed, where all the pixels belonging to one component have a unique value.

Watershed segmentation or region-growing on the equalised image are other possible segmentation techniques, which could perform better in cases where independent lumens incorrectly merge through a few connected pixels. However, they do tend to over-segment the image [**17**], resulting in the division of elongated nephron cross-sections.

6.4 Automatic Parameter Variation

A very important factor during pre-processing is accommodating for the change in morphology from the cortex to the medulla. The cortico-medullary boundary is an area where the proximal straight tubule (≈ 60μm in diameter) suddenly narrows to a diameter of 10-15μm to form the thin descending limb of the loop of Henle [**1**] [**2**]. This change requires almost all parameters of the pre-processing steps to vary accordingly to ensure that cross-sections of all sizes are extracted.

In order to accommodate for this relatively sudden change, the parameters are made to vary along the image set according to a modified sigmoid function $Y(n)$ (also known as a generalised logistic function [**38**]) as in equation 6.2, which has its inflection point set at the transition zone. This transition zone must be chosen empirically through observation during system calibration.

$$Y(z) = \frac{UL-LL}{1+e^{\frac{k-z}{\delta}}} + LL \qquad (6.2)$$

Where z is the image number, UL is the upper limit, LL is the lower limit, k is the z value at which the inflection point occurs and δ is a steepness coefficient [**38**].

The steepness coefficient is chosen by the best outcome during experimentation on a few images. The sigmoid function allows relatively constant parameter values in the cortex and inner medulla as shown in Figure 6.7. Some parameters of the tracking algorithm, such as the tracking radius, also make use of this function.

The suitability of this function can be validated by examining the change in certain characteristics along the image set, such as the number of cross-sections and average nephron cross-section width. As shown in Figure 6.8, these

characteristics have an inherent sigmoidal shape; hence the function models the change well. The curve seen near the initial images is due to the tissue slices progressively growing from the edge and is not due to differing tubule characteristics, and is therefore not modelled.

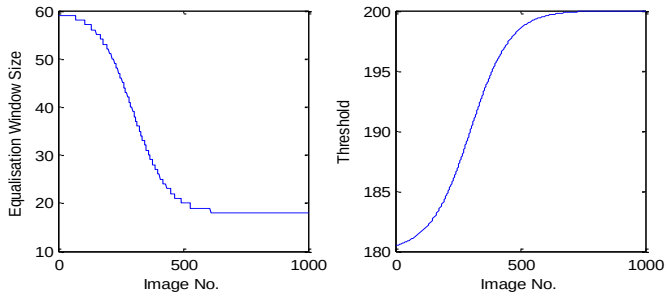


Figure 6.7: The equalisation window size and threshold value are made to vary according to custom sigmoid functions. The equalisation window is quantised. The parameters are for a mouse dataset with its inflection point at the 350th image.

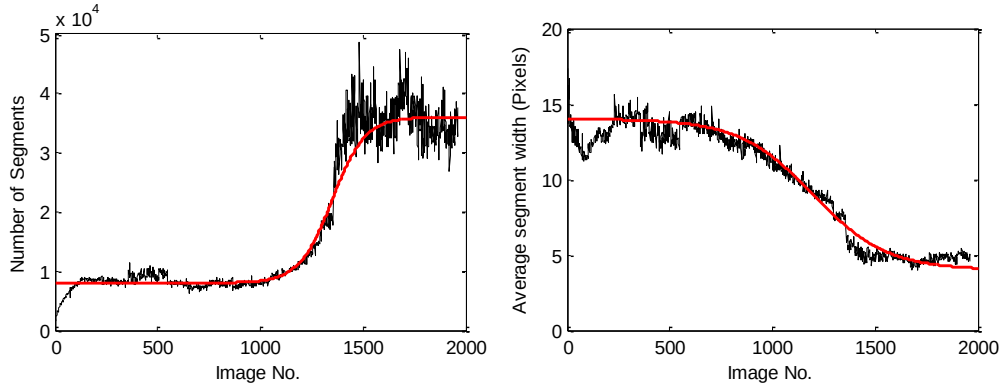


Figure 6.8: Features of the datasets (black) such as the average nephron cross-sectional width (right) and number of cross-sections per image (left) have an inherent sigmoidal characteristic, as shown in red. The graphs are of a rat dataset.

Modelling the processing parameters using tailored sigmoid functions for each image set results in uniform results (quality and accuracy of binary images) across all datasets, i.e. it serves as a calibration mechanism. Subsequent stages therefore do not have to cater for any particular dataset and can be designed in a generalised manner for standard input data.

CHAPTER 7

Feature Extraction

Feature extraction aims to simplify and concentrate useful information from raw data. It forms an *intrinsic image* of the input data, which is an array of information representing important physical characteristics of the objects involved [**37**]. Within the images, large amounts of the data are not useful, for example the large number of pixels making up the background. The pixel information can instead be condensed into a set of features per nephron cross-section, which represent the problem to a sufficient degree. Intuitively, the most useful information about a nephron cross-section is its size, shape, colour and location.

7.1 Node Allocation

A *node* is defined as a point coordinate in the 3D image space. The pixel locations per cross-section can be reduced into a set of nodes allocated along the cross-section, hence modelling their location. For example, a circular nephron cross-section can be represented by one central coordinate, instead of a few hundred pixel locations, and an elongated cross-section can have multiple nodes allocated along its length. This abstraction greatly simplifies the problem, reduces the size of the data, decreases the computational load on subsequent stages and concentrates significant information.

Node allocation was first achieved by using the circular Hough transform [**37**] to locate circles occurring within a certain radius range in the binary images. At first glance the results seemed good, but deeper inspection revealed that many cross-sections were not allocated a node, and elongated cross-sections were not handled adequately. This method was computationally expensive due to its iterative, analytical nature and offered no control over how many nodes would get allocated per cross-section.

Finally, K-means clustering was used to allocate nodes. K-means clustering is an unsupervised learning method which groups m observations into k clusters [**39**].

Each cluster is defined by its centroid, which is the mean of all the observations assigned to it. These centroids are used as the nodes and are found by solving the optimisation problem in equation 7.1.

$$\mathbf{C} = \arg\min_{\mathbf{S}} \sum_{i=1}^{k} \sum_{\mathbf{x} \epsilon \mathbf{S}_i} \|\mathbf{x} - \mu_i\|^2 \qquad (7.1)$$

Where **x** is the observations, **S** is the observations partitioned into *k* sets $\{S_1, S_2, \ldots S_k\}$, μ_i is the mean of the observations in S_i and $\mathbf{C} = \{c_1, c_2, \ldots c_k\}$ is the set of k centroids [**39**].

Each non-zero pixel on an isolated binary cross-section is made an observation. Shape criteria (area and circularity, which are discussed in Section 7.2.1) are used to decide on the number of nodes (K) to allocate to a cross-section. Cross-sections with an area below 5 pixels are ignored (K=0) as these are typically pixels due to equalisation noise. If the cross-section is circular (circularity>0.95) or small (area<300), one centroid is requested (K=1). For elongated cross-sections, the requested K value begins at 2 and is incremented until the average pairwise distance between adjacent nodes is less than a desired minimum value. This value is chosen to be close to the average cross-sectional width in the current image (≈ 20 pixels in the cortex and ≈ 8 pixels in the medulla). This ensures that an adequate number of nodes are allocated per cross-section depending on its size and shape. Figure 7.1 displays examples of allocated nodes on cross-sections of various shapes and sizes.

A characteristic of the K-means clustering method is that it groups data into Voronoi cells [**39**], which is suitable to the node allocation problem. Some disadvantages are the susceptibility to local minima and varying results due to randomised initial conditions [**28**]. This is overcome by running the clustering multiple times per cross-section until certain conditions are met. An important advantage of the method is that a desired number of centroids can be requested.

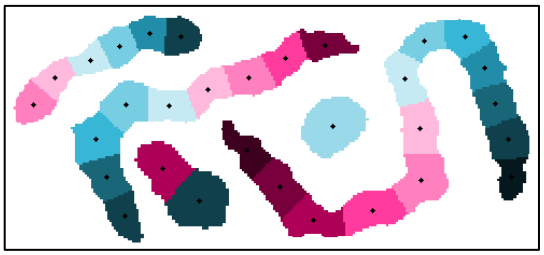


Figure 7.1: K-means clustering divides cross-sections into similarly sized Voronoi cells. The centroids from K-means clustering are used as nodes for the cross-sections. A suitable number of nodes are allocated automatically per cross-section.

7.2 Shape Measurements

Tracking of a nephron using only the 3D set of nodes results in the linkage of multiple nephrons, blood vessels and connective tissue due to the close proximity of the intertwining nephrons. By only considering the point cloud, the algorithm is blind to a large amount of available information. Shape information about each cross-section is thus also captured. The idea behind incorporating shape information into the tracking is to retain information about the original cross-sections so that the algorithm can make intelligent, confident and informed decisions at each incremental step of the process.

7.2.1 Shape Factors

A shape factor, or metric, refers to a dimensionless value that is dependent on an object's shape but is independent of its size [**40**]. It usually indicates the degree to which an object deviates from an ideal shape, such as a square or circle [**40**]. Various shape factors are calculated per cross-section to capture abstract information about each cross-section along with its nodes.

Shape metrics are calculated using various measurements of an object. Primary measurements include the object’s area, perimeter and chord lengths [**40**]. Secondary measurements which are useful include the convex area (the area of a polygon of the lowest degree which covers the object), equivalent diameter, centroid and moments of the object about the centroid [**37**] [**40**]. The first moment is the mean and the second is the variance. Moments are used to calculate an equivalent ellipse which has the same variance as that of the object [**37**]. The

minor and major axes are then the perpendicular axes of the aforementioned ellipse, and the orientation is the angle that the major axis makes with the horizontal axis [**37**].

Shape metrics combine these measurements to describe a number of shape characteristics. Circularity, eccentricity, solidity and aspect ratio were chosen as useful descriptors for the cross-sections. Area and minor axis length are also captured as absolute valued descriptors. These values are extracted using the MATLAB *regionprops* function [**33**]. The shape factors are briefly described:

The *aspect ratio* is the ratio of the major axis to the minor axis and indicates the length and symmetry of an object [**36**]. For a circle or bisymmetric object, the aspect ratio is 1 while it is >1 for an elongated object.

Circularity measures how close an object's shape is to a circle [**36**] and is given by equation 7.2. A value close to 1 indicates a very circular object while an elongated object has a value near 0.

$$f_{circ} = \frac{4\pi \text{Area}}{\text{Perimeter}^2} \tag{7.2}$$

Solidity is the ratio of the area to the convex area of an object, and measures its virtual hardness or density [**36**]. Generally, this area is larger than or equal to the area of the object. Objects with holes and multiple concave areas have a low solidity. Waviness is also sometimes used, where the perimeters of the object and polygon are used instead of the areas [**36**]. Waviness does not take holes into account.

Eccentricity is typically associated with an ellipse, being defined as the ratio of the distance between the foci of the ellipse and its major axis length [**37**]. Using moments, an equivalent ellipse can be calculated for an arbitrary shape, hence an eccentricity value. Eccentricity is a value between 0 and 1, with 0 indicating a circular shape and 1 indicating an elongated shape.

The *extent* is an area ratio between the object's area and the region area, where the region is the bounding box of the object [**37**]. It generally describes the degree to which the object fills the bounding box. Extent can give a combined

indication of elongation, orientation and solidity [**37**]. It is useful for detecting highly irregularly shaped objects. Note that this metric is orientation sensitive, e.g. an elongated object lying horizontally or vertically will have high extent, while it would have low extent if lying at 60^0.

7.2.2 Shape Profile

The shape factors are useful for cross-sections that are round and elliptical, but they do not adequately describe cross-sections that are more irregularly shaped, such as glomeruli or connective tissue cross-sections. Also, a move from one cross-section to another cannot be adequately described by merely comparing shape factors. As an additional feature the shape profile around each node is calculated.

The shape profile of an object is a polar plot of the distance to its boundaries with respect to a reference point [**36**]. It is commonly used to compare the structure of two objects. It transforms a 2D shape representation into a 1D plot [**36**], hence reducing computational effort during matching. It allows shape comparison of objects of different sizes by normalisation of the distance.

First, the edge of the cross-section is obtained using a Sobel edge detector [**37**]. This method produces a well-defined closed curve around the cross-section. The edge pixels are then processed into an ordered set of points. The angles and radii relative to the reference point are calculated as in equations 7.3 and 7.4.

$$\boldsymbol{\theta} = \arctan\left(\frac{\mathbf{P_{edge}}(y) - P_{ref}(y)}{\mathbf{P_{edge}}(x) - P_{ref}(x)}\right) \tag{7.3}$$

$$\mathbf{r}(\theta) = \left\|\mathbf{P_{edge}} - P_{ref}\right\| \tag{7.4}$$

where $\mathbf{P_{edge}}$ is the vector of edge coordinates, P_{ref} is the reference coordinate, $\boldsymbol{\theta}$ is the vector of angles and $\mathbf{r}(\theta)$ is the vector of radial distances.

If the object is concave, the plot will be multivalued [**36**]. This redundancy increases the order of the matching procedure back to 2D. This problem is solved by simply taking the nearest r value for a given θ, and is called *unwinding* [**36**]. In order to produce a consistent feature set, the radius at constant angle increments is

interpolated. The degree of abstraction is dependent on the chosen angle increment δ [**36**]. The unwinding and interpolation is shown in Figure 7.2.

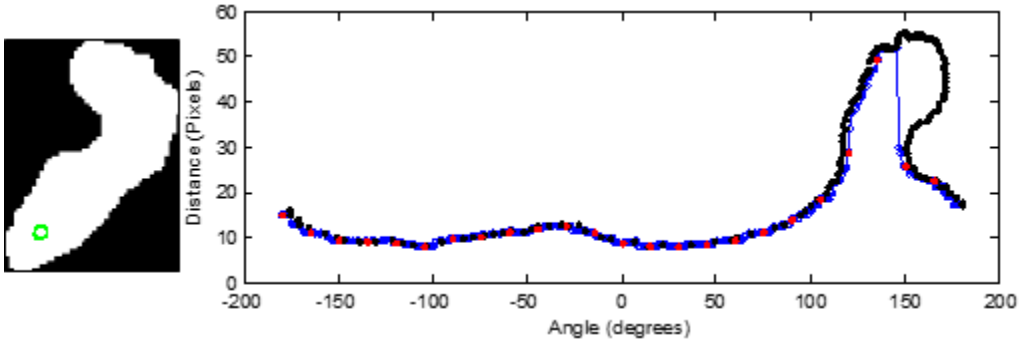


Figure 7.2: The shape profile of the cross-section on the left is shown, with the reference point indicated by the green dot. The original shape profile, shown in black, is unwound to form the plot in blue. This is then interpolated to acquire the distances at desired angle increments (15º), forming the final shape profile shown in red.

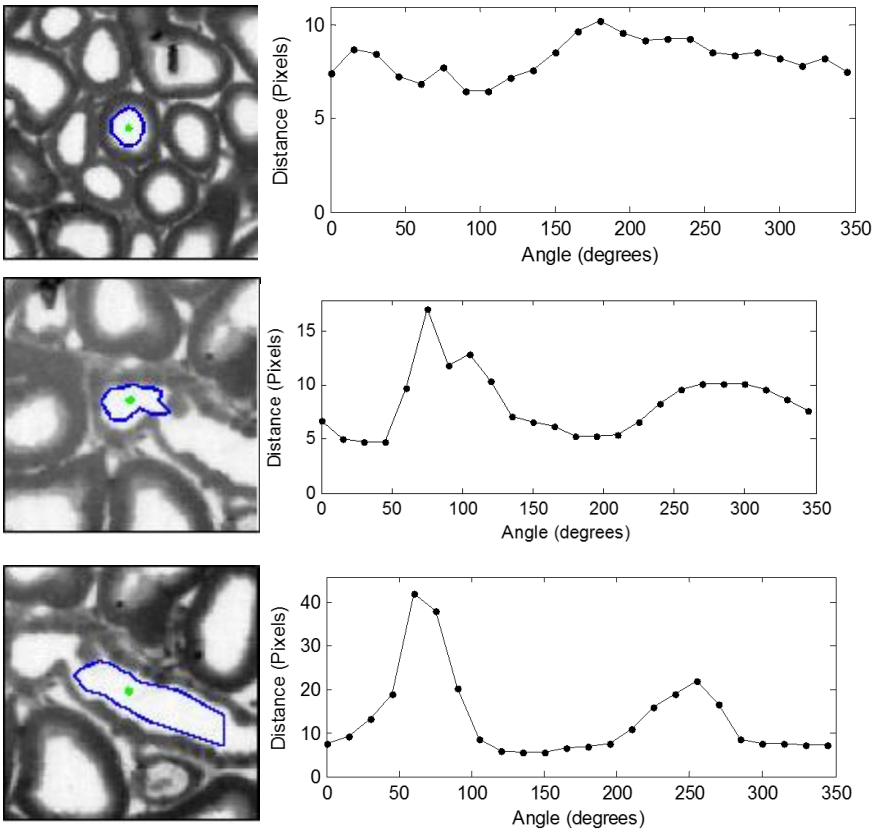


Figure 7.3: The shape profiles relative to nodes on the cross-sections **[**11**]** (green dots) are shown. 15 degree increments were chosen.

Another problem is pixel quantisation leading to discontinuous angles in small sized cross-sections. This problem is solved by scaling each cross-section up to 50x50 pixels prior to shape profile calculation. The final **r** vector is then down scaled. Examples of shape profiles are shown in Figure 7.3

The shape profile of a given object will be different depending on the chosen reference point. The centroid is commonly selected [**36**], but the allocated nodes have been chosen instead as they are more relevant to the problem and allow an accurate relative comparison of shape profiles between linked nodes. This is illustrated in Figure 7.4.

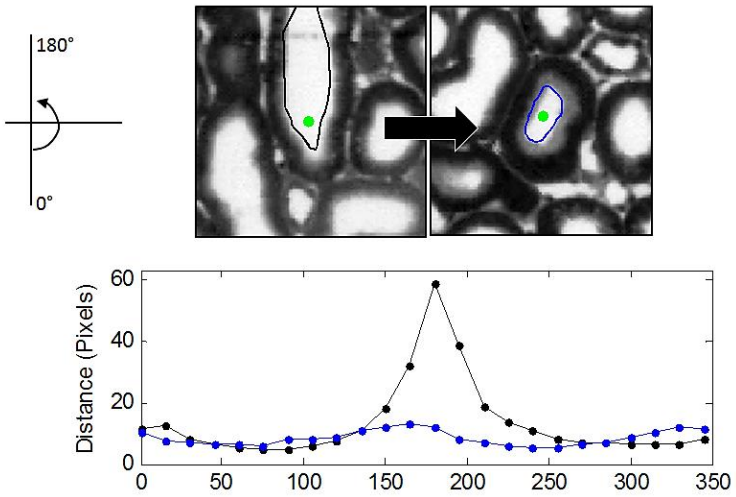


Figure 7.4: If a tracking iteration attempted to move from cross-section 1 (top left) to cross-section 2 (top right), the shape profile of the nodes involved would need to be compared, rather than the profiles around the centroids. As can be seen, when the respective nodes are used, the shape profile is similar for a large range of the angles and can hence be compared. The move shown is a legitimate move between nephron cross-sections of the same nephron. Images adapted from **[**11**]**.

Correlation of the shape profiles (equation 7.5) and a measure depending on their absolute difference (equation 7.6) are two similarity measures which can be used to compare a pair of shape profiles.

$$S_{r_1,r_2} = \mathrm{corr}(\mathbf{r_1}(\theta), \mathbf{r_2}(\theta)) \tag{7.5}$$

$$S_{r_1,r_2} = 100 \times \frac{360}{\delta} \times \mathrm{count}(|\mathbf{r_1}(\theta) - \mathbf{r_2}(\theta)| < 3) \tag{7.6}$$

Where S_{r_1,r_2} is the similarity measure between profiles r1(θ) and r2(θ), corr is the correlation function, count is a function returning the number of elements satisfying the condition in its argument and δ is the angle increment.

7.3 Data Structures

Each cross-section gets assigned a group of nodes and shape metrics with a shape profile per node as shown in Figure 7.5. Ideally, all the information belonging to a node should be stored along with it. However, the nodes, shape factors and shape profile occupy different amounts of memory and require different access speeds.

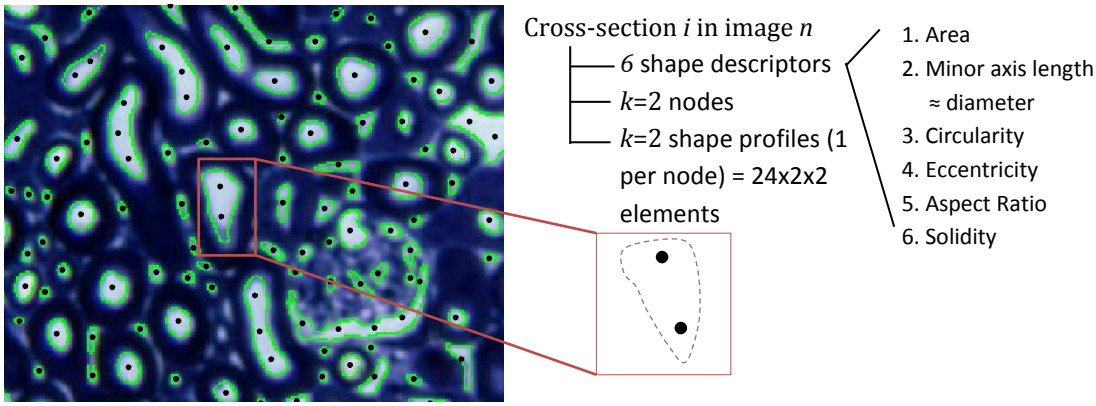


Figure 7.5: A clip of a raw image is shown **[**11**]**. The extracted cross-sections after pre-processing are highlighted in green and the allocated nodes are shown as black dots. Each cross-section will have k nodes, 6 shape factors and k shape profiles. As can be seen, many cross-sections in the cortex are not of actual nephrons but rather of the connective tissue between them. The glomeruli are also highly segmented.

The nodes need to be constantly and quickly accessed during tracking. Since they simply consist of x-y-z coordinates, they can be stored in working memory (RAM) for efficient access. In order to link the nodes with the other information, each cross-section is allocated an identity number, and each node is allocated an identity number relative to the cross-section it lies on. The nodes are stored in a cell array with each cell containing a fixed array for the node data for one image. Each row in the fixed array contains information about one node. Nodes lying on a common cross-section have the same identity number (ID), hence allowing for easy detection of elongated cross-sections during tracking.

The shape factors are accessed often and consume moderate memory. It is thus stored in working memory in the same manner as the nodes. Each row of the fixed arrays corresponds to the row for the nodes and is hence implicitly linked.

The shape profiles occupy a massive amount of memory as they consist of 50 elements per node (24 angles, 24 distances, 1 cross-section ID, 1 node ID). The shape profiles of an average rat dataset would occupy $2500\text{images} \times 20\,000 \frac{\text{nodes}}{\text{image}} \times 50 \frac{\text{elements}}{\text{node}} \times 8 \frac{\text{bytes}}{\text{element}} = 18.63\text{GB}$ if stored uncompressed in working memory. Since they are only accessed during machine learning validation, they are stored on hard disk and only the required elements are accessed using an input-output *matfile* structure in MATLAB.

The chosen storage format together with custom-coded utility functions (see Appendix) provides ease of access and complete traceability of information related to any particular nephron cross-section.

7.4 Glomeruli Detection

The glomeruli can be distinctly located by eye and possess unique characteristics which can distinguish them from other tissue. The glomeruli have a characteristic rough texture from the clump of tiny entangled blood vessels. A C-shaped white space is present in most glomeruli. This is the urinary pole which is often seen fusing with the nephron tubule in the span of 1-3 images. A glomerulus spans 15-30 images in the mouse datasets and 50-70 images in the rat datasets, depending on the depth into the kidney (cortical glomeruli are smaller while juxtamedullary glomeruli are larger [**1**]).

In order to fully model a glomerulus, an ellipsoid should be fitted in the 3D image space to cover its full volume. For purposes of tracking, the presence of a glomerulus must be detected when the nephron tubule merges with the urinary pole, so that tracking can be terminated. This termination is critical as it has been seen that the afferent/efferent arterioles are sometimes tracked out of the glomerulus, through connecting vessels and to the glomeruli of other nephrons.

This linking of multiple nephrons is highly undesirable. Two methods suitable to the given image sets have been devised for glomeruli detection.

The first method involves measuring the peak densities of an edge image. A log edge detector is applied to the binary image. In the edge image, the glomeruli have more edge pixels than the surrounding nephrons due to the numerous small binary fragments making up the glomerulus. Therefore, the average pixel density at the glomeruli is higher than elsewhere. An averaging filter is applied after which the peaks are extracted. Distance-based clustering is then used to obtain single point representations of the glomeruli.

The second method involves grouping alike cross-sections using shape factors as features during unsupervised learning. K-means clustering is used. If 5 clusters are chosen, clustering groups the cross-sections into those that are:

1. very circular – high circularity
2. slightly elliptical - low eccentricity and circularity
3. very elongated – high eccentricity and aspect ratio
4. very small in size – small area
5. concave, C-shaped structures – low solidity

The last class distinctly contains most of the C-shaped urinary poles of the glomeruli, along with some C-shaped nephron cross-sections.

An example of the results produced using these two methods is shown in Figure 7.6. Both methods have false positives and negatives but could perhaps be combined to improve accuracy. Issues with these methods include:

1. Nephrons become smaller deeper into the image set and hence their mean edge values go up as well, producing false positives. Connective tissue also sometimes looks like glomeruli. Parameters of the algorithm need to be varied per image to get accurate results.
2. Some glomeruli are missed (false negatives) as their edges are not pronounced or the C-shape is not present.

The glomeruli coordinates found from the union of the results of the two methods can be used as starting seeds for tracking. Another method was found to be more suitable for glomeruli termination during tracking. The method forms part of the validation and machine learning stages of the system and is discussed in Chapter 9. It basically involves detecting the fragments making up the glomerulus during tracking through a trained classifier.

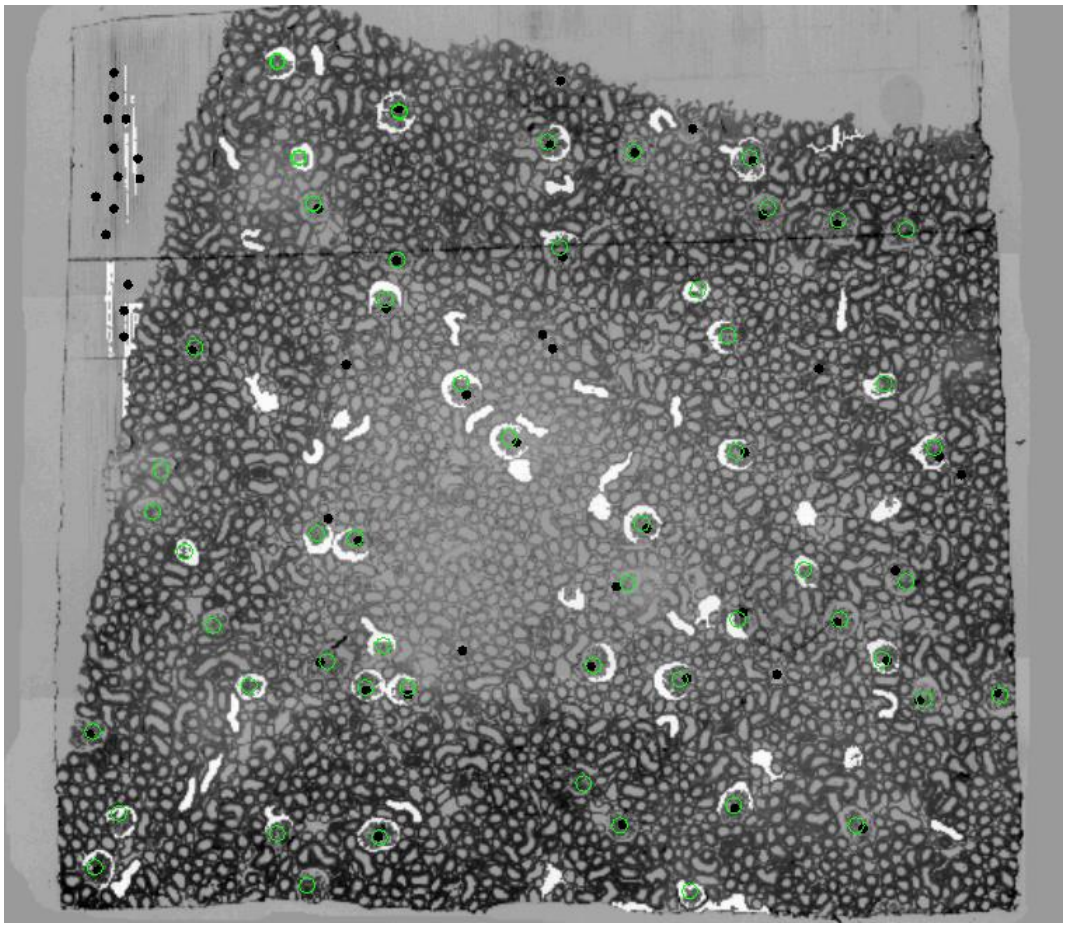


Figure 7.6: The glomeruli detection using the edge image density is shown by the black dots. The detection using shape clustering is shown by the cross-sections highlighted in white. True glomeruli locations were manually marked in green. Both methods can detect a large number of glomeruli and could be used to cross validate the results. Image from **[**11**]**.

CHAPTER 8

Tracking Algorithm

When a nephron is manually tracked by eye, an intuitive process is used by the brain. Once a nephron cross-section has been fixated on, a nephron cross-section within the same vicinity is searched for in the next image. Size, shape and colour are also subconsciously compared. The tracking algorithm uses a similar process, with a number of generalised rules to accommodate the tortuous paths taken by the many nephrons. The algorithm is highly dependent on the quality of pre-processing and accuracy of feature extraction stages. The major processes of the tracking algorithm are shown in Figure 8.1. Each activity is discussed in its respective section in this chapter.

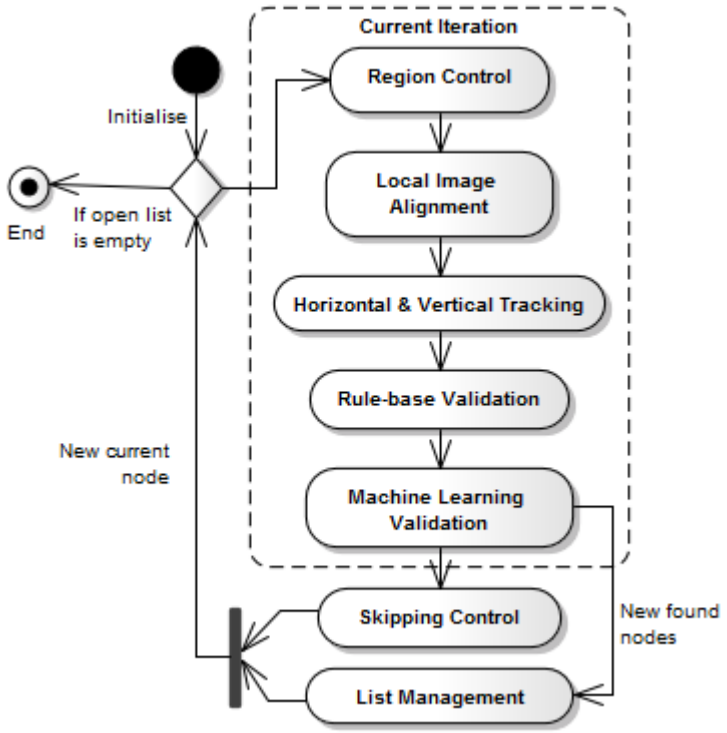


Figure 8.1: An activity diagram of the tracking algorithm. Each iteration explores possible child nodes from one parent node. Once the open list is empty, manual intervention is requested at the end points. Once the nephron is complete, the closed list is reconstructed into an ordered path.

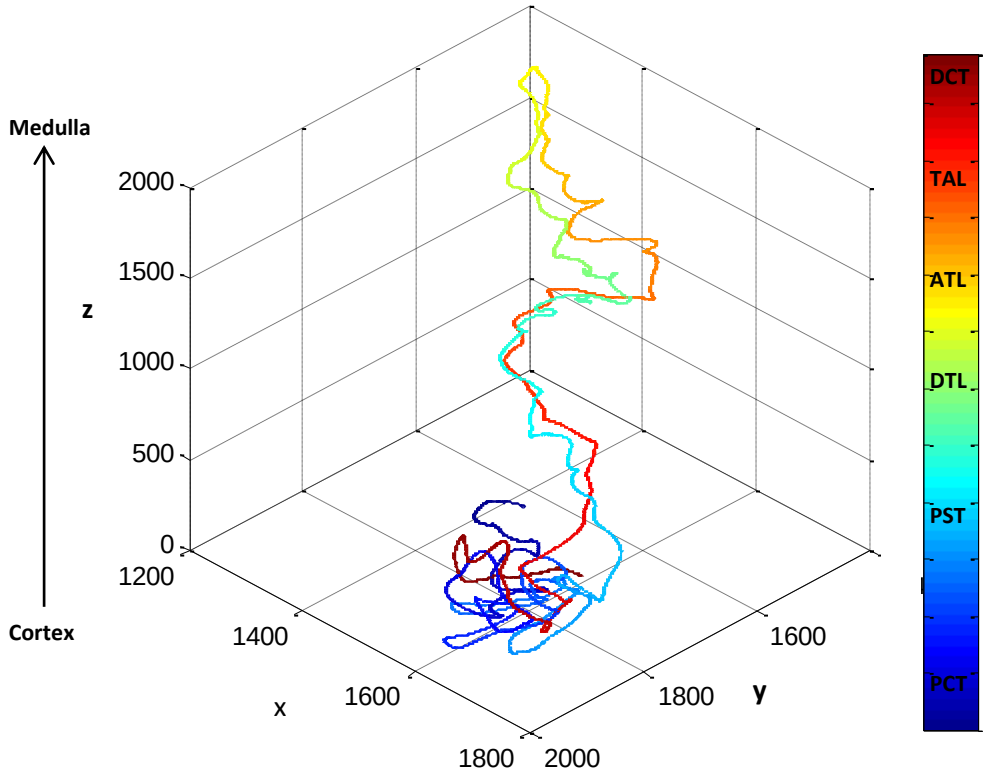


Figure 8.2: An example of a manually tracked rat nephron is shown **[**11**]**. The spatial dimensions of the x-y-z image space are illustrated. Note that the positive z direction extends from the cortex to the medulla as shown. Colour is used to highlight the different parts of the nephron as shown by the colormap on the right. This convention will be used for all 3D plots.

An example of a manually tracked nephron is shown in Figure 8.2. The tracking algorithm aims to produce such a result in an automatic manner. The z-dimension of the 3D image space refers to the dimension along the image set, where z=1 is the first image in the cortex. Note that 'upwards' (z+1) refers to movement towards the medulla while 'downward' (z-1) refers to movement towards the cortex. This is contrary to the standard orientation where the cortex is positioned superiorly and the loop of Henle is drawn extending downwards. Prior to proceeding, a few symbols are defined:

I_z	Image *z*
$\mathcal{C}_z$	The set of all nodes in image *z*
c_z^i	The set of nodes on cross-section *i* in image *z*
$c_{k_z}^{\ i}$	The kth node on cross-section *i* in image *z*

8.1 Local Image Registration

Local alignment is needed (in addition to the previous alignments [**9**] discussed in Section 6.1) due to the presence of local image distortions and progressive change in morphology. The procedure followed involves cropping images I_z, I_{z+1} and I_{z-1} around the current node location.

Image I_z is multiplied by a Gaussian function in order to give the centre of the image (the area around the current location) a higher weight than the surroundings. This is particularly useful when tissue folds occur, where two areas of the same sub-image can be aligned differently, such as in Figure 8.4. The preference is then given to the choice which best aligns the current node.

The sub images in I_{z+1} and I_{z-1} are cross-correlated against sub-image I_z, in order to obtain two pairs of translational offsets (x_{off}, y_{off}) between the images [**41**]. The offset is obtained by extracting the peak coordinate (global maximum) of the correlation result. The transform in equation 8.1 is applied to each pixel and each node in images I_{z+1} and I_{z-1}. This local alignment only takes translation into account; it is assumed that local rotational offsets are minimal as the previous alignment had largely compensated for rotational offsets.

$$T = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ x_{off} & y_{off} & 1 \end{bmatrix} \tag{8.1}$$

The offsets are typically only a few pixels, but they have a large impact on the accuracy of tracking since some nephron cross-sections are also just a few pixels wide. The example in Figure 8.3 shows how a small alignment can increase tracking accuracy.

After alignment, vertical tracking is attempted. The alignment is not cumulative; it is only with respect to the three images for the current iteration, so that accurate links can be made to the cross-sections above and below. After tracking, the newly found nodes (now in the transformed axes) are mapped back to the original axes using the inverse transform T^{-1}. The transformation must be reversed because each sub-image has a different transform which is not valid for other areas in the same image or other images.

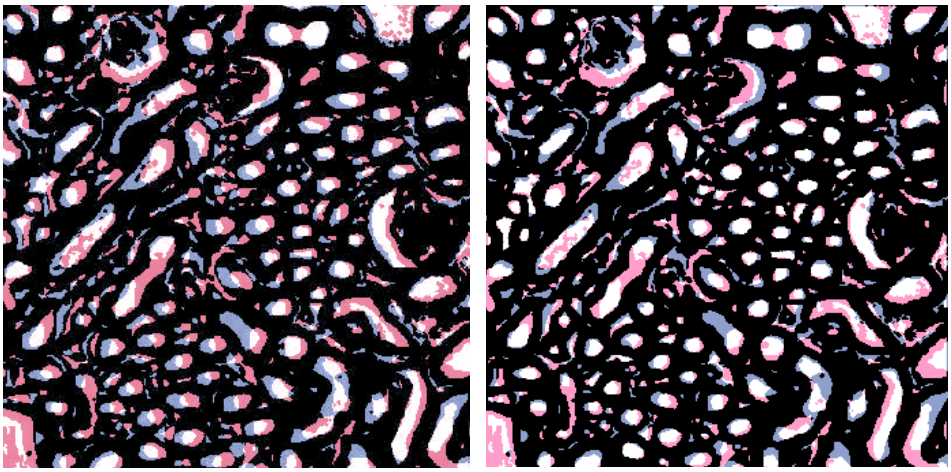


Figure 8.3: Although prior image registration was performed, there still exists an offset between adjacent images as shown on the left. After basic translational alignment (x_{off}=2, y_{off}=9) in the local region, the cross-sections are much better aligned (on the right), allowing more accurate tracking. Original image from **[**11**]**.

The transform is also applied to the sub-images themselves in order to obtain an alignment measure. This value is used to flag images which are highly misaligned even after the alignment procedure. This occurs when artefacts and tissue folds produce missing tissue and non-linear distortion such as in Figure 8.4.

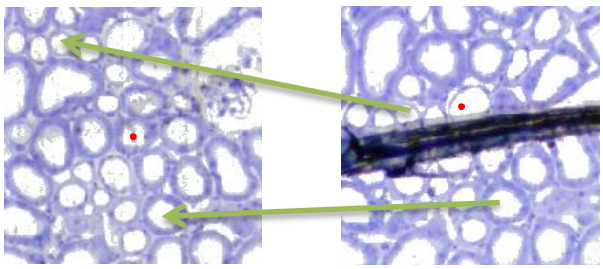


Figure 8.4: Some images cannot be aligned accurately due to artefacts. Such images can be flagged and thus ignored by measuring an alignment metric. If these images are not bypassed, tracking mistakes can easily occur as shown by the red dots. The green arrows indicate corresponding areas. Images from **[**11**]**.

8.2 Graph-based Tracking

Initial tracking attempts made use of hard-coded rules to handle bends and turns and explicitly controlled the tracking direction. Rules for a number of cases were created but these resulted in an inflexible system which was only capable of handling ideal turns and bends (which were modelled by the rules) and tracked only small pieces of the nephron. Inspection of the results showed that the majority of errors occurred when the tracking direction was not chosen correctly. The final tracking algorithm abandoned the concept of direction decision making. Instead, all directions are explored at each node and the final direction is implicitly determined in the reconstructed path.

A graph-based approach is employed for tracking that is similar to the structure of path finding algorithms [**42**]. The algorithm forms a graph in 3D space by establishing edges between the nodes previously allocated during feature extraction. Edge formation is described in Section 8.3.

The algorithm processes one node per iteration and continuously updates an open and closed list. It is given a starting seed, or node. Once the current node has been explored, it is added to a *closed* list, which contains all explored nodes. The new found nodes (child nodes of the current node) are added to an *open* list, which consists of all unexplored nodes. A new current node is selected from the open list at each iteration of the algorithm. This continues until the open list is empty. Ideally, given a starting seed, edges should be formed such that all nodes belonging to the nephron being tracked are collected in the closed list. Each node is stored along with its parent node, forming a linked list. The trajectory can then be reconstructed post-tracking.

By using linked lists and exploring all possible routes, the direction of tracking does not have to be explicitly controlled or limited in any way. The direction can be arbitrary, enabling the tracking of highly convoluted curves in variable directions, as well as branched structures. This accommodates for increased variability in the plane in which the nephrons are sliced, whereas previous methods did not have any tolerance for tissue that was not sliced in cross-section.

8.3 Edge Formation

The edges are established through a controlled set of criteria. Given a particular node $c_{k_z}^{\ i}$ in image I_z, it has the potential to connect to three other nodes through two types of edges as shown in Figure 8.3.

Vertical edge – Includes potential connections to cross-sections in the image above ($I_{z\text{-}1}$) and below (I_{z+1}) the current cross-section. Nodes are searched for which lie within some tracking radius around the current node, i.e. a node satisfying equation 8.2 will become a child node of the current node. The tracking radius varies from the cortex to medulla according to a sigmoidal function.

$$\min\left(\left\|\mathcal{C}_{z\pm1} - c_{k_z}^{\ i}\right\| < r_{track}\right) \tag{8.2}$$

Only one node is allowed to be formed in each direction. If multiple nodes satisfy the condition, the one with the smallest Euclidean distance is used. The confidence of a vertical edge is <1, as the possibility of linking to an incorrect cross-section exists due to the large number of closely packed nephrons in the presence of image distortions and misalignments.

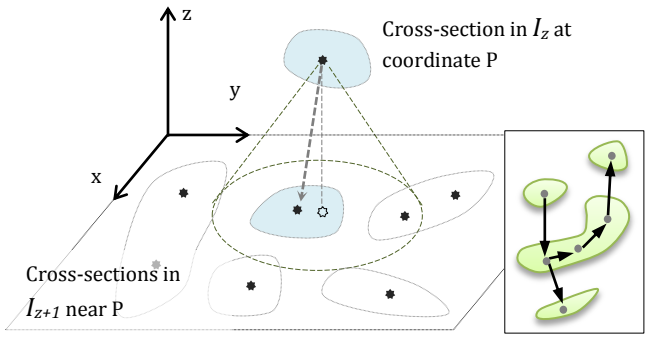


Figure 8.5: Each node in image *z* has the potential to connect to 2 nodes vertically (in images *z+1* and *z-1*) within some tracking radius and 1 node horizontally on the same cross-section as itself. This allows cross-sections to easily be linked through turns and bends.

Horizontal edge – This involves linking together all nodes that lie on the same cross-section as the current node, i.e. c_z^i. The current node is termed the 'entering' node. The pairwise distances between all nodes are used to establish the linkage from the entering node towards the outermost nodes. Only the outermost nodes

are added to the open list for exploration. The intermediate nodes are not explored above and below, i.e. they are 'locked'. This prevents incorrect linking.

8.4 Skipping Images

An image may be termed defective if it has a large amount of interfering artefacts or distortion, which obscures cross-sections of the nephron at hand. These images can safely be skipped while tracking the nephron. However, a maximum of 5 μm of the specimen may be skipped at a time (1 image for the mouse and 2 images for the rat datasets), as the morphology can change vastly in this span and would introduce too large a probability of error in tracking (e.g. jumping onto another nephron).

Since the tracking algorithm makes use of nodes rather than the images themselves, it is unaware of the presence of an image artefact. Skipping is thus attempted whenever tracking in the upward or downward direction is not successful. This results in skip attempts occurring too frequently at every dead end, for example on the last cross-section of a U-shaped bend. A set of skipping criteria are established to prevent skip attempts from occurring too frequently. A direction buffer is used to ensure that skips occur only on straight cross-sections, by checking if there have been recent successful tracking iterations in a particular direction. A refractory period is also used, which is the time (in number of iterations) after a successful skip during which other skips cannot occur.

8.5 Validation Steps

Tracking using the nodes alone would work if the images were exactly aligned and the data only contained information of the nephron cross-sections. However, many of the cross-sections actually belong to connective tissue (ICT) and blood vessels (BV), which are randomly dispersed between the nephron cross-sections and lie in close proximity to the nephrons. Even though the correct nephron path may be found, much interference is caused by connective tissue cross-sections, causing the path to branch from the nephron's path and even link onto other nephrons. A rule-base of three validation steps is incorporated in order to eliminate incorrect moves from one cross-section to another.

a. *Distance Validation* – The Euclidean distance (in x, y space) between a parent and potential child node must be less than the sum of their radii, as in equation 8.3. A coefficient D_{coeff} allows some leeway on this rule. This ensures that even if a cross-section lies within the tracking radius, consistency in terms of size and relative displacement is maintained. Many cases of ICT cross-sections linking to nephrons are eliminated by this rule as shown in Figure 8.6.

$$\|c_P - c_C\| < D_{coeff} \times \frac{MA_P + MA_C}{2} \tag{8.3}$$

Where c_p and c_c are the parent and child nodes and MA_p and MA_c are the minor axes of the parent and child node. The radii are approximated as half the minor axes lengths.

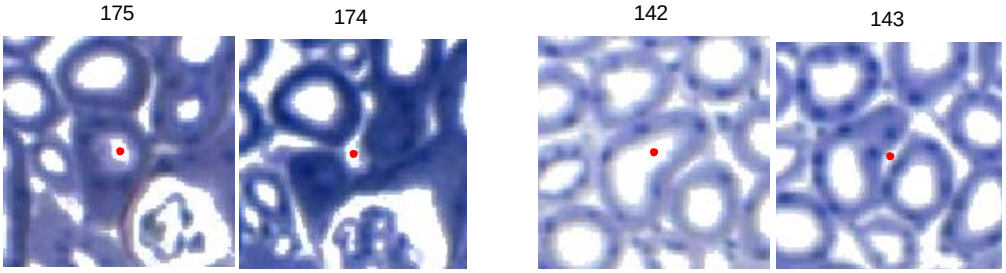


Figure 8.6: Examples of moves blocked by the distance validation rule. Images from **[**11**]**.

---

b. *Bidirectional Movement Validation* – If a move is made from node A in image I_z to node B in image I_{z+1}, then an attempted move from node B into image I_z must lead back to node A (i.e. bi-directionality must be maintained). If not, the move is discarded. Moves between ICT cross-sections and glomeruli are typically not bidirectional and are hence largely eliminated as in Figure 8.7.

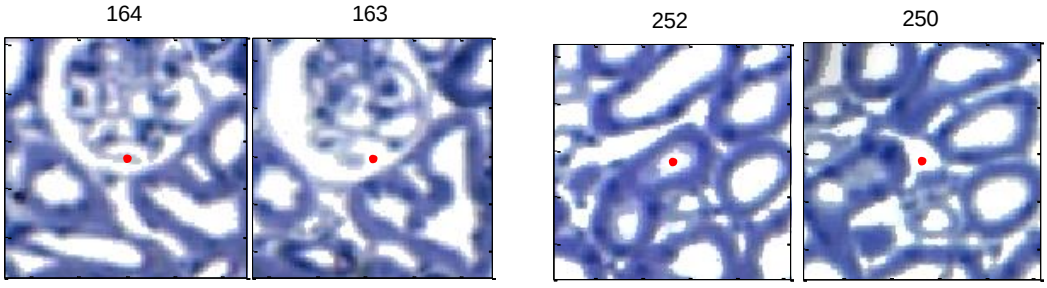


Figure 8.7: Examples of moves blocked by the bidirectional validation rule. Images from **[**11**]**.

---

c. *Skipping Validation* – This ensures that a move involving a skip is only allowed if the shape of the cross-section remains relatively constant during the skip according to equation 8.4. This is so that skips are not allowed on turns and bends, as this presents a high chance of error.

$$\Delta SF = \frac{100}{6}\sum_{i=1}^{6}\frac{|f_i^p - f_i^c|}{\min(f_i^p, f_i^c)} \tag{8.4}$$

Where ΔSF is the overall change in shape factors, f_i^p and f_i^c are the ith shape factors of the parent and child cross-sections, respectively. For a skip to be valid, ΔSF must be less than 30%. Figure 8.8 displays two invalid skip attempts.

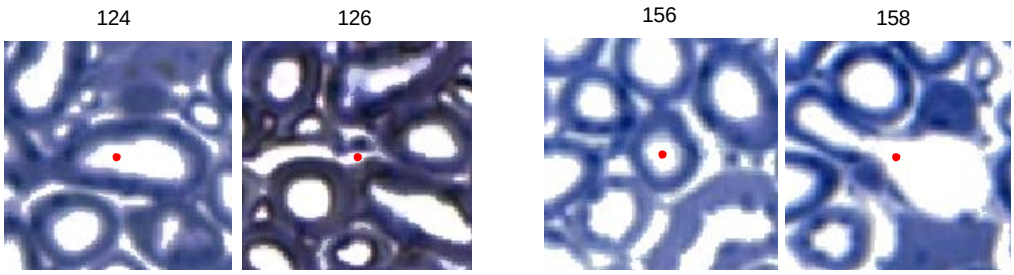


Figure 8.8: Examples of moves blocked by the skipping validation rule. Images from **[**11**]**.

8.6 Region Control

Certain control variables of the tracking algorithm are altered when a transition is made between the cortex and medulla, in order to make the algorithm more suitable to tracking in the respective regions. For example, once the PST narrows into the DTL, horizontal edge formation and vertical edge formation in the downward direction (from I_z to I_{z-1}) is disabled. This is so that only an upward, unidirectional path is allowed to be formed up to the loop of Henle. Using this known information about the nephrons structure prevents tracking errors in the inner medulla, which are common due to small nephron cross-sections merging (when separating walls are too thin) and being very close to one another. This of course assumes that no large bends occur in the medulla. The conditions activated for different regions are tabulated in Table 8.1.

Table 8.1: Different modes of tracking are created by altering conditions at transitions between different parts of the nephron.

Conditions	Cortex (PCT)	Cortex→Medulla (PST→DTL)	loop of Henle (DTL→ATL)	Medulla→Cortex (ATL→TAL)
Upward vertical tracking	Enabled	Enabled	Disabled	Enabled
Downward vertical tracking	Enabled	Disabled	Enabled	Enabled
Horizontal tracking	Enabled	Disabled	Disabled	Enabled
Size of cropping window for alignment	Initialised to 80	Reduce to 50	Stay at 50	Increase to 80

The transitions are detected using the 5th output of a machine learning classifier (discussed in Section 9.2) that produces a continuous valued output with '0' being a move in the cortex and '1' being a move in the medulla. A vector of this output along the tracking iterations produces a *real-time* 'region signal' which is used to indicate the transition from the cortex to the medulla. The signal is smoothed through a running average filter with an *m* sized window to produce a signal as in Figure 8.9. Hysteresis thresholding (with an upper medulla threshold and a lower cortex threshold) is applied to the signal to activate different modes of tracking. This method prevents a rapid fluctuation of activations.

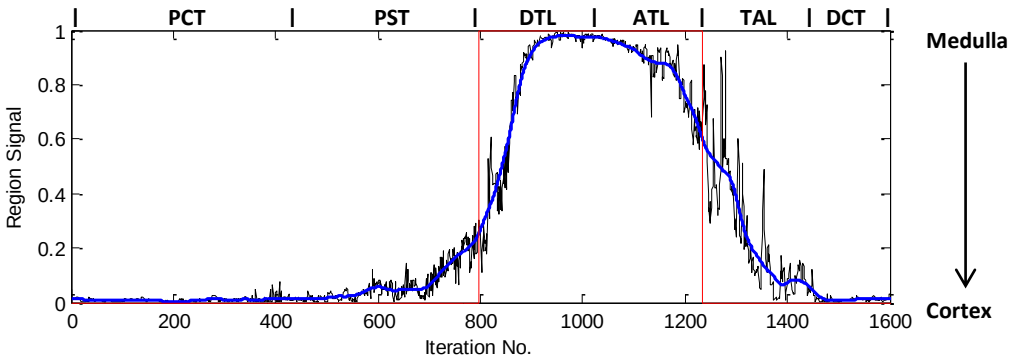


Figure 8.9: The output of the region classifier (black) is smoothed to form a region signal (blue). It is thresholded with hysteresis (red) to produce a binary decision for the activation of different conditions during tracking. The graph shown is for tracking of a whole mouse nephron from the glomerulus to the DCT.

8.7 Reconstruction

Once tracking is completed, the final path is reconstructed through inference of the parent-child node pairs. The longest path forms the nephron path, while shorter branches are eliminated as they are most likely ambiguous nephron paths near turns or pieces of connective tissue that were mistakenly linked. This reconstructed graph is a skeleton structure of the tracked nephron. Each coordinate in the skeleton can be linked to the original cross-section in the binary image as well as its shape factors, either of which can be used to reconstruct a 3D rendering of the tracked nephron.

Lastly, the automatically tracked path must be evaluated in 3D space. At this stage, known information about the kidney can be used, e.g. the proximal and distal convoluted tubules intertwine and must thus be in the same vicinity in the cortex [**10**], or the PCT is longer and more convoluted than the DCT [**10**]. If the results do not adhere to one or more of these expectations, it could then be that the result is incorrect.

Since each vertical move has some associated uncertainty, the reconstructed path can be seen as having weighted edges. The path can then be broken at points with a high uncertainty. In this way, multiple nephrons may be able to be separated if incorrectly linked during tracking.

8.8 Manual Intervention

Premature termination of tracking (due to non-ideal/inadequate pre-processing, feature extraction, image artefacts and distortions) commonly occurs in the inner medulla. One way to overcome premature termination during tracking without introducing error is to allow the user to manually link the end point/s of the automatically tracked path onto the correct path. This of course steers away from a purely automated system, but it still dramatically reduces the time and effort required for the manual tracking task. The degree of automation can be controlled by sensitivity of the validation stages as shown in Figure 8.10.

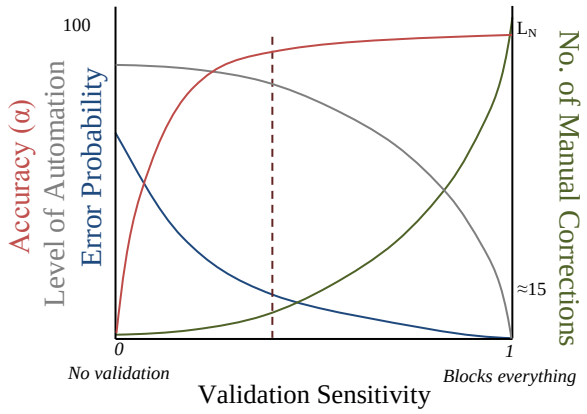


Figure 8.10: The number of false positives increases with increasing validation sensitivity, resulting in premature termination of tracking. This means only a portion of the nephron is tracked, but with a low error, where error refers to the deviation onto an incorrect path. If manual correction is used, the number of corrections required for continuation of tracking will increase with sensitivity (up to LN, the length of the nephron). This means a decreased level of automation but also a decreased chance of error. Note that this graph is merely conceptual.

More than two endpoints are sometimes detected as the last cross-section making up a bend is usually seen as an endpoint. Manual corrections are implemented by displaying to the user the main endpoints of the automatically tracked path, along with 2-3 cropped images before and after the problematic cross-section. The user then simply clicks on the cross-sections which should have been linked by the algorithm. These are added to the open list and the algorithm continues from those points. Correct endpoints (termination at the glomerulus) can simply be ignored.

CHAPTER 9

Machine Learning Validation

Tracking nephrons using the rule-base for validation results in some nephrons being correctly tracked while others are incorrectly linked to other nephrons, ICT cross-sections or blood vessel networks. A large amount of information is not yet taken into account, such as the shape profile and some shape metrics.

The purpose of the machine learning (ML) stage is to incorporate some form of intelligent decision making when linking one node to another during tracking. Machine learning is a suitable technique for this application, as it can automatically amalgamate the large amount of information into a generalised rule through training. The formed rule, or hypothesis, may be unintuitive to a human being and too complex to model using hard-coded logic or inflexible heuristics.

The rule-base eliminates invalid moves between pairs of cross-sections. Likewise, machine learning is incorporated such that a trained learning algorithm assesses the shape descriptors and other features of the cross-sections and classifies the move into one of five classes. This classification result is used by the tracking algorithm to make decisions during tracking. A supervised ANN and SVM are the chosen classifiers as they are non-linear and able to form complex hypotheses.

9.1 Feature Selection

The chosen features must fully characterise a move from one cross-section to another and provide a good degree of distinction between different types of examples. Since two cross-sections are being compared, it is useful to look at their combined features. A total of 67 features are formulated which include:

- x_1-x_6: the difference in the shape factors of area, eccentricity, solidity, aspect ratio, minor axis and circularity
- x_7-x_{12}: the mean of the shape factors of area, eccentricity, solidity, aspect ratio, minor axis and circularity
- x_{13}: the minimum area between the two cross-sections

- x_{14}: the Euclidean distance between the two nodes in the x-y plane
- x_{15}: the image difference, which is normally 1 but can be 2 or 3 if images have been skipped
- x_{16}: the magnitude of image alignment offset – a high offset coupled with other odd features may be a flag for an invalid move
- x_{17}: the position of the pair (average z coordinate) relative to the image set, which indicates depth into the kidney, i.e. cortex, outer medulla, inner medulla
- x_{18}: the correlation coefficient between the two shape profiles
- x_{19}: a correlation coefficient between the two sub-images around the location of the move
- x_{20}-x_{43}: shape profile at 15 degree intervals of cross-section 1
- x_{44}-x_{67}: shape profile at 15 degree intervals of cross-section 2

9.2 Training Set Formation

The training set is created by capturing moves, or pairs of nodes, during unsupervised tracking (without any machine learning validation) of a chosen set of nephrons. Five classes are chosen for classification as described in Table 9.1. Each parent-child pair is assigned a label as shown in the examples in Figure 9.1.

Table 9.1: The intermediate output classes of the learning functions and their combination into final classes

Final Class	Intermediate Class
Nephron (Valid)	1. A normal move between circular cross-sections 2. A normal move involving elongated cross-section/s
Non-nephron (Invalid)	3. An abnormal move typically involving ICT or blood vessel cross-sections 4. A move involving a glomerulus cross-section
x	5. A move in the inner medulla

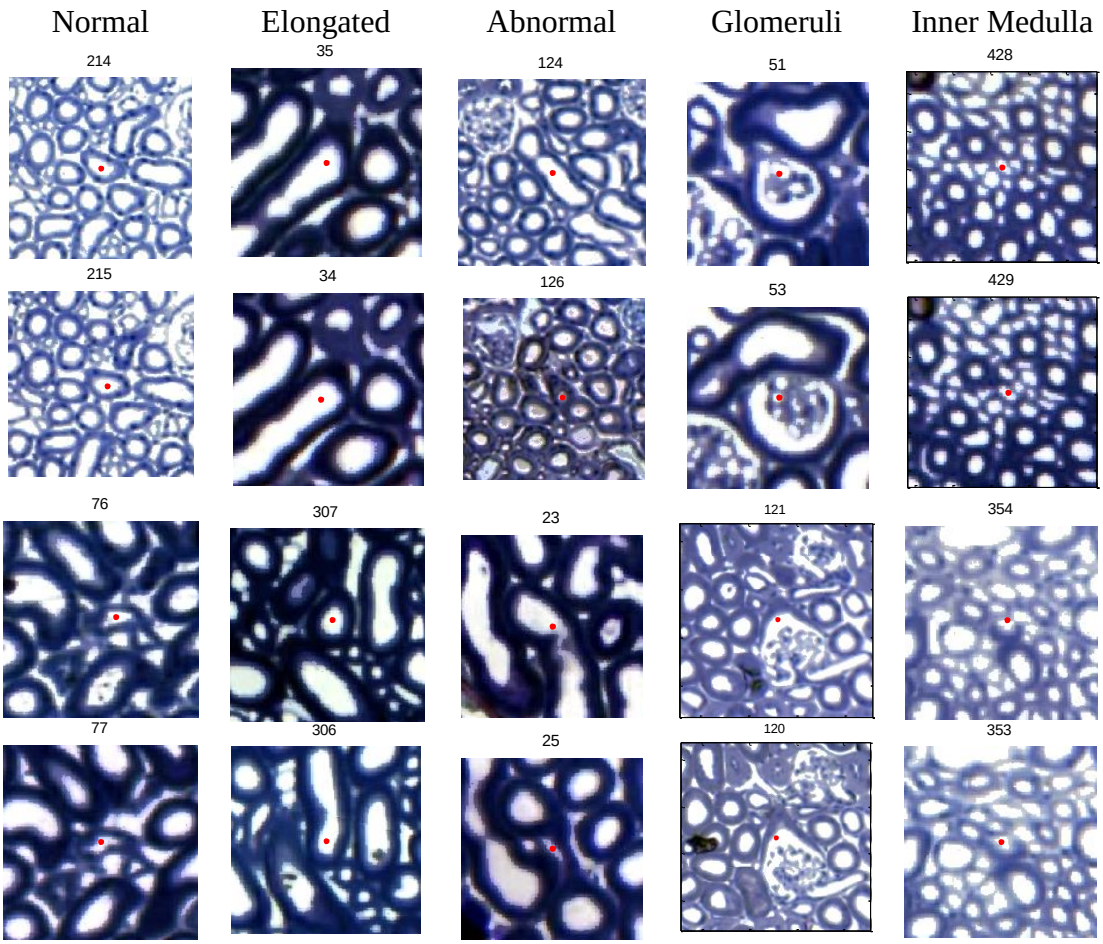


Figure 9.1: The moves attempted by the unregulated tracking algorithm are captured, displayed and labelled to form training examples for the machine learning algorithms. Two examples of moves from each of the five classes are shown. Images from **[**11**]**.

A voting scheme [**29**] between the five classes is then used to determine the final classification as valid or invalid. Class 4 is used to terminate tracking at the glomerulus while class 5 is used as a region signal to change the mode of tracking (parameters of the algorithm) from the cortex to inner medulla (as in Section 8.4). The shape factors and descriptors belonging to each cross-section in the pair can be extracted as required and the 67 features are then combined to form the input matrix. A multi-class classifier is produced using the one-vs.-all approach.

9.3 Training

The training set consisted of 9424 examples, with a ratio of 0.58 : 0.10 : 0.11 : 0.07 : 0.13 for classes 1 to 5, respectively. Although the types

of examples are skewed, there are a sufficient number of examples per class (at least 650). The input is randomised and each feature is normalised. The labelled data set was split into training, validation and test sets with a 0.7:0.15:0.15 ratio, respectively. Training of the ANN and SVM was carried out using built in MATLAB functions [**33**].

A threshold is applied to the continuous output of the ANN in order to deem the result positive or negative. This threshold has an impact on the sensitivity of invalid move rejection. For the SVM, the width of a radial basis function (RBF) kernel has the analogous effect. It is critical that false positives are minimised as a false positive would halt the tracking process by blocking a valid move along the path of the nephron, hence preventing the rest of the nephron from being tracked. A false negative on the other hand, would allow an incorrect path to be formed, but the incorrect branch is typically halted due to the presence of many invalid moves through connective tissue, and is therefore not as critical as a false positives.

9.4 Reinforced Learning

In addition to manual selection of examples, a method involving feedback from the tracking algorithm and the training process was used in order to collect a sufficient number of examples per output class. This reinforced learning procedure prevented the formation of a skewed dataset or the under-representation of a certain class, which may have affected classification accuracy. This feedback process is illustrated in Figure 9.2.

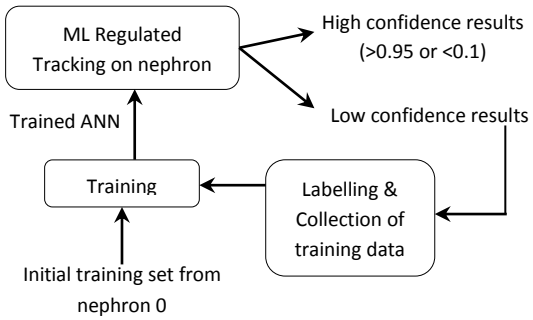


Figure 9.2: A schematic showing the method employed for reinforced learning, which aims to decrease skewness among the five output classes.

9.5 Feature Analysis

A number of the features may be redundant or irrelevant. A subset of the most useful features can be determined through various feature selection techniques. The RELIEFF feature selection method [**43**] has been used which ranks each feature by its importance. The analysis shows that shape profile correlation, average solidity, the pair's z-position and average eccentricity are the most useful features. The raw shape profiles have a medium importance, while the image difference, difference in area and difference in minor axis length have the least impact on classification and could be removed. However, the number of features is not too excessive (only 18 features and 2 shape profiles) and so all features are included.

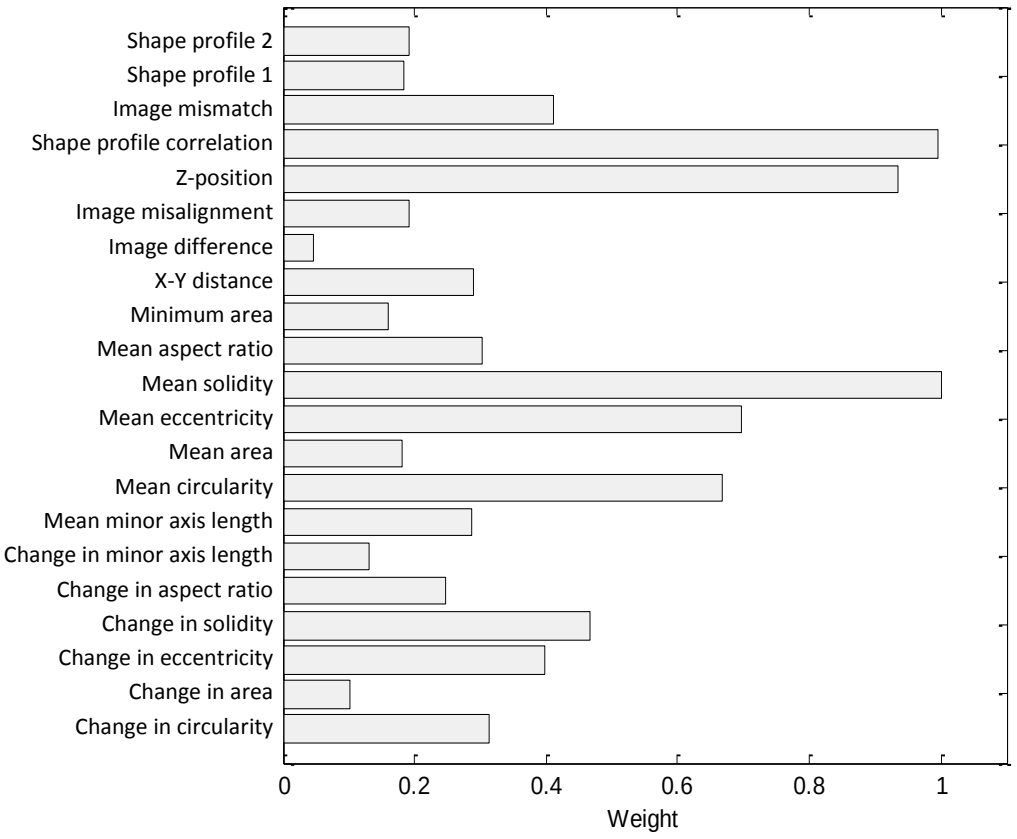


Figure 9.3: The RELIEFF feature selection method allocates a weight to each of the 67 features indicating its importance during classification of a move.

It is also useful to visualise the separability of the five classes and the impact of the features on the five classes of examples. One could simply view plots of two

or three features at a time and try to deduce their relationship and their impact on classification. However, this is limiting and does not represent the overall effect of all features on the classification problem.

Principal Component Analysis (PCA) is a dimensionality reduction technique, useful for visualising high dimensional data. It uses Single Value Decomposition (SVD) to obtain eigenvectors and eigenvalues. A training set of 9424 examples and 67 features was reduced to 2 dimensions yielding the plot in Figure 9.4.

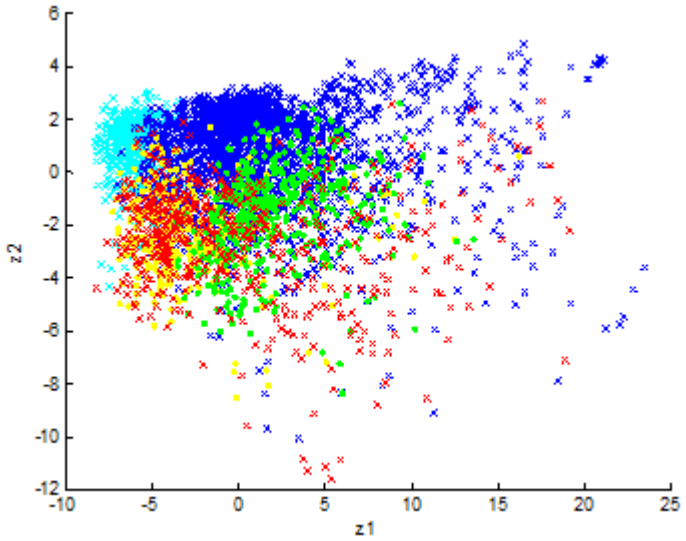


Figure 9.4: The scatter plot displays 9400 examples in terms of reduced features z1 and z2, which were projected from the first 2 eigenvectors of the data. Moves that are normal (blue), involving elongated cross-sections (green), connective tissue (red), glomeruli (yellow) and moves in the inner medulla (cyan) are shown.

Figure 9.4 shows that there are large overlaps between the different classes of moves especially between connective tissue and glomeruli types. However, this is expected as the cross-sections in these two classes are very similar. Despite overlapping, separating lines can be visualised between the other classes. Additionally, a machine learning algorithm using all 67 features will have better resolving ability between the classes than that seen in the plot. The PCA plot uses only two principal features which do not retain variance. That is, the number of eigenvectors (K) needed to adequately represent the original data should satisfy equation 9.1.

$$\frac{\sum_{i=1}^{K} S_i}{\sum_{i=1}^{n} S_i} \geq 0.99 \quad (9.1)$$

Where S_i is the ith eigenvalue. Additional feature analysis plots can be found in the Appendix. For K=2 and n=67, the value is only 0.4677, therefore variance is not retained and the two reduced features do not represent the information offered by all 67 features.

An SVM with an RBF kernel is well suited to creating the arbitrarily shaped hypothesis function required. A neural network with a large number of neurons could also form a complex hypothesis function.

9.6 Optimisation

Initially, the training data was categorised into valid and invalid moves (instead of the five intermediate classes). This resulted in low classification accuracy even with an increasing number of training examples and hidden units of the ANN. This was because there are different types of valid and invalid moves (as indicated by the five final classes), each of which has its own unique characteristics. The different types could not accurately be modelled into one hypothesis function and hence a multi-class classifier had to be used. This resulted in improved classification accuracy.

Under certain conditions, obtaining more training examples improves performance significantly. If the features provide sufficient information to accurately predict the output, and if the learning algorithm can fit a complex function (so that underfitting is addressed), then a large training set will optimise performance as it will minimise overfitting [**28**].

From the Principal Component Analysis of the features, it can be seen that the features do adequately model a move from one nephron cross-section to another. The features are also informative enough for a human operator to correctly classify a move. Both an ANN with a large number of hidden layers (50) and an SVM are capable of forming complex, non-linear hypotheses. The number of training examples was thus increased until there was no longer an increase in performance as shown in Figure 9.5.

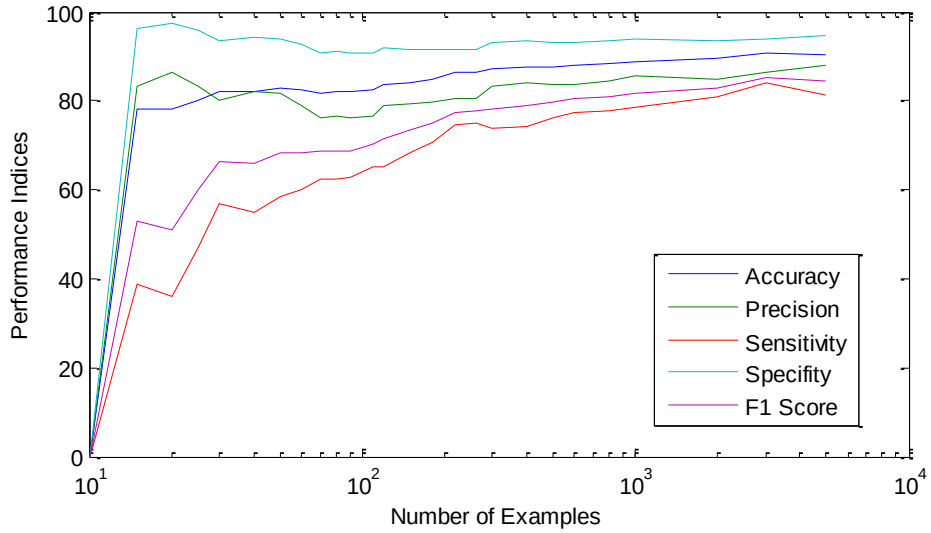


Figure 9.5: Five performance indices were measured on a test set after training the SVM (using an RBF kernel of a width of 5) with a varying number of randomised training examples. The performance converges around a 1000 examples.

CHAPTER 10

Results

The accuracy of each stage is dependent on the accuracy of previous stages as each stage's output is an input for the following stage. The results of each stage are analysed and discussed.

10.1 Pre-Tracking Stages

The pre-tracking stages include the pre-processing and feature extraction stages.

10.1.1 Pre-Processing

The goal of the pre-processing stage was to produce binary images in which each binary component represents one nephron cross-section. This has been achieved in the majority of cross-sections in each image by careful selection and variation of the pre-processing parameters using sigmoid functions.

These automatically varying parameters are suitable to the majority of images in the set but may not be suitable for a few outlying images resulting in the occasional merging of binary components which were meant to be independent. This is termed 'under-segmentation'. Also, the compromise between the equalisation window size and the threshold value sometimes results in the over-segmentation of some binary components which were meant to be whole. These two cases are depicted in Figure 10.1.

The performance of the pre-processing stage is thus evaluated by measuring *segmentation accuracy*, which is defined as the percentage of binary components correctly representing the nephron cross-sections. Due to the impracticality of manually evaluating each cross-section in any one image, 12 samples were selected from different datasets for evaluation. The samples were chosen such that the cross-sections contained were representative of all cross-sections in the respective area. Table 10.1 presents the results.

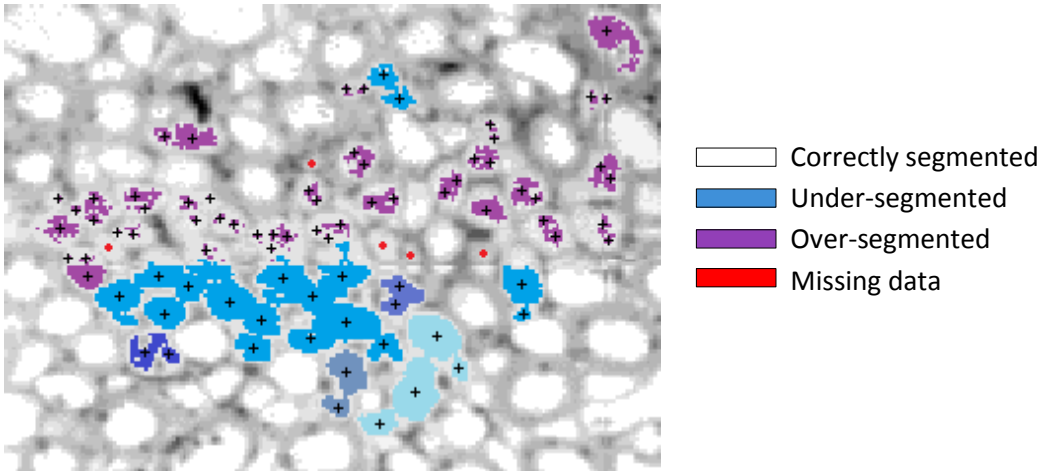


Figure 10.1: A clip of an image in the medulla of a rat dataset is shown with its binary image superimposed. Binary components correctly produced are shown in white. Purple components are those which were over-segmented, producing multiple binary components per nephron cross-section. Components in blue are those which have been incorrectly merged into single components. Red dots indicate nephrons which have no overlying binary components, thus producing missing data. Nodes allocated subsequent to binary image formation are shown by the '+' symbols. Original image from **[**11**]**.

Table 10.1: The segmentation accuracy of samples from 4 datasets in the cortex, outer and inner medulla.

	Segmentation Accuracy (%)		
	Cortex	**Outer Medulla**	**Inner Medulla**
Mouse 1	98.59	98.75	97.24
Mouse 2	98.48	96.62	95.01
Rat 5	95.13	82.92	88.86
Rat 4	96.01	93.61	90.30

The majority of extracted binary components correctly represents the structures in the original image. Under-segmentation occurs when groups of nephrons have very thin walls in comparison to surrounding nephrons (such as in the DTL) causing independent cross-sections to merge. Over- and under-segmentation

mostly occurs in images of the medulla of the rat datasets because the centre of the images is much brighter than the periphery.

10.1.2 Feature Extraction

K-means clustering in combination with the shape criteria has resulted in a highly flexible and adaptive node allocation method. The majority (>99%) of cross-sections receive an ideal number of nodes, especially since the randomly initialised K-means is repeated if the adjacent node criterion is not met. Even on under-segmented binary components (such as the large blue components in Figure 10.1), a node is correctly allocated at the location of each nephron cross-section. On incorrectly segmented binary components, nodes are not allocated per nephron cross-section as each binary component undergoes independent node allocation.

The degree to which the shape factors and shape profiles represent the original nephron cross-section in the colour image is dependent on the extracted binary components during pre-processing. Accuracy of feature extraction stage is thus dependant on the pre-processing stage or segmentation accuracy. Given ideal binary components, the measured shape factors and shape profiles are ideal (100% accurate).

10.2 Measuring Similarity between Paths

In order to evaluate the outcome of a tracking instance, the automatically tracked path must be compared to the corresponding manually tracked nephron. Single number evaluation metrics are used to indicate performance and similarity.

The accuracy of an automatically tracked nephron is measured against the manually tracked data, which forms the gold standard. The following is defined for ease of description:

Υ_f The manually tracked path of nephron *f*

Ψ_f The automatically tracked path of nephron *f*

where a path is a set of coordinates in 3D space.

When the result has a low degree of correctness, it is either because tracking terminated prematurely or the path deviates onto an incorrect one (linkage with another nephron, blood vessel, or ICT cross-sections), or a combination of these. The outcome of tracking a particular nephron is hence evaluated using two measures:

1. α_f = % of Ψ_f that is correct – 'accuracy'
2. β_f = % of Υ_f, that Ψ_f covers – 'extent'

These are calculated by obtaining the per image residuals as in equations 10.1-2.

$$\boldsymbol{r}_{\Psi\Upsilon}(i) = \min\left(\left\|\boldsymbol{\Psi}_{\mathrm{i}} - \boldsymbol{\Upsilon}_{\in z_{\Psi_{\mathrm{i}}}}\right\|\right) \quad \forall i \in \{1, \dots, m_{\Psi}\} \tag{10.1}$$

Where $\boldsymbol{\Psi}_{\mathrm{i}}$ is the ith coordinate in the automatically tracked path $\boldsymbol{\Psi}$, m_{Ψ} is the number of coordinates in $\boldsymbol{\Psi}$, $z_{\Psi_{\mathrm{i}}}$ is the z coordinate (image number) of the ith coordinate, $\boldsymbol{\Upsilon}_{\in z_{\Psi_{\mathrm{i}}}}$ is the subset of the manually tracked path containing all coordinates in image $z_{\Psi_{\mathrm{i}}}$ and $\boldsymbol{r}_{\Psi\Upsilon}$ is the per image residual of $\boldsymbol{\Psi}$ with respect to $\boldsymbol{\Upsilon}$. Similarly the residual of $\boldsymbol{\Upsilon}$ with respect to $\boldsymbol{\Psi}$ is:

$$\boldsymbol{r}_{\Upsilon\Psi}(i) = \min\left(\left\|\boldsymbol{\Upsilon}_{i} - \boldsymbol{\Psi}_{\in z_{\Upsilon_i}}\right\|\right) \quad \forall i \in \{1, \dots, m_{\Upsilon}\} \tag{10.2}$$

The residual indicates the minimum distance of each node in one set to nodes in the same image of the other set. It provides a measure of the discrepancy between the two paths on a per coordinate basis. In order to obtain a single valued similarity measure between whole paths, the residuals are thresholded at some tolerance. This is to allow for differences due to slight image misalignments and differing node positions (the manually placed coordinates may not be in the centre whereas the automatically allocated nodes are more towards the centre), as they should not technically contribute to the error. Accuracy and extent are then defined as in equations 10.3 and 10.4.

$$\alpha = \frac{100}{m_{\Psi}}\mathrm{count}(\boldsymbol{r}_{\Psi\Upsilon} < \mathrm{tol}) \tag{10.3}$$

$$\beta = \frac{100}{m_{\Upsilon}}\mathrm{count}(\boldsymbol{r}_{\Upsilon\Psi} < \mathrm{tol}) \tag{10.4}$$

Where tol is the tolerance in pixels and count is a function returning the number of elements satisfying the condition in its argument.

10.3 Possible Outcomes

The outcome of a tracking instance can be one (or a combination) of four types of cases as shown in Figure 10.2.

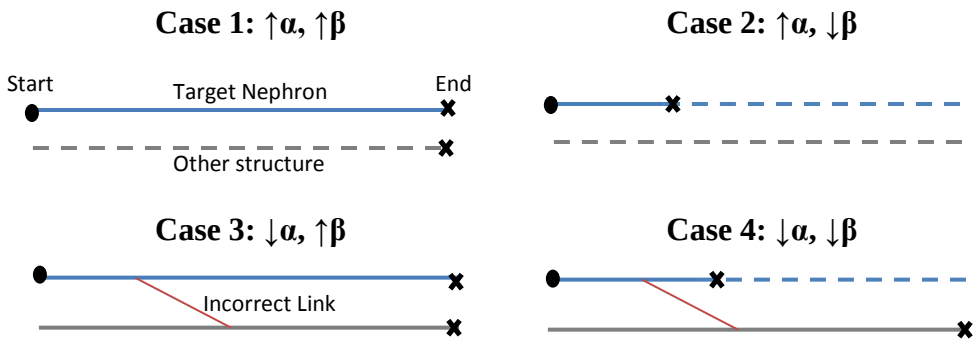


Figure 10.2: The target nephron's path is shown in blue while the path of other structures is indicated by the grey line. Solid lines indicate tracked paths and broken lines indicate untracked paths. The circles and crosses indicate the beginning and end of tracking, respectively.

Case 1: Ideally, the tracking algorithm should track the full length of the target nephron without mistakenly tracking the path of any other nephron or blood vessel (case 1). This would lead to a high α value (the tracked path is well correlated to the target path) and a high β value (a large percentage of the target path has been tracked). However, this is not achieved due to a number of hindering factors.

Case 2: Most often, only a portion of the target path is correctly tracked with no incorrect paths being formed (case 2). This produces a high α value but a low to medium β value depending on where along the path premature termination had occurred. This premature termination is usually caused by artefact interference and large local distortions, which trigger one of the validation steps, causing tracking to terminate. Examples can be seen in the middle column of Figure 10.5.

Cases 3 and 4: Sometimes incorrect links are made to one or more structures other than the target nephron (cases 3 and 4). This is caused by artefacts and distortions

as shown by examples in Figure 10.3. This leads to a low α value as the tracked path does not fully correlate to the target path. The β value can vary depending on how much of the target path has been found in addition to the incorrect paths.

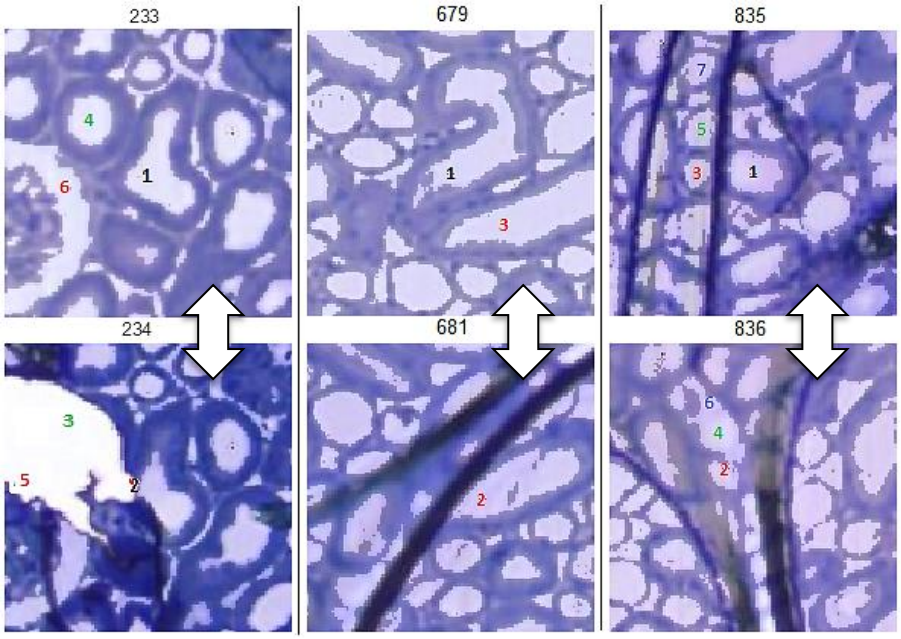


Figure 10.3: Three examples of incorrect linkage to multiple structures are shown. The numbers indicate the order of moves/iterations during tracking, while their colours indicate analogous structures. In the first example (left), the horizontal edges of the large cross-section created by the artefact results in the target nephron being linked to another nephron and an unrelated glomerulus. This move passes all validation stages, with the ANN output of (0.384 0.015 0.277 0.531 0.001) for the five classes. In the other two examples (middle and right), a tissue fold obscures the cross-section of the target nephron and brings a cross-section of another nephron directly beneath the current nephron cross-section. The algorithm sees these as valid moves and incorrect links are made. Images from **[**11**]**.

It is difficult to correct cases 3 and 4 as it cannot be detected without the use of manual data (there are no α or β measurements for unseen data) or manually inspecting the tracked path. Therefore, the algorithm has been designed to minimise the possibility of these cases by establishing the four validation steps. Despite these preventative measures, there are still some incidents of incorrect linking as it is difficult to model each of a variety of cases without hindering normal tracking. Also, some of the invalid moves appear to be valid according to

the model for each validation step, especially incorrect links made to other nephrons.

As a result of the 'strictness' imposed by the validation steps, there is a higher occurrence of premature termination (as opposed to tracking without the validation steps). If the problematic areas are not bypassed, the extent of tracking remains low (as only a portion of the nephron is tracked) even though the algorithm is capable of tracking the rest of the nephron. Manual intervention is used to manually bypass such points so that tracking can continue.

10.4 Tracking Results

The tracking algorithm successfully tracks large portions of the nephrons automatically, occasionally requiring manual correction in order to obtain full nephron paths. Different parts of the nephrons were tracked with varying accuracies and extents as shown in Tables 10.1 and 10.2 due to differing tubule characteristics. In particular, the PCT and PST were tracked well, while the DTL and ATL were more problematic in both the mouse and rat datasets.

16 nephrons from 2 mouse datasets and 11 nephrons from 2 rat datasets were chosen to form a test set to test the tracking algorithm. Only short-looped nephrons were chosen as the long-looped nephrons proved to be too error prone to track as the cross-sections become increasingly difficult to track deeper in the inner medulla. The chosen nephrons were ones for which manual tracking had been performed and ones that were not used to form the training set for the machine learning algorithms. The nephrons were tracked automatically to various points, i.e. some nephrons were only tracked to the DTL while others were tracked to the DCT, etc. This is because a number of consecutive cross-sections of some nephrons became so small that corresponding binary components were not extracted and tracking (without extensive manual intervention) could not proceed as a result.

The path of each tracked nephron was broken up into the six components (PCT, PST, DTL, ATL, TAL and DCT). α and β values were measured in isolation for

each component, along with the number of manual corrections required in each component. The Appendix contains a spreadsheet of the detailed results. Summarised results are presented in Tables 10.2 and 10.3.

Table 10.2: Test results on a chosen set of 16 mouse nephrons. The number of manual corrections is given as the mean ± 1 standard deviation.

	MOUSE				
Area of Nephron	**Accuracy - α_{MEAN} (%)**	**β_{MEAN} (%)**	**β_{IDEAL} (%) [9]**	**Extent - β_{MEAN}/ β_{IDEAL} (%)**	**Average Number of Manual Corrections**
PCT	95.14	27.36	25	109.44	1.20 ± 1.11
PST	98.24	16.33	18	90.72	0.50 ± 0.71
DTL	80.57	13.90	19	73.16	5.44 ± 1.69
ATL	85.67	14.94	14	106.71	2.46 ± 1.87
TAL	96.32	13.19	14	94.21	3.64 ± 1.55
DCT	72.13	14.29	10	142.90	5.86 ± 3.00
Full	87.49	100	100	100	19.09 ± 1.65
PCT to DTL		57.59	62		7.67 ± 1.25
PCT to TAL		87.38	90		13.25 ± 2.00

Table 10.3: Test results on a chosen set of 11 rat nephrons. The number of manual corrections is given as the mean ± 1 standard deviation.

	RAT				
Area of Nephron	**Accuracy - α_{MEAN} (%)**	**β_{MEAN} (%)**	**β_{IDEAL} (%) [9]**	**Extent - β_{MEAN}/ β_{IDEAL} (%)**	**Average Number of Manual Corrections**
PCT	96.32	28.48	25	113.92	5.20 ± 4.70
PST	90.17	14.64	18	81.33	5.00 ± 2.75
DTL	84.63	15.83	19	83.32	24.00 ± 8.19
ATL	88.47	15.63	14	111.64	13.50 ± 6.95
TAL	97.48	11.50	14	82.14	6.67 ± 3.09
DCT	95.23	13.91	10	139.10	4.33 ± 2.49
Full	80.85	100	100	100	58.70 ± 4.70

α indicates how much of the tracked nephron is correct by measuring similarity to the manually tracked nephron. It is low if the path deviates onto other structures and is high if the tracked path contains data of only the target nephron, be it a small or large portion. β measures how much of the target nephron is tracked; it is low (relative to the ideal β value per segment) if only a small portion is tracked. It can still be high if the path branches onto incorrect structures, as long as a large part of the target nephron is found.

Note that α and β are measured by comparing individual coordinates of the manual and automatically tracked nephrons. The automatically traced path typically contains more coordinates since the algorithm tracks all cross-sections related to the nephron rather than just those required, i.e. all 3-4 elongated cross-sections making up a bend are automatically tracked, while the manual path will label only 1-2 of the elongated cross-sections at a bend. The algorithm also tracks glomeruli cross-sections, whereas the manual path terminates on the last PCT cross-section. This has the overall effect of producing a higher than ideal β value.

The indistinct locations of some transitions (e.g. PCT to PST, or the end of the DCT) also results in the measured beta values being higher than the ideal values. Note that the beta values are relative to the entire nephron length and not the segment in question, for example the PST makes up 25% of the total length, and hence a measured beta value of 24% means that 96% of the PST was tracked.

The number of manual corrections varies with the sensitivity of the validation steps. For example, decreasing the ANN threshold, increasing the coefficient of the distance validation or turning bidirectional validation off will decrease the number of requests for manual correction by the algorithm. However, this increases the chance of branching onto incorrect structures (decreases α) as shown conceptually in Figure 8.10. The settings/conditions for the validation steps were therefore chosen such that the algorithm tracks with high accuracy while also not requesting for manual intervention at unnecessary/unreasonable points.

The frequency of manual intervention is dependent on the number of image artefacts and distortions encountered along the path of the nephron, and the

visibility of the cross-sections. Examples of manual corrections are shown in Figure 10.5. A longer path (in terms of the number of moves) requires more corrections, e.g. the rat nephrons are on average 4.7 times longer than mouse nephrons and long-looped nephrons are at least 1.5 times longer than short-looped nephrons from observation of the manual data. The cross-sections of the DCT in the mouse are very small and hence harder to track than the larger cross-sections in the rat. The number of corrections required in each area of the nephron in the mouse and rat is compared in Figure 10.4, where the rat data is normalised (divided by 4.7) in order to highlight differences other than the image set size.

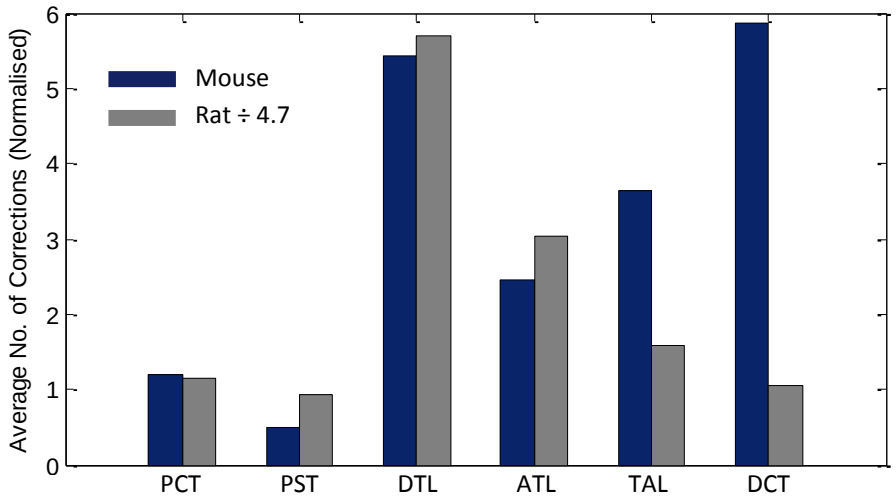
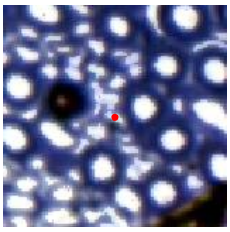
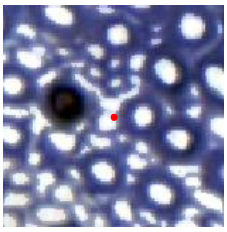
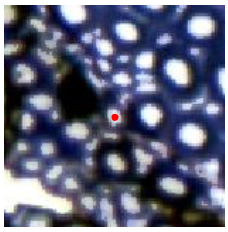
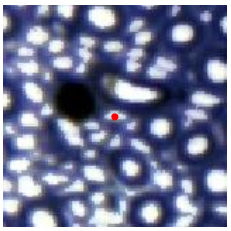
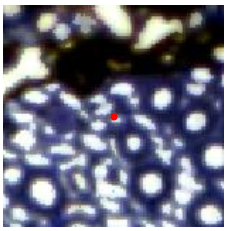
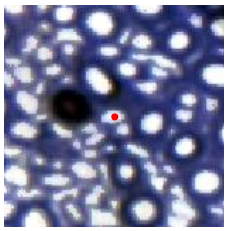
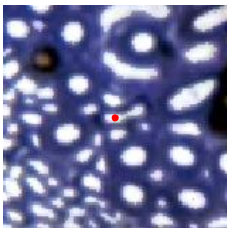
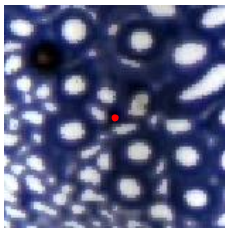
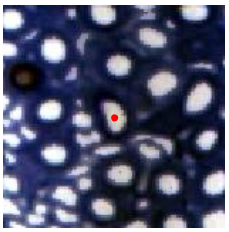
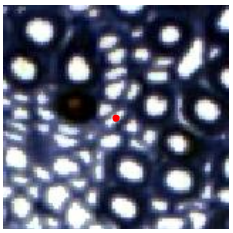
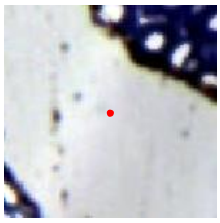
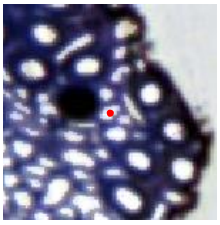
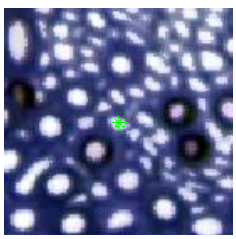
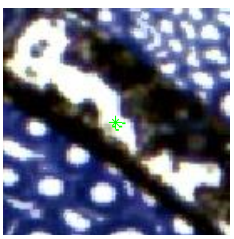
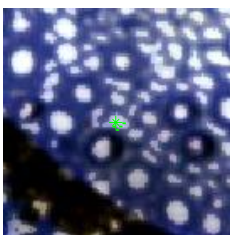


Figure 10.4: The number of manual corrections required for mouse and rat nephrons is shown. The rat data is normalised by the ratio of a mouse and rat dataset (1:4.7) in order to make a better comparison unrelated to image set size.

Automatic Endpoint	Problematic Image	Manual Correction	Notes
450 	451 	452 	Small cross-sections merged during pre-processing
465 	467 	468 	Large artefact prevents accurate alignment
517 	518 	519 	Drastic change in morphology within one image
489 	490 	491 	Missing tissue; A skip is not allowed due to sharp change in morphology
471 	473 	474 	Artefact in 472 skipped; Artefact in 473 produces a large binary cross-section which is linked to another nephron

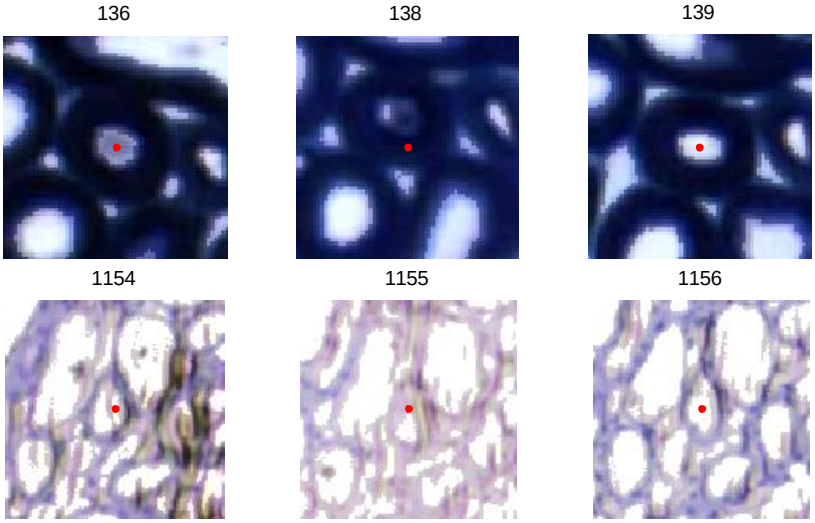


Figure 10.5: Examples of premature termination during tracking requiring manual intervention. The reasons summarised on the right are indicative of the variety of non-ideal situations encountered. Original images from **[**11**]**.

A number of examples of automatically tracked nephrons compared to their manually tracked versions are shown in Figures 10.6 to 10.9. The slight discrepancies seen between the automatic and manual paths are due to different image alignments and different point coordinates used by the two methods.

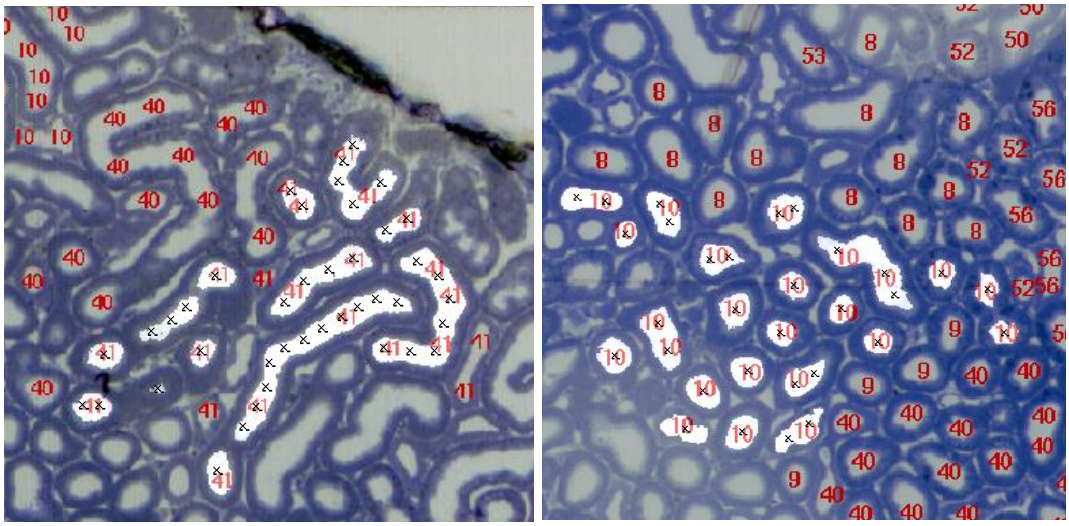


Figure 10.6: Examples of labelled images are shown with the red numbers indicating manually tracked nephrons. The automatically tracked nephrons are superimposed, shown in white with black crosses at the nodes. The automatically tracked cross-sections correspond to the manually labelled cross-sections of the PCT of nephron 41 (left) and nephron 10 (right). The cross-sections of 41 that are not highlighted are of the DCT. Original images from **[**11**]**.

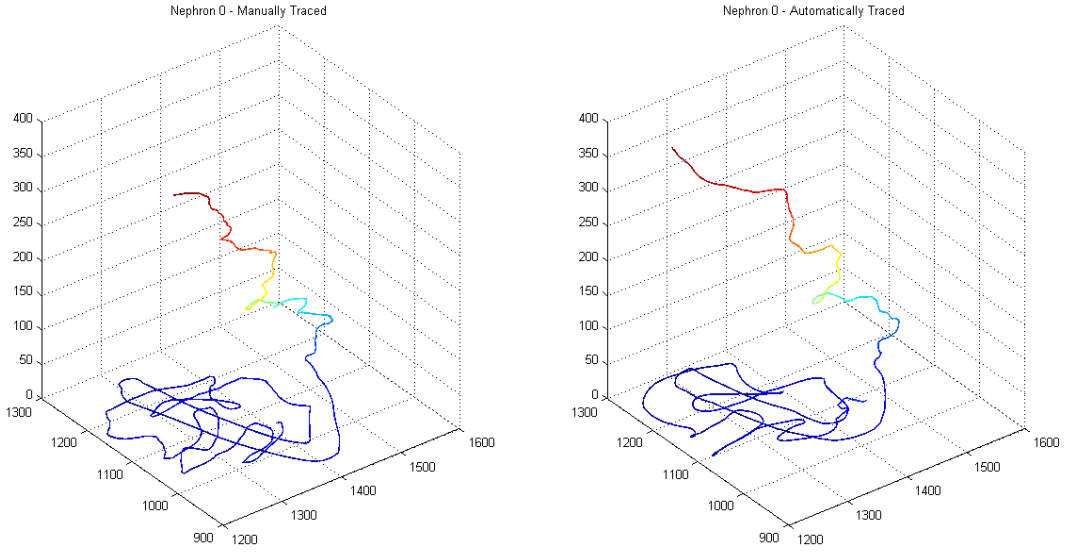


Figure 10.7: A manually tracked nephron (nephron 0 from mouse 1) is shown on the left. The same nephron is successfully tracked automatically by the tracking system. This nephron in particular was only manually tracked to the PST due to low visibility of the DTL cross-sections. The automatic tracking algorithm also experiences difficulty in tracking the DTL. The tracking terminates automatically at the glomerulus. α_0=97%; β_0>100%

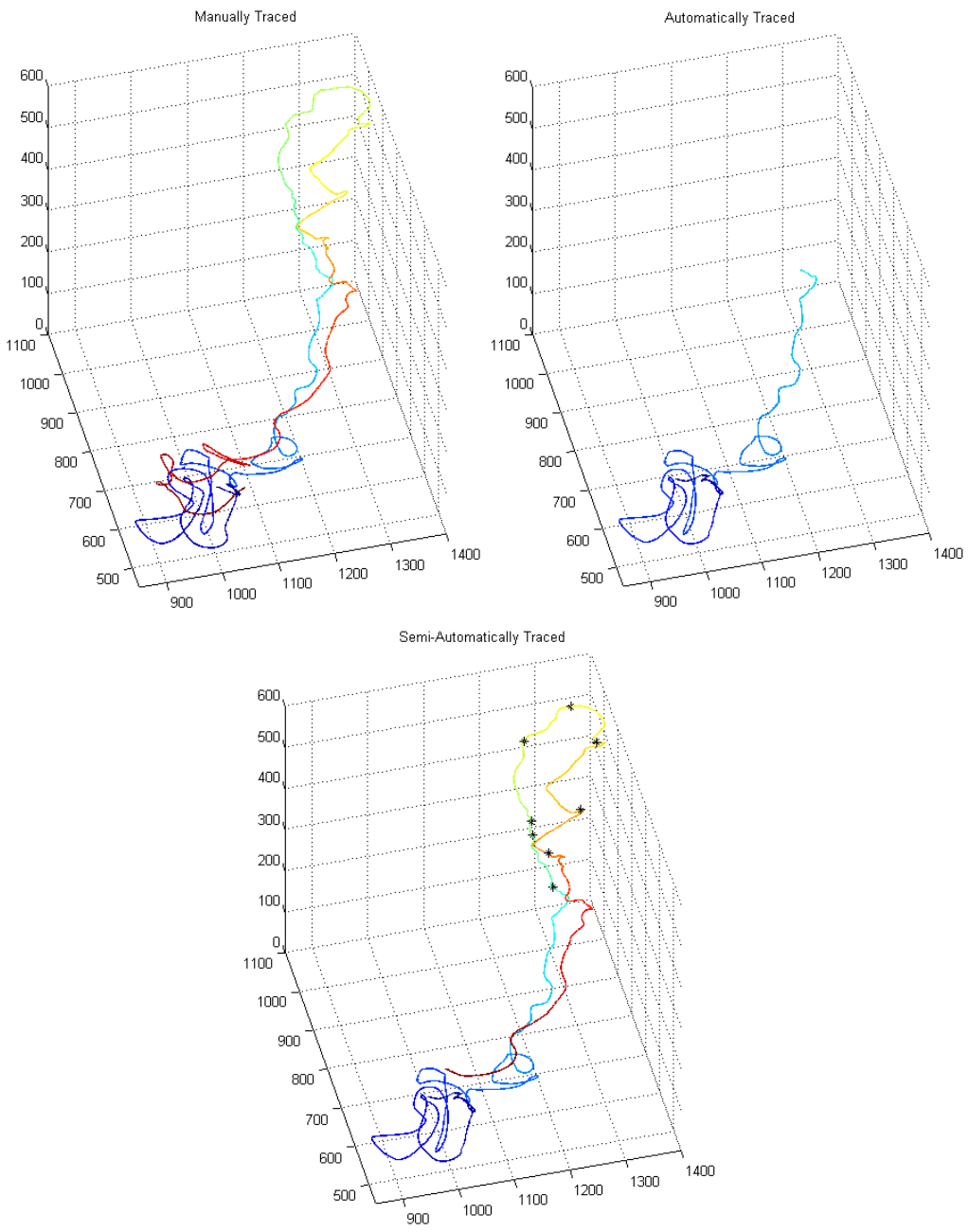


Figure 10.8: A manually tracked mouse nephron is shown on the top left. The PCT and PST are successfully tracked automatically as shown on the top right but the path terminates prematurely due to the presence of an artefact. A complete path is obtained with 5 manual corrections on the DTL and 3 on the ATL, as shown in the bottom image. This is minimal when considering a total of 1222 coordinates making up the path. α_{AUTO}=97.13%; β_{AUTO}=39.84%; $\alpha_{SEMI\text{-}AUTO}$=98.77%; $\beta_{SEMI\text{-}AUTO}$=90.23%

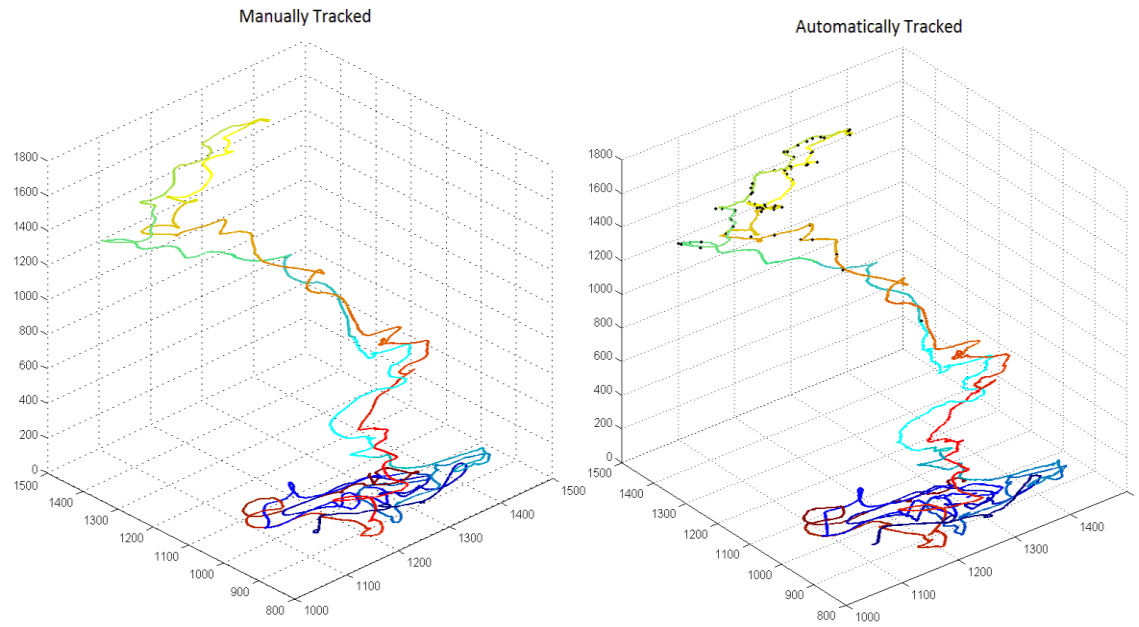


Figure 10.9: A manually tracked rat nephron is shown on the left. The same nephron is successfully tracked automatically by the tracking system with 56 manual corrections in the DTL and ATL. The paths can be seen to be almost identical.

10.5 Efficacy of Validation Steps

Although the types of invalid moves are diverse, the rule-base attempts to model the majority through hard-coded, direct rules while the machine learning validation attempts to model them in a more generalised, less rigid manner.

The validation steps for a particular move are carried out in a set sequence with the least computationally expensive step being first. This is so that if an invalid move is detected, it does not have to go through all of the subsequent stages. However, for testing, all validation steps were carried out. The rejection rates and accuracies are detailed in Table 10.4. The ‘accuracy’ of a validation step refers to the percentage of true positives (moves flagged as invalid that were actually invalid).

Table 10.4: The invalid move rejection rate by the various validation steps and their accuracy is shown. A total of 8017 moves were flagged as invalid in the test set of nephrons.

Validation Step		% of total invalid moves flagged	% of captured moves that are unique	% Accuracy
Distance Val.		40.21	25.94	99.67
Skip Val.	Total	38.59	25.38	90.01
	Skips	98.97		
Bidirectional Val.		29.92	18.94	92.05
ML Shape Val.		57.61	42.46	93.62

All four rules have produced accuracies above 90% with the distance validation rule being the most accurate (99.67%) and the machine learning validation being the most highly triggered (captures 57.61% of all invalid moves). Given a large set of detected invalid moves, certain fractions are uniquely captured by each of the validation steps as shown in Table 10.4. Of the 8017 invalid moves, 49.65% are captured by more than one rule.

Ideally, the machine learning validation stage should be able to perform the tasks of distance and skipping validation, as the rules should be spontaneously integrated into the learnt hypothesis. Since 57.54% of moves captured by the machine learning step are those captured by other rules, it can be said that it does perform the tasks of the rule-base to some degree. It can also be said that the rule-base models the abnormalities to a good degree since the majority of invalid moves are eliminated even without the machine learning component.

10.6 Machine Learning Classification

The trained machine learning algorithms eliminate a large number of invalid moves which would have otherwise resulted in multiple nephrons, ICT and blood vessels being linked (42.46% of its detections are not captured by the rule-base). Both the ANN and SVM were capable of forming complex hypotheses and have performed similarly, producing classification accuracies of approximately 93% on the test set. The confusion matrices are contained in Table 10.5.

Table 10.5: The confusion matrix and classification accuracies of the ANN and SVM on a test set of 712 examples. The examples were classified into one of the five classes. Combined results for classes 1 and 2 (valid moves) and classes 3 and 4 (invalid moves) are shown in bold.

Classification Algorithm		Predicted Class	Target Class y_1	y_2	y_3	y_4	y_5	Accuracy (%)	
ANN (threshold=0.7)		y_1	384	28	14	7	0	88.7	**94.8**
		y_2	23	36	4	1	0	56.3	
		y_3	7	3	48	6	0	75.0	**85.3**
		y_4	7	0	13	32	0	61.5	
		y_5	1	0	1	0	97	98.0	**98.0**
		Accuracy (%)	91.0	53.7	60.0	69.6	100.0	83.8	
			96.3		**78.6**		**100.0**		**93.7**
SVM (RBF kernel of width 5)			y_1	y_2	y_3	y_4	y_5		
		y_1	372	18	11	7	0	91.2	**95.1**
		y_2	32	48	6	0	0	55.8	
		y_3	17	0	63	26	7	55.8	**79.7**
		y_4	1	1	0	13	0	86.7	
		y_5	0	0	0	0	90	100.0	**100.0**
		Accuracy (%)	88.2	71.6	78.8	28.3	92.8	82.3	
			96.1		**80.9**		**100.0**		**93.0**

The class 5 output of both classifiers is highly accurate and is successfully used as the region signal. The classification accuracies of the first 4 output classes are variable between the ANN and SVM. For example, the ANN is better at predicted y_4 while the SVM is better at predicted y_2. For purposes of final classification, many false positives and negatives are irrelevant, as long as they belong to another acceptable class, e.g. a move involving elongated cross-sections (y_2) can be classified as y_1 and a glomerular move (y_4) can be classified as abnormal (y_3). The first four output classes (y_1-y_4) are thus combined into a final decision Y according to equation 10.1.

$$Y = \tfrac{1}{2}(y_1 + y_2 - \max(y_3, y_4) + 1) \qquad (10.1)$$

Y ranges from 0 (an invalid move) to 1 (a valid/normal move). The threshold applied to Y determines the binary decision on validity of a move.

The ANN was made purposely less sensitive and more precise by selecting a high threshold (0.7) in order to substantially minimise false positives. The width of the RBF kernel of the SVM was also chosen to minimise false positives, but it was not as flexible as the threshold of the ANN in manipulating the achieved sensitivity and precision. The confusion matrix and performances of this final classification are detailed in Tables 10.6 and 10.7.

Table 10.6: The confusion matrix of the final classification of the test set.

Classification Algorithm	**Predicted Class**	**Target Class**	
		Valid	Invalid
ANN (threshold=0.7)	Valid	444	34
	Invalid	19	120
SVM with RBF kernel (width=5)	Valid	439	39
	Invalid	2	137

Table 10.7: Various performance indicators for the ANN and SVM. The SVM shows slightly superior behaviour. Equations from [**28**].

		Performance (%)	
Indicator	**Equation**	ANN (threshold=0.7)	SVM with RBF kernel (width=5)
Accuracy	$(TP + TN)/Total$	91.41	93.35
Precision	$TP/(TP + FP)$	95.90	99.55
Sensitivity	$TP/(TP + FN)$	92.89	91.84
Specificity	$TN/(TN + FP)$	86.33	98.56
F1 Score	$2TP/(2TP + FP + FN)$	94.37	95.54

From the indices in Table 10.7, the SVM shows slightly better performance than the ANN. It produces fewer false positives on the test set. However, it is less flexible for use than the ANN due to its binary valued output. The continuous valued output of the ANN is advantageous as the four class outputs can be weighed against one another to produce a more accurate final classification. Examples of true and false positives and negatives produced by the ANN are shown in Figure 10.10.

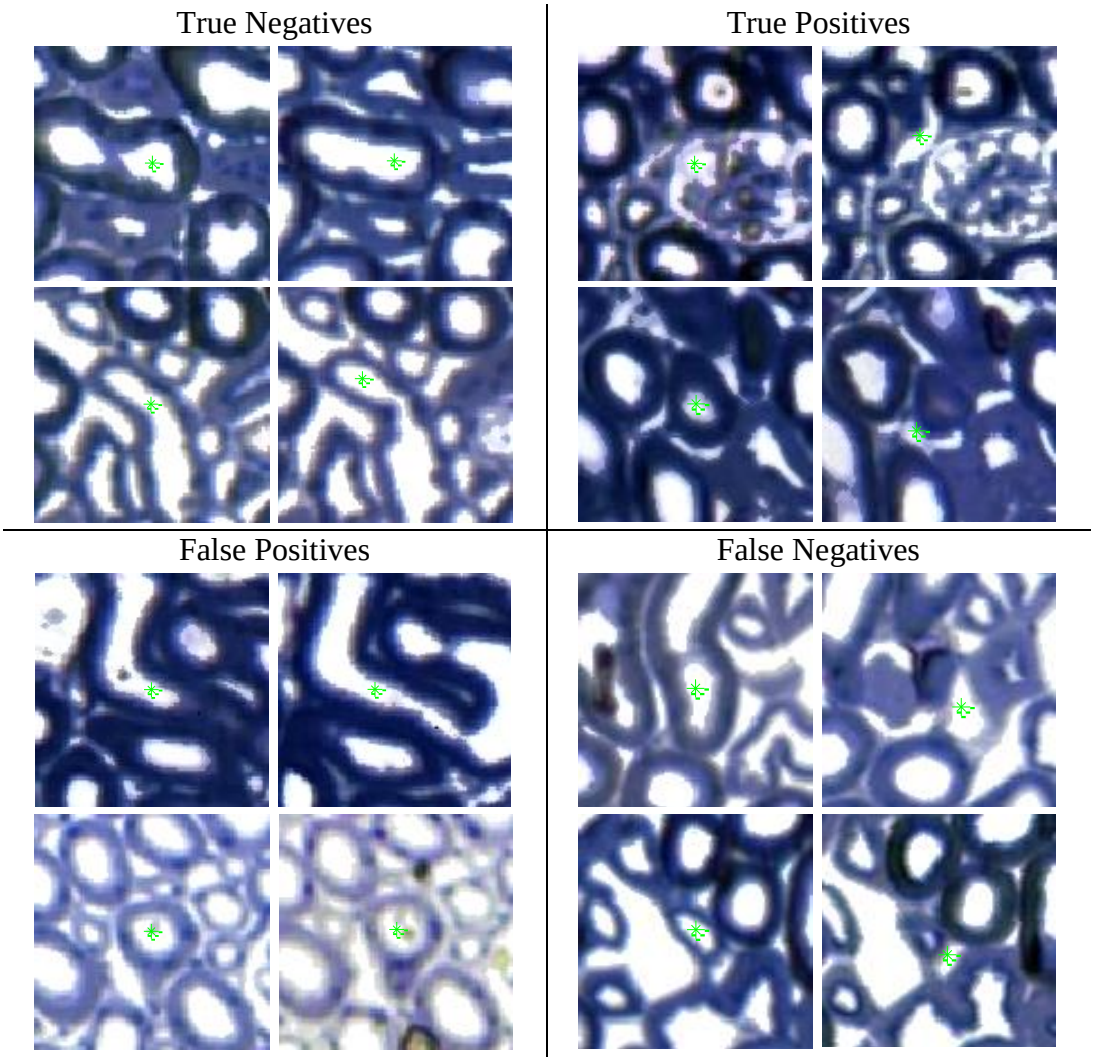


Figure 10.10: Examples of true and false positives and negatives produced by the ANN are shown. False negatives typically involve connective tissue cross-sections. False positives involve nephron cross-sections which have unusual characteristics.

Most false positives seem to consist of nephron cross-sections that were fragmented due to matter in the lumen or non-ideal pre-processing. Moves involving C-shaped, elongated nephron cross-sections were also sometimes mistaken for invalid moves due to their low solidity which is normally a characteristic of glomeruli cross-sections. False negatives typically involved ICT cross-sections which were similar in appearance to nephrons. These occur most frequently when skips are made at the last few cross-sections of a bend.

10.7 Monitoring Runtime Output

It is useful to monitor various variables and flags during the tracking process. For purposes of prototyping, verification and testing a live output log is created by the algorithm. The log is useful when analysing a path post-tracking. Figure 10.11 displays a snippet of an output log while tracking a nephron.

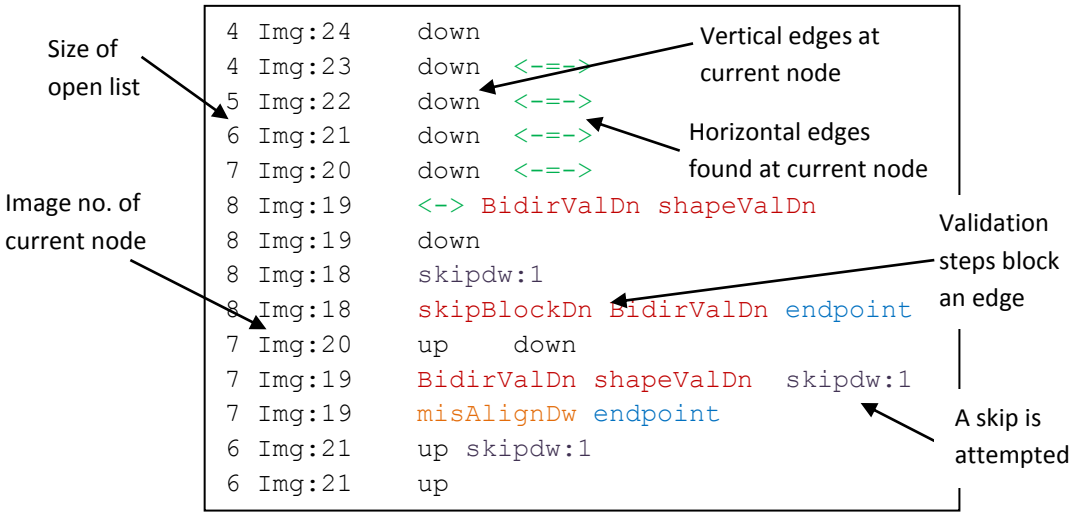


Figure 10.11: An example of an output log during the tracking of a nephron is shown. Each row is the output for one iteration of the tracking code. The left-most number is the size of the *open* list which indicates how many nodes are yet to be explored. The image number of the current node is then output. A number of strings representing the findings and validations at the current node are then shown.

The size of the open list is a useful indicator of the stability of the tracking instance, i.e. if the size diverges at a high rate (grows large very quickly), it is likely that the path has deviated onto another nephron or blood vessel. If many validation steps are being activated over a long period, it is likely that the glomerulus has been reached.

10.8 Processing Times

The main aim of this research was to develop the techniques required for automated tracking rather than to optimise efficiency for a user-end application. Nevertheless, good programming practices have been followed, such as the use of

functions, pre-allocation of memory and efficient use of available memory. Parallel processing was used for the pre-processing and feature extraction stages in order to decrease execution time (by using a MATLAB pool [**33**], or cluster, operating on a *parfor* loop). However, the current implementation can be made more efficient. Computational bottlenecks include the convolution (which uses a 2D FFT) required for image alignment, continuous calling of the ANN structure and reading in three images per iteration (which processes one node) of the algorithm. Specifications of the computer that was used are detailed in the Appendix. Using this computer, the algorithm processed 3 nodes per second.

MATLAB's high level language and built-in toolboxes enabled rapid prototyping and testing. However, it is generally slow at run-time in comparison to a possible implementation in C++ or another lower level language. Further parallelisation and use of a GPU for imaging operations would also improve speed. The execution times taken by various parts of the tracking code were measured using the MATLAB Profiler, the details of which are presented in Table 10.8.

Table 10.8: The distribution of time among the main components of the code is shown for the automatic tracking of the PCT and PST of a short-looped mouse nephron (no manual interventions).

Piece of Code (MATLAB function name)	**Time (%)**	**Time (sec)**
Reading in 1-3 images (imread)	21.52	95.31
Image alignment (conv2)	20.62	91.29
Machine learning validation (nnet)	24.13	106.84
Reading shape profiles (iomatfile)	19.30	85.47
Rest of tracking code	14.43	63.92
Total	**100**	**442.83**

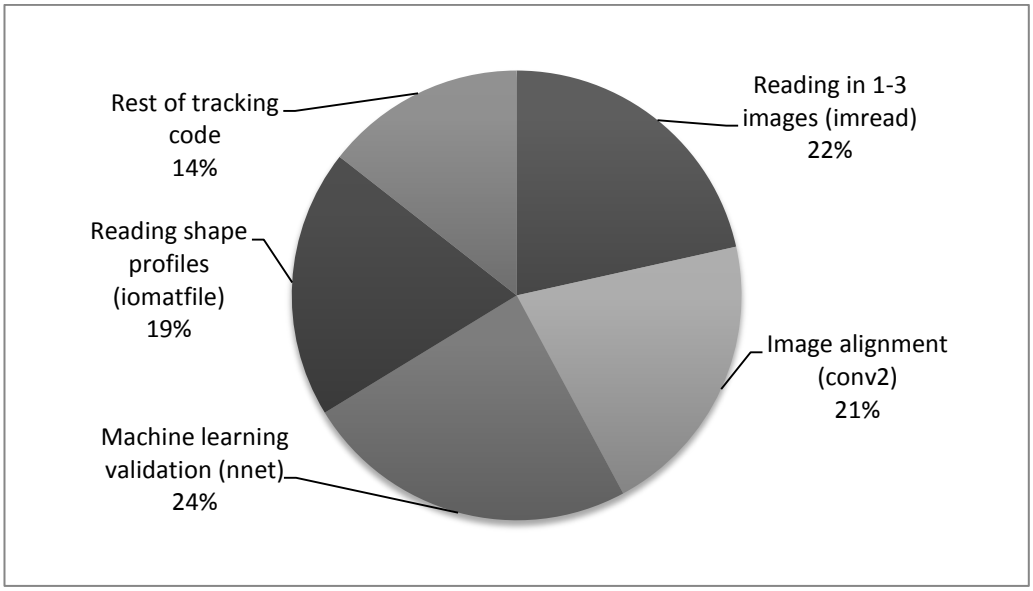


Figure 10.12: A pie chart of the distribution of time among the main components of the code is shown.

The times taken by the high level stages per mouse and rat image were also measured and are presented in Table 10.9. The timings are proportional to the number of cross-sections in the image, hence the longer processing times for inner medullary images (see Figure 6.8). While most operations in the pre-processing stage are image-wide (and therefore less dependent on the number of cross-sections), the feature extraction stage is highly dependent on the number of cross-sections as each cross-section is individually processed.

Table 10.9: The times taken to process cortical and medullary images of the mouse and rat datasets by the three stages of the nephron tracking system.

Process		**Average Time (sec/image)**	
		Mouse	Rat
Pre-processing	Cortex	2.31	3.64
	Medulla	3.94	6.21
Feature Extraction	Cortex	7.68	14.20
(using 8 parallel cores)	Medulla	13.08	55.28
Tracking one short-looped nephron		15 min/nephron	30 min/nephron

CHAPTER 11

Analysis & Discussion

The validation steps generally increase α (accuracy, or similarity to manually tracked nephron) while manual intervention increases β (the extent to which a nephron is tracked). Just as the validation steps eliminate invalid moves, they also block valid moves in the presence of artefacts, image distortions and misalignments, which cause normal morphology to appear abnormal. This is further described in Table 11.1.

Table 11.1: A summary of the implications and effects of different types of artefacts on the tracking process.

Artefact	Implication	Effect on Tracking
Bright centre	Stronger histogram equalisation needed, which over-segments larger nephron cross-sections, producing numerous independent binary components for a single nephron cross-section.	Bidirectional validation triggers which results in premature termination = more manual interventions
Tissue folds, Stretching, compression	Missing tissue and large misalignment which cannot be corrected is produced.	Jumping onto a cross-section of another structure as the fold brings other tissue directly underneath.
External matter	Obscures the nephron cross-sections either completely (missing cross-sections) or partially (change in shape of cross-sections)	Premature termination if obscuration is complete or shape validation is triggered and blocks movement for partial obscuration.
Bright spots	Under-segmentation (merging) of a small group of cross-sections	Other structures are linked to the nephron being tracked through the merged cross-sections

11.1 Performance per Area of the Nephron

Each portion of the nephron is discussed with reference to the results in Tables 10.1 and 10.2. A statement applies to both the mouse and rat datasets if it is not explicitly stated.

The majority of the cortical labyrinth is composed of the PCT and PST, which form 43% of a nephron's length (from measured β values). The algorithm is able to track the full length of the PCT and PST in the mouse and rat with averages of 2 and 10 manual corrections, respectively. The manual corrections are only required when large distortions and artefacts are encountered. Although the PCT was predicted to be the most challenging part to track due to its convoluted nature, it is tracked with high accuracy (95.14% in the mouse and 96.34 in the rat) as:

- The cross-sections are well isolated as they are large in diameter (15-30 pixels wide) and well-defined (they have thick walls).
- The average distance between neighbouring cross-sections (≈ 25 pixels) is larger than the average image misalignment of 4 pixels.

A class 2 move is successfully detected by the ML algorithms when the PCT of a nephron joins the glomerulus at its urinary pole, thus terminating the tracking. Without this, fragments in the glomerulus would be tracked towards the vascular pole, and tracking would continue through the adjoining blood vessel (afferent/efferent arterioles), which then joins blood vessel systems and other glomeruli, which is undesirable.

The PST of the mouse is also tracked well with 98.24% accuracy as the cross-sections are well isolated and defined, and the paths have a relatively straight course. In comparison, tracking of the rat PST produced a lower accuracy of 90.17% due to a higher frequency of tissue folds leading to incorrect linking with other nephrons.

As the PST narrows into the DTL, class 5 moves are successfully registered by the machine learning algorithm. The level of the class 5 output is used as a *region signal* to change the mode of tracking into a unidirectional one for the inner medulla. This reduces error in the inner medulla tremendously as ambiguity

decreases when only one unidirectional path is allowed to be formed. Without this, incorrect links are easily made to DTL cross-sections of other nephrons, especially where cross-sections incorrectly merge when bright spots are present in the image.

The DTL in the mouse and rat are tracked with moderate accuracies of 80.57% and 84.63%, respectively, as the cross-sections are very small in diameter (3-8 pixels) and very dense (≈ 6 pixels between neighbouring cross-sections). This results in a higher error probability during tracking as these values are comparable to the average misalignment of 4 pixels. Confusion is more likely among identical, closely packed nephrons which are not ideally aligned.

The DTL of the rat requires many manual corrections (≈ 24) to produce high tracking extent. Frequent premature termination occurs because over- and under-segmentation in the binary image cause the cross-sections to appear abnormal to the ANN, thus blocking many moves. Similarly, the cross-sections in the mouse are less well defined than in the cortex, making it more difficult to isolate them.

The ATL faces the same challenges as the DTL. However, these cross-sections are slightly larger (6-12 pixels) and have thicker walls, and are thus tracked more accurately in comparison to the DTL. It requires about half the number of manual corrections in both the mouse and rat datasets.

The TAL is tracked well (with 96.32% and 97.48% accuracies in the mouse and rat, respectively) as its cross-sections are well isolated and relatively large (8-12 in the mouse and 13-20 in the rat), and the path is straight.

The DCT differs vastly in the mouse and rat datasets. In the mouse, the DCT remains narrow as it progresses from the TAL. The small cross-sections making up a convoluted path are difficult to track. Fast changes in morphology (due to only having every second slice) combined with small-sized cross-sections trigger the distance validation rule. An average of 5 corrections is required in the mouse DCT.

The rat DCT is tracked well as its characteristics are comparable to the rat PCT. The cross-sections are much larger than in the mouse. Although the DCT is longer in the rat, it also requires an average of 5 corrections. Branching is correctly

handled when the DCT of multiple nephrons join through a common collecting duct.

Manual intervention is useful when the path terminates prematurely (usually due to image artefacts), as the user simply bypasses the problematic cross-section. In cases where incorrect links are made between different nephrons, manual intervention is not useful. The latter case is difficult to identify and correct without comparison to the manually tracked data or manual inspection.

In principle the automatically generated path could be more correct than the manually tracked path (due to the potential for human error especially in the inner medulla), but it was assumed that the manually tracked path is absolutely correct.

11.2 Effect of Image Properties on Performance

In general, the results are highly dependent on the quality of the images, the size of the nephron cross-sections and the amount of interfering connective tissue cross-sections. A larger slice thickness (e.g. every second slice in the mouse (5 μm) compared to 2.5 μm in the rat) produces less accurate results as the change in morphology is more abrupt from image to image. As shown in Figure 11.1, a certain slice thickness may be sufficient in the cortex where the nephron cross-sectional diameter is large, but it may cause too much of ambiguity for smaller cross sections in the inner medulla.

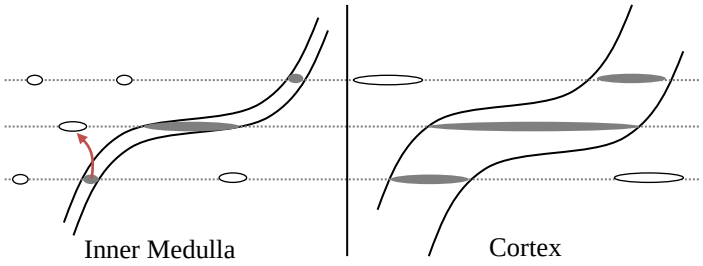


Figure 11.1: A chosen slice thickness has different implications for tracking in the cortex and inner medulla due to the different size of the structures. In the inner medulla, a larger change in morphology per image is perceived. This, along with misalignment and distortion, introduces tracking error.

This increased rate of change in morphology from the cortex to medulla can be measured by a simple ratio of the number of non-overlapping white pixels to the total number of white pixels in each pair of adjacent binary images. The measurements, as shown in Figure 11.2, indicate an increase of 20-25% in the change rate from the cortex to the inner medulla. This simple measure does, however, include the effects of misalignment and can therefore be seen as a *perceived* change in morphology. This is a large contributing factor to the high tracking error in the inner medulla.

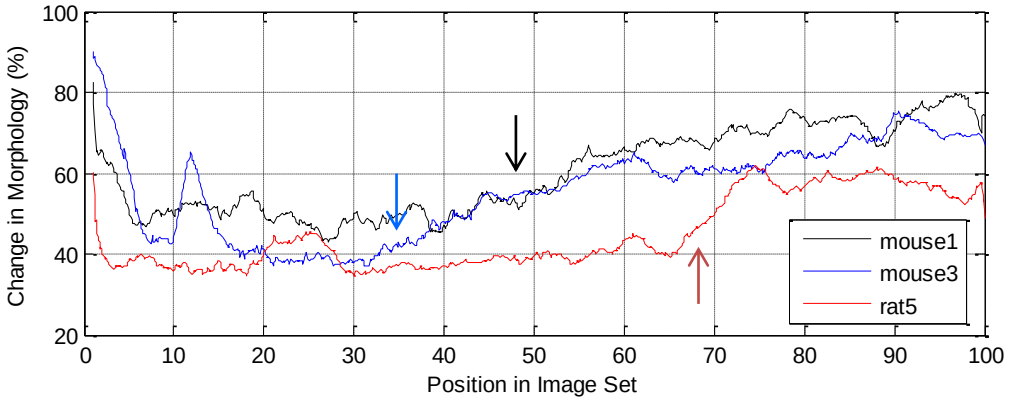


Figure 11.2: The changes in morphology for three image sets were measured. In each case, there is a noticeable increase in the morphology change rate during the transition from the cortex to inner medulla (indicated by arrows).

A high frequency of images containing artefacts and tissue folds decreases the accuracy of the findings tremendously, as it only requires a single incorrect move to cause the path to deviate from the nephron at hand onto another structure (i.e. the tracking process is chaotic or stability is completely dependent on results of the current iteration). This is especially applicable for tracking in the inner medulla, where the high tubule density coupled with an artefact may result in two nephron cross-sections joining incorrectly and the turn being mistaken for a loop of Henle. In conclusion, the amount of local image distortions, spatial resolution and slice thickness of images in the inner medulla are the main determining factors of the accuracy and extent of automated tracking in the inner medulla.

11.1 Ceiling Analysis

A ceiling analysis estimates the error due to each component of a pipeline system. It can provide a good idea on which modules of the system are worth expending additional time and effort on [**28**]. A singe real number evaluation metric representing the overall system performance is measured by progressively simulating the ground truth for the previous stages to artificially produce 100% accuracy. Table 11.2 contains the ceiling analysis for the nephron tracking system based on measured performances and projections.

Table 11.2: A high-level ceiling analysis of the system. These values have been determined through careful observation and assessment of the results of the three stages as well as the images themselves.

Component	**Overall accuracy with error carried over**	**Accuracy of stage given ideal previous stages**	**Overall accuracy given ideal previous stages (%)**
Image Quality	75	75	75
Image Pre-processing	73	95	90
Feature Extraction	70	99	94
Tracking Algorithm	63	90	95
Overall	**63**	**63**	**100**

The analysis shows that the quality and resolution of the images themselves are the main limiting upper bound on the accuracy and extent of the overall system. If high quality images are to be used, the systems accuracy goes up to 90%. Thereafter, improving the tracking algorithm would improve accuracy by about 5%.

CHAPTER 12

Recommendations & Future Work

12.1 Recommendations for Future Image Sets

Creating image sets that have minimal distortions and artefacts is essential if complete automation is to be achieved. Attaining an ideal image set of thousands of images may be impractical; however, the images can be artificially manipulated by manually removing and replacing defective images, or specially pre-processing a few images that have outlying characteristics.

The inner medulla poses the biggest challenge. Even by eye, tracking the small, thin-walled, lightly stained nephron cross-sections of the DTL in the midst of hundreds of identical cross-sections proves to be confusing and challenging. A thinner slice thickness (increased resolution in the z-direction) would improve tracking in the inner medulla as the change in position at turns and bends could be better resolved.

Higher resolution images would also offer improved accuracy in isolation and tracking of cross-sections in the inner medulla. An example of a higher resolution image is shown in Figure 12.1, where there are additional features that would be useful that were not visible on the lower resolution images, such as the tubule walls and brush borders. This of course would require more time and effort in capturing the images, as well as massive processing and memory resources if each image is to be at such a high resolution. Another useful addition would be using physical markers on the slides to aid automatic image alignment.

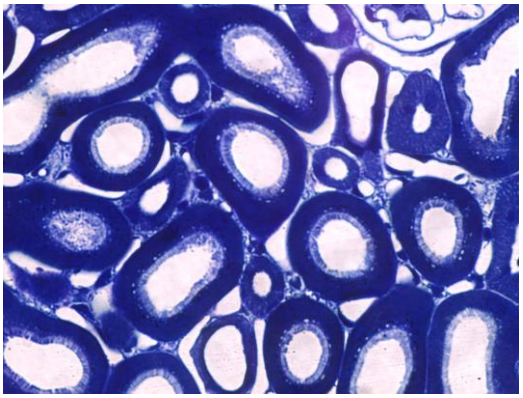


Figure 12.1: A clip of a slide from one of the mouse datasets taken at a much higher resolution is shown. Compared to the image sets used, the nephron cross-sections as well as the connective tissue could be better isolated on higher resolution images. Image from [11].

12.1.1 Staining Choice

The toluidine blue stain seems to have been ideal for the purpose of tracking as it leaves the lumens open (white in appearance), making it easy to isolate one nephron cross-section from another both by eye and automatically in software. The suitability of this stain for the tracking purpose may be emphasised by comparison with differently stained kidney specimens (such as H&E) where the lumens appear cloudy and nuclei are darkly stained [**1**] (which would then have to be compensated for during pre-processing).

One disadvantage of the toluidine blue stain is that different parts of the nephron cannot be easily distinguished. Techniques employed in the studies by Pannabecker and Dantzler [**4**] [**5**] on manually reconstructing the rat nephron may be advantageous in this regard. Immunohistochemistry techniques (using antibodies which bind to segment specific proteins) were used to stain various parts of the nephron differentially. This resulted in the DTL, ATL, collecting duct and blood vessels fluorescing with different colours. Using such staining methods would provide differentiating colour information and features to the tracking and machine learning algorithms, respectively. The confidence of results would increase as different types of cross-sections could be easily distinguished from one another and ICT interference would be virtually eliminated as only cross-

sections of interest would be highlighted. Fluorescent dyes which still leave the lumens open could therefore be the best staining choice for the tracking purpose.

12.1.2 Image Constraints

If the designed system is to be used on a new image set, the images must conform to the following constraints:

- All images must have the same resolution and a uniform scaling factor.
- The images must be a serial stack labelled sequentially.
- The resolution must be high enough such that nephron cross-sections can be easily distinguished, e.g. a DTL cross-section must be at least 5 pixels in diameter.
- At least every 5μm of the specimen (or preferably a slice thickness less than the diameter of the DTL) must be included in the dataset to adequately represent the change in morphology of the nephrons.
- Transverse sections through the kidney must be used (i.e. such that nephron cross-sections appear mostly circular). Sections producing longitudinally sliced cross-sections will not be accurately tracked, as the appearance of the cross-sections is then completely as shown in Figure 12.2. Tracking longitudinal sections even by eye is difficult and error prone. Also, this type of data is not available, and so training, testing and verifying an algorithm on longitudinal sections cannot be done. Due to inadequate training and tuning, the algorithm would not handle such cases with high accuracy.

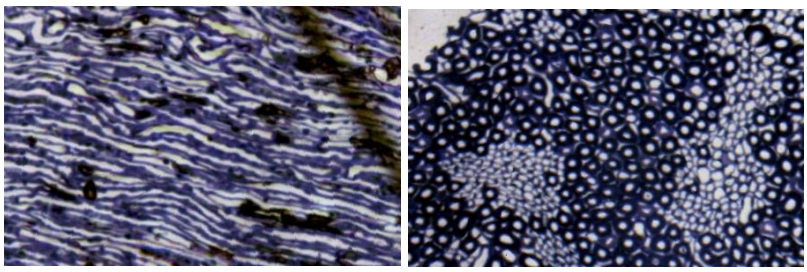


Figure 12.2: Examples of a longitudinal (left) and transverse (right) slice of the kidney. Images from **[**11**]**.

12.2 Future Work

The current approach looks at properties of the current cross-section and potential cross-sections in images above and below; it does not analyse patterns or properties of the local neighbourhood around it apart from using a local window for alignment. Looking at the surrounding area may be the key to solving problems especially in the inner medulla. Also, a machine learning algorithm that operates along a length of the detected path rather than only on a cross-section to cross-section basis may lead to more accurate results, especially in the DTL where the small cross-section diameter causes ambiguity.

The colour or intensity information from the original or equalised image may be used to compute additional features for machine learning algorithms. The colour images in combination with a full six-parameter homography could be used for a more accurate image registration, which may improve the tracking results.

Additional properties of the path around the current node, such as a 3D direction vector, can also be modelled and used for tracking conditions and validations. Even though the current algorithm can track nephrons orientated in various directions, the system still only has three degrees of freedom, for example a nephron segment that is angled 45° to the x-y plane will be tracked as a combination of horizontal and vertical edges rather than directly at 45° in the 3D image space. Future approaches for tracking could perhaps use more degrees of freedom.

This study focused on developing the methods required for (semi-) automated tracking. In order for the system to be used practically on a large number of nephrons, a more efficient version should be implemented in a language such as C++ using neural network and image processing libraries, many of which are open source. A user interface for system calibration, tracking (including manual intervention) and viewing of results should be included.

CHAPTER 13

Conclusion

The aim of the present study was to develop an automated system for the tracking of nephrons. A proposed methodology involving image processing and a custom tracking algorithm supervised by machine learning algorithms was presented. A number of features were extracted in order to retain shape information during the data abstraction process. The ANN and SVM have high classification accuracies of ≈ 93% and eliminate invalid moves caused by a number of hindering factors such as artefacts and distortions.

The system is successfully able to track large portions of the nephrons automatically through both highly convoluted and straight paths. Particularly, the PCT, PST and TAL (which form more than half of the nephron length) are tracked with high extents and accuracies in both the mouse and rat datasets. The DTL and ATL prove to be problematic due to image artefacts in combination with the small nephron cross-section size, thin walls and high tubule density in the inner medulla. These are tracked with good accuracy but require many manual corrections to achieve high extent. The DCT is tracked well in the rat but not in the mouse.

While only portions of the paths can be obtained automatically from the starting seed, full nephron paths can be obtained with an average of 17 and 62 manual corrections in the mouse and rat datasets, respectively. This is reasonable considering the thousands of coordinates making up a nephron path, each of which had to be previously manually tracked. Although complete automation is still elusive, the system saves a considerable amount of time and effort compared

with the manual tracking task as it performs 99% of the task automatically. Minimising image defects is crucial in improving performance and decreasing the amount of manual intervention required.

The developed system thus serves as a semi-automatic tool to aid the tracking process, decreasing the number of user interactions from 1100 to 17 per mouse nephron and 5000 to 62 per rat nephron. The methods developed during this study form a foundation for further development towards a fully automated nephron tracking system.

REFERENCES

[1] L. C. Junqueira and J. Carneiro, "Basic Histology - Text & Atlas," in *The Urinary System*.: McGraw-Hill, 2005.

[2] William A. Beresford. Histology Full-Text, Chapter 23 Urinary System, Anatomy Department, West Virginia University, Morgantown, USA. [Online]. http://wberesford.hsc.wvu.edu/histolch23.htm

[3] Pannabecker TL, Dantzler WH, Layton HE Layton AT, "Functional implications of the three dimensional architecture of the rat renal inner medulla," 2010.

[4] Thomas L. Pannabecker and William H. Dantzler, "Three-dimensional architecture of collecting ducts, loops of Henle, and blood vessels in the renal papilla," University of Arizona Health Sciences Center, Department of Physiology, Tucson, Arizona, American Physiological Society, 2007.

[5] Thomas L. Pannabecker, Diane E. Abbott, and William H. Dantzler, "Three-dimensional functional reconstruction of inner medullary thin limbs of Henle's loop," Department of Physiology, College of Medicine, University of Arizona, Tucson, Arizona, American Physiological Society, 85724-5051, 2003.

[6] W. Kriz, "The architectonic and functional structure of the rat kidney," Z Zellforsch Mikrosk Anat, 1967.

[7] Pannabecker Thomas L., "Comparative physiology and architecture associated with the mammalian urine concentrating mechanism: role of inner medullary water and urea transport pathways in the rodent medulla," Am J Physiol Regul Integr Comp Physiol, 2013.

[8] H Ren et al., "Direct Physical Contact between Intercalated Cells in the Distal Convoluted Tubule and the Afferent Arteriole in Mouse Kidneys," PLoS One, 2013.

[9] Erik I. Christensen et al., "Three-dimensional reconstruction of the rat nephron," Am J Physiol Renal Physiol, Department of Biomedicine, Anatomy, Aarhus University, Denmark, 2013.

[10] Xiao-Yue Zhai et al., "Three-Dimensional Reconstruction of the Mouse Nephron," Departments of Cell Biology, Connective Tissue Biology, and Neurobiology, Institute of Anatomy, University of Aarhus, Denmark, American Society of Nephrology, ISSN: 1046-6673/1701-0077, 2006.

[11] Image Sets of 3 Mouse and 3 Rat Kidneys, 2013, Departments of Cell Biology, Connective Tissue Biology, and Neurobiology, Institute of Anatomy, University of Aarhus, Denmark.

[12] Keith L. Moore, Arthur F. Dalley, and Anne M. R. Agur, *Clinically Orientated Anatomy, 6th ed.*: Lippincott Williams & Wilkins, ISBN 978-1-60547-652-0, 2010.

[13] RN Douglas-Denton, B Diouf, MD Hughson, WE Hoy, and JF Bertram, "Human nephron number: implications for health and disease," http://www.ncbi.nlm.nih.gov/pubmed/21604189, 2011.

[14] Arthur C. Guyton and John E. Hall, *Textbook of Medical Physiology, 11th ed.* Philadelphia, Pennsylvania: Elsevier, ISBN 0-7216-0240-1, 2006.

[15] SB. Nicholas, JM. Basgen, and S. Sinha, "Using stereologic techniques for podocyte counting in the mouse: shifting the paradigm.," *Am J Nephrol*, California, USA, 2011.

[16] Gokul Sridharan and Akhil A Shankar, "Toluidine blue: A review of its chemistry and clinical utility," *J Oral Maxillofac Pathol*, Department of Oral Pathology and Microbiology, YMT Dental College and Hospital, Maharashtra, India 2012.

[17] Jun Zhang and Jiulun Fan, "Medical Image Segmentation Based on Wavelet Transformation and Watershed Algorithm," Department of Information and Control, Xi 'an Institute of Post and Telecommunications, , Weihai, Shandong, China, IEEE, 2006.

[18] Cathy Merritt, Tony Kasvand Hiromitsu Yamada, "Recognition of Kidney Glomerulus by Dynamic Programming Matching Method," 1988.

[19] Hong Zhu, XueMing Qian, Tao Huang Jun Zhang, "Genetic Algorithm for Edge Extraction of Glomerulus Area," Department of System Integration, Institute of Information and Automation Engineering, Xi'an, Shannxi Province, China, 2004.

[20] Jun Zhang, Jinglu Hu Jiaxin Ma, "Glomerulus Extraction by Using Genetic Algorithm for Edge Patching," School of Information, Production and Systems, WASEDA University, Kitakyshu, Fukuoka, Japan, 2009.

[21] Paola Campadelli, Elena Casiraghi, and Stella Pratissoli, "Automatic segmentation of abdominal organs from CT scans," Universita' degli Studi di Milano, Department of Computer Science, Milano, IEEE, 2007.

[22] Hae-Yeoun Lee, Noel C. F. Codella, Matthew D. Cham, Jonathan W. Weinsaft, and Yi Wang, "Automatic Left Ventricle Segmentation Using Iterative Thresholding and an Active Contour Model With Adaptation on Short-Axis Cardiac MRI," *IEEE*, vol. 57, no. 4, 2010.

[23] Bahadir Karasulu, "Automatic Extraction of Retinal Blood Vessels: A Software Implementation," vol. 8, no. 30.

[24] Tamir Yedidya and Richard Hartley, "Tracking of Blood Vessels in Retinal Images Using Kalman Filter," The Australian National University and National ICT Australia, Australia,.

[25] Ali Can, Hong Shen, James N. Turner, Howard L. Tanenbaum, and Badrinath Roysam, "Rapid Automated Tracing and Feature Extraction from Retinal Fundus Images Using Direct Exploratory Algorithms," *IEEE Transactions on Information*

Technology in Biomedicine, New York, vol. 3, no. 2, 1999.

[26] Xin Kang et al., "Automatic Labelling of Liver Veins in CT by Probabilistic Backward Tracing," Children's National Medical Center, DC, USA, IEEE, 978-1-4673-1961-4, 2014.

[27] Michael A. Nielsen, "Neural Networks and Deep Learning," in *CHAPTER 2: How the backpropagation algorithm works*.: Determination Press, 2014. [Online]. http://neuralnetworksanddeeplearning.com/chap2.html

[28] Andrew NG. (2014, April) Coursera Online Courses: Machine Learning Course. [Online]. https://class.coursera.org/ml-005

[29] Ioannis Valavanisa, Stavroula G. Mougiakakoua, Spyretta Golematia, Alexandra Nikita, Konstantina S. Nikita John Stoitsisa, "Computer aided diagnosis based on medical image processing and artificial intelligence methods," School of Electrical and Computer Engineering & Medical School, National Technical University of Athens, Athens, Greece, 2006. [Online]. http://www.sciencedirect.com/science/article/pii/S0168900206015415

[30] Kenji Suzuki, "Pixel-Based Machine Learning in Medical Imaging," *International Journal of Biomedical Imaging*, vol. Department of Radiology, The University of Chicago, 5841 South Maryland Avenue, MC 2026, Chicago, IL 60637, USA, November 2011.

[31] Mashor M.Y., Esugasini S., Mat Isa N.A., and Othman N.H., "Classification of Breast Lesions Using Artificial Neural Network," *Proceedings of International Conference on Man-Machine Systems* , 2006.

[32] K Gayathri Devi and R Radhakrishnan, "Automatic Segmentation of Colon in 3D CT images and removal of opacified fluid using cascade feed forward neural network," *Institute of Technology, India*.

[33] MATLAB Version R2012a, MathWorks, Image Processing Toolbox; Neural Network Toolbox; Statistics Toolbox.

[34] Prof. Henrik Birn, "The Danish Ministry of Food, Agriculture and Fisheries; Ministeriet for Fødevarer," Institute of Anatomy/Biomedicine 2004/561-818.

[35] K. Wagholikar, "Modeling Paradigms for Medical Diagnostic Decision Support: A Survey and Future Directions," *Journal of Medical Systems*, Aug. 2012.

[36] E. R. Davies, *Computer & Machine Vision: Theory, Algorithms, Practicalities*. Egham, UK, Surrey: Elsevier, 2012.

[37] Dana H. Ballard and Christopher M. Brown, *Computer Vision*. Rochester, New York: Prentice Hall, 1982.

[38] Peter Henderson, Richard Seaby, and Robin Somes, "Growth II," in *Types of growth curve - Logistic curve*. Penington, Lymington, Hampshire: Pisces Conservation Ltd,

2006.

[39] Guojun Gan, Chaoqun Ma, and Wu Jianhong, "Data Clustering Theory, Algorithms and Applications," in *Chapter 9: Center-based Clustering Algorithms*. Philadelphia, ASA, Alexandria, VA: ASA-SIAM Series on Statistics and Applied Probability, 2007.

[40] L. Wojnar and K.J. Kurzydłowski, *Practical Guide to Image Analysis*.: ASM International, 2000.

[41] Barbara Zitova and Jan Flusser, "Image registration methods: a survey," *Elsevier: Image and Vision Computing*, Department of Image Processing, Institute of Information Theory and Automation, Academy of Sciences of the Czech Republic 2003.

[42] Amit Patel. (2014) Stanford Theory Group. [Online]. http://theory.stanford.edu/~amitp/GameProgramming/AStarComparison.html

[43] Kenji Kira and Larry Rendell, "The Feature Selection Problem: Traditional Methods and a New Algorithm," 1992.