

https://dspace.org/

Optical flow algorithms and background oriented schlieren for a supersonic wind tunnel setup

King, Timothy

2018

https://hdl.handle.net/10539/25714

King, Timothy James Adrian (2018) Optical flow algorithms and background oriented schlieren for a supersonic wind tunnel setup,University of the Witwatersrand, Johannesburg, https://hdl.handle.net/10539/25714

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

UNIVERSITY OF THE WITWATERSRAND, JOHANNESBURG
SCHOOL OF MECHANICAL, INDUSTRIAL AND AERONAUTICAL ENGINEERING



DISSERTATION

OPTICAL FLOW ALGORITHMS AND BACKGROUND ORIENTED SCHLIEREN FOR A SUPERSONIC WIND TUNNEL SETUP

Name: Mr. Timothy King

Student no.: 0108511R

Date: May 2018

Supervisor: Prof. Beric Skews

DECLARATION

I, the undersigned, affirm the following:

- I have carried out the work on this project myself without direct involvement by any other person
- Work performed by another person that has been used or referenced has been acknowledged appropriately within the text

_____

Timothy King

0108511R

ACKNOWLEDGEMENTS

I took a long time to get this work completed. Even so, here it is and although I will always wish for slightly more perfection, I am satisfied with what I have achieved so far. This work is for those who have enabled this, taken an interest, provided funding to related work, given me time at work to write up the Dissertation, put pressure on me to get work done and given me a break when I needed it.

Special thanks must go to the following people:

- **Mr. Steve Haselum**: For his interest and support
- **Mr. David King**: For his interest and support
- **Mrs. Stephanie King**: For her interest and support
- **Miss. Pamela King**: For her interest and support
- **Mrs. Sherian Lewis**: For her interest and support
- **Dr. David Maclucas**: For providing the traditional schlieren images
- **Mr. Andre Marais**: For his help with all things software related
- **Ms. Bronwyn Meyers**: For providing funding
- **Mr. Mauro Morelli**: For his support and funding of related work
- **Dr. Kaven Naidoo**: For providing funding and the opportunity to do this work
- **Miss. Lara Nel**: For helping with CFD on a related background task
- **Prof. Beric Skews**: For his patience, understanding and help with navigating the bureaucracy
- **Mr. Peter Skinner**: For taking a detailed interest and providing helpful discussion
- **Dr. Glen Snedden**: For funding related work

ABSTRACT

The work details a comparison between various GPU accelerated optical flow algorithms that the OpenCV open source computer vision library provides as implemented in a Background Oriented Schlieren (BOS) setup for a supersonic wind tunnel. The chosen test case is the visualisation of supersonic flow (Mach 2.5) over a conical wind tunnel model. The theory for predicting the distortion of the images is developed and used for comparison, as well as a correlation based brute-force block matching algorithm, which is typical of current BOS setups. Additionally, two background images were designed and tested with all algorithms to investigate the interaction between these backgrounds and the various algorithms. The theory behind the optical flow calculations is outlined and discussed in the BOS context and in particular, how it can be used for future background designs. It is also shown that there exists a general compromise between optical flow detail and noise rejection in the processed BOS images and that the more modern algorithms that have been implemented in the OpenCV toolbox generally outperform the traditional correlation based block-matching methods when their accuracy is compared.

TABLE OF CONTENTS

Declaration ..... **i**

Acknowledgements ..... **ii**

Abstract ..... **iii**

Table of contents ..... **iv**

List of figures ..... **vi**

List of tables ..... **ix**

Abbreviations ..... **x**

List of symbols ..... **xi**

1 Background and introduction **1**
1.1 BOS system 3
1.1.1 BOS system hardware 3
1.1.2 BOS system software 4
1.2 Objectives 4

2 Theory **5**
2.1 Theoretical distortion of background images 5
2.1.1 Supersonic flow around a cone 8
2.1.2 Light-ray deflections 13
2.1.3 Background distortions 28
2.1.4 Optical flow predictions 30
2.2 Image processing - optical flow 35
2.2.1 Common optical flow concepts 40
2.2.2 Optical flow algorithms 41
2.3 Background design 57
2.3.1 Speckled backgrounds 57
2.3.2 Wavelet noise backgrounds 59
2.3.3 Synthesis of b-spline wavelet noise 60

3 Method **69**
3.1 Overview 69
3.2 Test setup 70
3.2.1 Background design 70
3.2.2 Equipment 70
3.2.3 Setup 72
3.3 Test procedure 75
3.4 Image processing 75
3.5 Theoretical image generation 76
3.5.1 Relationship between distortion angle and image distortion 76

4 Results **79**
4.1 Flow properties 79

4.2 Raw images 79
4.3 Theoretical distortions 83
4.4 Colour schlieren 87
4.5 Experimental images 88
4.5.1 Processed results 90

5 Discussion 94
5.1 Discussion on optical flow algorithms 94
5.1.1 Block matching 94
5.1.2 Lucas and Kanade 101
5.1.3 Brox 108
5.1.4 Farnebäck 118
5.1.5 TV-L1 127
5.2 General differences between backgrounds 134
5.3 The compromise between noise rejection and detail 134
5.4 The consequences of the aperture problem 135
5.5 Camera spatial resolution 135

6 Recommendations for future work 136
6.1 Camera greyscale resolution and noise 136
6.2 Speckled backgrounds with gradient based algorithms 136
6.3 Application of optical flow knowledge to background design 139
6.3.1 Checkerboard pattern 139
6.4 Colour and black and white image comparison 141

7 Conclusion 142

8 References 143

Appendix A Block-matching algorithm development 145
A.1 Introduction 145
A.2 Implementation 147
A.2.1 Parallel block matching approach 147
A.2.2 Sequential correlations with parallel algorithm components 149
A.3 Conclusion 149

Appendix B Horn and Schunk's algorithm 150

Appendix C Index to processed images 153
C.1 Down sampling 153
C.2 Block matching results 154
C.3 Lucas and Kanade algorithm results 159
C.4 Brox algorithm results 162
C.5 Farnebäck Results 172
C.6 TV-L1 Results 181

LIST OF FIGURES

Figure 1: Illustration of light bending in a BOS setup 1
Figure 2: A cross-section with streamlines illustrating supersonic flow over a cone 8
Figure 3: Plot of density against angle 12
Figure 4: Plot of Mach number against angle 13
Figure 5: Illustration of geometry used for calculating ray deflections 14
Figure 6: Deflection angles 14
Figure 7: Vectors used in the vector formulation of Snell's law 15
Figure 8: Example of lateral deflection of light rays at the shock 17
Figure 9: Example of vertical deflection of light rays at the shock 17
Figure 10: Graph illustrating an example of lateral deflection angle with projected ray angle 24
Figure 11: Graph illustrating an example of vertical deflection angle with projected ray angle 24
Figure 12: Example of lateral deflection of rays with projected angle 27
Figure 13: Example of vertical deflection of rays with deflection angle 27
Figure 14: Example of horizontal deflection angle in radians 29
Figure 15: Example of vertical deflection angle in radians 29
Figure 16: Example of horizontal deflection in pixels 31
Figure 17: Example of vertical deflection in pixels 31
Figure 18: Undistorted background image 33
Figure 19: Distorted background image 34
Figure 20: Reference Image, $I0$ 38
Figure 21: Distorted image, $I1$ 38
Figure 22: Optical flow definitions 39
Figure 23: Illustration of the aperture problem 41
Figure 24: An example of speckled background noise 58
Figure 25: An example of b-spline wavelet noise 59
Figure 26: Illustration of weighted spline basis functions approximating a parabola 61
Figure 27: Four third order b-spline basis functions 62
Figure 28: Illustration of basis function refinement 63
Figure 29: Illustration of the generation of wavelet noise 65
Figure 30: Initial two-dimensional b-spline image with random weights 66
Figure 31: Down-sampled b-spline image 67
Figure 32: B-spline wavelet noise image (one layer) 67
Figure 33: Power spectral density of one layer of wavelet noise 68
Figure 34: Front end of the wind tunnel model 71
Figure 35: BOS setup geometry 73
Figure 36: Averaged reference Image for the speckled background 80
Figure 37: Averaged distorted image for the speckled background 80
Figure 38: Averaged reference image for the wavelet background 81
Figure 39: Averaged distorted image for the wavelet background 81
Figure 40: Last single frame captured during the speckled image test 82
Figure 41: Last single frame captured during the wavelet background test 82
Figure 42: Predicted vertical deflection in pixels with a range of -4 to +4 pixels 83
Figure 43: Predicted horizontal deflection in pixels with a range of -4 to +4 pixels 84
Figure 44: Predicted vertical deflection in pixels with a range of -1 to +1 pixels 85
Figure 45: Predicted horizontal deflection in pixels with a range of -1 to +1 pixels 86
Figure 46: Colour schlieren image 87
Figure 47: Sample measured vertical deflection in pixels with a range of -4 to +4 pixels 89
Figure 48: Sample measured vertical deflection in pixels with a range of -1 to +1 pixels 89

Figure 49: Block matching algorithm, vertical pixel displacement, 4 by 4 window, Gaussian sub-pixel interpolation, speckled background 95
Figure 50: Block matching algorithm, vertical pixel displacement, 32 by 32 window, Gaussian sub-pixel interpolation, speckled background 95
Figure 51: Block matching algorithm, vertical pixel displacement, 16 by 16 window, Gaussian sub-pixel interpolation, speckled background 97
Figure 52: Block matching algorithm, vertical pixel displacement, 16 by 16 window, quadratic sub-pixel interpolation, speckled background 97
Figure 53: Block matching algorithm, vertical pixel displacement, 16 by 16 window, centroid sub-pixel interpolation, speckled background 98
Figure 54: Block matching algorithm, vertical pixel displacement, 16 by 16 window, no sub-pixel interpolation, speckled background 98
Figure 55: Block matching algorithm, vertical pixel displacement, 8 by 8 window, Gaussian sub-pixel interpolation, wavelet background 100
Figure 56: Block matching algorithm, vertical pixel displacement, 8 by 8 window, Gaussian sub-pixel interpolation, speckled background 100
Figure 57: Lucas and Kanade algorithm, vertical pixel displacement, 5 by 5 window, single layer, speckled background 102
Figure 58: Lucas and Kanade algorithm, vertical pixel displacement, 33 by 33 window, single layer, speckled background 102
Figure 59: Lucas and Kanade algorithm, vertical pixel displacement, 9 by 9 window, single layer, wavelet background 104
Figure 60: Lucas and Kanade algorithm, vertical pixel displacement, 9 by 9 window, 11 layers, down sample factor 0.5, wavelet background 104
Figure 61: Lucas and Kanade algorithm, vertical pixel displacement, 17 by 17 window, single layer, speckled background 106
Figure 62: Lucas and Kanade algorithm, vertical pixel displacement, 17 by 17 window, single layer, wavelet background 106
Figure 63: Lucas and Kanade algorithm, vertical pixel displacement, 5 by 5 window, single layer, speckled background 107
Figure 64: Lucas and Kanade algorithm, vertical pixel displacement, 5 by 5 window, single layer, wavelet background 107
Figure 65: Brox algorithm, vertical pixel displacement, $\alpha = 0.1$, $\gamma = 50$, single layer, speckled background 109
Figure 66: Brox algorithm, vertical pixel displacement, $\alpha = 0.4$, $\gamma = 50$, single layer, speckled background 109
Figure 67: Brox algorithm, vertical pixel displacement, $\alpha = 0.2$, $\gamma = 0$, single layer, speckled background 111
Figure 68: Brox algorithm, vertical pixel displacement, $\alpha = 0.2$, $\gamma = 100$, single layer, speckled background 111
Figure 69: Brox algorithm, vertical pixel displacement, $\alpha = 0.1$, $\gamma = 100$, single layer, wavelet background 113
Figure 70: Brox algorithm, vertical pixel displacement, $\alpha = 0.1$, $\gamma = 100$, 11 layers, down sample factor = 0.5, wavelet background 113
Figure 71: Brox algorithm, vertical pixel displacement, $\alpha = 0.1$, $\gamma = 100$, 35 layers, down sample factor = 0.8, wavelet background 114
Figure 72: Brox algorithm, vertical pixel displacement, $\alpha = 0.1$, $\gamma = 100$, single layer, speckled background 114
Figure 73: Brox algorithm, vertical pixel displacement, $\alpha = 0.1$, $\gamma = 100$, 11 layers, down sample factor = 0.5, speckled background 115

Figure 74: Brox algorithm, vertical pixel displacement, $\alpha = 0.1$, $\gamma = 100$, 35 layers, down sample factor = 0.8, speckled background 115
Figure 75: Brox algorithm, vertical pixel displacement, $\alpha = 0.2$, $\gamma = 50$, single layer, wavelet background 117
Figure 76: Brox algorithm, vertical pixel displacement, $\alpha = 0.2$, $\gamma = 50$, single layer, speckled background 117
Figure 77: Farnebäck algorithm, vertical pixel displacement, 5 by 5 greyscale window, 4 by 4 optical flow window, single layer, speckled background 119
Figure 78: Farnebäck algorithm, vertical pixel displacement, 5 by 5 greyscale window, 32 by 32 optical flow window, single layer, speckled background 119
Figure 79: Farnebäck algorithm, vertical pixel displacement, 7 by 7 greyscale window, 4 by 4 optical flow window, single layer, speckled background 120
Figure 80: Farnebäck algorithm, vertical pixel displacement, 5 by 5 greyscale window, 32 by 32 optical flow window, single layer, speckled background 120
Figure 81: Farnebäck algorithm, vertical pixel displacement, 7 by 7 greyscale window, 32 by 32 optical flow window, single layer, speckled background 121
Figure 82: Farnebäck algorithm, vertical pixel displacement, 5 by 5 greyscale window, 8 by 8 optical flow window, single layer, wavelet background 123
Figure 83: Farnebäck algorithm, vertical pixel displacement, 5 by 5 greyscale window, 8 by 8 optical flow window, 11 layers, down sample factor = 0.5, wavelet background 123
Figure 84: Farnebäck algorithm, vertical pixel displacement, 5 by 5 greyscale window, 8 by 8 optical flow window, 35 layers, down sample factor = 0.8, wavelet background 124
Figure 85: Farnebäck algorithm, vertical pixel displacement, 5 by 5 greyscale window, 16 by 16 optical flow window, single layer, wavelet background 125
Figure 86: Farnebäck algorithm, vertical pixel displacement, 5 by 5 greyscale window, 16 by 16 optical flow window, single layer, speckled background 126
Figure 87: TV-L1 algorithm, vertical pixel displacement, $\lambda = 0.08$, single layer, speckled background 128
Figure 88: TV-L1 algorithm, vertical pixel displacement, $\lambda = 0.32$, single layer, speckled background 128
Figure 89: TV-L1 algorithm, vertical pixel displacement, $\lambda = 0.16$, single layer, speckled background 130
Figure 90: TV-L1 algorithm, vertical pixel displacement, $\lambda = 0.16$, 11 layers, down sample factor = 0.5, speckled background 130
Figure 91: TV-L1 algorithm, vertical pixel displacement, $\lambda = 0.16$, 35 layers, down sample factor = 0.8, speckled background 131
Figure 92: TV-L1 algorithm, vertical pixel displacement, $\lambda = 0.16$, single layer, wavelet background 133
Figure 93: TV-L1 algorithm, vertical pixel displacement, $\lambda = 0.16$, single layer, wavelet background 133
Figure 94: Variation of fraction of edges in an image with black speckle probability 140
Figure 95: Illustration of a Bayer mask 141
Figure 96: Image illustrating cross correlation of translated window 145

LIST OF TABLES

Table 1: BOS system specifications 3
Table 2: List of symbols for the theoretical distortion of background image 6
Table 3: List of symbols for optical flow 35
Table 4: Parameters used in the block-matching algorithm 44
Table 5: Parameters used to tune the Lucas and Kanade algorithm 46
Table 6: Description of parameters used in the Brox algorithm 48
Table 7: Parameters used in the Farnebäck algorithm 51
Table 8: Parameters use to tune the TV-L1 algorithm 56
Table 9: Camera specifications 72
Table 10: Camera lens specifications 72
Table 11: Dimensions of BOS setup 73
Table 12: List of symbols for the relationship between distortion angle and image distortion 76
Table 13: Magnification calculation value 77
Table 14: Equivalent pixel size calculation values 77
Table 15: Angular deflection calculation values 78
Table 16: Wind tunnel flow parameters 79
Table 17: Number of images averaged 79
Table 18: Optical flow parameters effect on noise rejection and detail 134
Table 19: Down sampling parameters 153
Table 20: Block matching algorithm results 154
Table 21: Lucas and Kanade algorithm results 159
Table 22: Brox algorithm results 162
Table 23: Farnebäck algorithm results 172
Table 24: TV-L1 algorithm results 181

ABBREVIATIONS

BOS	Background oriented schlieren
CCD	Charge-coupled device
CMOS	Complementary metal–oxide–semiconductor
CPU	Central processing unit
CSIR	Council for Scientific and Industrial Research
FFT	Fast Fourier transform
GPU	Graphics processing unit
HSWT	High Speed Wind Tunnel
LED	Light emitting diode
MP	Megapixels
MSWT	Medium Speed Wind Tunnel
NMISA	National Metrology Institute of South Africa
OEM	Original equipment manufacturer
PIV	Particle image velocimetry
RAM	Random access memory
SSD	Solid state drive

LIST OF SYMBOLS

dB	Decibels
e^-	Electrons
fps	Frames per second
GiB	Gigabytes
ke^-	Thousand electrons
px	Pixels
TiB	Terabytes

Note: Due to the large number of symbols used in this text, symbols other than those given above are provided per chapter where necessary.

1 BACKGROUND AND INTRODUCTION

Background Oriented Schlieren (BOS), developed by Meier [1], is a technique that is used to visualise refractive index gradients within a field of interest. The Gladstone-Dale relation relates a medium's density to its refractive index. This means that images of refractive index gradient can be interpreted as images of density gradient.

BOS works by comparing two images, one undistorted image that was acquired prior to the presence of the varying refractive index field and one distorted image that was taken through the field of varying refractive index. In the context of the visualisation of high-speed flows in wind tunnels, this amounts to taking a picture with the air off (undistorted image) and then with the air on (distorted image). The distortion arises due to the light, which is being reflected from the background, bending when it travels through the field of varying refractive index. This is illustrated in Figure 1.

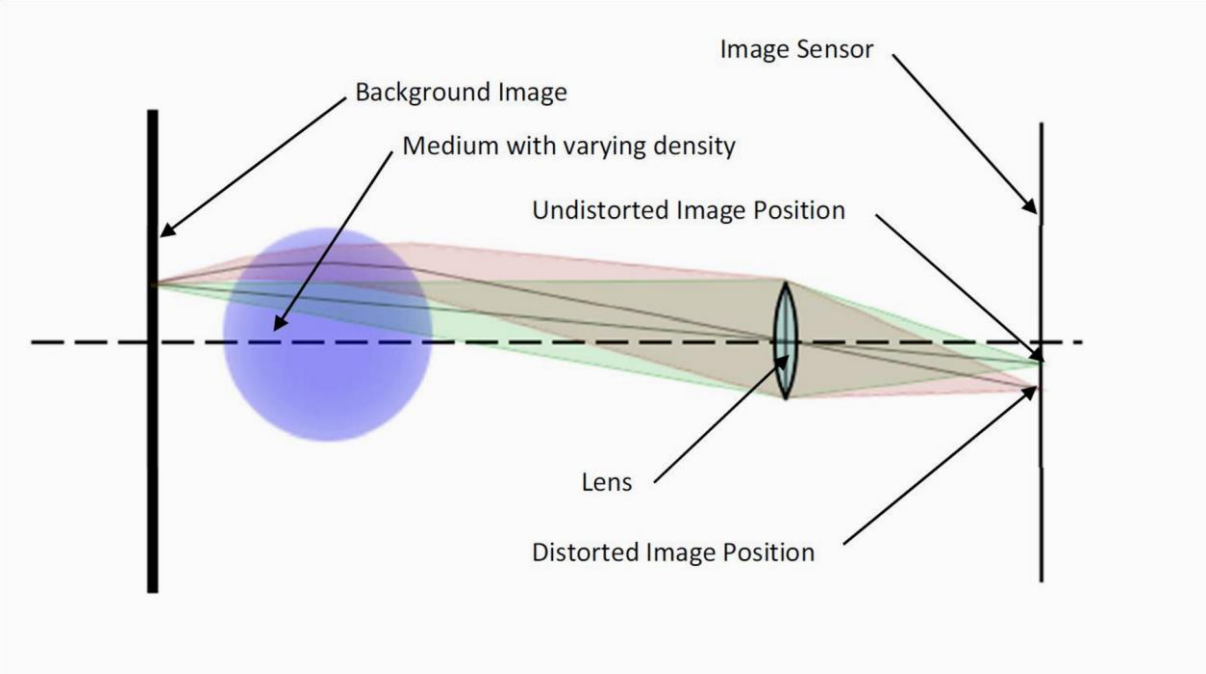


Figure 1: Illustration of light bending in a BOS setup

A BOS System has been developed by the author at the Council for Scientific and Industrial Research (CSIR) with the ultimate goal of providing a flow visualisation system for the CSIR's Medium Speed Wind Tunnel (MSWT). The system has been developed and tested in the High Speed Wind Tunnel (HSWT) using supersonic flow (with a nominal Mach number of 2.5) over a cone. The HSWT provided a more cost effective development environment than the MSWT.

The open source computer vision library, OpenCV, provides several graphics processing unit (GPU) accelerated optical flow methods for tracking movements between images. These methods differ from the traditional block-matching approach (otherwise known as a cross-correlation approach) often used in BOS setups [2]. The methods studied in this work are those that have been incorporated into the CSIR BOS system that include the GPU accelerated optical flow algorithms in the OpenCV library and one purpose-developed block-matching algorithm.

Each optical flow algorithm has parameters that are used to tune the results. This study concentrates on the primary parameters affecting the global results while numerical parameters, such as those that affect convergence of the algorithms, are set pragmatically. The values that were used are given in Appendix C.

While the standard practice is to use a random speckled background in conjunction with a block-matching processing algorithm, there has been some study regarding the use of other optical flow algorithms in BOS setups. Atcheson *et al.* [2] studied various optical flow algorithms in conjunction with a greyscale wavelet background. More recently, Hayasaka *et al.* [3], used an optical flow method for processing a background with randomly placed spots, similar to a speckled pattern, and compared these results to a block matching method. The optical flow method by Horn and Schunk [4] used in their work has been improved upon, for example, by Brox *et al.* [5] and that by Zach *et al.* [6].

Some of these methods used for calculating the displacements between the images make use of greyscale gradients in the images. Gradients are present in the wavelet images while there are only zero gradients and discontinuities present in the speckled backgrounds. For this study, one background was created using a traditional speckled pattern while the other was created using a tiled b-spline wavelet image. Both backgrounds were processed by all of the algorithms used by the BOS system for comparison.

Although the use of coloured backgrounds have been covered in the literature [7] [8], only grey-scale images were considered for this study. While the method used by Leopoldt *et al.* [7], which relied on a block-matching technique, can be used to extend this work, that used by Ota *et al.* [8] relied on a structured coloured grid background and used a finite-fringe technique for processing a single image and is therefore not applicable in this context. The focus of this work is on the parameters that are used to tune the algorithms and the difference between results from the two chosen background images. Incorporating the use of colour with the optical flow algorithms is left for future study.

1.1 BOS system

The BOS system being studied can be separated into hardware and software elements. An outline of each is given in the sections that follow.

1.1.1 BOS system hardware

The system hardware specifications are listed in Table 1.

Table 1: BOS system specifications

Item	Specification
CPU	Intel® Core ™ i7-4790K CPU @ 4.00 GHZ x 8
GPU 1	Nvidia GeForce GTX 750 Ti
GPU 2	Nvidia Tesla K20c
RAM	14.6 GiB
Hard Disk 1	1.8 TiB
Hard Disk 2	1.8 TiB
Hard Disk 3	111.8 GiB SSD
Camera	BASLER ACE acA2000-50gc
Lens	RICOH 25 mm FL-CC2514-2M
Lights	GS Vitech GmbH multiLED
Background	Printed vinyl

1.1.2 BOS system software

Development environment

The operating system for the BOS system is Ubuntu 16.04 LTS, which was chosen because it is well supported by large range of software development tools.

The software was programmed by the author in the CUDA, QT and C++ languages, borrowing from specific libraries. CUDA is an extension of the C++ language that has been developed to allow for programming of highly parallel algorithms on NVIDIA GPU's. NVIDIA NSight Eclipse Edition integrated development environment (IDE) was used for custom CUDA code development. QT is another extension to the C++ language enabling the use of signals and slots. The QT Creator IDE was used, which enabled easy graphical user interface (GUI) development.

Software libraries

The BOS system software makes extensive use of the OpenCV library. This library is implemented in the C++ and CUDA languages with wrappers for other languages. The library is a computer vision library that makes use of GPU's for the acceleration of various routines. Only the block-matching algorithm was coded from scratch, while all others form part of the OpenCV software library. A modified version of OpenCV 3.1.0 was used. The modification was required to fix a memory-addressing bug in the Lucas and Kanade algorithm. The bug fix was submitted to OpenCV and included in OpenCV 3.3.0. The interested reader is referred to the OpenCV web site [9].

The Basler Pylon library was used for interfacing with the camera. This library is provided by the camera original equipment manufacturer (OEM) for setting camera parameters and for image acquisition. Pylon4 was the version used in this software.

QCustomPlot is a plotting library that was used to display the processed results in the QT developed GUI. QCustomPlot version 2.0.0-beta was used in the software used for this analysis.

1.2 Objectives

The purpose of this study is to provide insight into the accuracy of the GPU accelerated algorithms that have been implemented in the OpenCV library used in the BOS context, how they compare to the traditional block-matching techniques, and how each of these interact with the chosen backgrounds images. To this end, a theoretical model is developed for comparison with the processed results so that the various experimental results can be evaluated in context.

2 THEORY

This section covers the theoretical predictions of the distorted backgrounds for the chosen test case, a theoretical overview of each of the algorithms used and the theory involved in generating the two chosen backgrounds for the study.

2.1 Theoretical distortion of background images

In this section, the method for predicting the distortion caused by capturing the background images through the flow field of interest is shown. First, the Taylor-Maccoll equations are solved in order to predict the density variation of the air around the cone. The Gladstone-Dale relation is used to find the optical properties of the air. The angles by which rays that are entering the camera are distorted are then predicted and, by taking cognisance of the test setup geometry, the position on the background image from which each ray originated is predicted, thereby quantifying the predicted distortion. The flow behind the expansion fans generated when the cone transitions into the cylinder by which the cone is supported is not modelled. A list of symbols used in this section is given in Table 2.

Table 2: List of symbols for the theoretical distortion of background image

Symbol	Description
f	Convenience function f
g	Convenience function g
i	Lateral position in image
i_{max}	Maximum lateral position in image
i_0	Lateral reference position in image
I_0	Undistorted image greyscale value
I_1	Distorted image greyscale value
j	Vertical position in image
j_{max}	Maximum vertical position in image
j_0	Reference vertical position in image
K	Gladstone-Dale constant
L	Size of angle array
M	Mach number
M_{n_1}	Component of free-stream Mach number normal to the shock
M_1	Free-stream Mach number
M_2	Mach number just behind the shock
M_{n2}	Component of Mach number just behind the shock that is normal to the shock
n	Refractive index
n_0	Surrounding medium refractive index
n_i	Refractive index of initial medium (for the incoming ray)
n_o	Refractive index of final medium (for the outgoing ray)
$\boldsymbol{p}$	Normal to the shock-cone surface
$\boldsymbol{r}$	Position vector relative to the cone point
r	The magnitude of $\boldsymbol{r}$
r	The distance from the centreline of the test section to the background
$\boldsymbol{r_i}$	Incoming ray vector
$\boldsymbol{r_o}$	Outgoing ray vector
s_m	Element in angle array at position m
u	Lateral component of optical flow
V	Non-dimensional flow velocity
v	Vertical component of optical flow
V_2	Non-dimensional flow velocity just behind the shock
V_r	Radial component of non-dimensional flow velocity
V_{r2}	Radial component of Non-dimensional flow velocity just behind the shock
V_θ	Tangential component of non-dimensional flow velocity
$V_{\theta 2}$	Tangential component of non-dimensional flow velocity just behind the shock
$\boldsymbol{w}$	Optical flow vector
x	Component of $\boldsymbol{r}$ along the axis of revolution of the cone
x_o	x component of $\boldsymbol{r_0}$
y	Component of $\boldsymbol{r}$ parallel to the assumed direction of incoming light ray travel
y_o	y component of $\boldsymbol{r_0}$
z	Component of $\boldsymbol{r}$ perpendicular to x and y
z_o	z component of $\boldsymbol{r_0}$
β	Shock angle

Symbol	Description
γ	Ratio of specific heats
δ	Flow-turn angle
ϵ_{x_g}	Half the lateral angular deflection of the light ray due to refractive index gradients behind the shock
ϵ_{x_s}	Half the lateral angular deflection of the light ray due to the refractive index discontinuity at the shock
ϵ_{y_g}	Half the vertical angular deflection of the light ray due to refractive index gradients behind the shock
ϵ_{y_s}	Half the vertical angular deflection of the light ray due to the refractive index discontinuity at the shock
ϵ_x	The total lateral angular deflection of the light ray
ϵ_y	The total vertical angular deflection of the light ray
ζ	The distance between the positions on the background that correspond to pixel positions on the image
θ	The angle between $\boldsymbol{r}$ and the x-axis
$\boldsymbol{\theta}$	Vector perpendicular to $\boldsymbol{r}$ and in the same plane as the x-axis
θ_0	θ at the start of integration
θ_1	θ at the end of integration
θ_{cone}	The halfi-angle of the model cone
θ_k	θ at position k
ρ	Density
ρ_1	Density in the free-stream
ρ_2	Density just behind the shock
ϕ	The angle that the projection of $\boldsymbol{r}$ on the x-y plane makes with the y axis
ψ	The angle that the projection of $\boldsymbol{r}$ on the x-y plane makes with the x axis

2.1.1 Supersonic flow around a cone

Supersonic flow around a cone first passes through an oblique conical shock, turning outward partially in the direction of the cone's surface, where after it is isentropically compressed and turned further toward the tangent of the solid cone surface [10]. The boundary layer is not modelled. Figure 2 uses streamlines to show the flow turning in the direction of the cone's surface. The shock position is shown as a dashed black line.

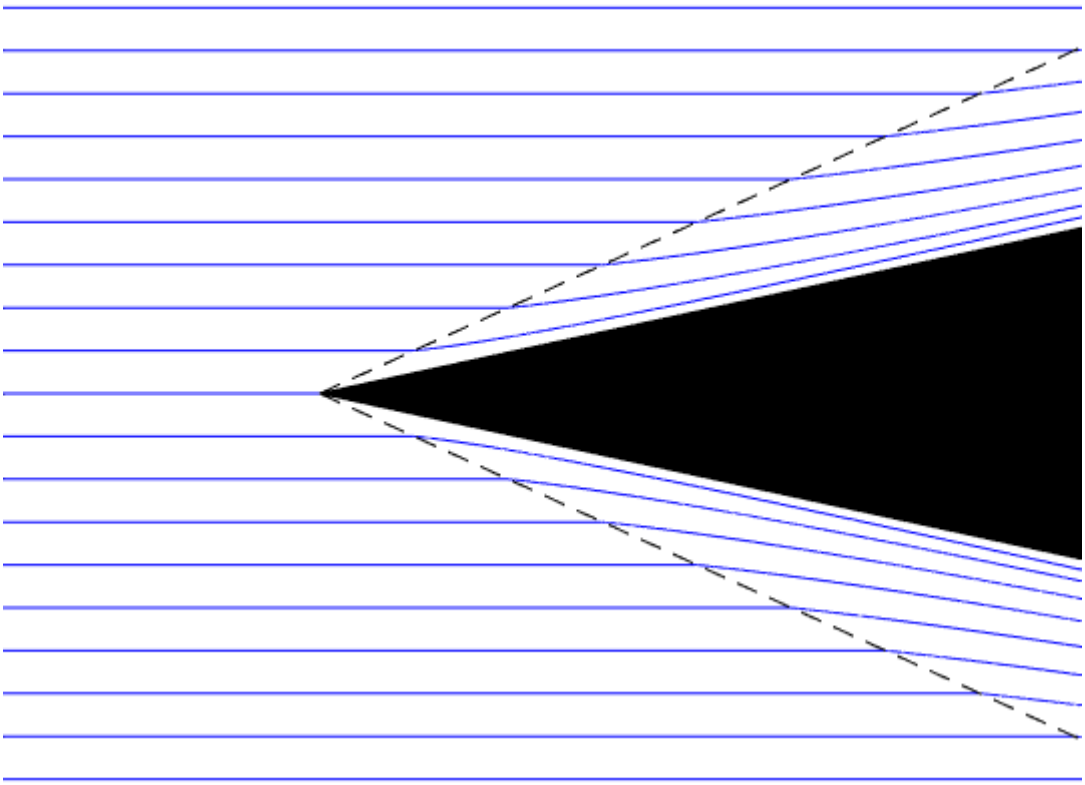


Figure 2: A cross-section with streamlines illustrating supersonic flow over a cone

In this subsection, the flow property changes across the shock are calculated, after which the Taylor-Maccoll equations are used to solve for the flow properties behind the shock.

Flow property changes across a conical shock

For a given Mach number, and desired shock angle, the Mach number behind the shock is calculated using the oblique shock relations [11] with some slight manipulations as outlined below.

First, the component of the Mach number normal to and upstream of the Shock is calculated.

$$M_{n1} = M_1 \sin(\beta) \tag{1}$$

Next, the normal component of the Mach number just downstream of the shock is calculated.

$$M_{n2} = \sqrt{\frac{M_{n1}^2 + \left(\frac{2}{\gamma - 1}\right)}{\frac{2\gamma}{\gamma - 1} M_{n1}^2 - 1}} \tag{2}$$

The flow deflection angle is also calculated.

$$\delta = \operatorname{atan}\left(2\,cot(\beta)\left(\frac{M_{n1}^2 - 1}{M_1^2\left(\gamma + cos(2\beta)\right) + 2}\right)\right) \tag{3}$$

The Mach number just downstream of the shock is then calculated.

$$M_2 = \frac{M_{n2}}{sin(\beta - \delta)} \tag{4}$$

The ratio of the densities across the shock can then be calculated [11].

$$\frac{\rho_2}{\rho_1} = \frac{(\gamma + 1)M_{n_1}^2}{(\gamma - 1)M_{n_1}^2 + 2} \tag{5}$$

Flow properties behind the shock

Anderson recommends using velocities scaled by the maximum theoretical velocity that the flow can obtain to solve the Taylor-Maccoll equations [10].

To calculate the non-dimensional velocity just behind the shock, the following equation is used [10].

$$V_2 = \sqrt{\frac{2}{(\gamma-1)M_2} + 1} \tag{6}$$

Using the non-dimensional velocity just behind the shock and the geometry, the non-dimensional velocity components are calculated [10].

$$V_{\theta 2} = -V_2 \sin(\beta - \delta) \tag{7}$$

$$V_{r2} = V_2 cos(\beta - \delta) \tag{8}$$

The Taylor-Maccoll equation re-cast in terms of non-dimensional velocities, as recommended by Anderson, is given below [10].

$$\frac{\gamma-1}{2}\left[1 - {V_r}^2 - \left(\frac{dV_r}{d\theta}\right)^2\right]\left[2\,V_r + cot(\theta)\frac{dV_r}{d\theta} + \frac{d^2V_r}{d\theta^2}\right] - \frac{dV_r}{d\theta}\left[V_r\frac{dV_r}{d\theta} + \frac{dV_r}{d\theta}\frac{d^2V_r}{d\theta^2}\right] = 0 \tag{9}$$

Additionally, the following two equations are used as implied or in a slightly modified form from Anderson [10].

$$V_\theta = \frac{dV_r}{d\theta} \tag{10}$$

$$V = \sqrt{{V_r}^2 + {V_\theta}^2} \tag{11}$$

These equations are arranged to form a system of first order ordinary differential equations, which is suited to easy numerical integration with common tools.

$$\begin{bmatrix} \frac{dV_r}{d\theta} \\ \frac{dV_\theta}{d\theta} \end{bmatrix} = \begin{bmatrix} V_\theta c \\ \frac{\frac{\gamma-1}{2}\left[1 - {V_r}^2 - {V_\theta}^2\right]\left[2V_r + cot(\theta)V_\theta\right] - V_r{V_\theta}^2}{{V_\theta}^2 - \frac{\gamma-1}{2}\left[1 - {V_r}^2 - {V_\theta}^2\right]} \end{bmatrix} \tag{12}$$

For this study, MATLAB's ode45 was used to perform the numerical integration. The initial condition is the non-dimensional velocity behind the shock and the integration ends where $V_\theta = 0$, indicating flow along a solid cone surface. An iterative scheme is used to find this point.

The density ratio at each angle is required. For this, the Mach number at each angle must first be calculated.

The non-dimensional velocity in terms of Mach number is given in the equation that follows [10].

$$V = \left[\frac{2}{(\gamma - 1)M^2} + 1\right]^{-\frac{1}{2}} \tag{13}$$

This equation can be re-arranged to make the Mach number the subject of the formula.

$$M = \sqrt{\frac{2V^2}{(\gamma - 1)(1 - V^2)}} \tag{14}$$

While the density ratio behind the shock, due to the flow being subject to isentropic compression [10], is calculated using the following equation, which is a manipulation of the equations in Anderson [12].

$$\frac{\rho}{\rho_2} = \left(\frac{1 + \frac{(\gamma - 1)}{2}M^2}{1 + \frac{(\gamma - 1)}{2}{M_2}^2}\right)^{\frac{-1}{\gamma - 1}} \tag{15}$$

The density just behind the shock is then calculated in the following equation.

$$\rho_2 = \rho_1 \frac{\rho_2}{\rho_1} \tag{16}$$

While the density at an arbitrary point behind the shock is finally calculated using the following equation.

$$\rho = \rho_2 \frac{\rho}{\rho_2} \tag{17}$$

Having solved the Taylor-Maccoll equations and used the oblique shock and isentropic flow relations, for each angle θ behind the shock, the density ρ is known. Figure 4shows the variation in Mach number while Figure 3 shows the variation in density for a sample case where the shock angle is prescribed to be $28.67°$, the Mach Number is 2.48, the ratio of specific heats is set to 1.4 and the free-stream density is $0.446\ kg/m^3$. These values are representative of the test setup outlined in section 3.2.

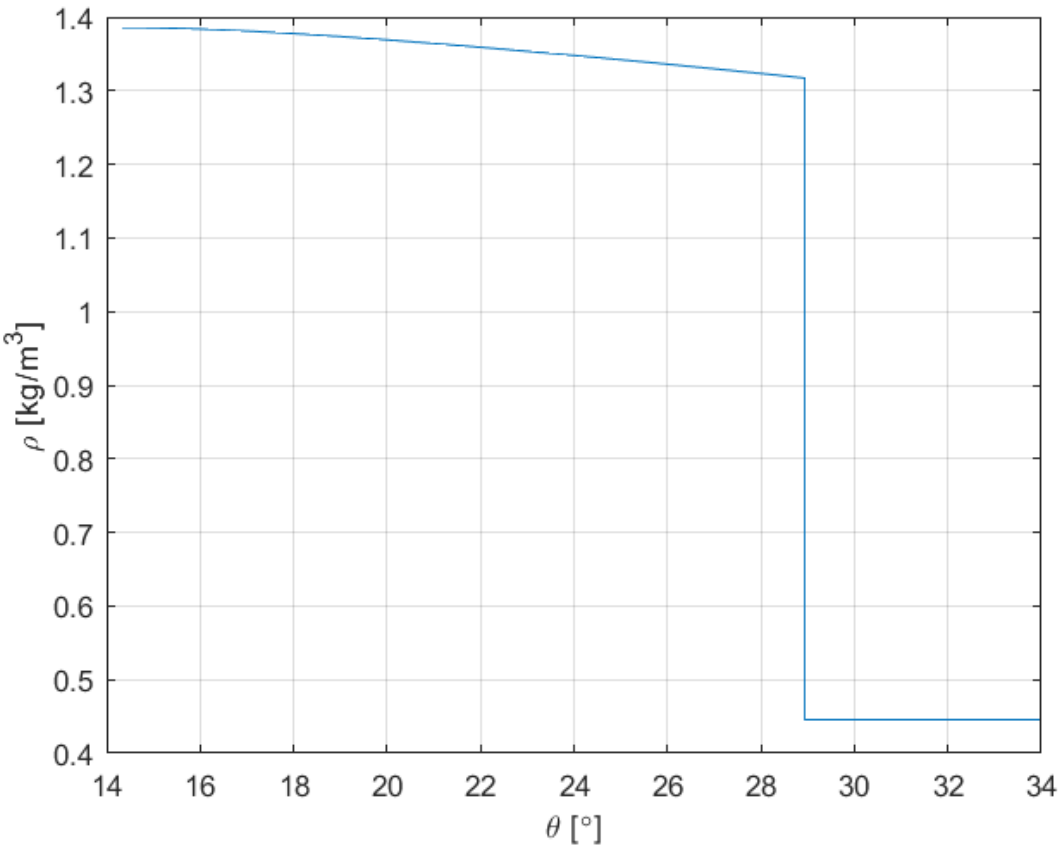


Figure 3: Plot of density against angle

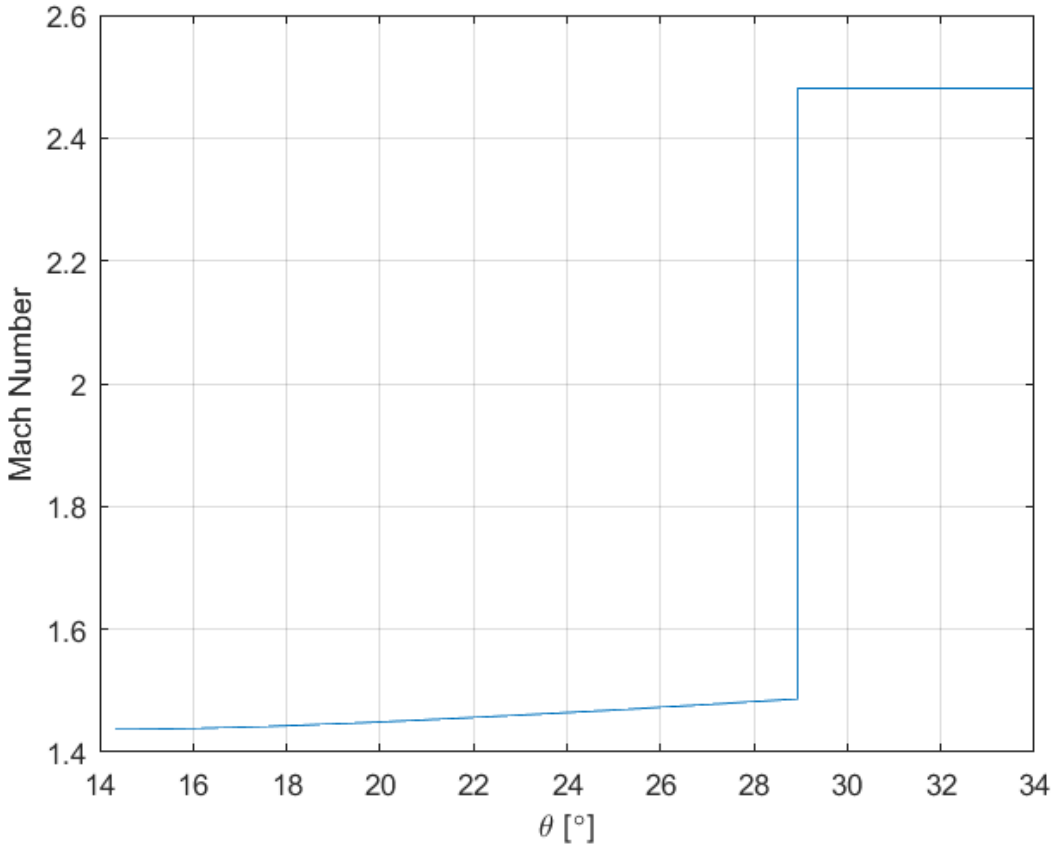


Figure 4: Plot of Mach number against angle

2.1.2 Light-ray deflections

Geometrical optics is used to predict how light rays will deflect when moving from the background, through the test section and into the camera. The angles of ray deflection are calculated and used to predict how the background images will distort. The angles of deflection are broken down into two components, one component due to the discontinuity in refractive index at the conical shock surface and one due to density gradients behind the shock.

The model used assumes parallel rays entering the camera, an assumption that becomes more valid for cameras with high zoom lenses. The Helmholtz reciprocity principle is used to trace the rays backward from the camera toward the background. For this study, the calculations are performed in the quadrant where x, y and z values are positive, and symmetry is used to complete the estimation of the distortion angles. Figure 5 and Figure 6 illustrate the geometry used in this solution to the problem. The parameters illustrated in Figure 5 and the deflection angles illustrated in Figure 6 are used throughout this section.

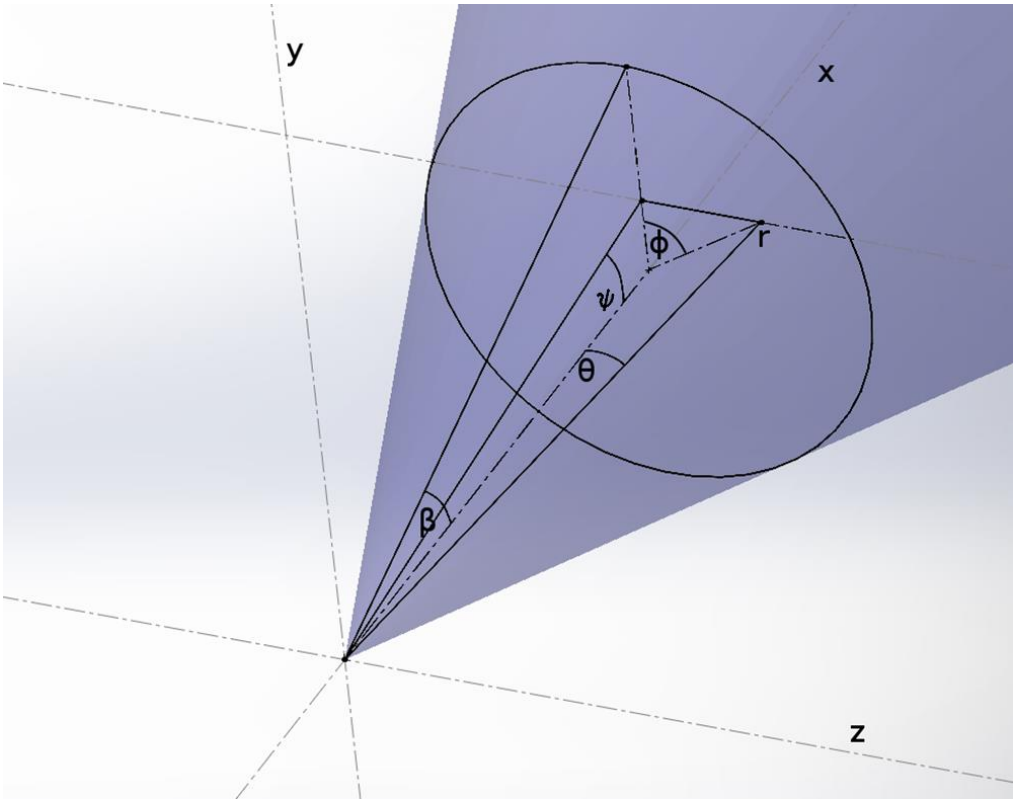


Figure 5: Illustration of geometry used for calculating ray deflections

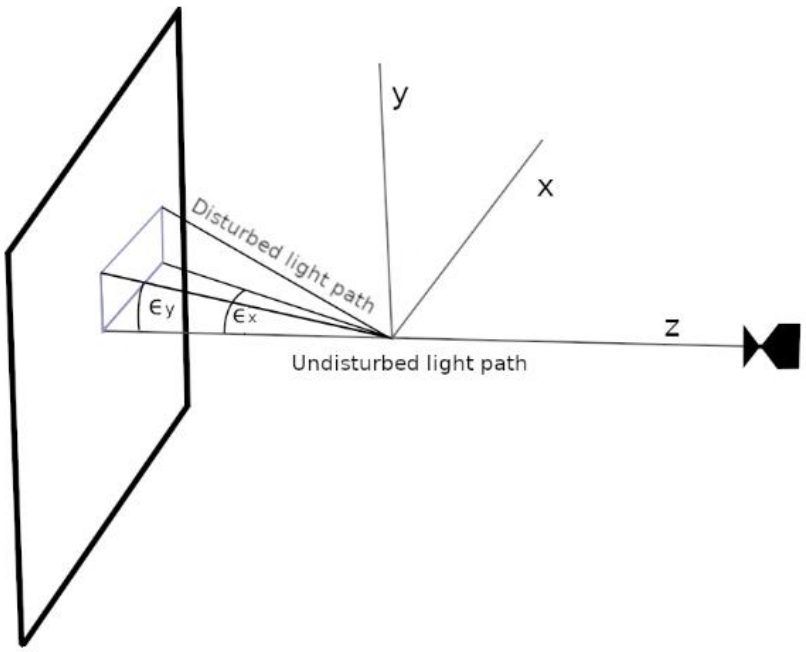


Figure 6: Deflection angles

Deflections due to refraction index discontinuity at the shock boundary

The vector form of Snell's law is used to predict the deflection of the light rays at the shock cone surface.

$$\boldsymbol{r_o} = \frac{n_i}{n_o}\boldsymbol{r_i} + \left(-\frac{n_i}{n_o}\boldsymbol{p}\cdot\boldsymbol{r_i} - \sqrt{1-\left(\frac{n_i}{n_o}\right)^2 (1-(\boldsymbol{p}\cdot\boldsymbol{r_i})^2)} \right)\boldsymbol{p} \qquad (18)$$

Here, $\boldsymbol{r_o}$ is the outgoing ray vector, $\boldsymbol{r_i}$ is the incoming ray vector, $\boldsymbol{p}$ is the unit normal to the cone surface. The refractive indices of the initial and final medium are denoted n_i and n_o respectively. This is illustrated in Figure 7.

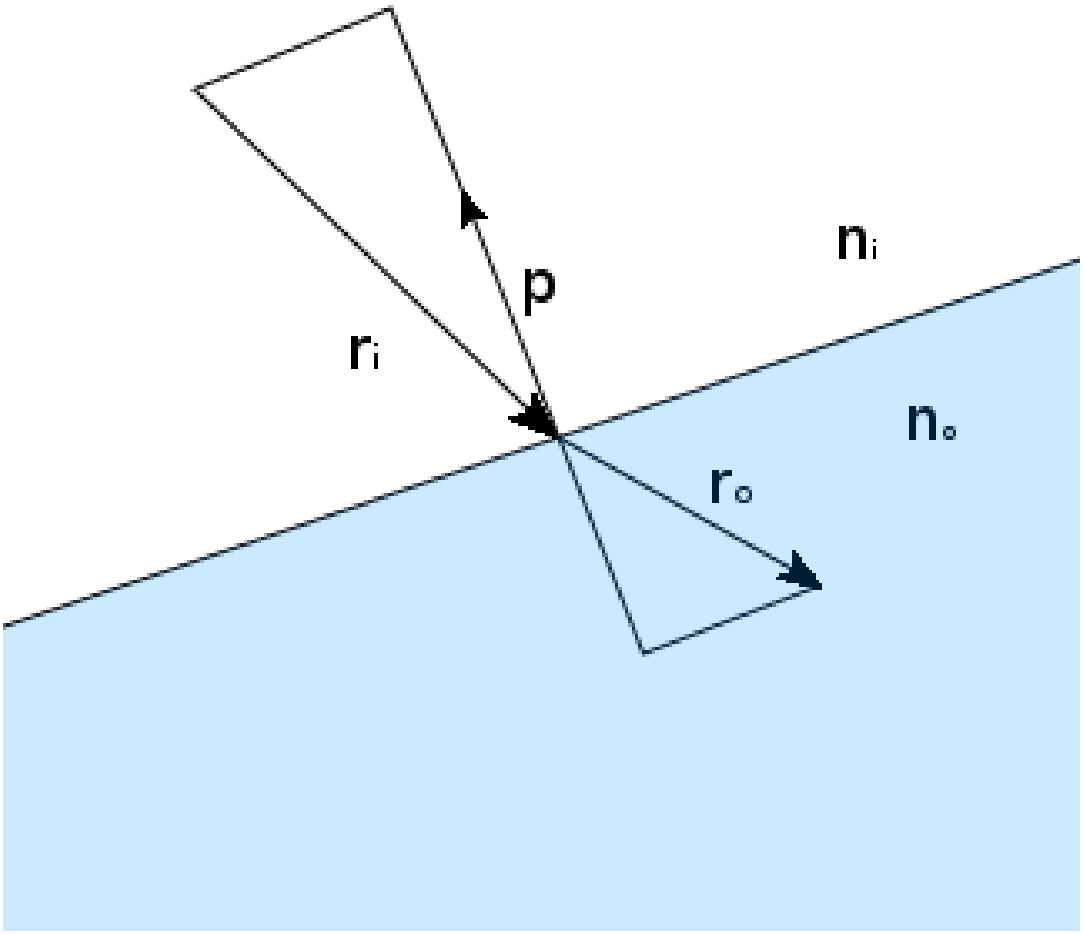


Figure 7: Vectors used in the vector formulation of Snell's law

The normal to the shock surface, $\boldsymbol{p}$, for a given angle ϕ can be determined. The result is shown in equation 19.

$$\boldsymbol{p} = [-\sin(\beta) \quad \cos(\beta)\cos(\phi) \quad \cos(\beta)\sin(\phi)] \tag{19}$$

The equation for angle ϕ is given below.

$$\phi = \operatorname{acos}\left(\frac{tan\theta}{tan\beta}\right) \tag{20}$$

The incident ray,$\boldsymbol{r_i}$, is chosen in the negative z direction in order to trace the ray backwards from the camera toward the background using Helmholtz reciprocity principle.

$$\boldsymbol{r_i} = [0 \quad 0 \quad -1] \tag{21}$$

The angles of deflection in lateral and vertical directions can be obtained according to equations 22 and 23.

$$\epsilon_{x_s} = \operatorname{atan}\left(\frac{x_o}{-z_o}\right) \tag{22}$$

$$\epsilon_{y_s} = \operatorname{atan}\left(\frac{y_o}{-z_o}\right) \tag{23}$$

Here, x_o, y_o and z_o, are the components of $\boldsymbol{r_o}$.

Figure 8 and Figure 9 show sample results for the deflection at the shock surface as ψ changes for lateral and vertical deflections respectively. For this case, the Mach number was set to 2.5 and the shock angle to 30°.

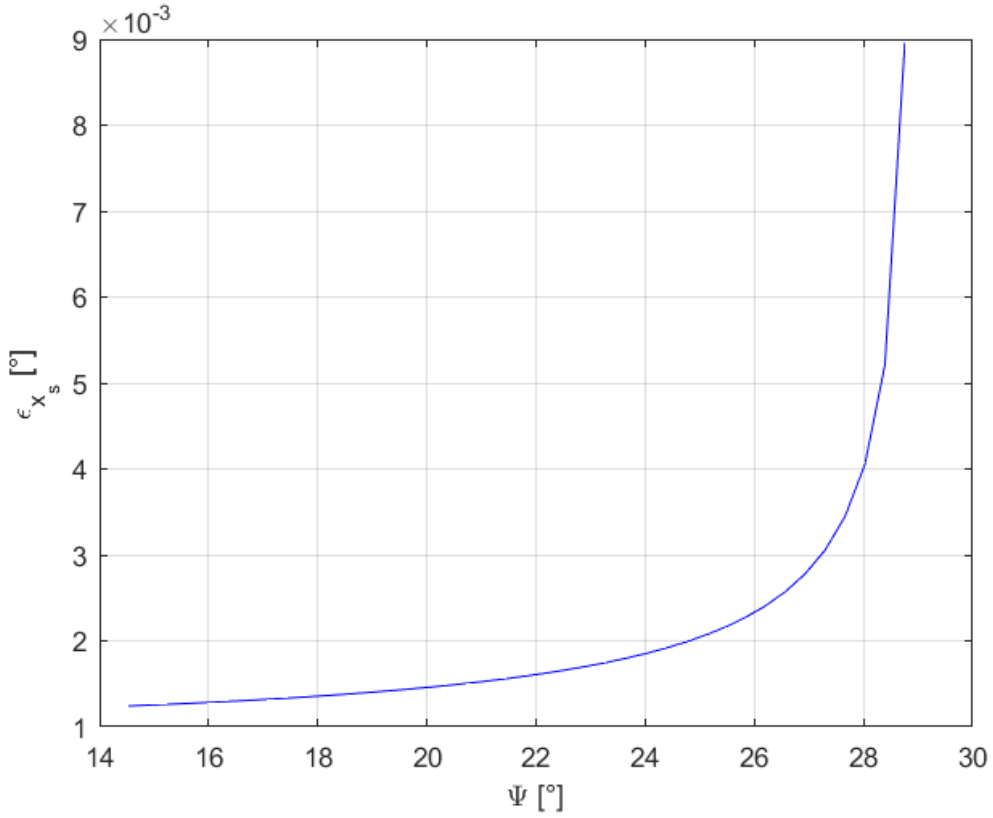


Figure 8: Example of lateral deflection of light rays at the shock

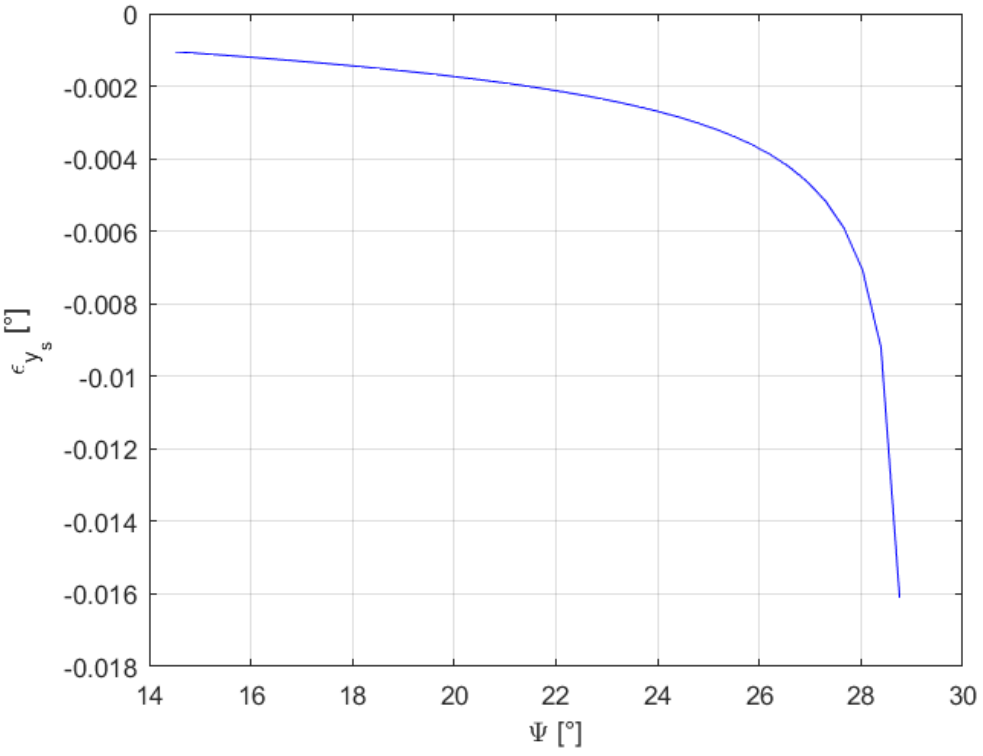


Figure 9: Example of vertical deflection of light rays at the shock

Deflections due to density gradients behind the shock

For typical schlieren setups where the light ray does not deviate significantly from its original path while moving through the schlieren object (known as the paraxial approximation), the lateral and vertical angular deflection of a light ray travelling along the z axis may be calculated as follows. [13]

$$\epsilon_{x_l} = \frac{1}{n_0}\int_{z_0}^{z_1}\frac{\partial n}{\partial x}dz \tag{24}$$

$$\epsilon_{y_l} = \frac{1}{n_0}\int_{z_0}^{z_1}\frac{\partial n}{\partial y}dz \tag{25}$$

Using Helmholtz reciprocity principle, a ray entering the camera can be traced back to its origin. The angle of deflection for this case is then opposite in sign to tracing the ray from its origin.

$$\epsilon_{x_g} = -\frac{1}{n_0}\int_{z_1}^{z_0}\frac{\partial n}{\partial x}dz \tag{26}$$

$$\epsilon_{y_g} = -\frac{1}{n_0}\int_{z_1}^{z_0}\frac{\partial n}{\partial y}dz \tag{27}$$

This is a useful form of the equation for modelling background distortions in the context of BOS.

With some manipulation and a change of variable, it can be shown that the lateral angular deflection may be calculated as follows.

$$\epsilon_{x_g} = -\frac{1}{n_0}\int_{\theta_1}^{\theta_0}\frac{\partial n}{\partial \theta}\frac{\partial \theta}{\partial x}\frac{\partial z}{\partial \theta}d\theta \tag{28}$$

Similarly, the vertical angular deflection may be calculated as follows.

$$\epsilon_{y_g} = -\frac{1}{n_0}\int_{\theta_1}^{\theta_0}\frac{\partial n}{\partial \theta}\frac{\partial \theta}{\partial y}\frac{\partial z}{\partial \theta}d\theta \tag{29}$$

For this study, the integrations are performed in the quadrant where x, y and z are positive and symmetry is used to complete the estimation of the distortion angles.

The path of integration goes from just behind the shock, where the ray enters the area of changing density gradient in the negative z-direction, and continues in the negative z-direction until it reaches the x-y plane. Symmetry will be used later to complete the deflection through the rest of the flow region. With this path of integration, the bounds of the integration are set as follows.

$$\epsilon_{x_g} = -\frac{1}{n_0}\int_{\beta}^{\psi} \frac{\partial n}{\partial \theta}\frac{\partial \theta}{\partial x}\frac{\partial z}{\partial \theta} d\theta \tag{30}$$

$$\epsilon_{y_g} = -\frac{1}{n_0}\int_{\beta}^{\psi} \frac{\partial n}{\partial \theta}\frac{\partial \theta}{\partial y}\frac{\partial z}{\partial \theta} d\theta \tag{31}$$

The gradient $\frac{\partial n}{\partial \theta}$ can be calculated from the density distribution of the flow while $\frac{\partial \theta}{\partial x}$, $\frac{\partial \theta}{\partial y}$ and $\frac{\partial z}{\partial \theta}$ need to be calculated. These calculations follow in the sections below.

Derivative of θ with respect to x

From the geometry of the problem, the angle θ has the relationship with the Cartesian co-ordinates as stated in the following equation.

$$\theta = \text{atan}\left(\frac{\sqrt{(z^2 + y^2)}}{x}\right) \tag{32}$$

The variables x, y and z are the components of $\boldsymbol{r}$, the vector extending from the point of the cone to the point on the line of integration that is of interest.

The required derivative is given below.

$$\frac{d\theta}{dx} = \left(\frac{y^2 + z^2}{x^2} + 1\right)^{-1}\left(-\frac{\sqrt{z^2 + y^2}}{x^2}\right) \tag{33}$$

This derivative is simplified as follows.

$$\frac{d\theta}{dx} = -\frac{\sqrt{z^2 + y^2}}{y^2 + z^2 + x^2} \tag{34}$$

By denoting the magnitude of $\boldsymbol{r}$ as r, $\frac{d\theta}{dx}$ can be cast as follows.

$$\frac{d\theta}{dx} = -\frac{\sin(\theta)}{r} \tag{35}$$

To check that the derivative has been calculated correctly, the following logic can be followed.

When $\theta = 90°$, $\frac{d\theta}{dx} = -\frac{1}{r}$, which is consistent with a small angle approximation when a small movement in x is perpendicular to the vectors $\boldsymbol{r}$ and $\boldsymbol{\theta}$, as it is when $\theta = 90°$. The sign is correct as θ decreases with increasing x.

When $\theta = 0°$, $\frac{d\theta}{dx} = 0$ which is consistent with x being parallel to $\boldsymbol{r}$, which it is when $\theta = 0°$.

Derivative of $\boldsymbol{\theta}$ with respect to $\boldsymbol{y}$

From the geometry of the problem, the angle θ has the relationship with the Cartesian co-ordinates as stated in the following equation.

$$\theta = \operatorname{atan}\left(\frac{\sqrt{z^2 + y^2}}{x}\right) \tag{36}$$

The gradient $\frac{d\theta}{dy}$ is calculated and simplified in the steps that follow.

$$\frac{d\theta}{dy} = \left(\frac{y^2 + z^2}{x^2} + 1\right)^{-1} \left(\frac{1}{2}\frac{2y}{x\sqrt{z^2 + y^2}}\right) \tag{37}$$

$$\frac{d\theta}{dy} = \frac{y}{\left(\frac{y^2 + z^2}{x^2} + 1\right)\left(x\sqrt{z^2 + y^2}\right)} \tag{38}$$

$$\frac{d\theta}{dy} = \frac{yx}{(x^2 + y^2 + z^2)\left(\sqrt{y^2 + z^2}\right)} \tag{39}$$

$$\frac{d\theta}{dy} = \frac{1}{r}\frac{x}{r}\frac{y}{\sqrt{z^2 + y^2}} \tag{40}$$

$$\frac{d\theta}{dy} = \frac{1}{r}\cos(\theta)\, cos(\phi) \tag{41}$$

To check that the derivative has been calculated correctly, the following logic can be followed.

When $\phi = 90°$, the gradient $\frac{d\theta}{dy} = 0$. This is consistent with $\boldsymbol{r}$ having no y component, which it is when $\phi = 90°$.

When $\phi = 0°$ and $\theta = 90°$, $\frac{d\theta}{dy} = 0$, which is consistent with y being parallel to $\boldsymbol{r}$, which it is when $\theta = 90°$.

When $\phi = 0°$ and $\theta = 0°$, $\frac{d\theta}{dy} = \frac{1}{r}$, which is consistent with a small angle approximation when a small movement in y is perpendicular to the vectors $\boldsymbol{r}$ and $\boldsymbol{\theta}$,as it is when $\theta = 0°$. The sign is correct as θ increases with an increase in y.

Derivative of z with respect to θ

From the geometry, z is given according to the following equation.

$$z = r\sin(\theta)\sin(\phi) \tag{42}$$

Taking the derivative $\frac{dz}{d\theta}$ results in the equation that follows.

$$\frac{dz}{d\theta} = r\cos(\theta)\sin(\phi) \tag{43}$$

In order to remove ϕ, the following substitution is made.

$$sin(\phi) = \frac{\sqrt{\tan^2(\theta) - \tan^2(\psi)}}{\tan(\theta)} \tag{44}$$

The resulting equation for $\frac{dz}{d\theta}$ is given below.

$$\frac{dz}{d\theta} = rcos(\theta)\frac{\sqrt{\tan^2(\theta) - \tan^2(\psi)}}{\tan(\theta)} \tag{45}$$

To check that the derivative has been calculated correctly, the following logic can be followed:

When $\psi = 0°$, $\frac{dz}{d\theta} = rcos(\theta)$.

When $\psi = 0°$ and $\theta = 0°$, $\frac{dz}{d\theta} = r$, which is consistent with the small angle approximation when $\boldsymbol{r}$ is parallel to the x axis.

When $\psi = 0°$ and $\theta = 90°$, $\frac{dz}{d\theta} = 0$, which is consistent with the small angle approximation when $\boldsymbol{r}$ is parallel to the z axis.

When $\psi = 90°$, θ is also 90°, and $\boldsymbol{r}$ lies anywhere in the y-z plane. The gradient $\frac{dz}{d\theta}$ is therefore undefined, as is $rcos(\theta)\frac{\sqrt{\tan^2(\theta)-\tan^2(\psi)}}{\tan(\theta)}$.

Multiplying the derivatives

Some simplification can be achieved by multiplying $\frac{\partial\theta}{\partial x}$ with $\frac{\partial z}{\partial\theta}$ and $\frac{\partial\theta}{\partial y}$ with $\frac{\partial z}{\partial\theta}$.

For the former, the multiplication is performed.

$$\frac{\partial\theta}{\partial x}\frac{\partial z}{\partial\theta} = -\frac{\sin(\theta)}{r}rcos(\theta)\frac{\sqrt{\tan^2(\theta) - \tan(\psi)}}{\tan(\theta)} \tag{46}$$

This can be simplified.

$$\frac{\partial\theta}{\partial x}\frac{\partial z}{\partial\theta} = -\left[cos^2(\theta)\sqrt{\tan^2(\theta) - \tan(\psi)}\right] \tag{47}$$

For the latter, the multiplication results in the following equation.

$$\frac{\partial\theta}{\partial y}\frac{\partial z}{\partial\theta} = \frac{1}{r}\cos(\theta)\, cos(\phi) r cos(\theta)\frac{\sqrt{\tan^2(\theta) - \tan(\psi)}}{\tan(\theta)} \tag{48}$$

To simplify the product, the following substitution is required.

$$\frac{\tan(\psi)}{\tan(\theta)} = cos(\phi) \tag{49}$$

After substitution, the product can be simplified to give the following result.

$$\frac{\partial\theta}{\partial y}\frac{\partial z}{\partial\theta} = \frac{\tan(\psi)}{\tan^2(\theta)}\left[\cos^2(\theta)\sqrt{\tan^2(\theta) - \tan(\psi)}\right] \tag{50}$$

Performing the integration

Substituting the solutions for the multiplied derivatives into the equations for the deflection of the rays leads to the following two equations.

$$\epsilon_{x_g} = \frac{1}{n_0}\int_{\beta}^{\psi}\frac{\partial n}{\partial\theta}\, cos^2(\theta)\sqrt{\tan^2(\theta) - \tan(\psi)}\; d\theta \tag{51}$$

$$\epsilon_{y_g} = -\frac{1}{n_0}\int_{\beta}^{\psi}\frac{\partial n}{\partial\theta}\,\frac{\tan(\psi)}{\tan^2(\theta)}\left[\,\cos^2(\theta)\sqrt{\tan^2(\theta) - \tan(\psi)}\,\right] d\theta \tag{52}$$

These two equations are solved numerically using trapezoidal integration for the integrals and finite differences for the derivative inside the integral. Figure 10 and Figure 11 illustrate the results for the sample case.

It is interesting to note that for a given shock angle, both ϵ_x and ϵ_y are functions of ψ only. That is, the distortion along a ray emanating from the point of the cone is constant, regardless of distance from the point.

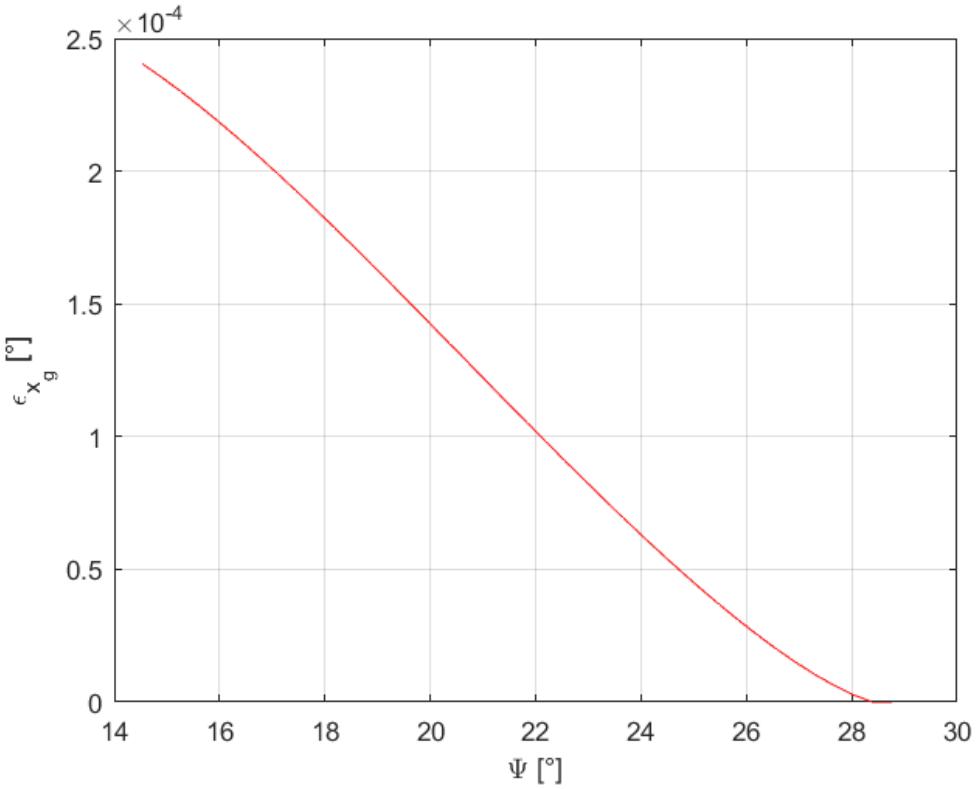


Figure 10: Graph illustrating an example of lateral deflection angle with projected ray angle

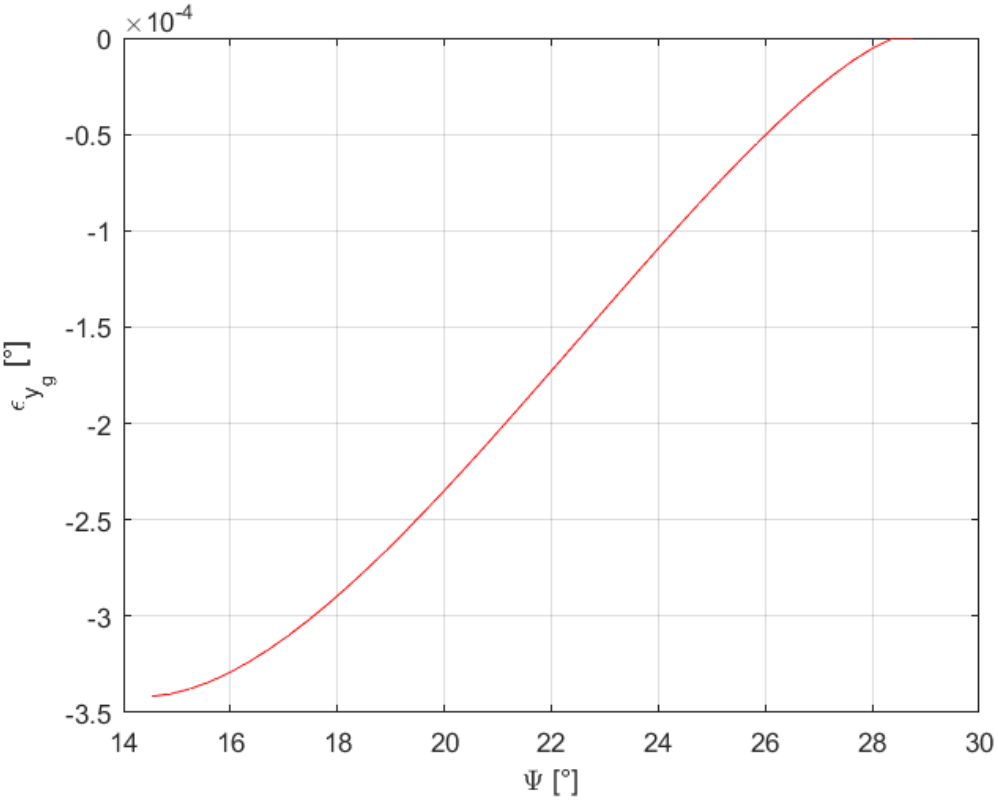


Figure 11: Graph illustrating an example of vertical deflection angle with projected ray angle

Numerical solution of distortion angles

Equations 51and 52 are solved numerically as outlined below.

Firstly, the refractive index n is calculated using the Gladstone-Dale relation below.

$$n = K\rho + 1 \tag{53}$$

For air, $K = 0.000227\frac{m^3}{kg}$.

The density required for the equation is obtained for the various values of θ using the $\rho(\theta)$ values that were calculated in the solution of the Taylor-Maccoll equations.

Central differences is then used to approximate $\frac{\partial n}{\partial \theta}$.

$$\left.\frac{\partial n}{\partial \theta}\right|_{s_l} \approx \frac{n_{k+1} - n_k}{\theta_{k+1} - \theta_k} \tag{54}$$

$$s_l \equiv \frac{1}{2}(\theta_{k+1} + \theta_k) \tag{55}$$

For each combination of $\theta = s_l$ and $\psi = s_m$, $cos^2(\theta)\sqrt{\tan^2(\theta) - \tan(\psi)}$,for equation 51, and $\frac{\tan(\psi)}{\tan^2(\theta)}\left[\cos^2(\theta)\sqrt{\tan^2(\theta) - \tan(\psi)}\right]$, for equation 52, are calculated.

Then for each $\psi = s_m$, the trapezoidal rule is used to approximate the integrals.

The following two functions are defined to aid in simplification.

$$f(\psi, \theta) = \frac{\partial n}{\partial \theta}\, cos^2(\theta)\sqrt{\tan^2(\theta) - \tan(\psi)} \tag{56}$$

$$g(\theta_0,\theta)=\frac{\partial n}{\partial\theta}\,\frac{\tan(\psi)}{\tan^2(\theta)}\left[\cos^2(\theta)\sqrt{\tan^2(\theta)-\tan(\psi)}\right] \tag{57}$$

The following two equations result.

$$\epsilon_{x_g}(s_m)\approx\frac{1}{n_0}\left(\frac{1}{2}\sum_{l=1}^{l=L-1}\left[\Big(\big(f(s_m,s_{l+1})+f(s_m,s_l)\big)(s_{l+1}-s_l)\Big)\right]\right) \tag{58}$$

$$\epsilon_{y_g}(s_m)\approx\frac{1}{n_0}\left(\frac{1}{2}\sum_{l=1}^{l=L-1}\left[\Big(\big(g(s_m,s_{l+1})+g(s_m,s_l)\big)(s_{l+1}-s_l)\Big)\right]\right) \tag{59}$$

This gives an angular distortion for any point found along a line at an angle s_m to the centreline of the model lying in the plane perpendicular to the viewing direction. For angles in between these discrete points, the values are interpolated.

Total angular deflection

The total deflection is calculated by adding the deflections caused by the discontinuity in refractive index at the shock boundary with that caused by the density gradients behind the shock. It was chosen to take advantage of symmetry in the problem by calculating the angular deflection up to the centre plane of the cone and then doubling the resulting angle. This is shown in the following two equation.

$$\epsilon_x=2\left(\epsilon_{x_s}+\epsilon_{x_g}\right) \tag{60}$$

$$\epsilon_y=2\left(\epsilon_{y_s}+\epsilon_{y_g}\right) \tag{61}$$

The total deflections and the two components from which they are composed are shown in the figures that follow.

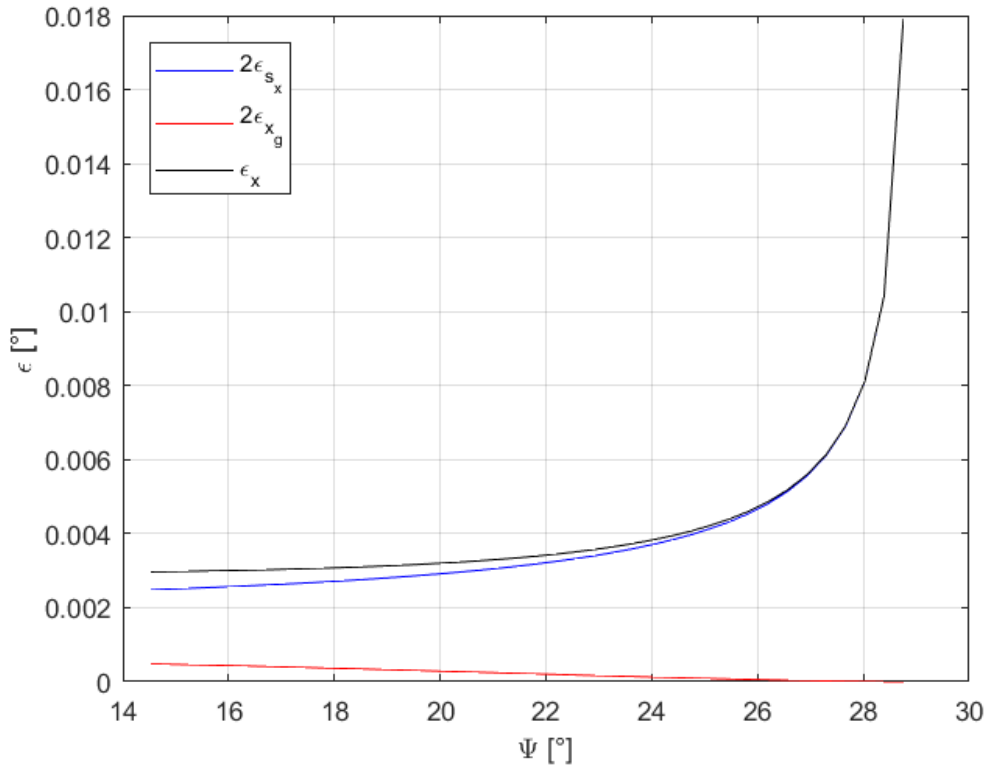


Figure 12: Example of lateral deflection of rays with projected angle

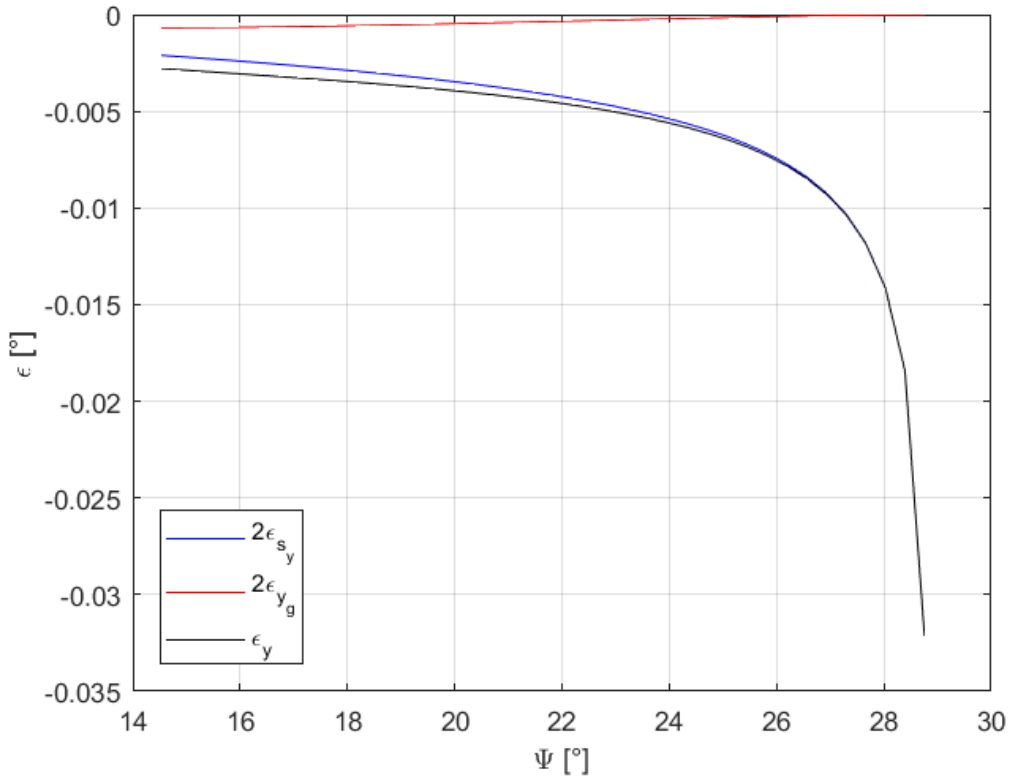


Figure 13: Example of vertical deflection of rays with deflection angle

2.1.3 Background distortions

Background distortion calculations make use of the following assumptions:

- Parallel rays entering the camera
- Angular distortion occurs at the centre of the test section
- Pixel positions correspond to points spread equally along the background in both lateral and vertical directions

The point of the cone in the synthetic images is placed at $\frac{1}{3}$ the width of the image from the left hand side and half way down the image. For a particular pixel then, ψ may be calculated.

$$\psi(i,j) = \arctan\left(\frac{j_0 - j}{i - i_o}\right) \tag{62}$$

The variable i is the column index and j is the row index of the pixel. The choice $i_0 = \frac{1}{3} i_{max}$ and $j_0 = \frac{1}{2} j_{max}$ was made to set the point of the cone one third laterally into, and half way down the resulting image.

At each pixel, $\epsilon_x(i,j)$ and $\epsilon_y(i,j)$ are calculated as follows.

- For $\theta_{cone} \leq \psi(i,j) \leq \beta$, the values $\epsilon_x(\psi)$ and $\epsilon_y(\psi)$ were interpolated from $\epsilon_x(s_m)$ and $\epsilon_y(s_m)$.
- For $-\theta_{cone} \leq \psi(i,j) \leq -\beta$, the values $\epsilon_x(-\psi)$ and $\epsilon_y(-\psi)$ were interpolated from $\epsilon_x(s_m)$ and $\epsilon_y(s_m)$ with a change in sign for ϵ_y.
- Otherwise, the values for ϵ_x and ϵ_y were set to zero.

The results for a sample case are shown in Figure 14 and Figure 15.

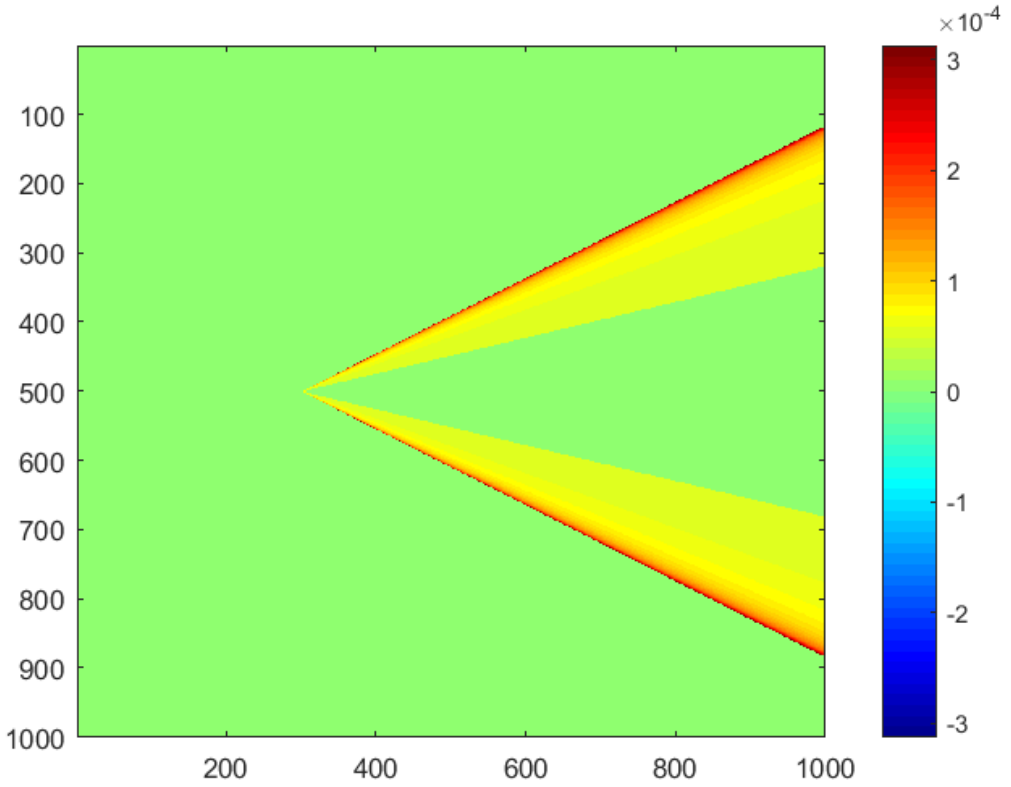


Figure 14: Example of horizontal deflection angle in radians

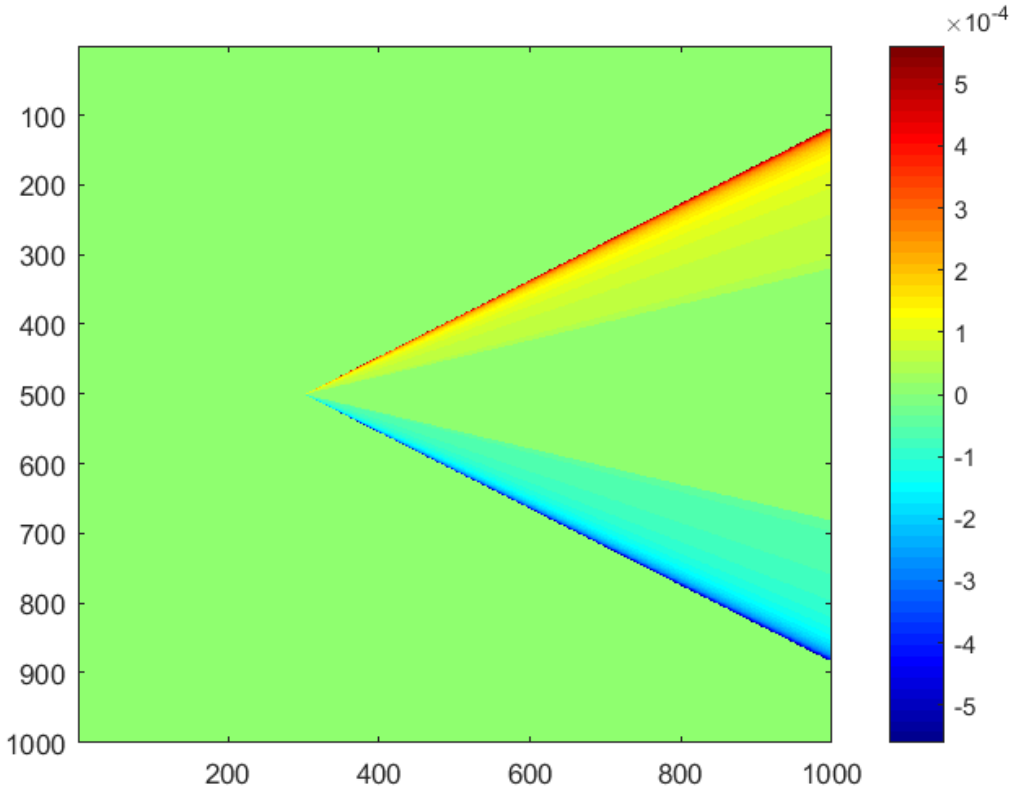


Figure 15: Example of vertical deflection angle in radians

2.1.4 Optical flow predictions

Optical flow predictions are calculated according to the total angular deflections of the incoming light rays and the geometry of the test setup. The effect of the assumption that rays exiting the test section toward the camera are parallel becomes less severe the higher the zoom the camera uses and the further away from the test section the camera is placed.

With this assumption, it is not necessary to take in to account the physical size of the background since angular deflection is only a function of ψ, however, the distance that the background is placed away from the test section and the distance on the background that relates to a change from one pixel to the next in the image are important. The distortion in pixels for each position in the image is calculated as follows.

$$\boldsymbol{w} = -\frac{r}{\zeta}\left(\epsilon_x, \epsilon_y\right) \tag{63}$$

In equation 63, the symbol ζ denotes the distance between the positions on the background that correspond to pixel positions on the image and r is the distance from the centreline of the test section to the background. The negative sign appears because the apparent movement in the background has the opposite sense to that of the light ray deflection. For example, if a light ray is traced backward from the camera to the background and is deflected downward, the portion of the background object being sampled in the distorted image lies below the portion of the background object that was sampled in the undistorted image. The apparent movement (from undistorted image to distorted image) is therefore in the upward direction.

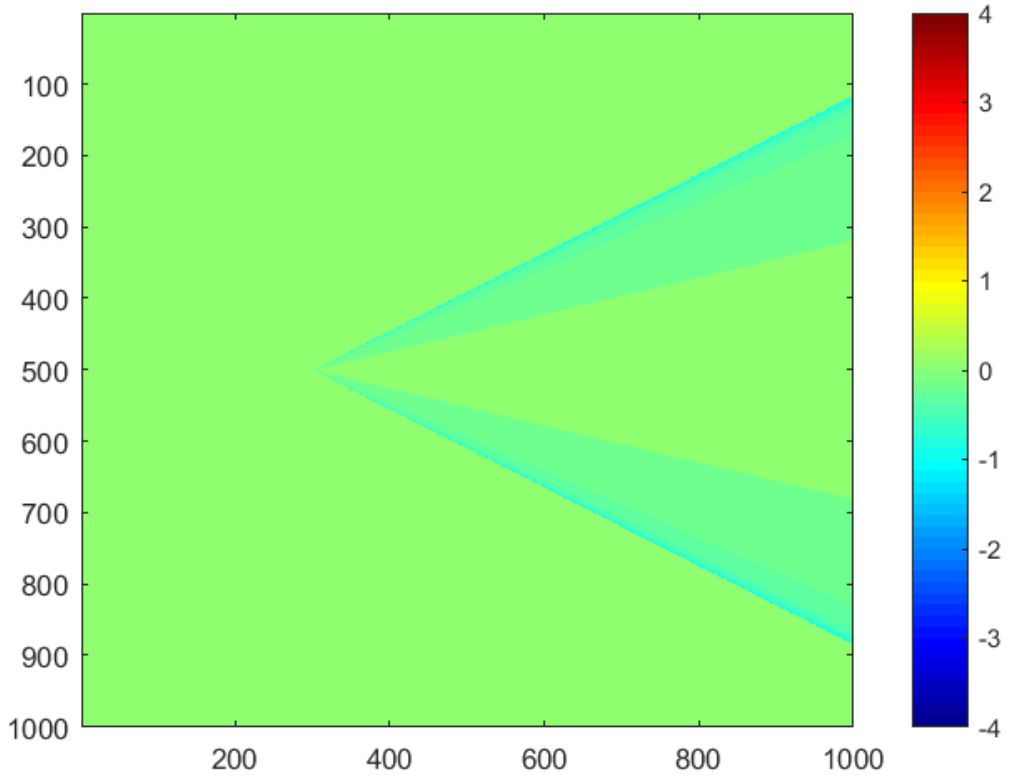


Figure 16: Example of horizontal deflection in pixels

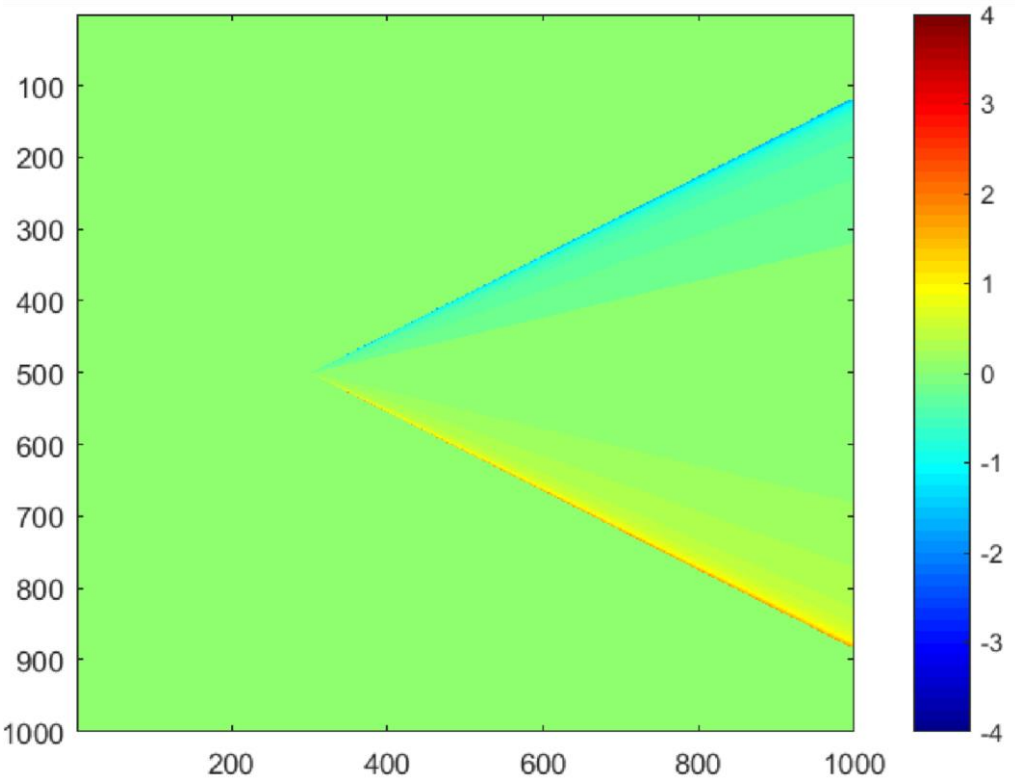


Figure 17: Example of vertical deflection in pixels

In order to visualise this movement, the distorted image can then be sampled according to the following equation.

$$I_1(i,j) = I_0(i+u, j+v) \tag{64}$$

Fractional pixel values are handled by interpolation of the image values.

It is easy to interpret this distortion when viewed using a background with lateral and horizontal lines. This will be different to the backgrounds that are eventually analysed in this study. To visualise the movement of the components of the image, the deflections from Figure 14 and Figure 15 are used with r and ζ such that the maximum vertical deflections are 8 pixels, Figure 18 (with grid spacing of 16x16 pixels) distorts to create Figure 19.

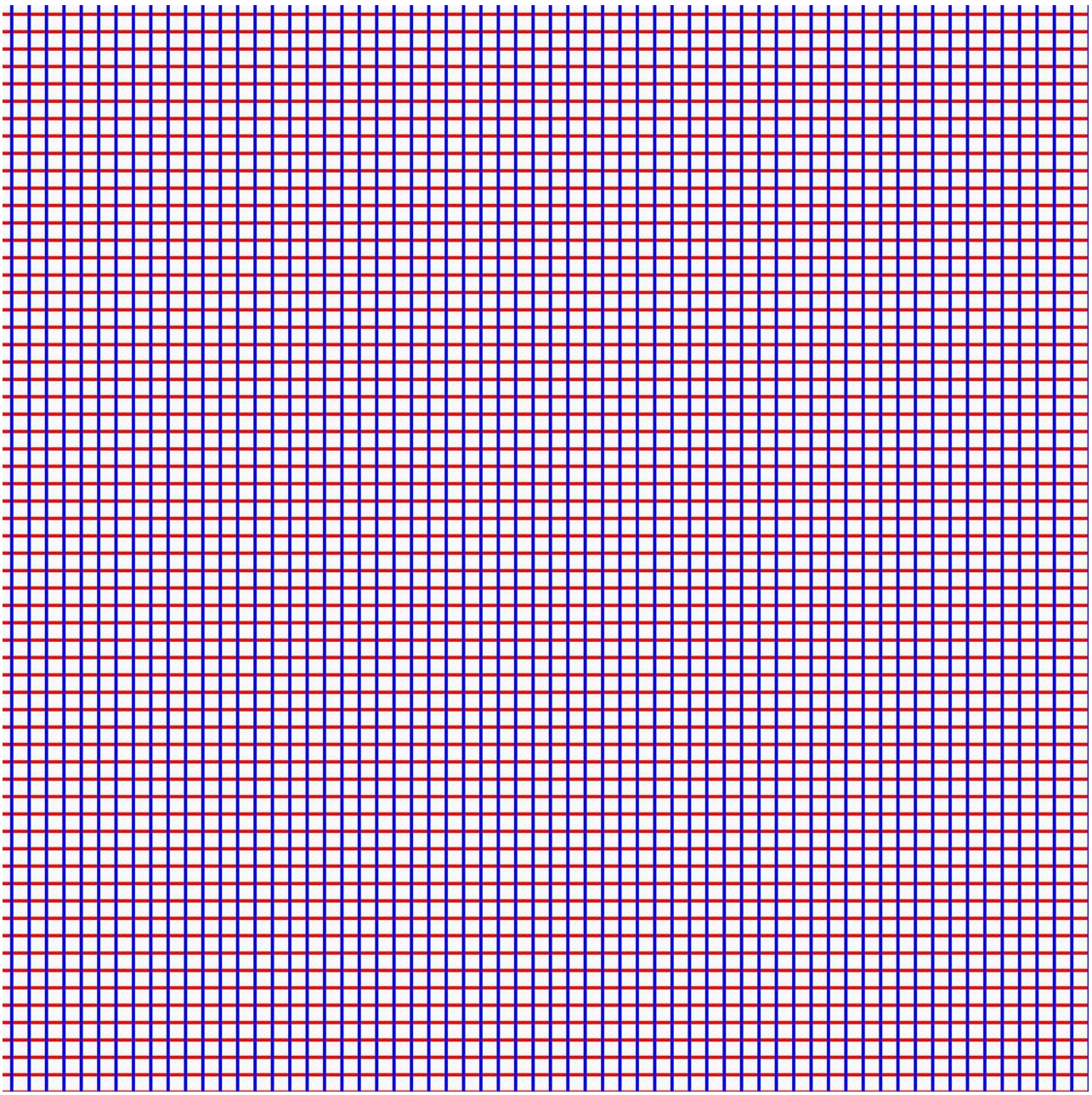


Figure 18: Undistorted background image

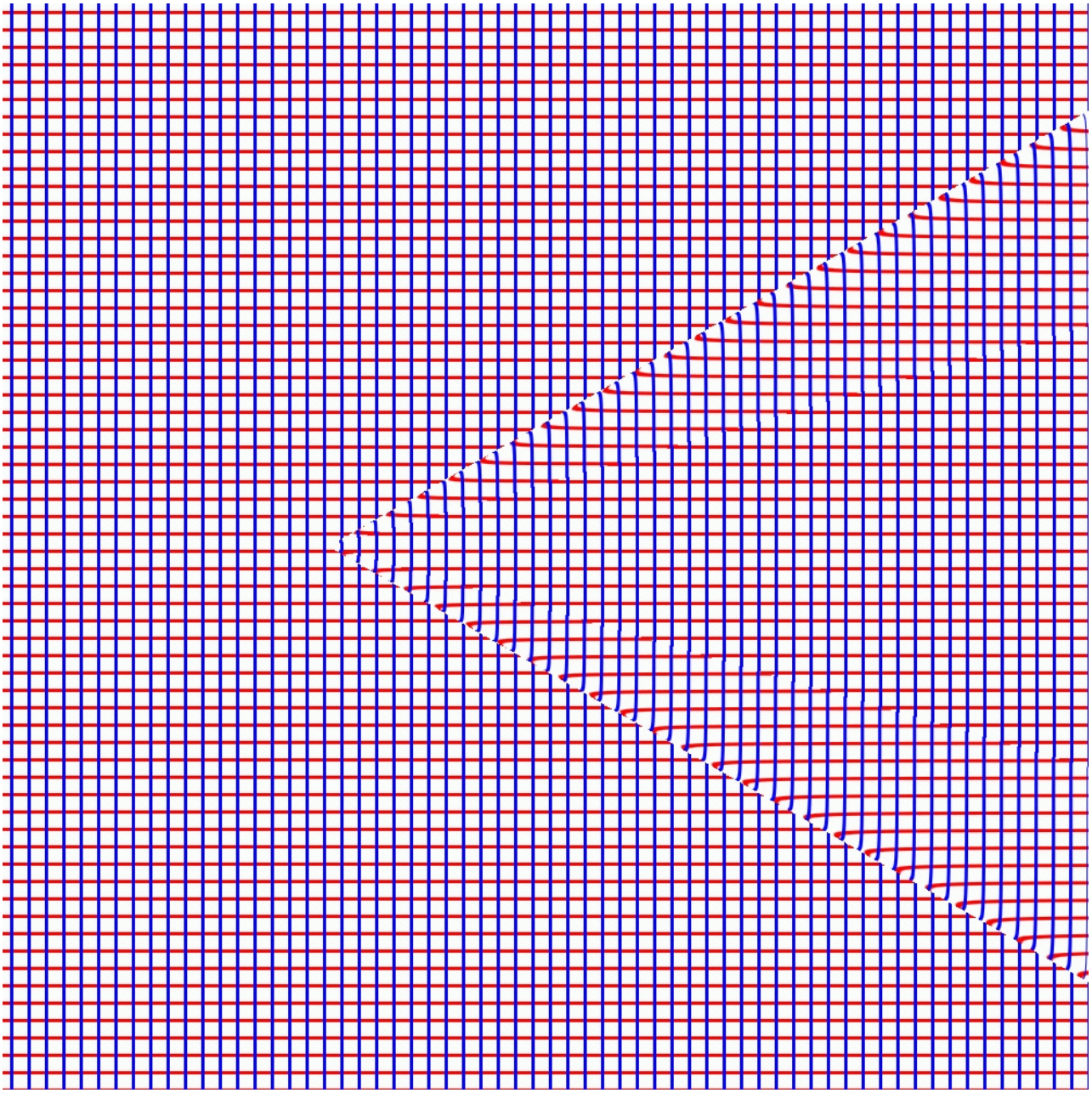


Figure 19: Distorted background image

2.2 Image processing - optical flow

In this section, the theory for the various optical flow algorithms being used in this study is outlined. Table 3 lists the various symbols used in this section.

Table 3: List of symbols for optical flow

Symbol	Description
$\boldsymbol{a}$	Optical flow vector
a	Relative position of left hand edge of interrogation window with respect to the reference location
$\boldsymbol{A}$	Matrix for brightness model in reference image
$\boldsymbol{A}'$	Matrix for brightness model in distorted image
$\boldsymbol{A_k}$	Matrix for brightness model at pixel location k
a_n	Weight n in vector $\boldsymbol{p}$
$\boldsymbol{b}$	Optical flow vector perpendicular to greyscale gradient
b	Relative position of left hand edge of interrogation window with respect to the reference location
$\boldsymbol{b}$	Vector for brightness model in reference image
$\boldsymbol{b}'$	Vector for brightness model in distorted image
$\boldsymbol{c}$	Optical flow vector parallel to greyscale gradient
c	Relative position of top edge of interrogation window with respect to the reference location
c	Scalar for brightness model in reference image
c'	Coefficient for brightness model in distorted image
d	Relative position of right hand edge of interrogation window with respect to the reference location
d	Optical flow component index
i	Lateral position in image
i_0	Lateral position of feature in image 0
I_0	Greyscale value in image 0
i_1	Lateral position of feature in image 1
I_1	Greyscale value in image 1
i_k	Lateral position of pixel k
I_n	Greyscale value in image n
i_r	Lateral component of reference location in image
j	Vertical position in image
j_0	Vertical position of feature in image 0
j_1	Vertical position of feature in image 1
j_k	Vertical position of pixel k in window
j_r	Vertical component of reference location in image
k	Index for pixel location in window
k	Brightness index
n	Number of samples in window
N	Number of pixels in window

$\boldsymbol{p}$	Vector of weights for optical flow model
$\boldsymbol{p}_d^k$	Numerical variable for component d at iteration k
R	Correlation coefficient
$R_{u,v}$	Correlation coefficient with window displacement $\boldsymbol{w} = (u, v)$
s	Scalar variable
$\boldsymbol{S}$	Matrix for optical flow model
$\boldsymbol{S_k}$	Matrix for optical flow model at pixel location k
u	Lateral optical flow component
$\hat{u}$	Estimated lateral component of optical flow
u^n	lateral component of optical flow estimate at iteration n
v	Vertical optical flow component
$\hat{v}$	Estimated vertical component of optical flow
v^n	Vertical component of optical flow estimate at iteration n
$\boldsymbol{w}$	Optical flow vector
w	Brightness component of optical flow
$\boldsymbol{w}^*$	Auxiliary optical flow vector
$\boldsymbol{w}^{*k}$	Auxiliary optical flow vector at iteration k
w^{*k}_d	Auxiliary optical flow vector component d at iteration k
$\boldsymbol{w_0}$	Reference optical flow vector
w_k	Weight for pixel location k
$\boldsymbol{w}^k$	Optical flow extimate at iteration k
$\boldsymbol{w}^n$	Optical flow estimate at iteration n
$\boldsymbol{x}$	Position on image
$\boldsymbol{x_0}$	Position of feature in image 0
$\boldsymbol{x_1}$	Position of feature in image 1
$\boldsymbol{x_k}$	Position of pixel k in window
$\boldsymbol{x_r}$	Reference location in image
μ_0	Mean value in window of image 0
μ_1	Mean value in window of image 1
α	Smoothness Weight
γ	Gradient consistency weight (Brox)
γ	Brightness gradient (TV-L1)
$\Delta\boldsymbol{b}$	Difference between $\boldsymbol{b}$ vectors
$\Delta\boldsymbol{b_k}$	Difference between $\boldsymbol{b}$ vectors at position k
$\boldsymbol{\delta w}$	Iterative change in optical flow vector
ϵ	Error value
$\boldsymbol{\epsilon}$	Error values vector
ϵ	Small number
θ	Numerical parameter
λ	Greyscale consistency weighting
ρ	Linearized optical flow greyscale consistency constraint estimation error
σ_0	Variance of greyscale values in window of image 0
σ_1	Variance of greyscale values in window of image 1
τ	Numerical scheme time step
Ψ	Nonlinear function
Ω	Image domain

Optical flow can be described as the apparent movement in an image sequence [4]. It is a two-dimensional vector field in two-dimensional space. At each $\boldsymbol{x} = (i, j)$ location, there is a $\boldsymbol{w}(\boldsymbol{x}) = (u, v)$ optical flow vector that corresponds to the movement in a sequence of images at a given time. For the BOS application a reference image with no distortion, I_0, and an image that is distorted due to being sampled through the flow field of interest, I_1,are compared. At each $\boldsymbol{x}$ location in the images a vector $\boldsymbol{w}$ can be determined that corresponds to the apparent movement between the two images. This vector field quantifies the distortion of the background image. To illustrate this, a reference image shown in Figure 20 is distorted to create the image in Figure 21. The dot located at $\boldsymbol{x_0} = (i_0, j_0)$, illustrated in Figure 20 has moved by $\boldsymbol{w}(\boldsymbol{x}) = (u, v)$ to $\boldsymbol{x_1} = (i_1, j_1)$, as illustrated in Figure 21. Figure 22 illustrates the relationship between the variables described above. An optical flow algorithm is used to compute the optical flow field.

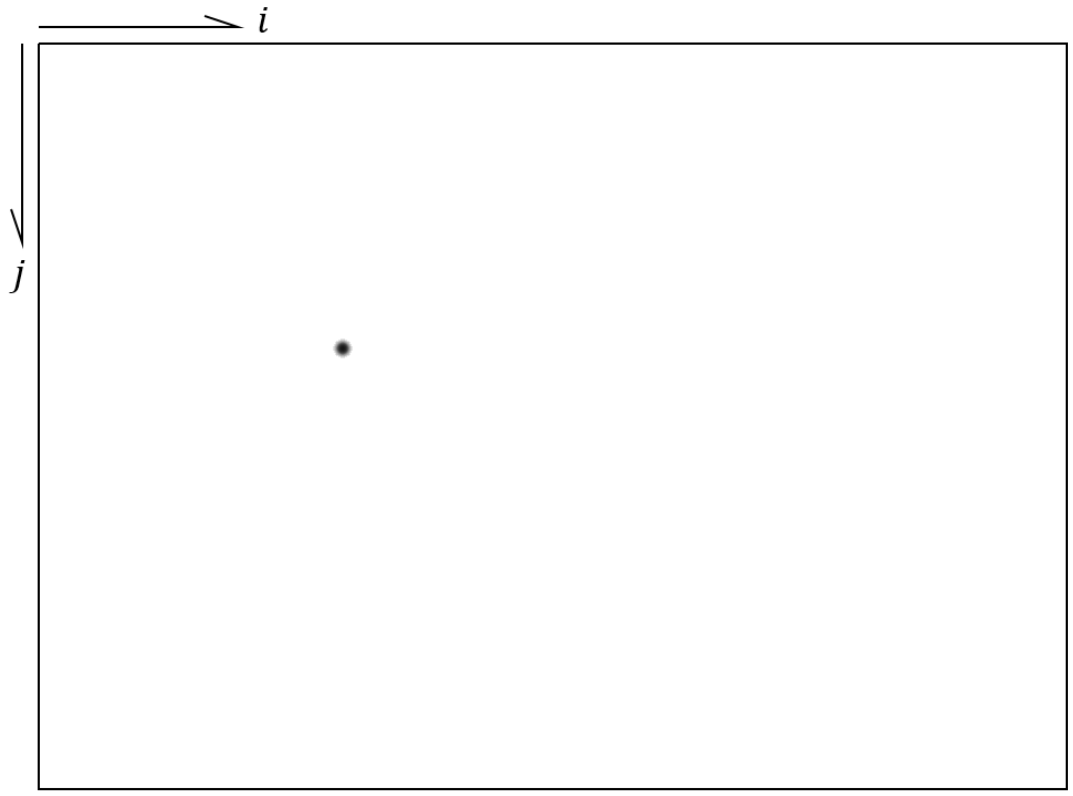


Figure 20: Reference Image, I_0

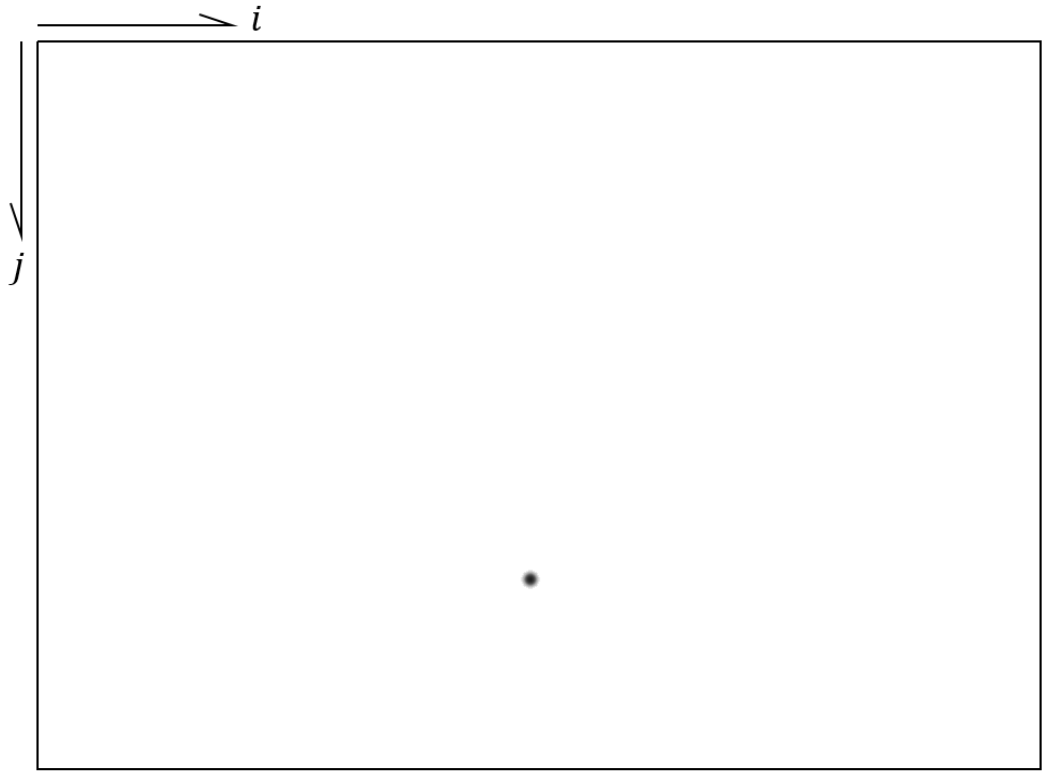


Figure 21: Distorted image, I_1

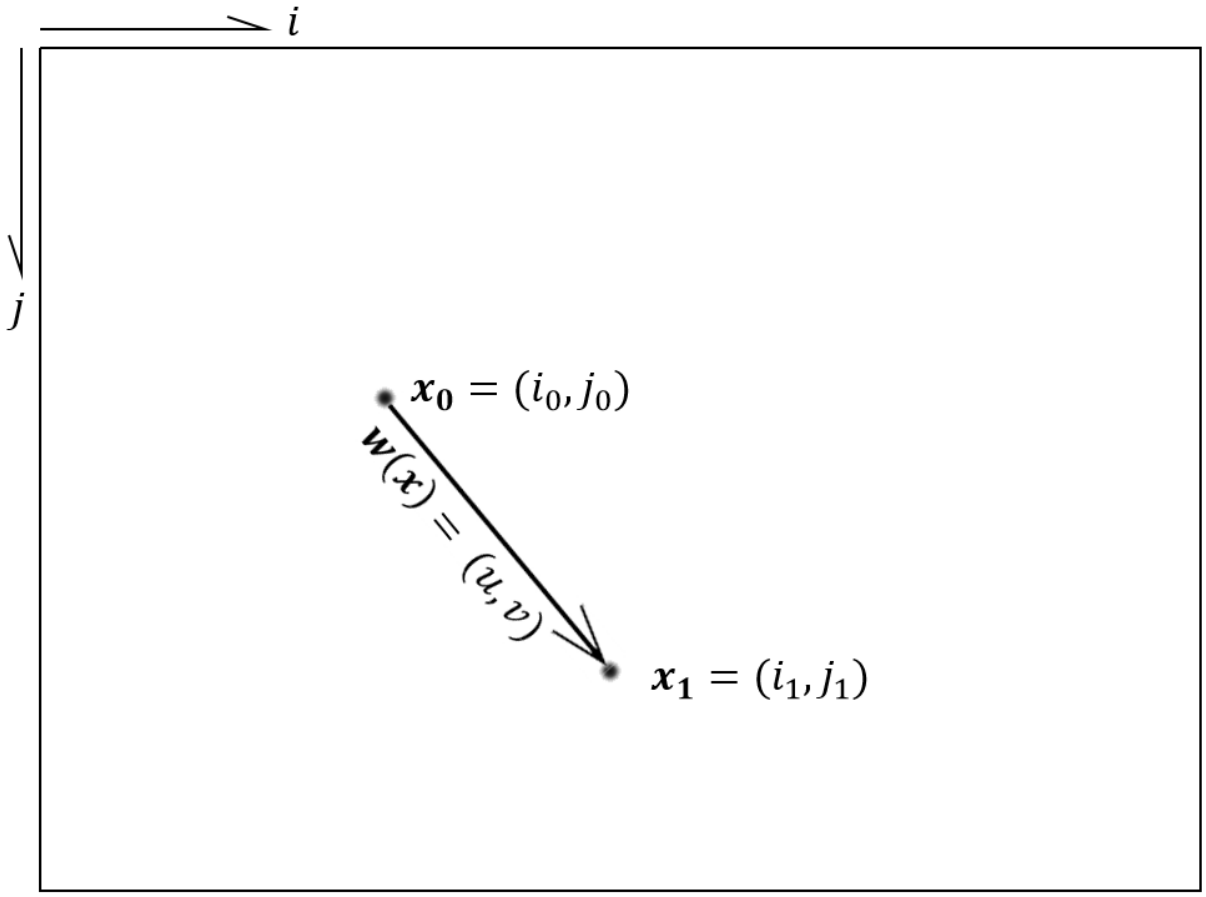


Figure 22: Optical flow definitions

2.2.1 Common optical flow concepts

The brightness consistency assumption

The assumption of brightness consistency is used with many optical flow algorithms [4] [5] [6] [14][1] [15][2]. The brightness consistency assumption assumes that the pixels that are sampled from the same point on an object have the same intensity value at each corresponding point of each sampled image. This is described mathematically in equation 65.

$$I_0\big(\boldsymbol{x} + \boldsymbol{w}(\boldsymbol{x})\big) = I_1(\boldsymbol{x}) \tag{65}$$

Here $\boldsymbol{x} = (i, j)$ is the reference position in the image, $\boldsymbol{w}(\boldsymbol{x}) = (u, v)$ is the optical flow corresponding to the location $\boldsymbol{x}$ and $I_n(\boldsymbol{x})$ returns the brightness value of image n at position $\boldsymbol{x}$.

The aperture problem

Optical flow algorithms work by matching measurements at (i, j) locations between images. The quantity measured is typically greyscale image intensity but does not have to be. At a particular point in an image, the image intensity gradient will point in a particular direction. A change in image intensity can be used to determine the component of the movement in this direction but no information can be gained regarding movement perpendicular to this direction. This is illustrated in Figure 23. Here, vector $\boldsymbol{a}$ illustrates the actual movement, vector $\boldsymbol{b}$ is the component of $\boldsymbol{a}$ perpendicular to the gradient for which no information can be obtained from a change in greyscale value (the grayscale value remains constant), and vector $\boldsymbol{c}$ is the component of $\boldsymbol{a}$ in the direction of the grayscale gradient[3], for which a change in greyscale value can give information. Determining the movement of the image from the change in greyscale value is underdetermined and more information is needed to constrain the problem. This phenomenon is known as the aperture problem.

[1] Lucas and Kanade also make provision for a brightness and contrast adjustment, improving on the brightness consistency assumption.
[2] Farnebäck implicitly assumes signal value consistency between frames, which equates to grayscale consistency in a grayscale image.
[3] As a consequence, $\boldsymbol{a}$ is perpendicular to $\boldsymbol{b}$.

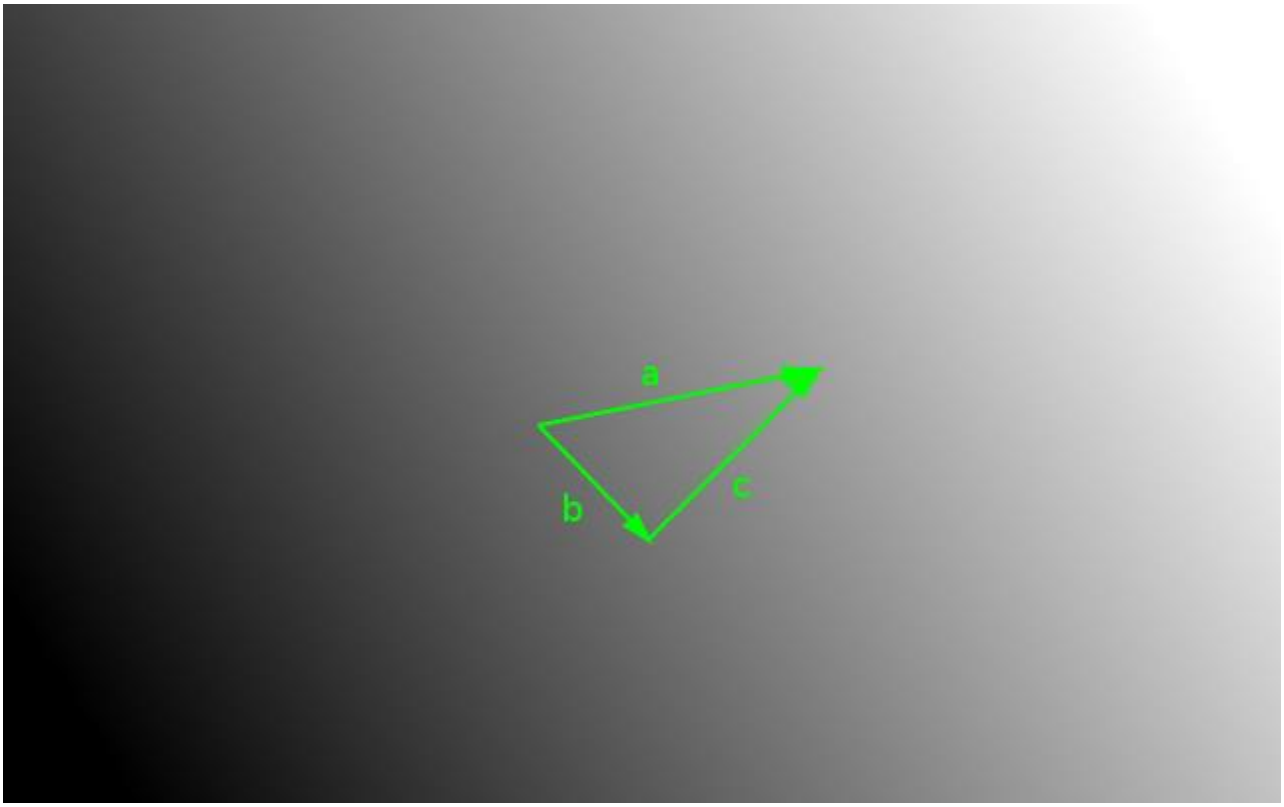


Figure 23: Illustration of the aperture problem

The aperture problem can be solved by imposing additional constraints that incorporate information from points surrounding the point of interest. The way in which optical flow algorithms differ can be related to the way in which they solve the aperture problem.

Coarse to fine iteration

In order to prevent finding local minima, many optical flow algorithms use a coarse to fine image processing strategy [5] [6] [15]. The original image is filtered using a low-pass filter and down sampled. The algorithm of choice is used to process the filtered images. The results from the processing of the filtered images are then used as a starting point for the algorithm in the higher resolution image. By down sampling several times, then solving from the coarsest image to the finest image, the robustness of the algorithm is improved. In this text, we refer to this as multi-resolution processing.

2.2.2 Optical flow algorithms

Various optical flow algorithms are described in the sections that follow. The level of detail of each description has been tailored to understanding the parameters that are used to tune each algorithm. For more details regarding of the algorithms, the reader is referred to the relevant appendix, original

papers or source code for each respective algorithm. This study concentrates on the primary parameters effecting the result while numerical parameters such as those that affect convergence of the algorithms are set pragmatically.

Block-matching algorithms

The most common algorithms used in BOS setups are block-matching algorithms. These algorithms also find wide use in particle image velocimetry (PIV). The canonical method involves brute-force correlation of image intensity values over small regions in the images called windows. Improvements in performance can be made through the use of fast Fourier transforms (FFT), smart stopping rules when computing correlations and other methods, each with associated costs in algorithm complexity and, or solution accuracy.

For this study, a brute-force algorithm [16] was implemented on a GPU in order to find the displacement with the maximum correlation coefficient for each pixel. In order to improve the performance of the GPU some limitations were imposed:

- The search window height had to be a power of two
- The search window width had to be a power of two
- The number of pixels traversed laterally had to be a power of two
- The number of pixels traversed longitudinally had to be a power of two

The theory is outlined below while the details of the algorithm can be found in Appendix A.

For a rectangular window with reference point $\boldsymbol{x_r} = (i_r, j_r)$ with the top left corner $(i_r - a, j_r - b)$ and the bottom right corner $(i_r + c, j_r + d)$, the correlation coefficient is defined according to the equation 66.

$$R(\boldsymbol{x_r}, \boldsymbol{w}) \equiv \sum_{i=i_r-a}^{i_r+c} \sum_{j=j_r-b}^{j_r+d} \frac{(I_0(\boldsymbol{x}) - \mu_0)(I_1(\boldsymbol{x} + \boldsymbol{w}) - \mu_1)}{\sqrt{\sigma_0}\sqrt{\sigma_1}} \tag{66}$$

The parameters required to calculate this coefficient are given below.

$$\mu_0 = \sum_{i=i_r-a}^{i_r+c} \sum_{j=j_r-b}^{j_r+d} \frac{I_0(\boldsymbol{x})}{n} \tag{67}$$

$$\mu_1 = \sum_{i=i_r-a}^{i_r+c} \sum_{j=j_r-b}^{j_r+d} \frac{I_1(\boldsymbol{x}+\boldsymbol{w})}{n} \tag{68}$$

$$\sigma_0 = \sum_{i=i_r-a}^{i_r+c} \sum_{j=j_r-b}^{j_r+d} \frac{(I_0(\boldsymbol{x})-\mu_0)^2}{n-1} \tag{69}$$

The objective is then to find $\boldsymbol{w}(\boldsymbol{x})$ such that $R(\boldsymbol{x},\boldsymbol{w}) = \max_{\boldsymbol{w}} R(\boldsymbol{w})$ for each $\boldsymbol{x}$ location.

In practice, the search is performed at integer values of $\boldsymbol{x}$ and $\boldsymbol{w}$ which produces an optical flow with integer-valued components. By assuming that the correlation coefficient response has a particular form, a sub-pixel interpolation is possible. Those implemented in this study include a Gaussian shaped peak, a parabolic peak, and a method that calculates the weighted centroid of the correlation peak.

For each $\boldsymbol{x}$ location, using the notation that $R_{u,v}$ is the maximum correlation coefficient with u and v in the subscript indicating the position of maximum correlation, the equations for finding these interpolated values of $\widehat{\boldsymbol{w}} = (\hat{u},\hat{v})$ for each method are as follows. [16]

Centroid:

$$\hat{u} = \frac{(u-1)R_{u-1,v} + uR_{u,v} + (u+1)R_{u+1,v}}{R_{u-1,v} + R_{u,v} + R_{u+1,v}} \tag{70}$$

$$\hat{v} = \frac{(v-1)R_{u,v-1} + vR_{u,v} + (v+1)R_{u,v+1}}{R_{u,v+1} + R_{u,v} + R_{u,v+1}} \tag{71}$$

Parabolic:

$$\hat{u} = u + \frac{R_{u-1,v} - R_{u+1,v}}{2R_{u-1,v} + 4R_{u,v} + 2R_{u+1,v}} \tag{72}$$

$$\hat{v} = v + \frac{R_{u,v-1} - R_{u,v+1}}{2R_{u,v+1} + 4R_{u,v} + 2R_{u,v+1}} \tag{73}$$

Gaussian:

$$\hat{u} = u + \frac{\ln R_{u-1,v} - \ln R_{u+1,v}}{2\ln R_{u-1,v} + 4\ln\ R_{u,v} + 2\ln R_{u+1,v}} \tag{74}$$

$$\hat{v} = v + \frac{\ln R_{u,v-1} - \ln R_{u,v+1}}{2\ln R_{u,v+1} + 4\ln R_{u,v} + 2\ln R_{u,v+1}} \tag{75}$$

The parameters used for tuning the algorithm are outlined in Table 4. While it is possible to use rectangular windows, only square windows were used in this study to limit the number of required permutations to a manageable quantity.

Table 4: Parameters used in the block-matching algorithm

Parameter	Effect
w	Window width
h	Window height
u_{min}	Minimum lateral displacement to search
u_{max}	Maximum lateral displacement to search
v_{min}	Minimum vertical displacement to search
v_{max}	Maximum vertical displacement to search
$type$	Sub-pixel interpolation type: • None • Centroid • Parabolic • Gaussian

Lucas and Kanade

Lucas and Kanade [14] use image intensity gradients to direct the search for matching windows by using a least squares approach. A truncated Taylor series expansion of the brightness consistency assumption at $\boldsymbol{x}$ yields the following equation.

$$I_1(\boldsymbol{x}) - I_0(\boldsymbol{x}) + \boldsymbol{w}(\boldsymbol{x}) \cdot \nabla I_0(\boldsymbol{x}) \approx 0 \tag{76}$$

The result is that at each pixel $\boldsymbol{x_k} = (i_k, j_k)$ within the region of interest the following holds true.

$$\boldsymbol{w}(\boldsymbol{x_k}) \cdot \nabla I_0(\boldsymbol{x_k}) = I_0(\boldsymbol{x_k}) - I_1(\boldsymbol{x_k}) + \epsilon(\boldsymbol{x_k}) \tag{77}$$

If it is assumed that $\boldsymbol{w}$ is constant over the region, the following equation results.

$$\begin{bmatrix} \frac{\partial I_0(\boldsymbol{x_1})}{\partial i} & \frac{\partial I_0(\boldsymbol{x_1})}{\partial j} \\ \frac{\partial I_0(\boldsymbol{x_2})}{\partial i} & \frac{\partial I_0(\boldsymbol{x_2})}{\partial j} \\ \vdots & \vdots \\ \frac{\partial I_0(\boldsymbol{x_n})}{\partial i} & \frac{\partial I_0(\boldsymbol{x_n})}{\partial j} \end{bmatrix} \boldsymbol{w} = \begin{bmatrix} I_0(\boldsymbol{x_1}) - I_1(\boldsymbol{x_1}) \\ I_0(\boldsymbol{x_2}) - I_1(\boldsymbol{x_2}) \\ \vdots \\ I_0(\boldsymbol{x_n}) - I_1(\boldsymbol{x_n}) \end{bmatrix} + \boldsymbol{\epsilon} \tag{78}$$

Where $\boldsymbol{\epsilon}$ is the error term that has to be minimised. The vector $\boldsymbol{w}$ has the least squares solution given in equation 79.

$$\begin{bmatrix} u \\ v \end{bmatrix} = \begin{bmatrix} \sum \left(\frac{\partial I_0(\boldsymbol{x_k})}{\partial i}\right)^2 & \sum \frac{\partial I_0(\boldsymbol{x_k})}{\partial i} \frac{\partial I_0(\boldsymbol{x_k})}{\partial j} \\ \sum \frac{\partial I_0(\boldsymbol{x_k})}{\partial i} \frac{\partial I_0(\boldsymbol{x_k})}{\partial j} & \sum \left(\frac{\partial I_0(\boldsymbol{x_k})}{\partial j}\right)^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum \left[\frac{\partial I_0(\boldsymbol{x_k})}{\partial i} (I_0(\boldsymbol{x_k}) - I_1(\boldsymbol{x_k}))\right] \\ \sum \left[\frac{\partial I_0(\boldsymbol{x_k})}{\partial j} (I_0(\boldsymbol{x_k}) - I_1(\boldsymbol{x_k}))\right] \end{bmatrix} \tag{79}$$

Nonlinearities can be catered for by iterating the solution above from each new estimate. If the linearization is performed about the newest estimation of the vector $\boldsymbol{w}^n = \begin{bmatrix} u^n \\ v^n \end{bmatrix}$, the next step in the iteration can be solved . The procedure is shown in the three equations that follow, defining A and B to aid in making the text more readable.

$$A = \begin{bmatrix} \sum \left(\frac{\partial I_0(\boldsymbol{x_k} + \boldsymbol{w}^n)}{\partial i}\right)^2 & \sum \frac{\partial I_0(\boldsymbol{x_k} + \boldsymbol{w}^n)}{\partial i} \frac{\partial I_0(\boldsymbol{x_k} + \boldsymbol{w}^n)}{\partial j} \\ \sum \frac{\partial I_0(\boldsymbol{x_k} + \boldsymbol{w}^n)}{\partial i} \frac{\partial I_0(\boldsymbol{x_k} + \boldsymbol{w}^n)}{\partial j} & \sum \left(\frac{\partial I_0(\boldsymbol{x_k} + \boldsymbol{w}^n)}{\partial j}\right)^2 \end{bmatrix} \tag{80}$$

$$B = \begin{bmatrix} \sum \left[\frac{\partial I_0(\boldsymbol{x_k} + \boldsymbol{w}^n)}{\partial i} (I_0(\boldsymbol{x_k} + \boldsymbol{w}^n) - I_1(\boldsymbol{x_k}))\right] \\ \sum \left[\frac{\partial I_0(\boldsymbol{x_k} + \boldsymbol{w}^n)}{\partial j} (I_0(\boldsymbol{x_k} + \boldsymbol{w}^n) - I_1(\boldsymbol{x_k}))\right] \end{bmatrix} \tag{81}$$

$$\boldsymbol{w}^{n+1} = \boldsymbol{w}^n + [A]^{-1}[B] \tag{82}$$

The parameters used to tune this algorithm are outlined in Table 5.

Table 5: Parameters used to tune the Lucas and Kanade algorithm

Parameter	Use
w	Window width
h	Window height
n_l	Levels
n_{its}	Iterations

Brox

Brox *et al.* [5] use the following assumptions.

Brightness Consistency

$$I_0(\boldsymbol{x}) = I_1(\boldsymbol{x} + \boldsymbol{w}) \tag{83}$$

Gradient Consistency

$$\nabla I_0(\boldsymbol{x}) = \nabla I_1(\boldsymbol{x} + \boldsymbol{w}) \tag{84}$$

Smooth Flow

$$|\nabla u|^2 + |\nabla v|^2 = 0 \tag{85}$$

These assumptions are combined with weights into a cost function as which has the following result.

$$\int \psi(|I_0(\boldsymbol{x}) - I_1(\boldsymbol{x} + \boldsymbol{w})|^2 + \gamma|\nabla I_0(\boldsymbol{x}) - \nabla I_1(\boldsymbol{x} + \boldsymbol{w})|^2) + \alpha\psi(|\nabla u|^2 + |\nabla v|^2)\, d\Omega \tag{86}$$

Here $\psi(s^2) = \sqrt{s^2 + \epsilon^2}$ is used to prevent outliers from having a large influence on the results while the addition of ϵ^2 is a small positive number that prevents the singularity in the derivative of $\psi(s^2)$ when $s = 0$. Brox chose $\epsilon = 0.001$ arbitrarily.

The minimisation problem is solved sing the calculus of variations and then solving the resulting equations numerically. Two levels of fixed-point iteration are used, resulting in an inner loop and an outer loop. The reader is referred to the original paper [5] for details. Table 6 shows the parameters used to tune the algorithm.

Table 6: Description of parameters used in the Brox algorithm

Parameter	Description
α	Weighting assigned to the regularisation term
γ	Weighting assigned to the gradient consistency portion of the optical flow constraint term
n	Number of inner loop iterations
m	Number of outer loop iterations
l	Number of layers

Farnebäck

Farnebäck's algorithm [15] works by approximating the image brightness values with a second order polynomial approximation as well as approximating the flow in a local region with a model that is linear in its coefficients. In other words, locally the brightness in the reference image is approximated by equation 87, while the brightness in the distorted image is approximated by equation 88.

$$\boldsymbol{x}^T\boldsymbol{A}\boldsymbol{x} + \boldsymbol{x}^T\boldsymbol{b} + \boldsymbol{c} = 0 \tag{87}$$

$$(\boldsymbol{x} + \boldsymbol{w})^T\boldsymbol{A}(\boldsymbol{x} + \boldsymbol{w}) + (\boldsymbol{x} + \boldsymbol{w})^T\boldsymbol{b} + c = 0 \tag{88}$$

In the simplest case, $\boldsymbol{w}$ is constant, but more generally is approximate by the equation that follows.

$$\boldsymbol{w} = \boldsymbol{S}\boldsymbol{p} \tag{89}$$

The result is that in the distorted image, the brightness is approximated by the following equation.

$$(\boldsymbol{x} + \boldsymbol{S}\boldsymbol{p})^T\boldsymbol{A}(\boldsymbol{x} + \boldsymbol{S}\boldsymbol{p}) + (\boldsymbol{x} + \boldsymbol{S}\boldsymbol{p})^T\boldsymbol{b} + \boldsymbol{c} = 0 \tag{90}$$

Two options for $\boldsymbol{S}$ and $\boldsymbol{p}$ include constant optical flow and the eight-parameter model as follows.

Constant flow

$$\boldsymbol{S} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \tag{91}$$

$$\boldsymbol{p} = \begin{bmatrix} u \\ v \end{bmatrix} \tag{92}$$

Eight Parameter Model

$$\boldsymbol{S} = \begin{bmatrix} 1 & i & j & 0 & 0 & 0 & i^2 & ij \\ 0 & 0 & 0 & 1 & i & j & ij & j^2 \end{bmatrix} \tag{93}$$

$$\boldsymbol{p} = \begin{bmatrix} a_1 \\ a_2 \\ a_3 \\ a_4 \\ a_5 \\ a_6 \\ a_7 \\ a_8 \end{bmatrix} \tag{94}$$

OpenCV uses the eight-parameter model.

With this mathematical model, the local displacement field can be solved for as shown below

$$(\boldsymbol{x} + \boldsymbol{Sp})^T \boldsymbol{A}(\boldsymbol{x} + \boldsymbol{Sp}) + (\boldsymbol{x} + \boldsymbol{Sp})^T \boldsymbol{b} + c = 0 \tag{95}$$

$$\boldsymbol{x}^T \boldsymbol{Ax} + \boldsymbol{p}^T \boldsymbol{S}^T \boldsymbol{Ax} + \boldsymbol{x}^T \boldsymbol{ASp} + \boldsymbol{x}^T \boldsymbol{b} + \boldsymbol{p}^T \boldsymbol{Sb} + c = 0 \tag{96}$$

$$\boldsymbol{x}^T \boldsymbol{Ax} + \boldsymbol{x}^T (2\boldsymbol{ASp} + \boldsymbol{b}) + \boldsymbol{p}^T \boldsymbol{Sb} + c = 0 \tag{97}$$

Equation 97 can be written in the following form.

$$\boldsymbol{x}^T \boldsymbol{A'x} + \boldsymbol{x}^T \boldsymbol{b'} + c' = 0 \tag{98}$$

The parameters used in the equation are given below.

$$A' = A \tag{99}$$

$$b' = 2ASp + b \tag{100}$$

$$c' = p^T Sb + c \tag{101}$$

The displacement parameters can then be solved from the fits of the two equations by rearranging equation 100. The vector p can be solved for each point, however this gives rise to noisy results and so, Farnebäck minimised the following weighted average over a region.

$$\sum_k w_k [2\boldsymbol{A_k S_k p} - \Delta \boldsymbol{b_k}]^2 \tag{102}$$

The parameter $\Delta \boldsymbol{b}$ is defined as follows.

$$\Delta \boldsymbol{b} = \boldsymbol{b} - \boldsymbol{b}' \tag{103}$$

This minimisation has the following result.

$$p = \left(\sum_i w_k \boldsymbol{S_k^T A_k^T A_k S_k} \right)^{-1} \sum_k w_k \boldsymbol{S_k^T A_k^T \Delta b_k} \tag{104}$$

The OpenCV library implements the 8-parameter model above with either constant or Gaussian weights. The parameters used to tune the model are shown in Table 7.

Table 7: Parameters used in the Farnebäck algorithm

Parameter	Description
$Scale$	Scaling factor for down sampling
$Size$	Size of window used for fitting greyscale polynomial.
σ	The standard deviation of the Gaussian used to smooth the values used in the calculation of derivatives for the polynomial expansion
Its	Number of iterations
$Levels$	Number of levels used in multi-resolution processing
$Use\ Fast\ Pyramids$	A flag used to improve processing efficiency for Scale values of 0.5
$Use\ Initial\ Flow$	A flag used to set the iterations to start from the values in the initialised optical flow images
$Use\ Gaussian\ Filter$	A flag used to set the use of a Gaussian filter instead of a box filter to set the weights in the optical flow estimation.

TV-L1

The total variation algorithm by Zach *et al.* [6] is based on that of Horn and Schunck [4] but instead of using an L2 norm, it uses an L1 norm. This allows for piecewise continuous optical flow whereas that of Horn and Schunck only allows for smooth flow. The algorithm implemented in OpenCV extends the algorithm that was published in the original paper by Zach *et al.* by adding an illumination variation term. First, the original formulation is discussed. The original minimisation problem is given in the following equaiotn.

$$\min_{\boldsymbol{w}} \int (\lambda |I_0(\boldsymbol{x}) - I_1(\boldsymbol{x}+\boldsymbol{w})| + |\nabla \boldsymbol{w}|)\, d\Omega \tag{105}$$

In order to make the minimisation problem more tractable, an auxiliary variable $\boldsymbol{w}^*$ is introduced to the functional and the minimisation problem is approximated according to the following equation.

$$\min_{\boldsymbol{w},\boldsymbol{w}^*} \int \left(\lambda |I_0(\boldsymbol{x}) - I_1(\boldsymbol{x}+\boldsymbol{w})| + \frac{1}{2\theta}(\boldsymbol{w}-\boldsymbol{w}^*)^2 + |\nabla \boldsymbol{w}^*|\right) d\Omega \tag{106}$$

Setting θ to a small value drives $\boldsymbol{w}$ and $\boldsymbol{w}^*$ toward each other. This allows the problem to be separated into two parts, which can be solved iteratively by alternating between solving for $\boldsymbol{w}$ and the solving for $\boldsymbol{w}^*$ while holding the other constant. These two cost functions are given below.

$$\min_{\boldsymbol{w}^*} \int \left(\frac{1}{2\theta}(\boldsymbol{w}-\boldsymbol{w}^*)^2 + |\nabla \boldsymbol{w}^*|\right) d\Omega \tag{107}$$

$$\min_{\boldsymbol{w}} \int \left(\lambda |I_0(\boldsymbol{x}) - I_1(\boldsymbol{x}+\boldsymbol{w})| + \frac{1}{2\theta}(\boldsymbol{w}-\boldsymbol{w}^*)^2\right) d\Omega \tag{108}$$

For the details of the numerical scheme used to solve the minimisation problems the reader is referred to the original paper by Zach *et al.* [6]. Only the results are shown here

The first minimisation problem can be solved with the following fixed-point iteration scheme.

$$\boldsymbol{p}_d^{k+1} = \boldsymbol{p}_d^k + \frac{\frac{\tau}{\theta}\nabla\left({w^*}_d^{k+1} + \theta div(\boldsymbol{p}_d^k)\right)}{1 + \frac{\tau}{\theta}\left|\nabla\left({w^*}_d^{k+1} + \theta div(\boldsymbol{p}_d^k)\right)\right|}, d \in \{1,2\} \tag{109}$$

Here τ is the time step in the numerical scheme. Convergence has been proven for $\tau \leq 0.125$.

The variable $\boldsymbol{p}$ is introduced to aid the numerical solution and is solved for first, after which $\boldsymbol{w}$ is recovered from the following equation.

$$w_d^{k+1} = w^*{}_d^{k+1} + \theta div\left(\boldsymbol{p_d^k}\right), d \in \{1,2\} \tag{110}$$

The second minimisation problem is solved by approximating the penalty term, $|I_0(\boldsymbol{x}) - I_1(\boldsymbol{x}+\boldsymbol{w})|$, by using a truncated Taylor series about $\boldsymbol{w_0}$ with $\boldsymbol{w} = \boldsymbol{w_0} + \boldsymbol{\delta w}$.

$$I_1(\boldsymbol{x}+\boldsymbol{w}) = I_1(\boldsymbol{x}+\boldsymbol{w_0}+\boldsymbol{\delta w}) \tag{111}$$

$$I_1(\boldsymbol{x}+\boldsymbol{w_0}+\boldsymbol{\delta w}) \approx \nabla \mathrm{I}_1(\boldsymbol{x}+\boldsymbol{w_0}) \cdot (\boldsymbol{\delta w}) + \mathrm{I}_1(\boldsymbol{x}+\boldsymbol{w_0}) \tag{112}$$

The approximation to the term $I_1(\boldsymbol{x}+\boldsymbol{w})$- $I_0(\boldsymbol{x})$ is then approximated by the equation that follows.

$$I_1(\boldsymbol{x}+\boldsymbol{w}) - I_0(\boldsymbol{x}) \approx \nabla \mathrm{I}_1(\boldsymbol{x}+\boldsymbol{w_0}) \cdot (\boldsymbol{\delta w}) + \mathrm{I}_1(\boldsymbol{x}+\boldsymbol{w_0}) - \mathrm{I}_0(\boldsymbol{x}) \tag{113}$$

The notation $\rho(\boldsymbol{w}, \boldsymbol{w_0})$is adopted for this term, resulting in the following equation.

$$\rho(\boldsymbol{w}, \boldsymbol{w_0}) = \nabla \mathrm{I}_1(\boldsymbol{x}+\boldsymbol{w_0}) \cdot (\boldsymbol{\delta w}) + \mathrm{I}_1(\boldsymbol{x}+\boldsymbol{w_0}) - \mathrm{I}_0(\boldsymbol{x}) \tag{114}$$

The following iterative scheme is then used to solve the second optimisation problem.

$$\boldsymbol{w}^{*k+1} = \begin{cases} \boldsymbol{w}^{k+1} + \lambda\theta\nabla I_1(\boldsymbol{x}+\boldsymbol{w_0}), \ if \rho(\boldsymbol{w}, \boldsymbol{w_0}) < -\lambda\theta|\nabla I_1(\boldsymbol{x}+\boldsymbol{w_0})|^2 \\ \boldsymbol{w}^{k+1} - \lambda\theta\nabla I_1(\boldsymbol{x}+\boldsymbol{w_0}), \ if \rho(\boldsymbol{w}, \boldsymbol{w_0}) > \lambda\theta|\nabla I_1(\boldsymbol{x}+\boldsymbol{w_0})|^2 \\ \boldsymbol{w}^{k+1} - \rho(\boldsymbol{w}, \boldsymbol{w_0}) \frac{\nabla I_1(\boldsymbol{x}+\boldsymbol{w_0})}{|\nabla I_1(\boldsymbol{x}+\boldsymbol{w_0})|^2}, \ if |\rho(\boldsymbol{w}, \boldsymbol{w_0})| \leq \lambda\theta|\nabla I_1(\boldsymbol{x}+\boldsymbol{w_0})|^2 \end{cases} \tag{115}$$

The method above is as originally reported in the paper [6] by Zach *et al.* The algorithm implemented in OpenCV uses an illumination variation term. Inspection of the source code reveals am addition to the original algorithm. This addition is outlined below.

The initial problem was posed as follows.

$$\min_{u} \int (\lambda |I_0(\boldsymbol{x}) - I_1(\boldsymbol{x} + \boldsymbol{w})| + |\nabla \boldsymbol{w}|)\, d\Omega \tag{116}$$

The following variables were defined.

$$\boldsymbol{w} = \begin{bmatrix} u \\ v \end{bmatrix} \tag{117}$$

$$\boldsymbol{x} = \begin{bmatrix} i \\ j \end{bmatrix} \tag{118}$$

It is also possible to define these variables as follows.

$$\boldsymbol{w} = \begin{bmatrix} u \\ v \\ w \end{bmatrix} \tag{119}$$

$$\boldsymbol{x} = \begin{bmatrix} i \\ j \\ k \end{bmatrix} \tag{120}$$

Here, the third dimension is conceived as a brightness variation term. It should be noted here that this extra dimension is additional to the dimension in which the image frames are said to change (the time dimension in a video sequence of images.) The result is to allow for an extra degree of freedom that allows for brightness changes that are not due to image movement. The inclusion of these brightness changes in the smoothness term, $|\nabla u|$, penalises high frequency brightness changes between frames. Brightness changes in a BOS setup could arise from global illumination changes or noise in the brightness measurements but the latter would tend not to be smooth.

The gradient ∇I_1 requires an extension to include the brightness function.

$$\nabla I_1 = \begin{bmatrix} \frac{\partial I_1}{\partial i} \\ \frac{\partial I_1}{\partial j} \\ \frac{\partial I_1}{\partial k} \end{bmatrix} \tag{121}$$

The gradient $\frac{\partial I_1}{\partial z}$ is defined to be a set valueγ

$$\nabla I_1 \equiv \begin{bmatrix} \frac{\partial I_1}{\partial i} \\ \frac{\partial I_1}{\partial j} \\ \gamma \end{bmatrix} \tag{122}$$

The optical flow constraint, $|I_0(\boldsymbol{x}) - I_1(\boldsymbol{x} + \boldsymbol{u})|$, when linearized appears as shown in the following.

$$|I_0(\boldsymbol{x}) - \nabla \mathrm{I}_1(\boldsymbol{x} + \boldsymbol{w_0}) \cdot (\boldsymbol{\delta w}) - \mathrm{I}_1(\boldsymbol{x} + \boldsymbol{w_0})| \tag{123}$$

With the newly defined parameters, the following optical flow constraint is the result.

$$\left| I_0(\boldsymbol{x}) - \mathrm{I}_1(\boldsymbol{x} + \boldsymbol{w_0}) - \left(\frac{\partial \mathrm{I}_1(\boldsymbol{x} + \boldsymbol{w_0})}{\partial i} \cdot \delta u + \frac{\partial \mathrm{I}_1(\boldsymbol{x} + \boldsymbol{w_0})}{\partial j} \cdot \delta v + \gamma \cdot \delta w \right) \right| \tag{124}$$

So that γ is seen as a weighting factor that increases the sensitivity of the cost function to brightness changes and will therefore reduce the allowable brightness changes in the solution. It is noted here that when γ is set to 0 in the OpenCV implementation of the algorithm, the original problem is solved. (ie, w is excluded from the smoothness constraint)

The study of the OpenCV extension to the TV-L1 algorithm is left for future work. The parameters used to tune this algorithm are set in Table 8.

Table 8: Parameters use to tune the TV-L1 algorithm

Parameter	Use
λ	Weights the grey value consistency term
θ	Couples the regularisation term and grey value consistency term. (smaller values create a more tightly couples system)
τ	The time-step of the numerical scheme. Convergence has been proved for values ≤ 0.125 [6]
γ	The derivative of I in the third direction *"coefficient for additional illumination variation term"*

2.3 Background design

Two backgrounds were used, one speckled background and one b-spline wavelet background. It was thought that the speckled background with high frequency content and discontinuous greyscale values would be better suited to block-matching algorithms while the b-spline wavelet noise backgrounds with continuous gradients would be better suited to the algorithms that require greyscale gradients to be present in the images. Both backgrounds were tested and analysed with each of the algorithms with unexpected results.

2.3.1 Speckled backgrounds

Speckled backgrounds are created by randomly colouring background pixels to be either black or white. This creates a high contrast image with high frequency content. It has been shown that the optimal fill (the ratio of black squares to white squares) is between 30 % and 70 % [17].

For this study speckled backgrounds with a fill of 50% and square speckles were chosen. Figure 24 gives an example of such a background pattern.

To generate this pattern, an array of squares is produced. Each square is assigned a random number between 0 and 1. If the random number is greater than 0.5, the square is coloured white otherwise the square is coloured black.

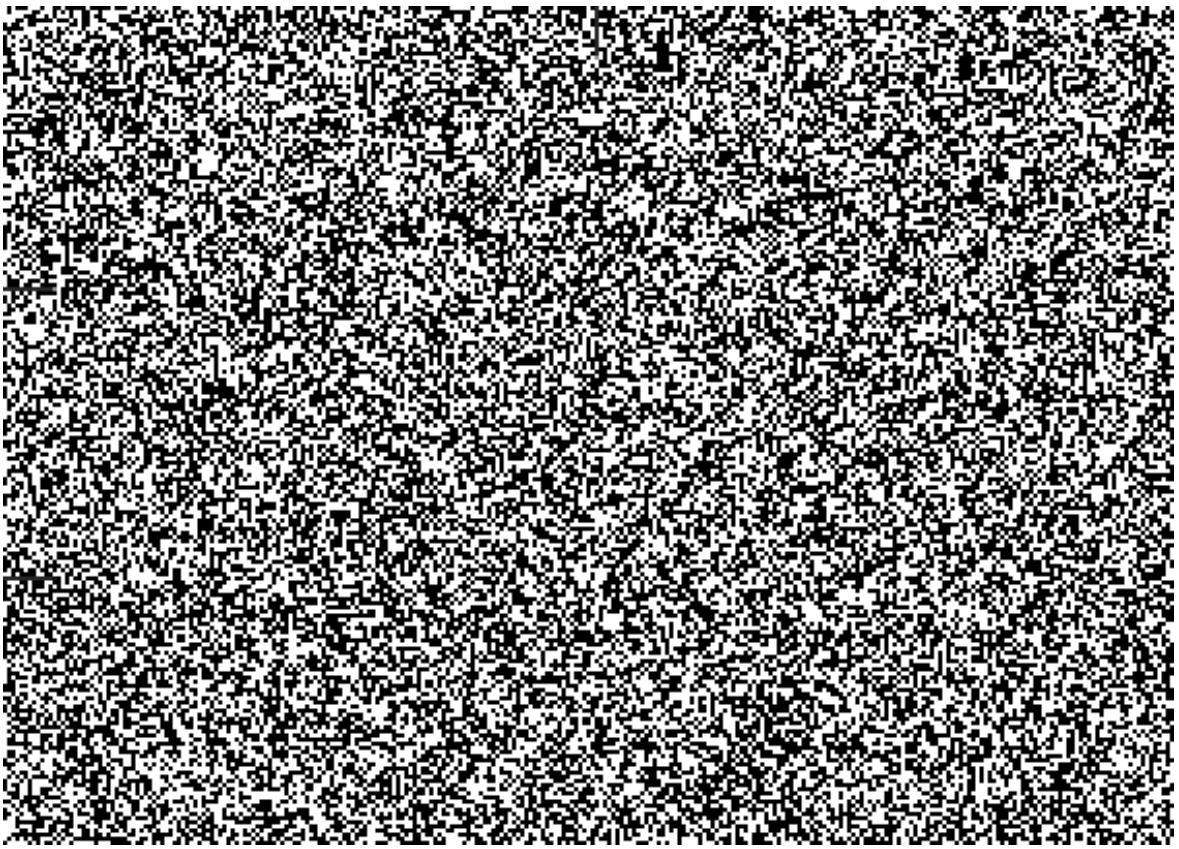


Figure 24: An example of speckled background noise

2.3.2 Wavelet noise backgrounds

B-Spline wavelet noise was developed by Pixar [18] in order to prevent aliasing of textures used in computer-animated films. When random noise is down-sampled, the distribution of the pixel intensities changes causing a reduction in contrast. Wavelet noise is designed to have frequency content at many scales and is resilient to this reduction in contrast [2]. Optical flow algorithms benefit from images of higher contrast. This makes the ability to retain contrast when down-sampled a desirable property in background images that are to be used with optical flow algorithms that employ multi-scale approaches. An example of b-spline wavelet noise is shown in Figure 25.

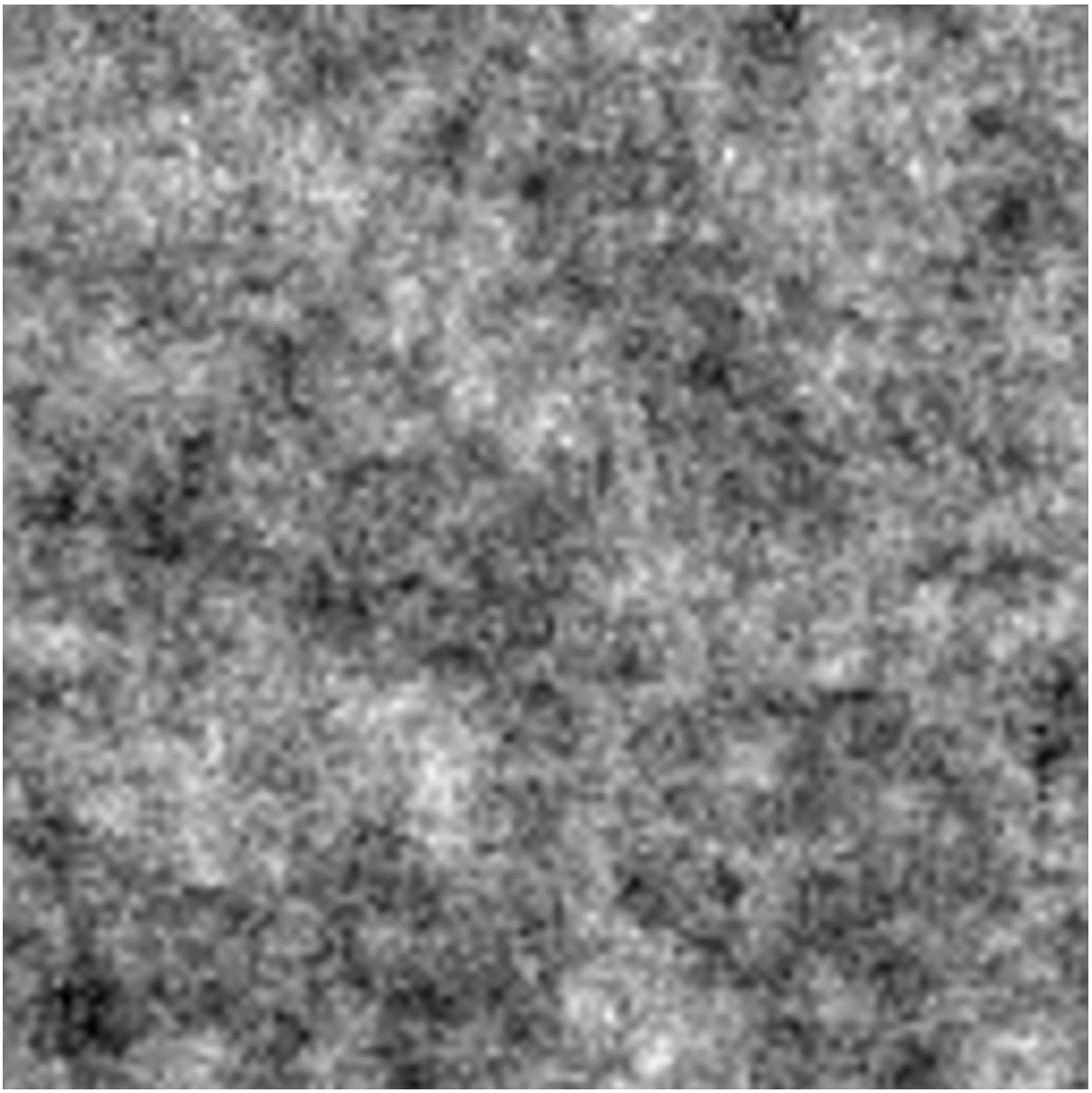


Figure 25: An example of b-spline wavelet noise

2.3.3 Synthesis of b-spline wavelet noise

B-Spline wavelet noise is produced in the following way:

- A texture tile is produced by using b-spline basis functions with random weights.
- A least squares fit is used to down sample the b-spline noise, first in the lateral and then in the vertical direction.
- The tile of the down-sampled noise is subtracted from the original noise tile.
- What is left is a tile with band limited frequency content.
- Tiles produced at varying scales can be added together and used to produce noise with detail in a controlled frequency band.

The frequency content of a single tile can be seen in the power spectral density image of one layer of wavelet noise shown in Figure 33.

The process for creating b-spline wavelet noise will first be shown in one dimension and then extended to two dimensions.

Wavelet noise in one dimension

A b-spline of order k is defined as the linear combination of each of the J basis functions of order k as follows.

$$S_k(t) = \sum_{j}^{J} \alpha_j B_{j,k}(t) \tag{125}$$

Each basis function is a linear interpolation of the two adjacent basis functions of order k-1 over the k knot spans, where a knot span is defined as the span from from t_j to t_{j+1}, where each t_j is a fixed value. The interpolation is, according to the Cox-de Boor recursion formula, as follows.

$$B_{j,k}(t) = \left(\frac{t - t_j}{t_{j+k} - t_j}\right) B_{j,k-1}(t) + \left(\frac{t_{j+k+1} - t}{t_{j+k+1} - t_{j+1}}\right) B_{j+1,k-1}(t) \tag{126}$$

$$B_{j,1}(t) = \begin{cases} 1 & t_j < t \leq t_{j+1} \\ 0 & Otherwise \end{cases} \tag{127}$$

Figure 26shows how four weighted basis functions of degree 3 with fixed knot intervals (dotted lines) are added together to form the spline function (solid red line) which approximates a parabola (solid black line). In this case, the spline and the parabola are exactly equal between x-values of -2 and 2. If more basis functions are added to the spline, the range where the parabola is modelled can be extended.

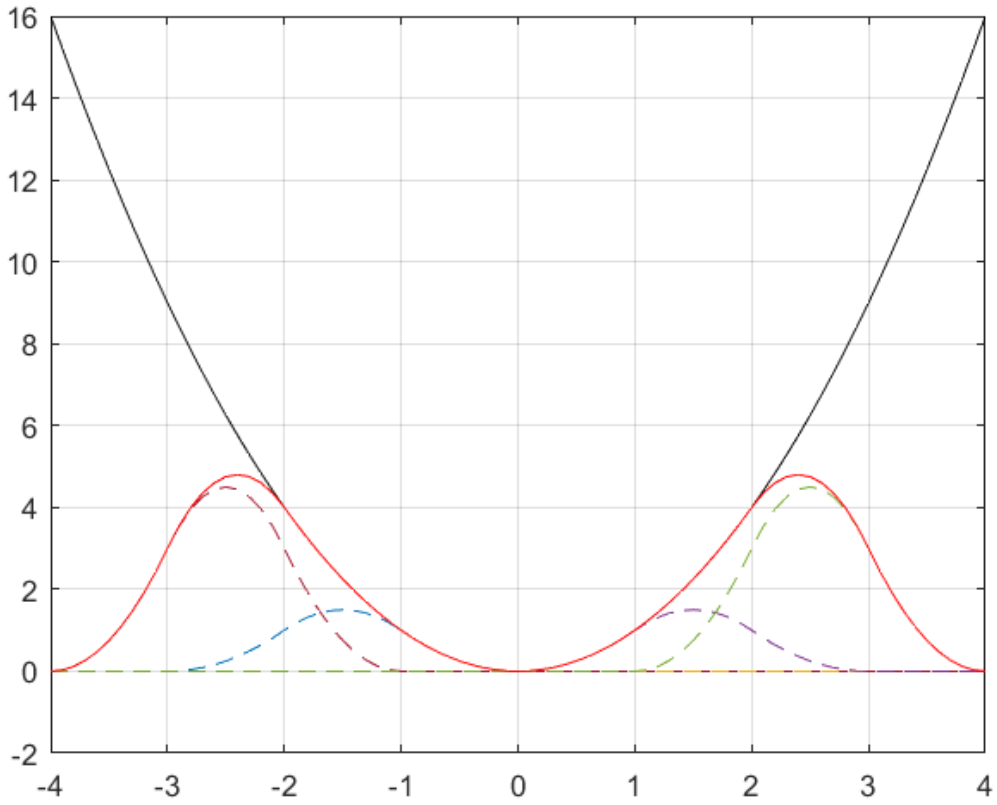


Figure 26: Illustration of weighted spline basis functions approximating a parabola

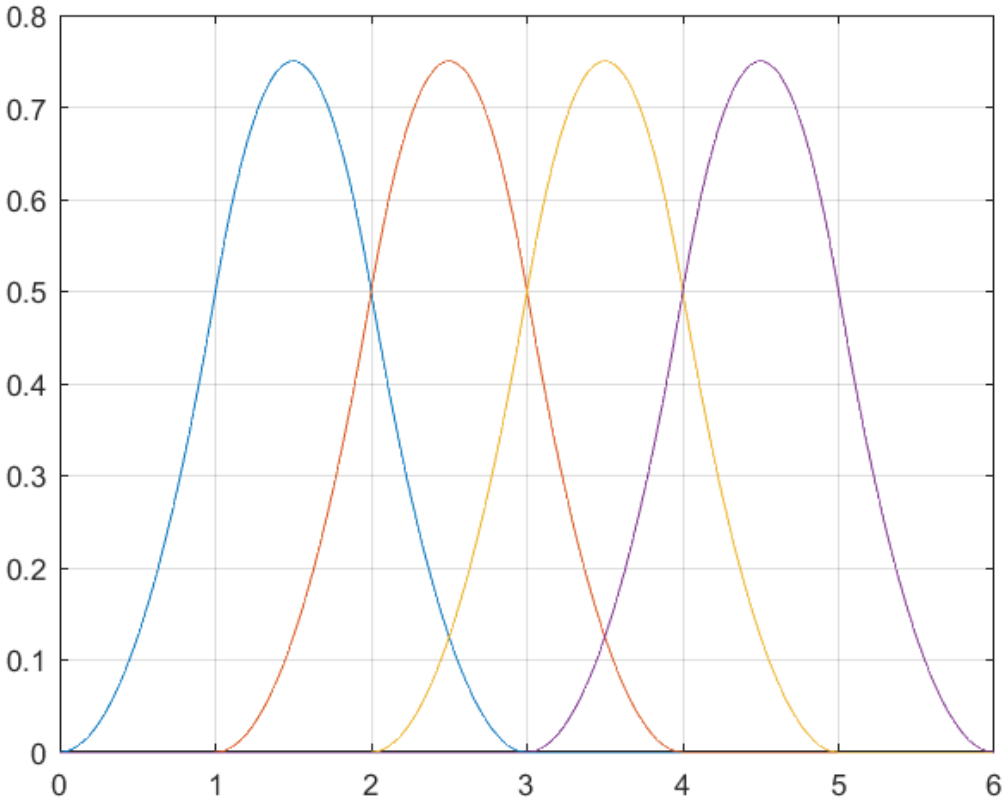


Figure 27: Four third order b-spline basis functions

One property of b-splines is that they may be refined by using a basis with smaller knot spans. An example of the refinement of a basis function is shown in Figure 28where four b-splines of order 3 and knot spans of $\frac{1}{2}$ are added together to form one basis function of order 3 and a knot span of 1. It can be seen that the finer scale spline can be used to create any of the coarser scale spline functions but the coarser scale spline can only approximate a finer scale spline in the general case. The portion of the spline that lies purely in the fine scale, which cannot be approximated by the coarse scale spline, is said to lie in the wavelet space of the spline [18]. B-spline wavelet noise consists of b-splines with randomly weighted basis functions that lie in the wavelet space.

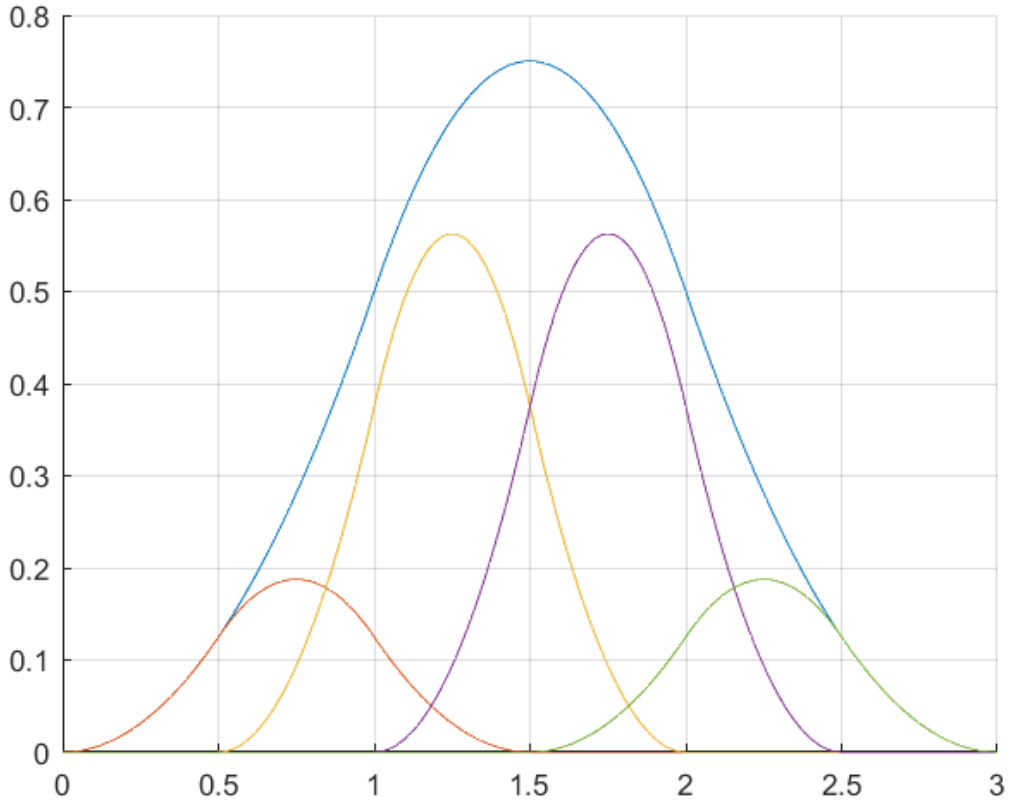


Figure 28: Illustration of basis function refinement

For the creation of b-spline wavelet noise, fixed knot intervals are used. The space between knots differs by a factor of two at each frequency level.

The initial spline is created by using random weights for a spline with uniformly spaced knots. Figure 29 illustrates an example of the initial random b-spline with a striped blue line.

Down-sampling is performed by using a linear least squares fit of the lower frequency basis functions to sampled data from the higher frequency spline function. The weights of the lower frequency spline function are calculated as follows.

$$\boldsymbol{\alpha} = [X'X]^{-1}X'\boldsymbol{y} \tag{128}$$

Here, X is a matrix that is constructed with the entries $B_{t,k}(t_s)$, with t_s indicating the value of the s^{th} sample of parameter, t.

$$X = \begin{bmatrix} B_{0,k}(t_0) & B_{0,k}(t_1) & \cdots & B_{T,k}(t_S) \\ B_{1,k}(t_0) & B_{1,k}(t_1) & \cdots & B_{T,k}(t_S) \\ \vdots & \vdots & \ddots & \vdots \\ B_{T,k}(t_0) & B_{T,k}(t_1) & \cdots & B_{T,k}(t_S) \end{bmatrix} \quad (129)$$

The definition of y, which is a column vector consisting of the values of the higher frequency spline function, is shown below.

$$y = \begin{bmatrix} S_k^{\uparrow}(t_0) \\ S_k^{\uparrow}(t_1) \\ \vdots \\ S_k^{\uparrow}(t_S) \end{bmatrix} \quad (130)$$

Figure 29 shows down-sampled b-spline with a dotted black line. The final wavelet noise spline is produced by subtracting the down-sampled spline from the initial spline. Figure 29 illustrates the b-spline wavelet noise as a solid red line.

Note: New weights can also be found for the down-sampled spline in the higher frequency space and these weights subtracted instead of the spline values.

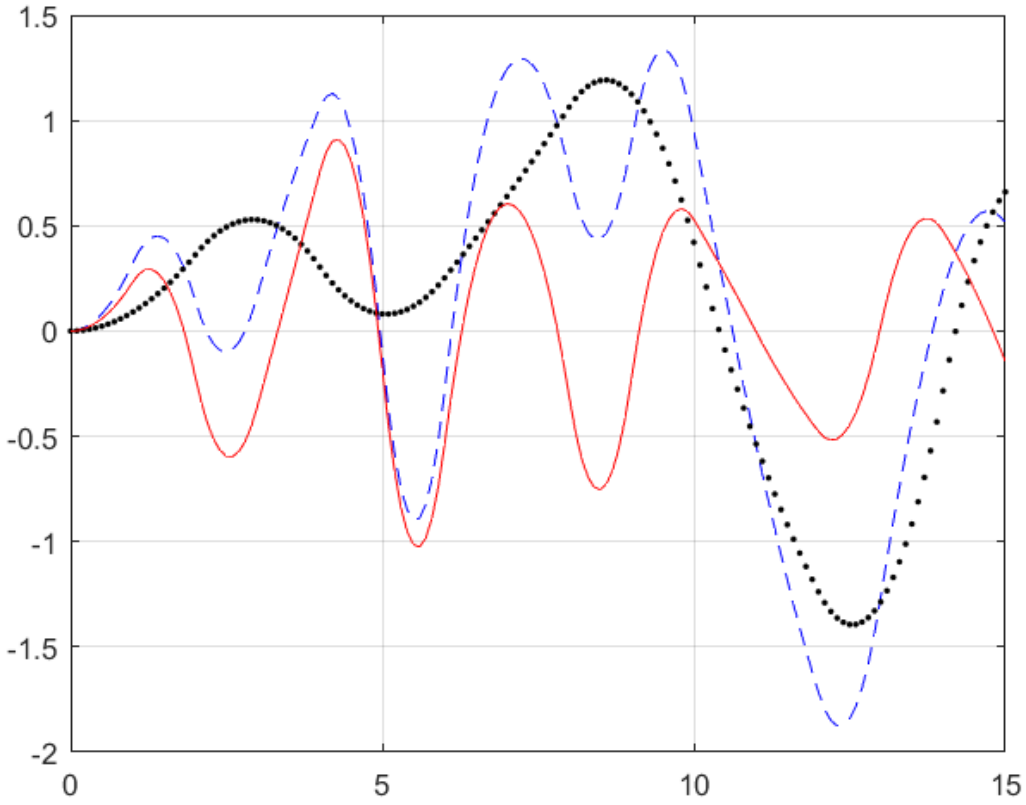


Figure 29: Illustration of the generation of wavelet noise

Wavelet noise in two dimensions

A two dimensional b-spline is defined as follows.

$$S_k(s,t) = \sum_{j=0}^{J}\sum_{i=0}^{I} \alpha_{i,j} B_{i,k}(s) B_{j,k}(t) \tag{131}$$

Here the entries of $\alpha_{i,j}$ can be represented in a matrix. The down sampling is first performed on the rows of the $\alpha_{i,j}$ matrix and then on the columns of the resulting matrix. The resulting image is then subtracted from the initial image.

Figure 30 shows an example of the initial image created using a two-dimensional b-spline with random weights. Figure 31 shows a down-sampled two-dimensional b-spline image and Figure 32 shows the final b-spline wavelet noise tile. The frequency properties of the noise tile are shown in the power spectral density of the image, which is shown in Figure 33.

Figure 25 shows the final image of many images with different scales of b-spline wavelet noise being added together.

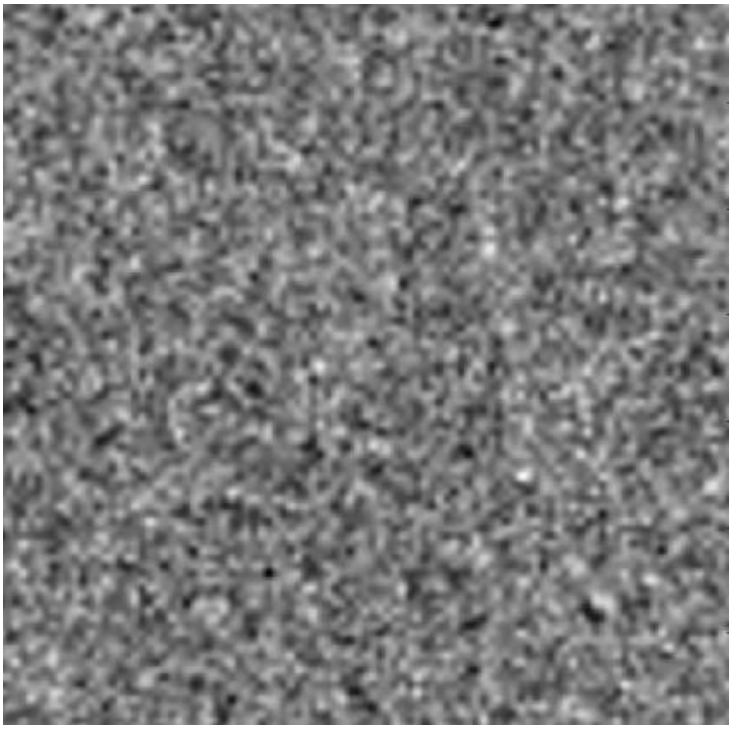


Figure 30: Initial two-dimensional b-spline image with random weights



Figure 31: Down-sampled b-spline image

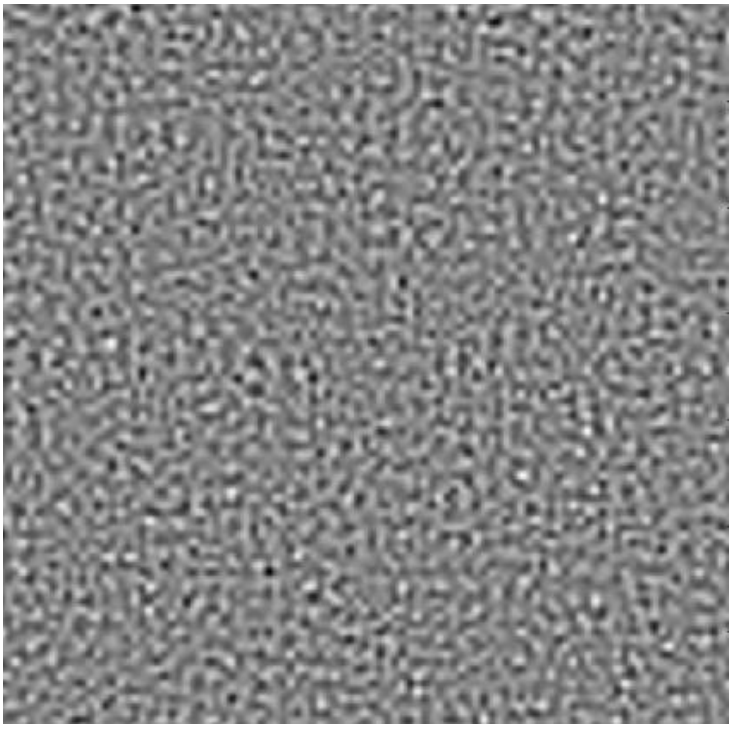


Figure 32: B-spline wavelet noise image (one layer)

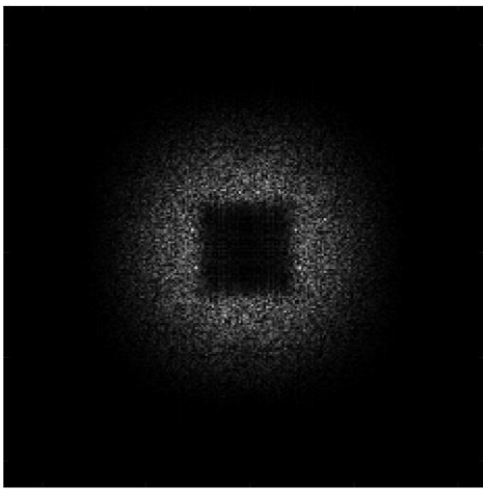


Figure 33: Power spectral density of one layer of wavelet noise

Tiling wavelet noise tiles

If the b-splines are created such that at the basis functions wrap around the boundaries, the b-spline tiles may be tiled together. This allows for the creation of a large b-spline wavelet image while reducing the computational effort required to create it.

3 METHOD

3.1 Overview

This study looks into using the GPU accelerated optical flow algorithms that were implemented in the OpenCV open source computer vision toolbox for the processing of BOS images. In order to judge the results of these algorithms, comparisons were needed. The chosen comparisons were:

- Theoretically predicted optical flow
- A standard cross-correlation (block matching) optical flow algorithm
- Colour schlieren images

Conical flow was chosen as a test case for the BOS system because the flow field around the cone (in front of the expansion fan) can be solved for using the Taylor-Maccoll equations. Additionally edge effects that two-dimensional test cases may suffer from are inherently taken care of. The chosen test condition was a nominal Mach number of 2.5. This operating point was reported to have good flow quality by the HSWT operating personnel.

The global procedure used was as follows:

- Two backgrounds (one wavelet background and one speckled background) were designed and used to capture the pairs of images required for BOS analysis.
    - The selected algorithms are used to process the distorted and undistorted pairs of images.
    - The parameters affecting convergence of these algorithms were set pragmatically and remained fixed for the subsequent analyses.
- The parameters affecting the results of the analyses were varied to study the effects on the results.
- A schlieren image of the same test setup was captured.[4]
- The theoretical optical flow was predicted using the measured test setup geometry and flow conditions.
- OpenCV algorithm results, the results of the block-matching algorithm, the optical schlieren image, and the theoretical predictions are then compared.

[4] Special thanks to Dr. David Maclucas for acquiring this image during a demonstration of the HSWT

3.2 Test setup

3.2.1 Background design

Speckled background

The speckled background was designed to have squares of approximately 4 mm x 4 mm. This would ensure that at least one full Bayer mask per block was used. A Bayer mask is a 2 x 2 ensemble of pixels, with each pixel having a specific colour filter. This was considered the design with the best compromise between high frequency while ensuring at each square filled at least one full Bayer mask.

Wavelet background

A wavelet background was created with bumps on the 2 mm, 4 mm, and 8 mm scale. This choice resulted from 4 mm spacing (as discussed with the speckled backgrounds) as well as a half-scale and double scale noise tile.

3.2.2 Equipment

The equipment used for these experiments is outlined in the sections that follow.

Wind tunnel

The experiments were performed at the CSIR's HSWT. The HSWT is a blow-down wind tunnel that can operate between Mach 0.6 and Mach 4.3 and has a cross-section of 450 mm by 450 mm. The wind tunnel instruments are calibrated regularly according to a calibration schedule. Either these instruments are calibrated by the National Metrology Institute of South Africa (NMISA) or by secondary instruments that were themselves calibrated by the NMISA.

Model

The wind tunnel model used was a cone designed with an integral sting to fit directly into the HSWT pitch mechanism with no balance. The model had a circular cross section with a conical point to create a conical shock. The model was measured with a Vernier protractor to have a total cone angle of 28°56'. This equates to 14°28' from the centreline to the surface of the cone. The front end of the wind tunnel model is shown in Figure 34.



Figure 34: Front end of the wind tunnel model

Camera

The BOS system uses a BASLER ace acA2000-50gc camera with the following specifications.

Table 9: Camera specifications

Parameter	Value
Sensor Type	CMOS
Sensor Size	11.3 mm x 6 mm
Resolution	2046 pixels by 1086 pixels
Resolution	2 MP
Pixel Size	5.5 µm x 5.5 µm
Frame Rate	50 fps
Mono/Colour	Colour
Pixel Bit Depth	12 bits
EMVA Quantum Efficiency	67.5 %
Dark Noise	13.9 e^-
Saturation Capacity	9.3 ke^-
Dynamic Range	56.5 dB
Signal to Noise Ratio	39.7 dB

The camera used a Ricoh FL-CC2514-2M lens. The specifications for this lens are outlined in Table 10.

Table 10: Camera lens specifications

Parameter	Specification
Format	2/3”
Focal Length	25 mm
Aperture	F1.4
Iris Type	Manual Iris
Mount	C-Mount

Lighting

Three GS Vitech GmbH multiLED lights were used. These are high power LED lights.

3.2.3 Setup

General layout

The test setup geometry is illustrated in Figure 35. The camera was positioned so that the top and bottom test section walls were just on the edge of the of the camera’s view. This resulted in a position that was 905 mm from the closest schlieren window.

The wind tunnel test section half-width is 289 mm including the thickness of the schlieren glass. The backgrounds were place approximately 3 m from the nearest schlieren window and this position measured. The resulting measured distances were as follows.

- Speckled Background: 3000 mm
- Wavelet Background: 3011 mm

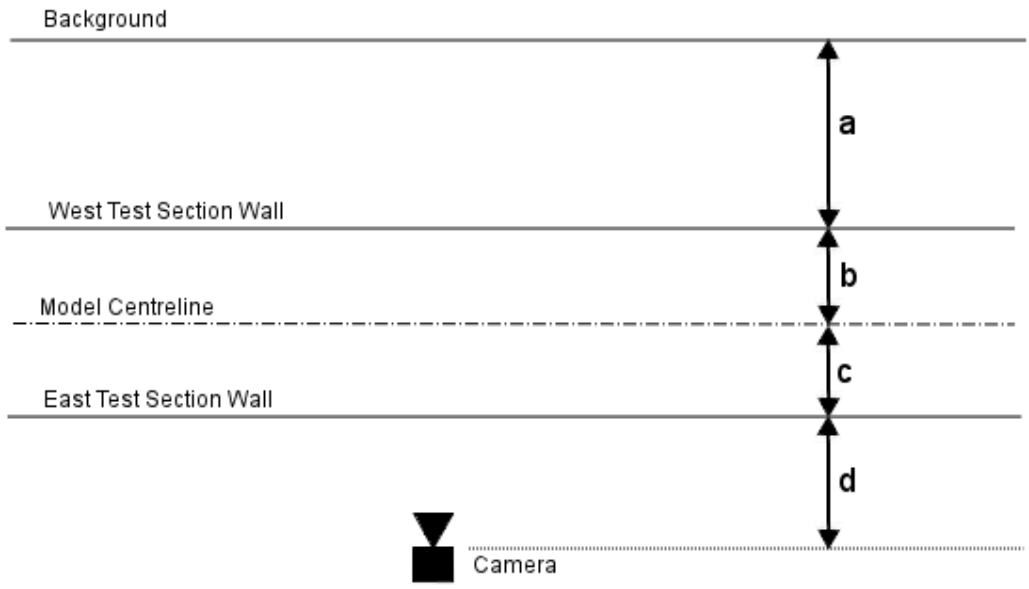


Figure 35: BOS setup geometry

The dimensions for each setup are outlined in Table 11 below.

Table 11: Dimensions of BOS setup

Parameter	Speckled Background	Wavelet Background
a	$3.000\ m$	$3.011m$
b	$0.289\ m$	$0.289\ m$
c	$0.289\ m$	$0.289\ m$
d	$0.905\ m$	$0.905\ m$
$a+b$	$3.289\ m$	$3.300\ m$
$a+b+c+d$	$4.483\ m$	$4.494\ m$

Model alignment

The model was installed on to the HSWT pitch actuator. A Vernier inclinometer was used to set the intercept for the pitch sector calibration and the pitch sector was controlled to 0° for the tests. This corresponds to having the longitudinal axis of the model aligned with the flow. The inclinometer had a resolution of 1 arc minute with a reported accuracy of 0.5 arc minutes.

Lighting

The lighting was arranged to give the backgrounds maximum illumination while still maintaining a qualitatively constant illumination.

Camera setup

The camera was focussed on the background and the aperture closed until a marking on the outside of the closest schlieren window came into focus. The exposure time was adjusted to accommodate these settings. This ensured that the depth of field was sufficient to capture the flow field in focus.

For each background, the camera exposure time was initially set automatically using the OEM supplied camera software and then set to a nearby constant value for the remainder of the test.

The raw camera gain was set to 36, with the camera mode being set to 12-bit greyscale. Exposure times were 0.5 s for the wavelet background and 0.25 s for the speckled background. The larger exposure for the wavelet background was required due to it being a generally darker image.

The distance of the camera from the test section was chosen such that the upper and lower test section walls lay just outside the view of the camera.

Camera alignment was achieved by marking the centres of each schlieren window and aligning these marks in the camera image.

3.3 Test procedure

For each test, the following procedure was followed:

- Test setup was performed
- Approximately 30 images were acquired prior to starting the tunnel.
- The wind tunnel was started and allowed to settle.
- Approximately 30 images were acquired manually using the camera OEM supplied software.
- The tunnel was stopped.

3.4 Image processing

In order to reduce the possibility of camera noise affecting the results, an average of the images acquired pre-test and an average of the images acquired post-test was calculated for each set of images corresponding to each background. Section 4.5.1 details the combinations of parameters in the various algorithms that were used to process the results.

3.5 Theoretical image generation

The theoretical optical flow images are generated using the method outlined in the theory section. The method for determining the gain that relates the angular distortion to the per pixel distortion, $\frac{r}{\zeta}$, is outlined in this section.

3.5.1 Relationship between distortion angle and image distortion

The symbols outlined in Table 12 are used in this section in addition to those shown in Figure 35.

Table 12: List of symbols for the relationship between distortion angle and image distortion

Symbol	Description
d_o	Distance to object
d_i	Distance to background
f	Focal length
m	Magnification
Δx_{ccd}	Distance between pixels on CCD
Δx_{bg}	Distance on background object relating to a 1 pixel change on the image
$\Delta\theta$	Change in deflection angle relating to a 1 pixel change in the image

The following method was used to find the relationship between angular distortion of the light rays and the per pixel distortion of the raw BOS images.

The lens equation is shown below.

$$\frac{1}{d_o}+\frac{1}{d_i}=\frac{1}{f} \tag{132}$$

The definition of magnification is given by the following equation.

$$m=\frac{d_i}{d_o} \tag{133}$$

Using the two equations above, the following relationship can be derived.

$$m = 1 - \frac{d_i}{f} \tag{134}$$

The results for the two background setups are shown in Table 13

Table 13: Magnification calculation value

Parameter	Speckled Background	Wavelet Background
di	$4.483\ m$	$4.494\ m$
f	$0.025\ m$	$0.021\ m$
m	-178	-179

To find the length on the object that relates to the length between pixels on the image, Δx_{bg}, the pixel size on the charge-coupled device (CCD), Δx_{ccd}, is scaled by the magnification.

$$\Delta x_{bg} = -m\Delta x_{ccd} \tag{135}$$

The negative sign is appropriate since the image appears upside down on the CCD.

Table 14: Equivalent pixel size calculation values

Parameter	Speckled Background	Wavelet Background
Δx_{ccd}	$5.5\ \mu m$	$5.5\ \mu m$
m	-178	-179
Δx_{bg}	$0.979\ mm$	$0.984\ mm$

To find the angular deflection per pixel at the model, the small angle approximation is used.

$$\Delta\theta = \frac{\Delta x_{bg}}{a+b} \tag{136}$$

The results are shown in Table 15.

Table 15: Angular deflection calculation values

Parameter	Speckled Background	Wavelet Background
Δx_{bg}	$0.979\ mm$	$0.984\ mm$
$a+b$	$3.289\ m$	$3.300\ m$
$\Delta\theta$	$0.298\ .10^{-3}\ rad/pix$	$0.298\ .10^{-3}\ rad/pix$

The reference case for the theoretical calculations is therefore chosen have $\Delta\theta = 0.298.10^{-3}\ \frac{rad}{pix}$

Here $\Delta\theta$ is equivalent to $\frac{r}{\zeta}$ in equation 63. This value is used to scale the predicted deflection angles to predicted per-pixel optical flow values.

4 RESULTS

4.1 Flow properties

The achieved operating conditions are outlined in Table 16

Table 16: Wind tunnel flow parameters

Parameter	Speckle Value	Wavelet Value	Average Value
Mach Number	2.477	2.484	2.481
Static Pressure [kPa]	17.34	17.16	17.25
Total Pressure [kPa]	286.11	286.15	286.13
Static Temperature [K]	135.0	134.7	134.8
Total Temperature [K]	300.6	301.0	300.8
Free Stream Density [$kg/m3$]	0.448	0.444	0.446

4.2 Raw images

The images used for processing were average of the images captured pre-test for the reference images and during the test for the distorted images. The number of images averaged in each is captured in Table 17.

Table 17: Number of images averaged

Background	Reference Image	Distorted Image
Speckle	30	29
Wavelet	31	19

The following features were noted by inspection of the raw images:

- The model moves slightly (approximately 5 pixels) vertically during the test, this results in some of the background pattern above the model being occluded while some previously occluded background pattern is observable below the model. When inspecting flow features close to the model therefore, the flow features shown on the upper surface would be more valid.
- Oil streaks are formed on the schlieren windows during the tests. The averaging reduces this to some degree. These oil streaks can be seen clearly in the single images shown in Figure 40 and Figure 41.

The resolution of the model movement would involve a detailed analysis of the model control system while the resolution of the oil-streaks would involve a detailed study and maintenance of the nozzle-positioning jacks in the HSWT nozzle. These tasks lie outside the scope of this study and are being investigated by the HSWT personnel.

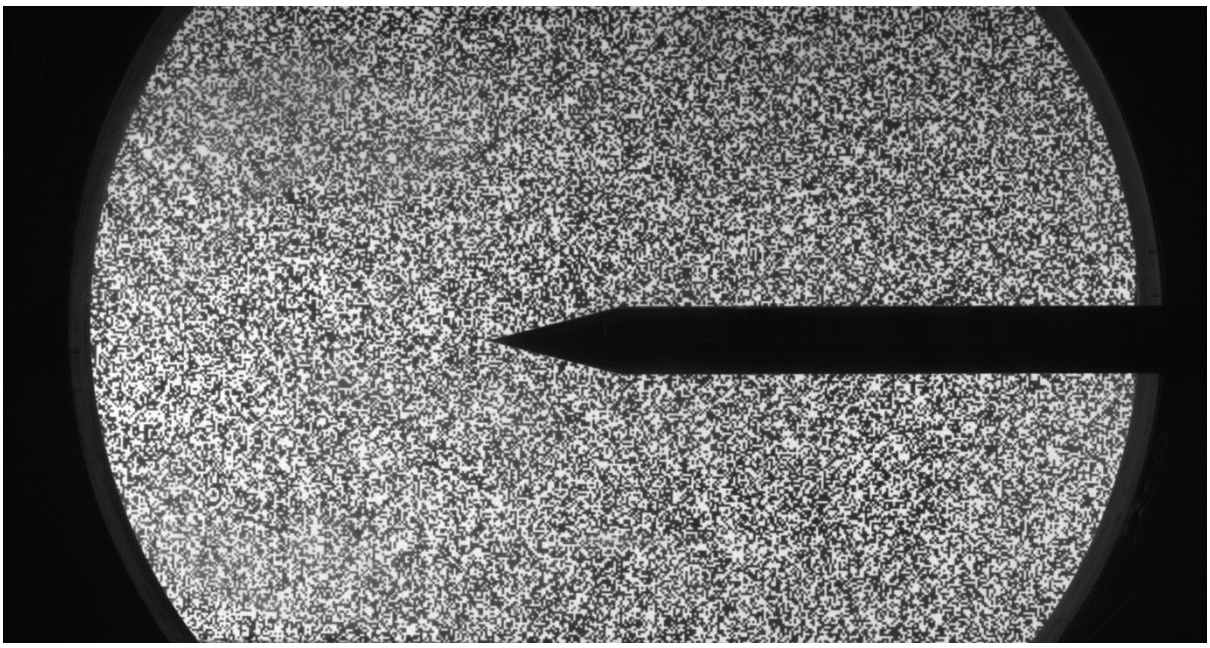


Figure 36: Averaged reference Image for the speckled background

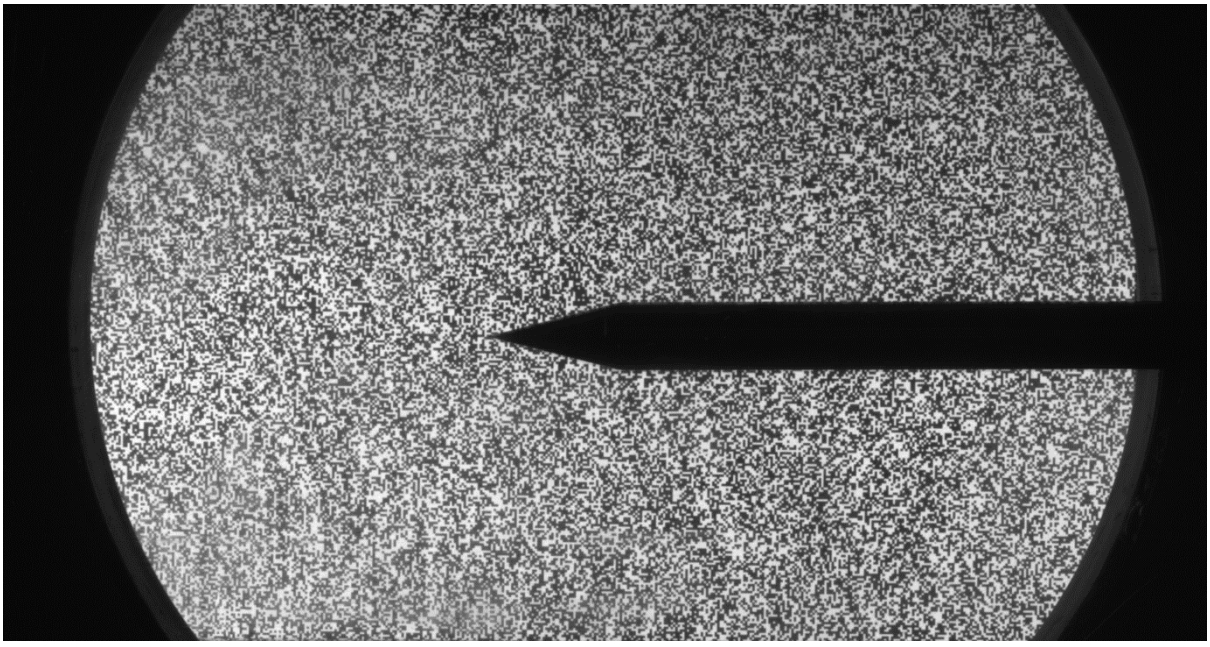


Figure 37: Averaged distorted image for the speckled background

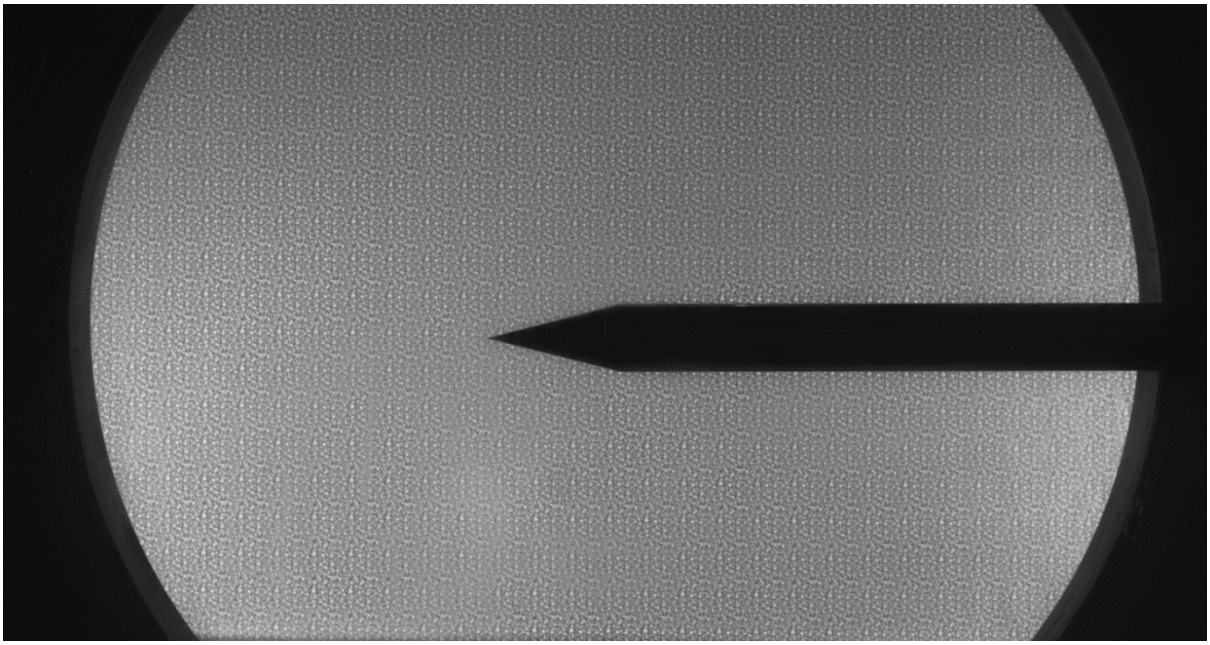


Figure 38: Averaged reference image for the wavelet background

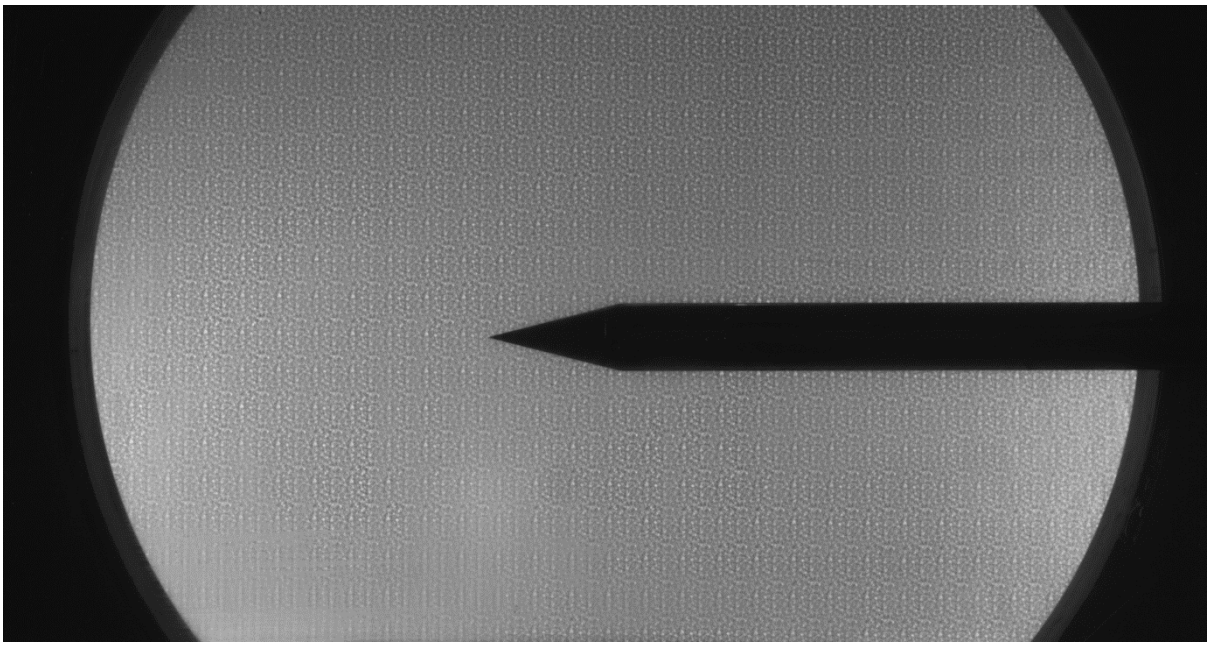


Figure 39: Averaged distorted image for the wavelet background

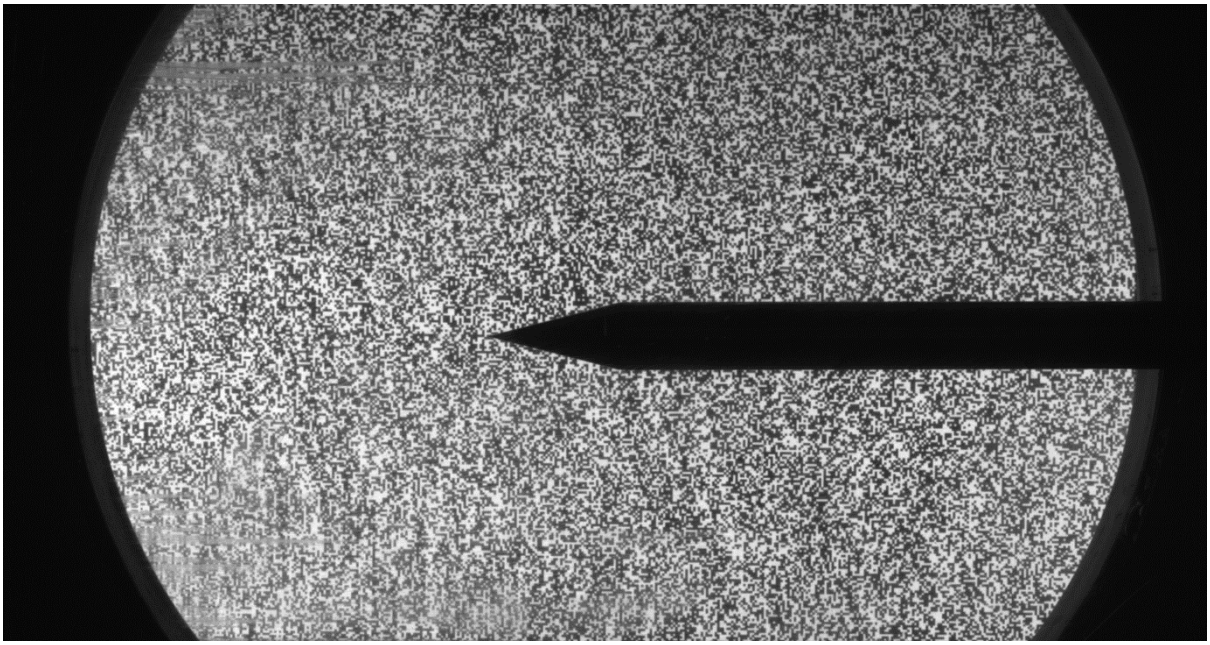


Figure 40: Last single frame captured during the speckled image test

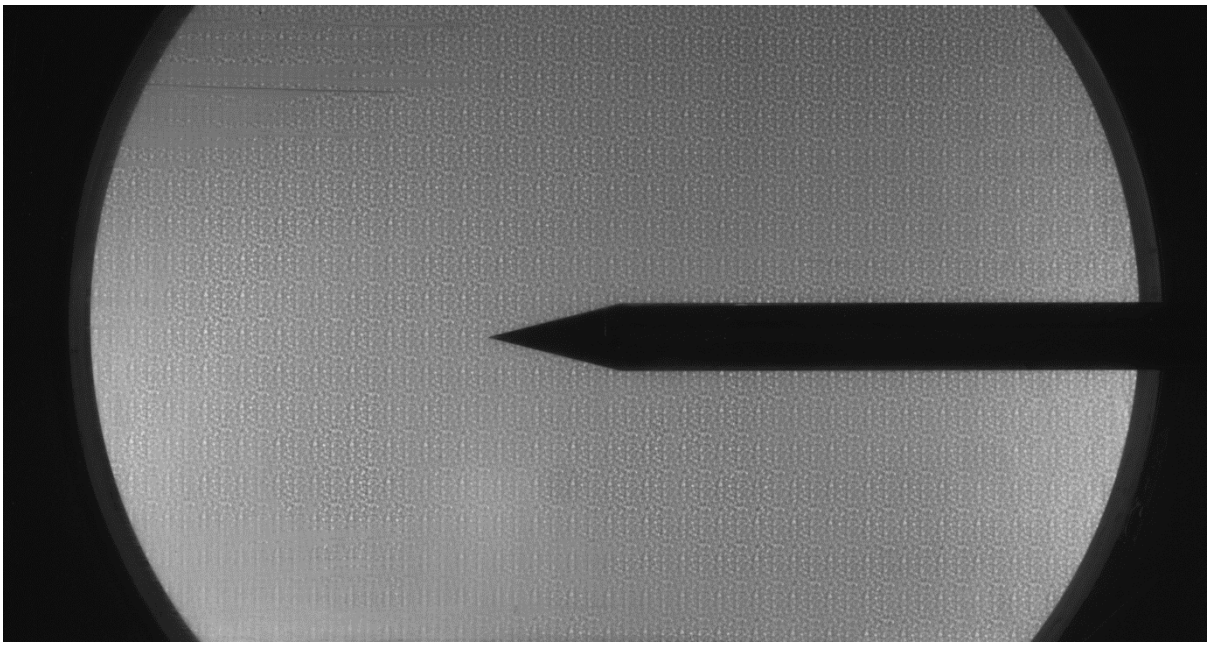


Figure 41: Last single frame captured during the wavelet background test

4.3 Theoretical distortions

Using the average values from Table 16, the predicted image distortions are illustrated in the figures that follow.

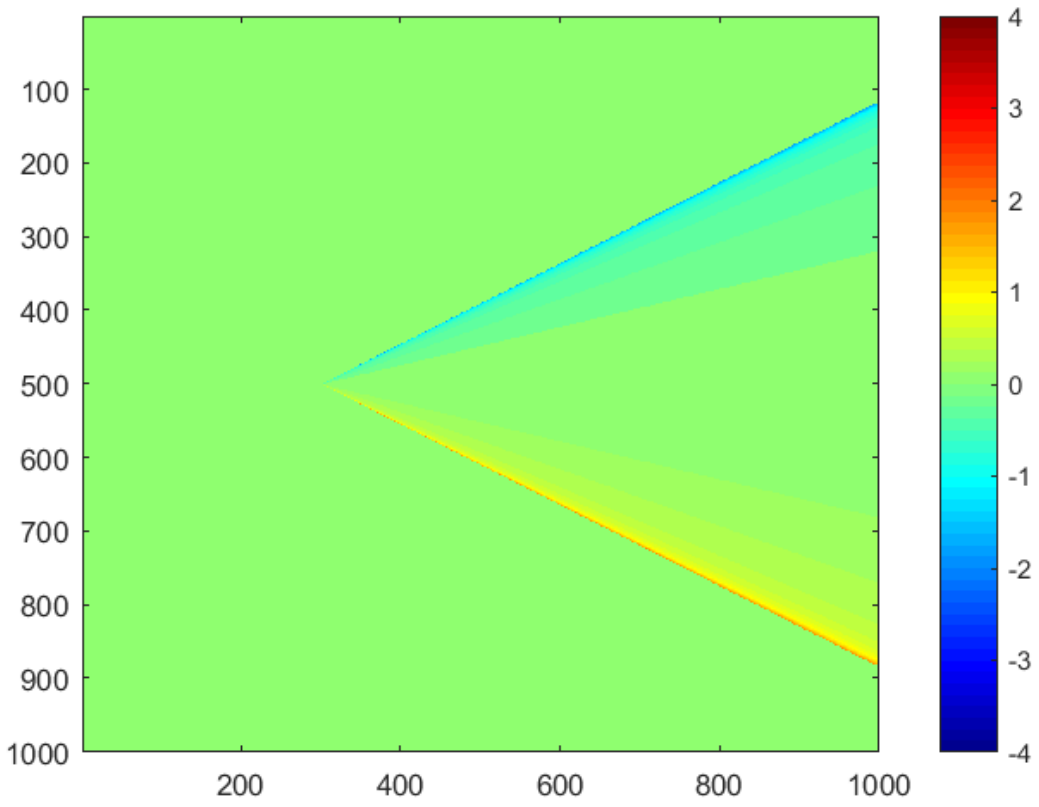


Figure 42: Predicted vertical deflection in pixels with a range of -4 to +4 pixels

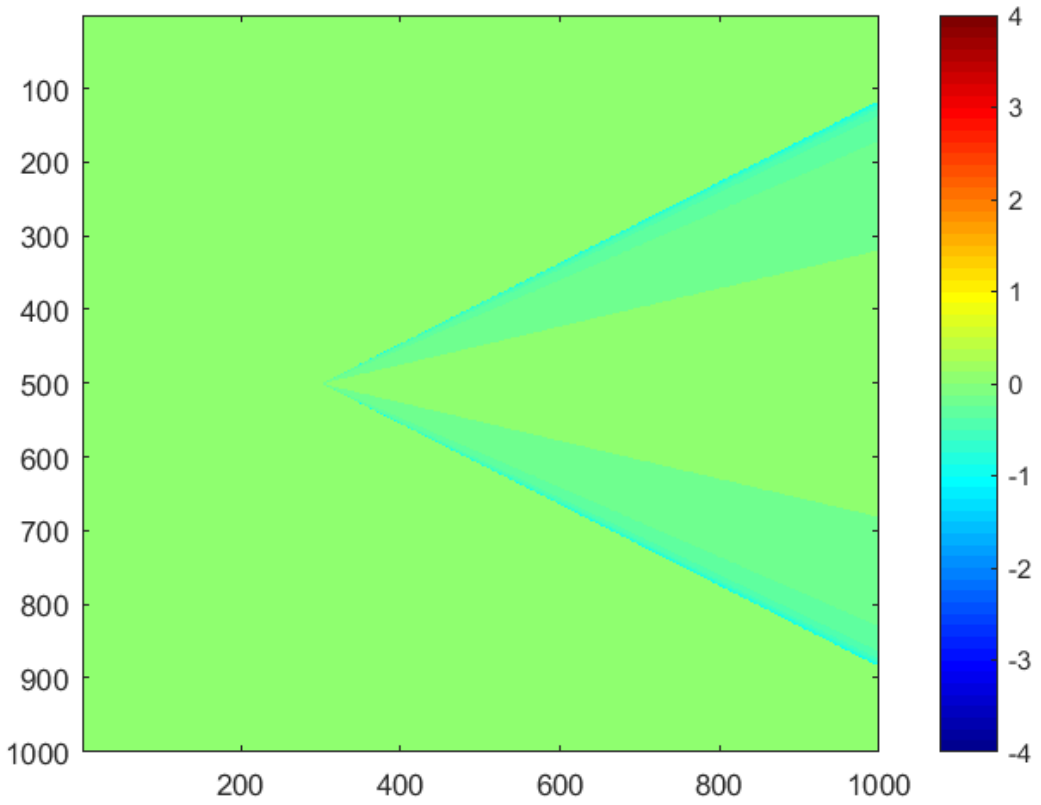


Figure 43: Predicted horizontal deflection in pixels with a range of -4 to +4 pixels

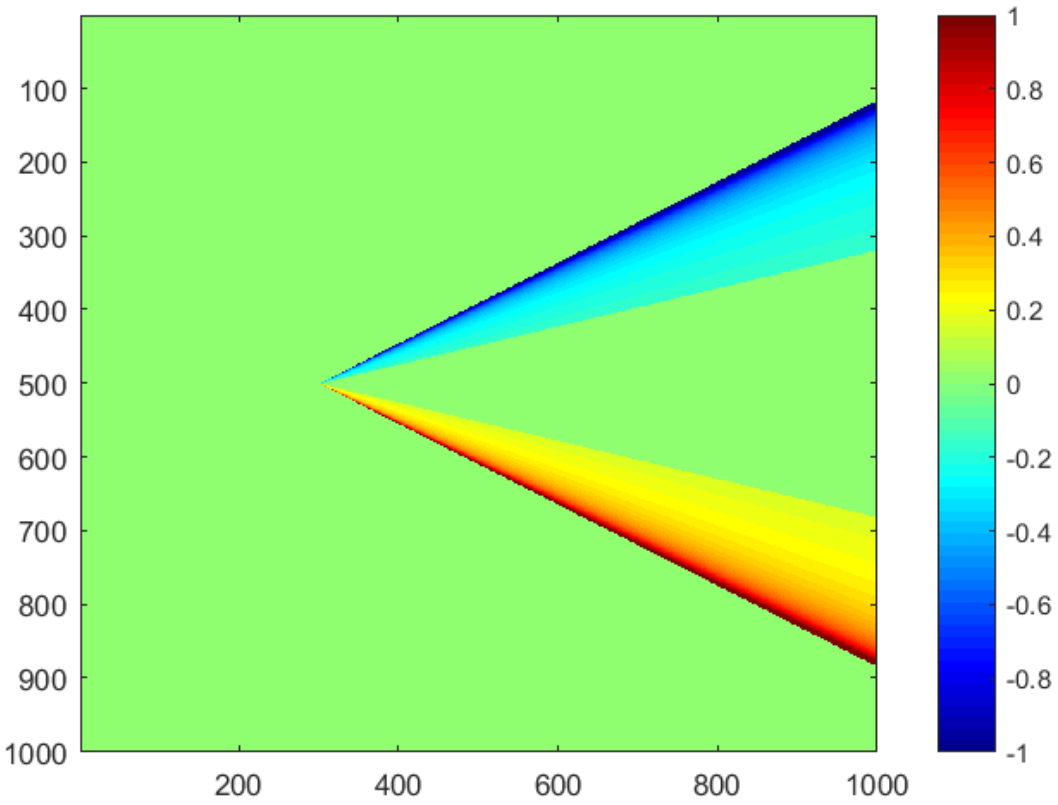


Figure 44: Predicted vertical deflection in pixels with a range of -1 to +1 pixels

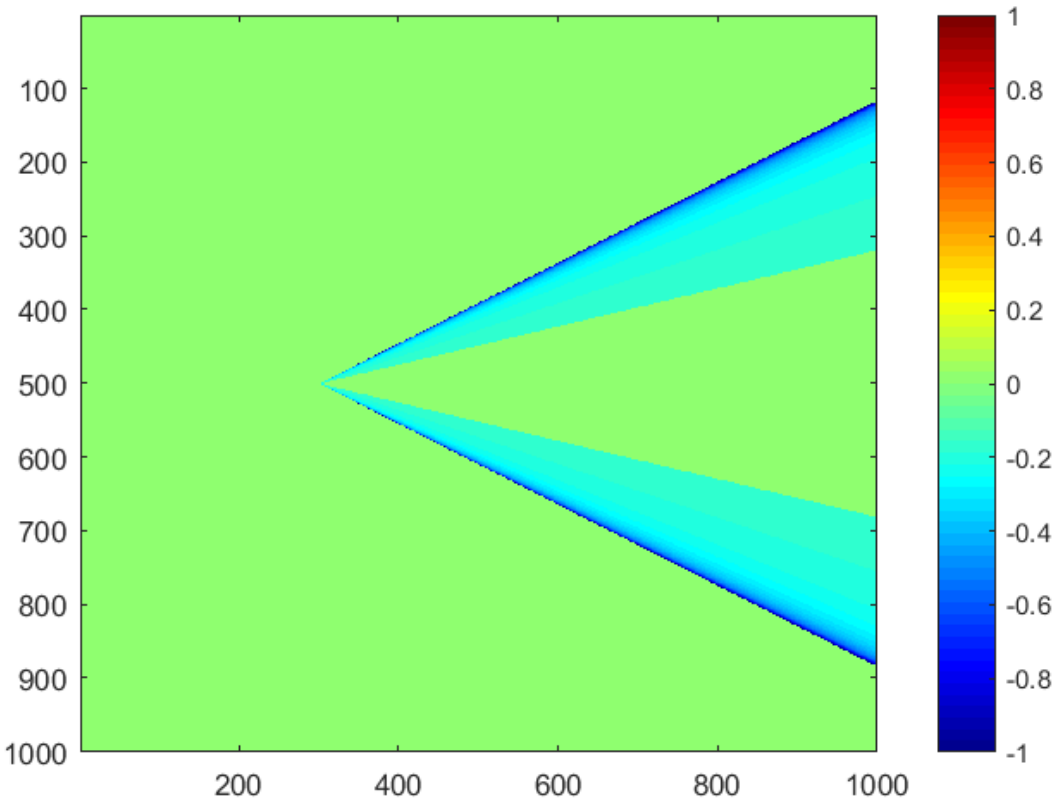


Figure 45: Predicted horizontal deflection in pixels with a range of -1 to +1 pixels

4.4 Colour schlieren

The colour schlieren image in Figure 46 shows some features existing in the flow for comparison with BOS images. The conical shock followed by the expansion that emanates from the base of the cone can be observed. Additionally, weak compression waves emanating from the HSWT nozzle are also observed. This image has been flipped horizontally to have the same orientation as the BOS images.

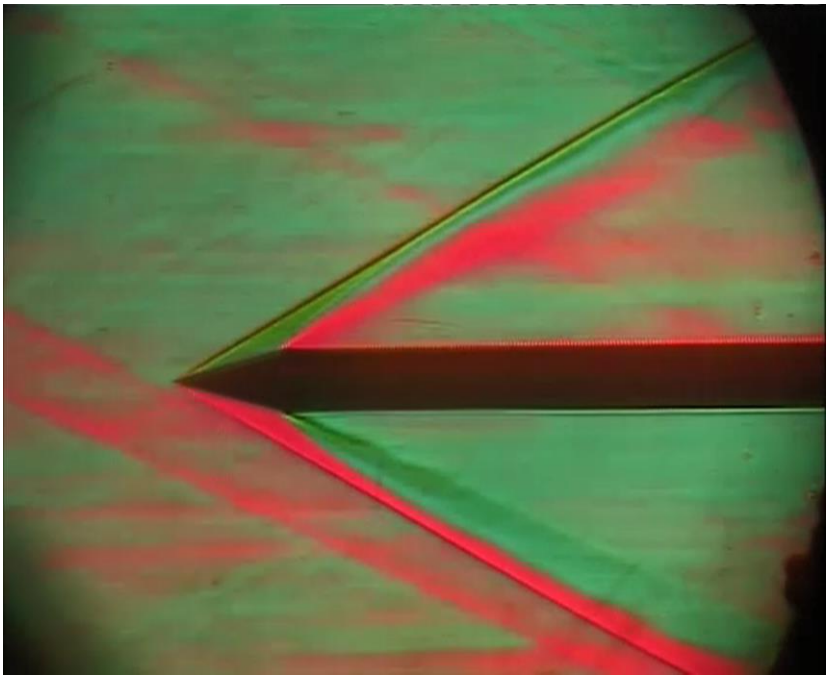


Figure 46: Colour schlieren image

4.5 Experimental images

The results contained in the digital appendix are arranged in a tree structure as follows:

- At the top level, the results from each algorithm are grouped together
- The first sub layer groups the down sampling strategy used. Three were studied in this case:
    - A single layer was processed with no down sampling
    - 11 Layers were processed, each with a down sampling factor of 0.5
    - 35 layers were processed, each with a down sampling factor of 0.8
- The second sub layer groups the results obtained from the analysis of either the wavelet or the speckled background.
- Further sub layers group results according to algorithm specific parameters.
- The final directory contains four images, two illustrating the lateral component of optical flow, and two illustrating the vertical component. Each component is illustrated using a range of either -4 to 4 pixel displacement or -1 to 1 pixel displacement. The former is useful to see the global result with the latter being useful to inspect the finer detail with only a few pixels exceeding this range. Illustrations that included larger pixel displacements were found not to add any practical value.

These results contain 672 images.

A sample result is shown in the two images that follow.

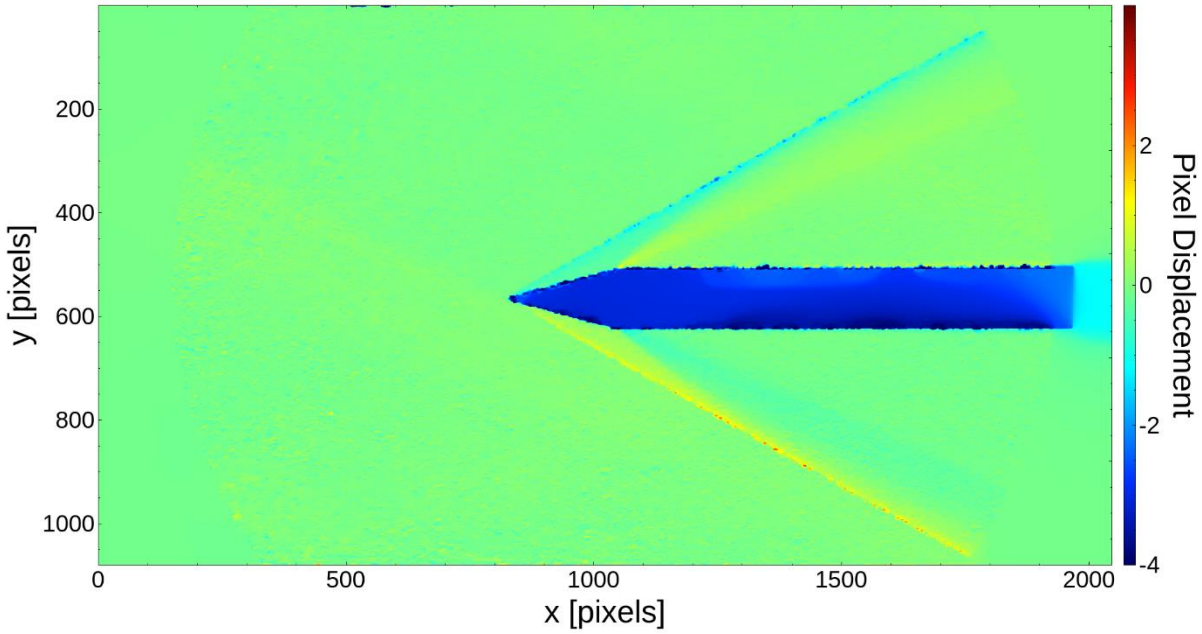


Figure 47: Sample measured vertical deflection in pixels with a range of -4 to +4 pixels

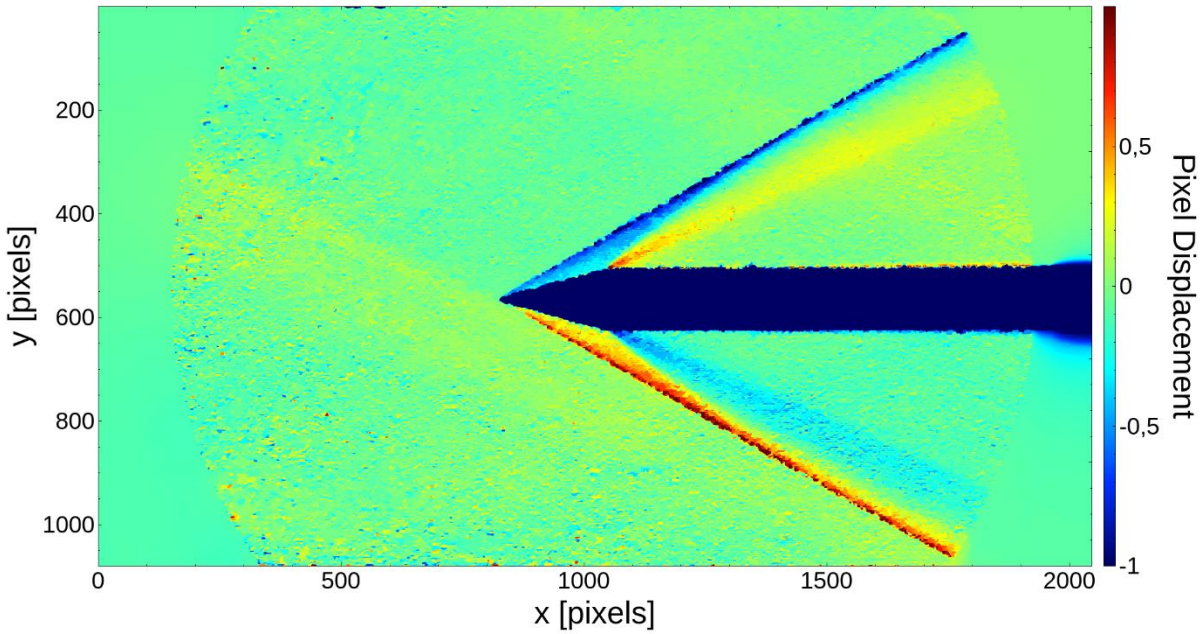


Figure 48: Sample measured vertical deflection in pixels with a range of -1 to +1 pixels

4.5.1 Processed results

Images with colour scales indicating deflection of between -1 and 1 pixel and -4 and 4 pixels were generated for all permutations of each of the parameters outlined in the lists below. A detailed breakdown of the results contained in the digital appendix is contained in Appendix C.

Block matching

Window Size

- 4x4
- 8x8
- 16x16
- 32x32

Sub-Pixel Interpolation Type

- None
- Centroid
- Parabolic
- Gaussian

Background Type

- Wavelet
- Speckle

Lucas and Kanade

Window Size[5]

- 5x5
- 9x9

[5] A bug identified in the Lucas and Kanade algorithm implemented in the OpenCV version used for this study meant that only window sizes containing an odd number of pixels produced reliable results. The window sizes chosen were specifically chosen to be close to that of those used in the block matching method.

- 17x17
- 33x33

Down Sampling Strategy

- None
- 11 Layers with a down sample factor of 0.5

Background Type

- Wavelet
- Speckle

Brox

Gradient Consistency Weighting

- 0
- 50
- 100

Regularisation Term Weighting

- 0.1
- 0.2
- 0.4

Down Sampling Strategy

- None
- 11 Layers with a down sample factor of 0.5
- 35 Layers with a down sample factor of 0.8

Background Type

- Wavelet
- Speckle

Farnebäck

Greyscale Polynomial Approximation[6]

- 5x5 window with Gaussian standard deviation of 1.1 pixels
- 7x7 window with Gaussian standard deviation of 1.5 pixels

Optical Flow Averaging Window Size

- 4x4
- 8x8
- 16x16
- 32x32

Down Sampling Strategy

- None
- 11 Layers with a down sample factor of 0.5
- 35 Layers with a down sample factor of 0.8

Background Type

- Wavelet
- Speckle

[6] These are the recommended combinations published in the OpenCV documentation.

TV-L1

Greyscale Consistency Weight

- 0.08
- 0.16
- 0.32

Down Sampling Strategy

- None
- 11 Layers with a down sample factor of 0.5
- 35 Layers with a down sample factor of 0.8

Background Type

- Wavelet
- Speckle

5 DISCUSSION

First, the results from each of the various optical flow algorithms are discussed, followed by a discussions on general findings of various aspects.

5.1 Discussion on optical flow algorithms

Each algorithm and the parameters excluding those that affect the convergence of the numerical schemes are discussed in the sections that follow. The reader is referred to the colour schlieren image (Figure 46) and the theoretical optical flow image (Figure 44) within the area for which it is applicable for comparison with the presented optical flow images in the discussions below. The level of detail in the processed images can be deduced from these comparisons, particularly with the observation of the optical flow resulting from the density changes at and behind of the shock, the density changes due to the expansion fans and the compression waves emanating from the nozzle of the tunnel (as seen in the colour schlieren image).

5.1.1 Block matching

The block-matching algorithm is the most common algorithm used in BOS setups. This particular algorithm was developed with accuracy as the primary driver. The main parameters that have an influence on the results are the window size and sub-pixel interpolation type. The influence that these parameters have on the results is discussed below. Additionally, results from the two different backgrounds are compared.

Window size

With an increase of window size, an increase in noise rejection was observed with a corresponding decrease in image detail. This can be observed when comparing Figure 49 with Figure 50, which were processed using 4x4 and 32x32 windows respectively. Both images shown use Gaussian sub-pixel interpolation but this effect is seen in all cases.

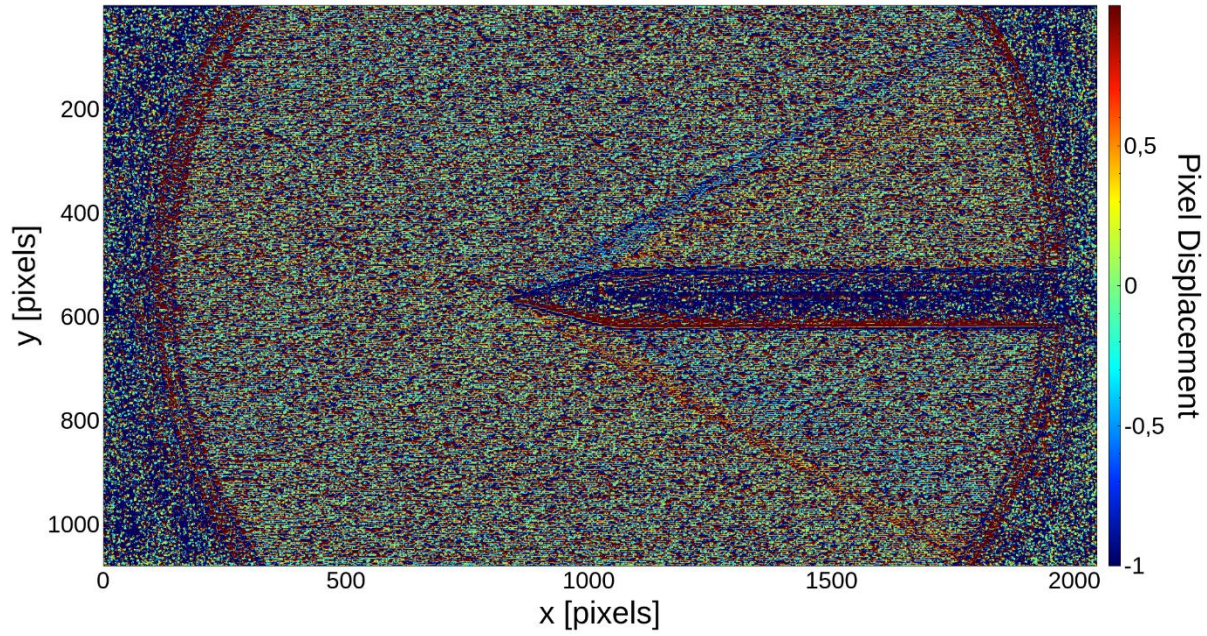


Figure 49: Block matching algorithm, vertical pixel displacement, 4 by 4 window, Gaussian sub-pixel interpolation, speckled background

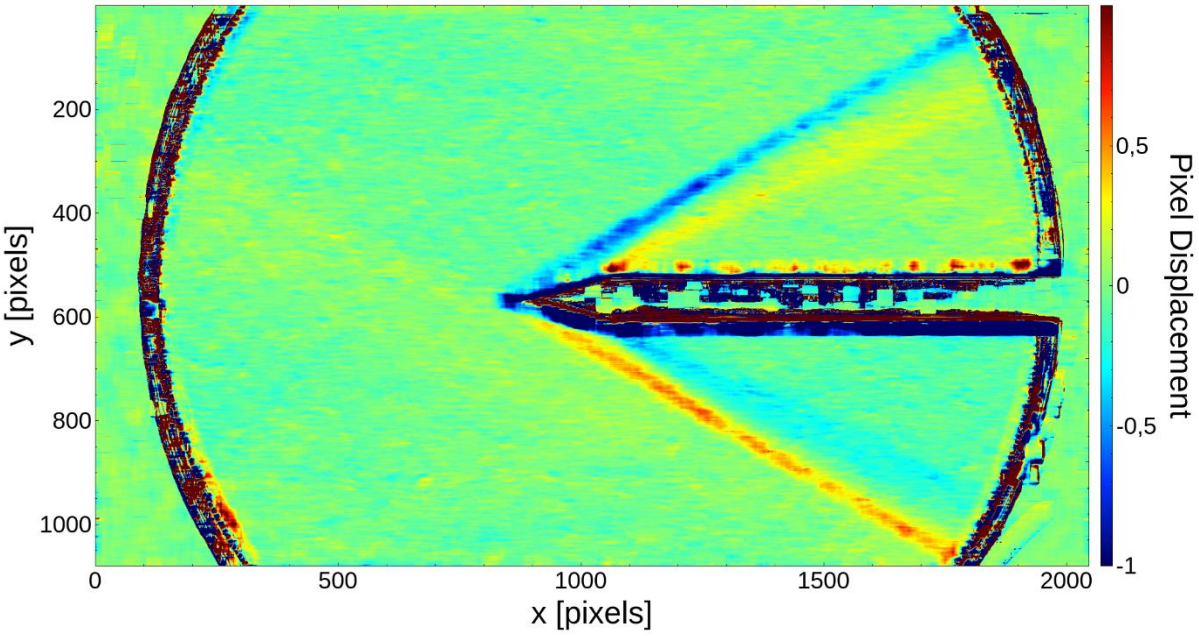


Figure 50: Block matching algorithm, vertical pixel displacement, 32 by 32 window, Gaussian sub-pixel interpolation, speckled background

Sub-pixel interpolation

The quadratic and Gaussian sub-pixel interpolation methods showed better results than the centroid sub-pixel interpolation. As would be expected, no sub-pixel interpolation performed the worst. Figure 51 through to Figure 54 show sample results that used a 16x16 window to illustrate results using the different interpolation schemes

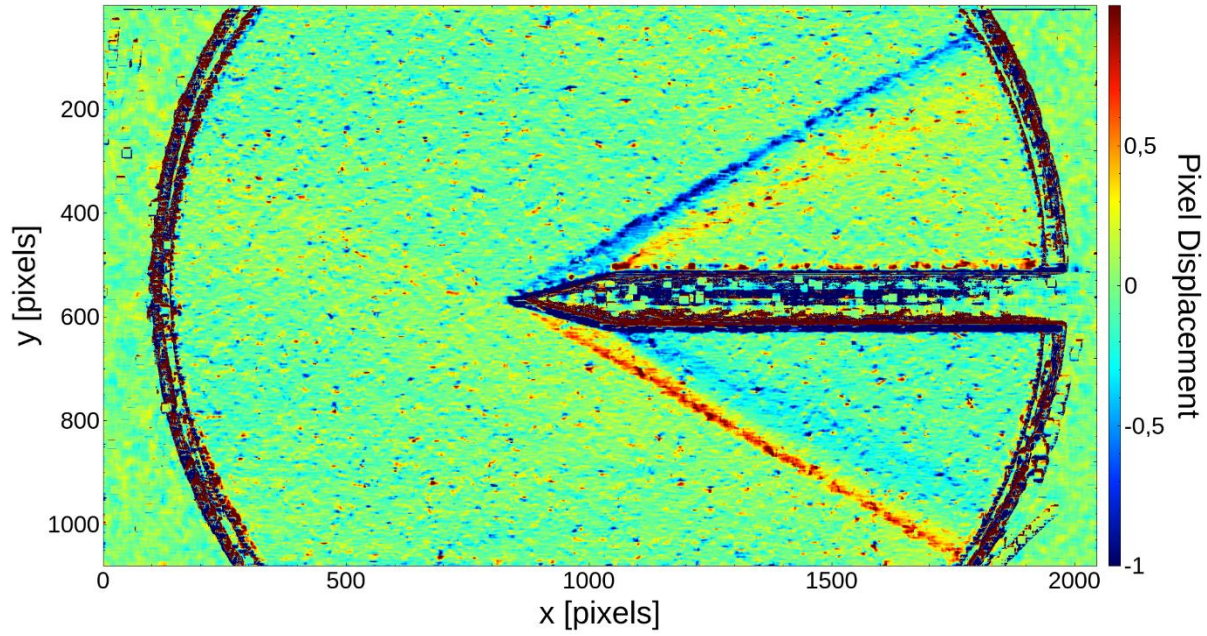


Figure 51: Block matching algorithm, vertical pixel displacement, 16 by 16 window, Gaussian sub-pixel interpolation, speckled background

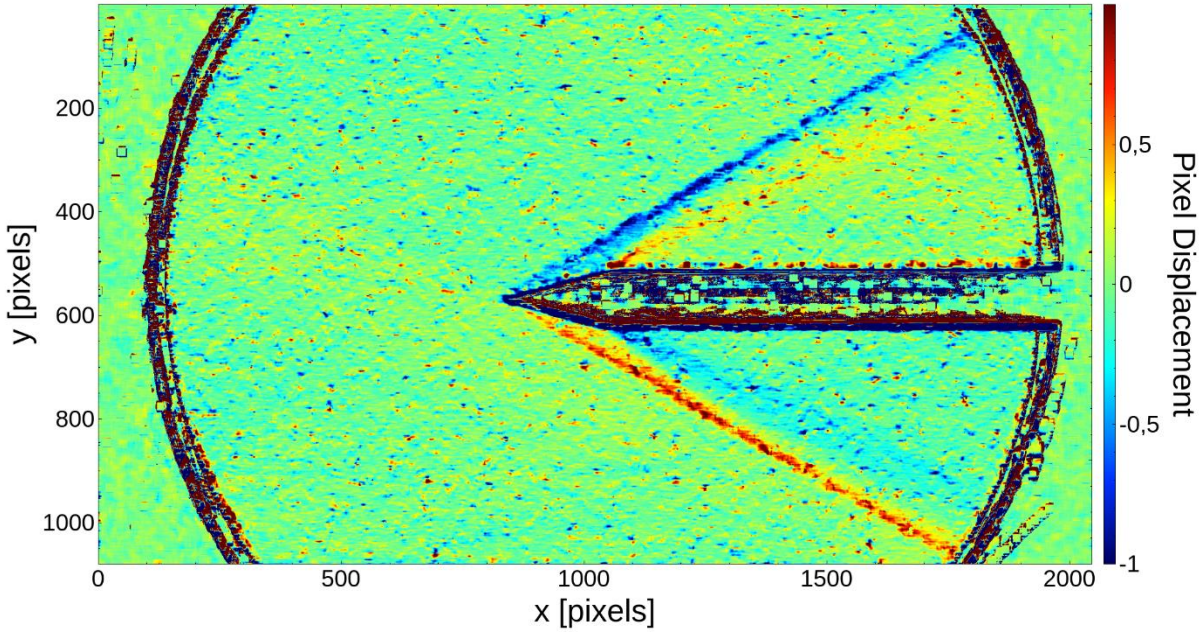


Figure 52: Block matching algorithm, vertical pixel displacement, 16 by 16 window, quadratic sub-pixel interpolation, speckled background

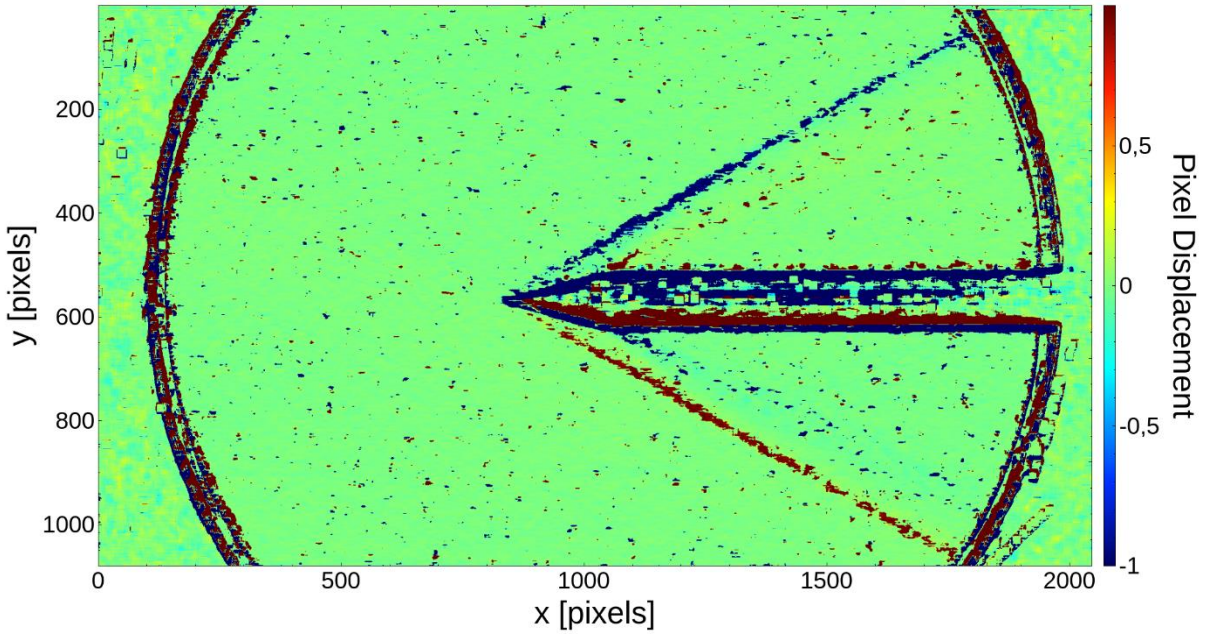


Figure 53: Block matching algorithm, vertical pixel displacement, 16 by 16 window, centroid sub-pixel interpolation, speckled background

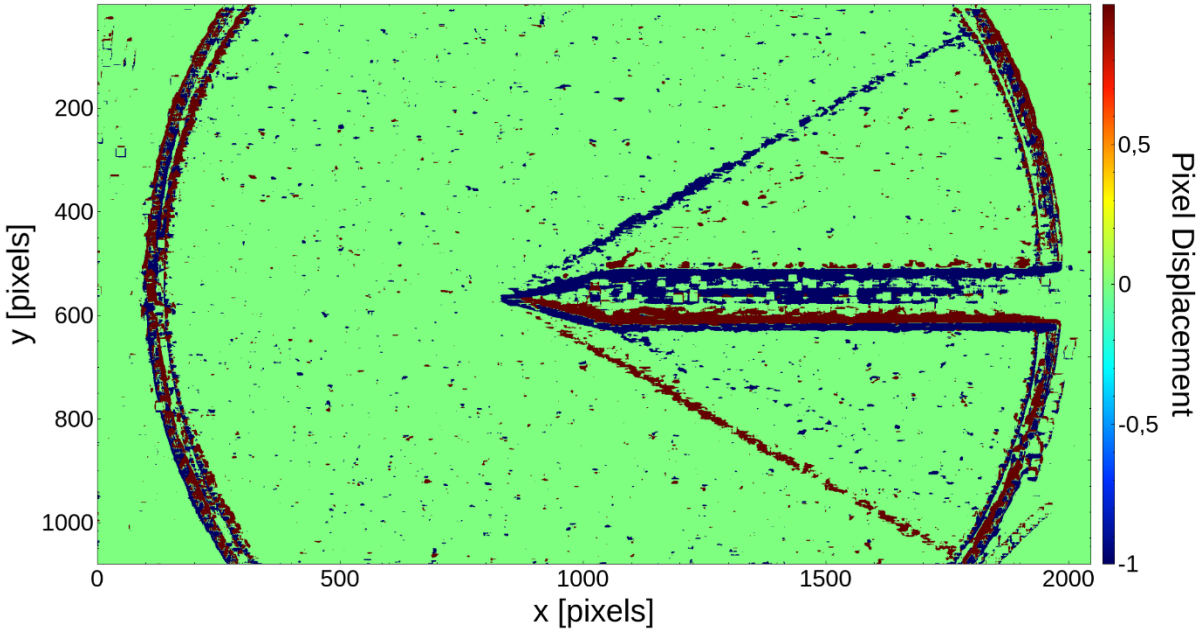


Figure 54: Block matching algorithm, vertical pixel displacement, 16 by 16 window, no sub-pixel interpolation, speckled background

Backgrounds

An example of the different results obtained by the wavelet and speckled backgrounds are shown in Figure 55 and Figure 56 respectively. There is a regular pattern present in the errors of the wavelet pattern coinciding with the tiles used to create this background. The streaks from the oil on the schlieren window are evident in the wavelet background BOS image but not in the speckled background BOS image. Both backgrounds produce a similar level of detail, with those from the wavelet background being perceived as having slightly higher detail with qualitatively less noise. This may be due to the regularity in the noise of the BOS images produced from the wavelet backgrounds.

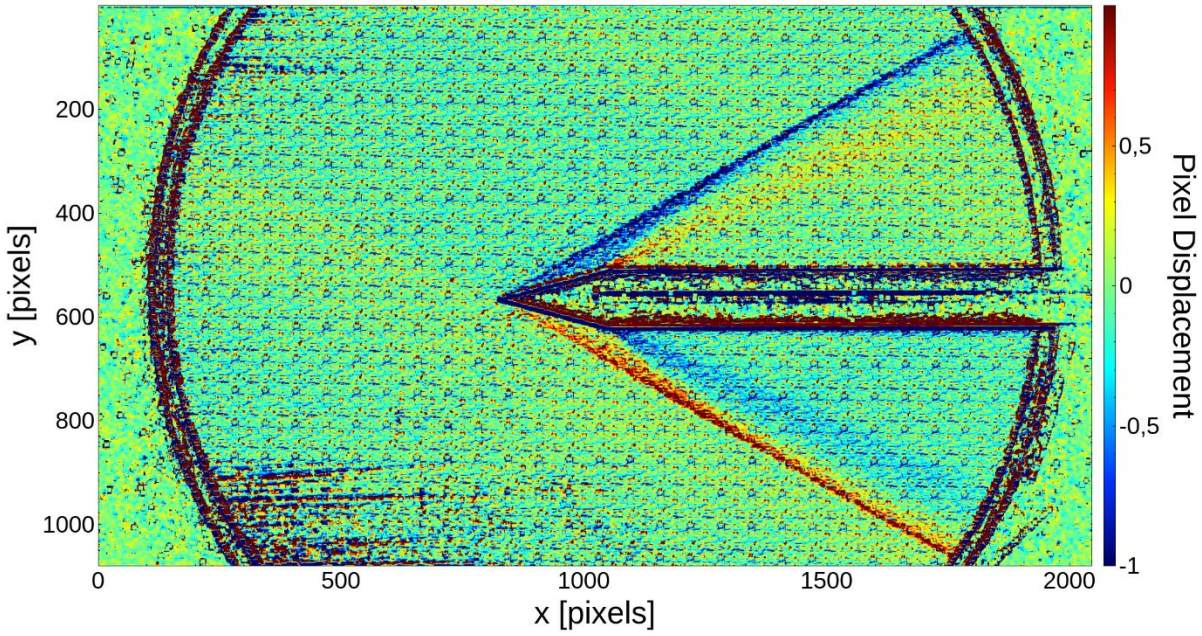


Figure 55: Block matching algorithm, vertical pixel displacement, 8 by 8 window, Gaussian sub-pixel interpolation, wavelet background

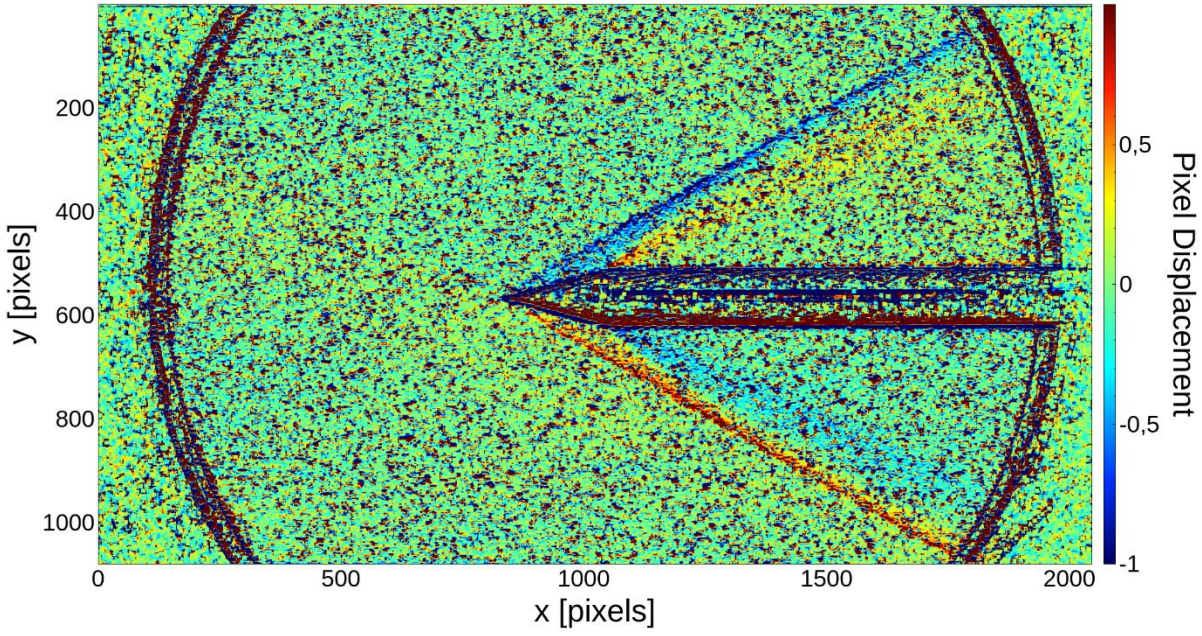


Figure 56: Block matching algorithm, vertical pixel displacement, 8 by 8 window, Gaussian sub-pixel interpolation, speckled background

5.1.2 Lucas and Kanade

The main factors that have an influence on the results from the Lucas and Kanade algorithm are the window size and whether or not a multi-resolution processing scheme is used. The influence that these parameters have on the results is discussed below. Additionally, results from the two different backgrounds are compared. In general, this algorithm produced BOS images with more detail and less noise than it's block-matching counterpart.

Window size

Window size had the effect of reducing noise at the expense of reducing detail in the processed images. This is can be seen by comparing Figure 57 and Figure 58, which use a 5x5 and 33x33 window respectively.

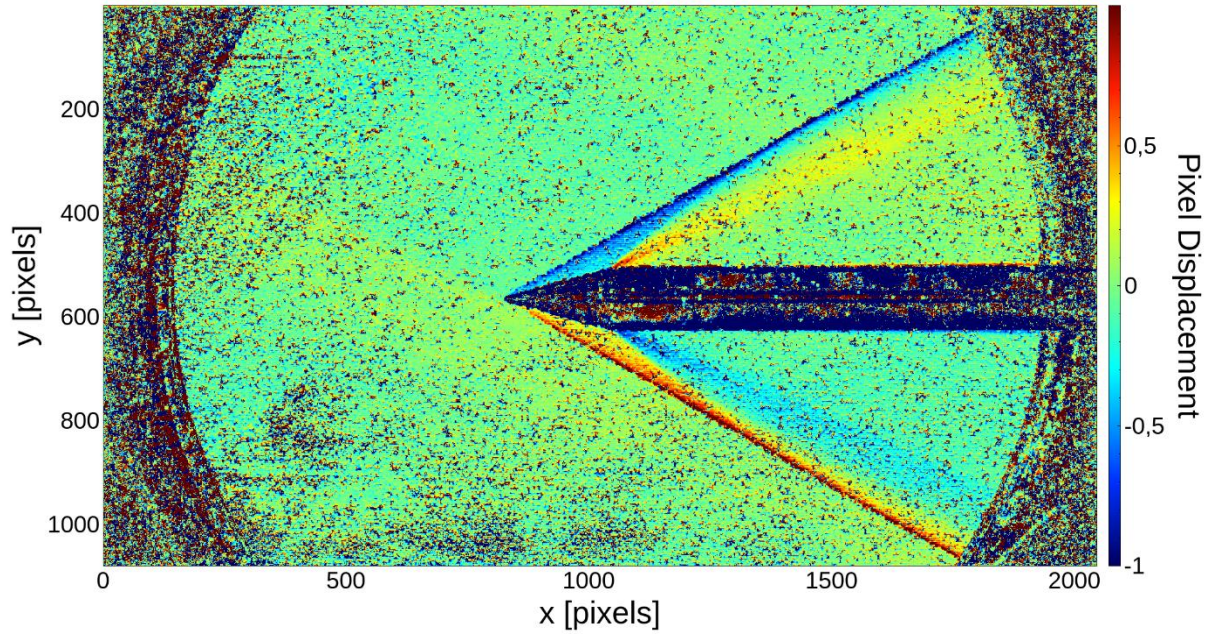


Figure 57: Lucas and Kanade algorithm, vertical pixel displacement, 5 by 5 window, single layer, speckled background

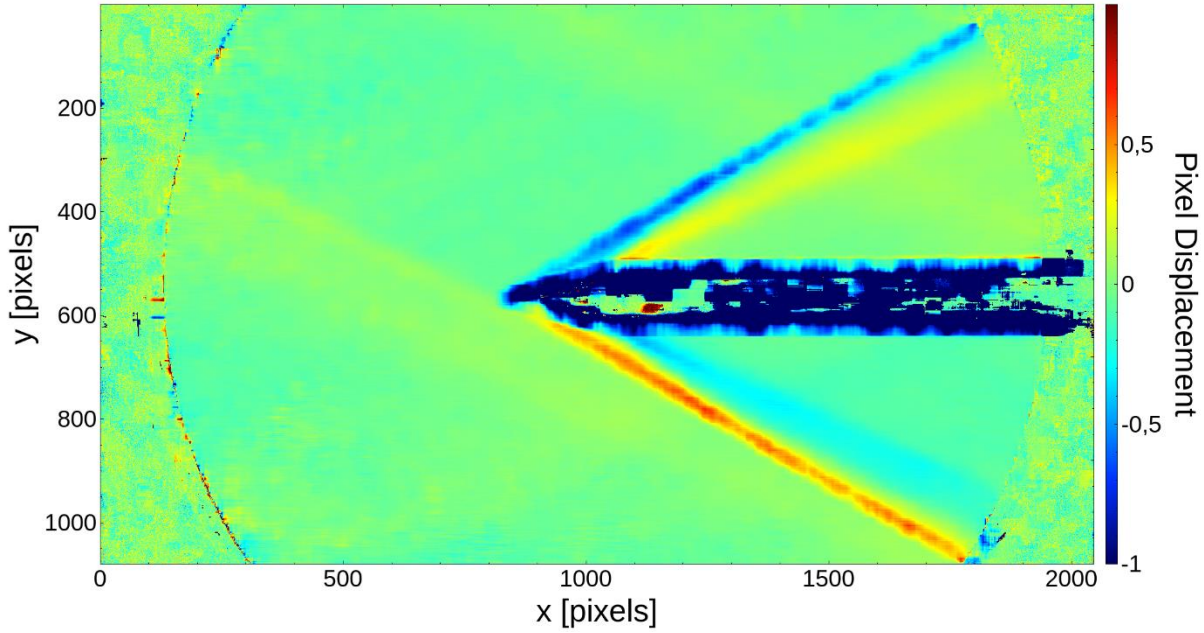


Figure 58: Lucas and Kanade algorithm, vertical pixel displacement, 33 by 33 window, single layer, speckled background

Layered processing

Single layer processed images performed better than the multiple layer processing strategy. More artefacts were present in the latter. This phenomenon can be seen when comparing Figure 59 (single layer) and Figure 60 (multiple layers). The down-sampling scale was 0.5 only as the OpenCV implementation of this algorithm does not support other scales.

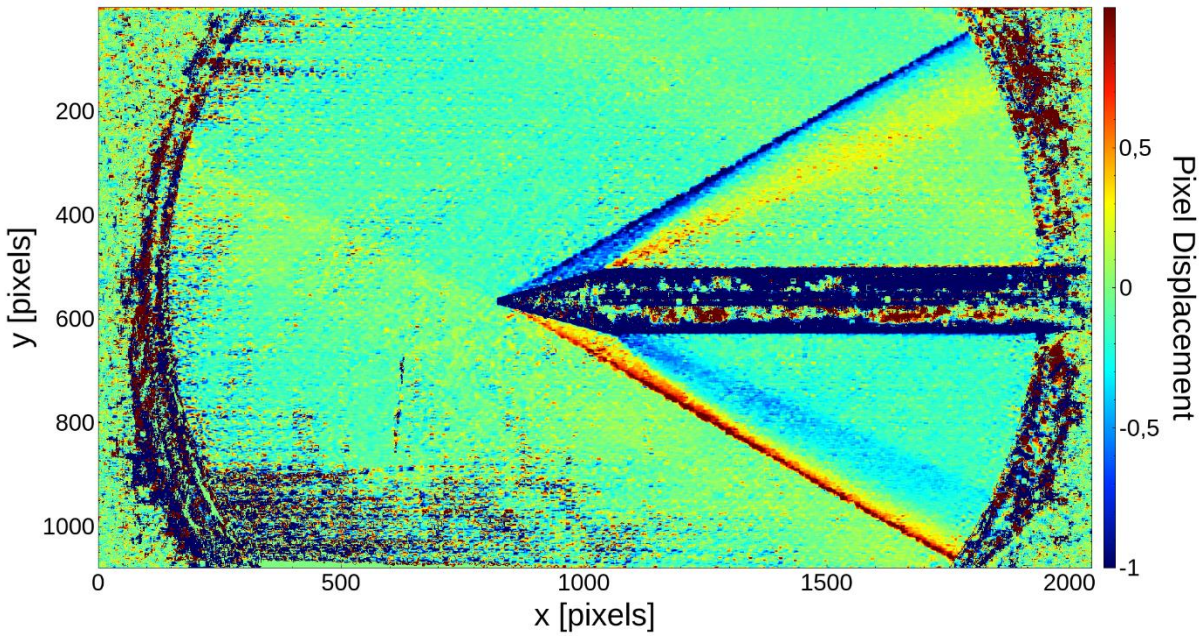


Figure 59: Lucas and Kanade algorithm, vertical pixel displacement, 9 by 9 window, single layer, wavelet background

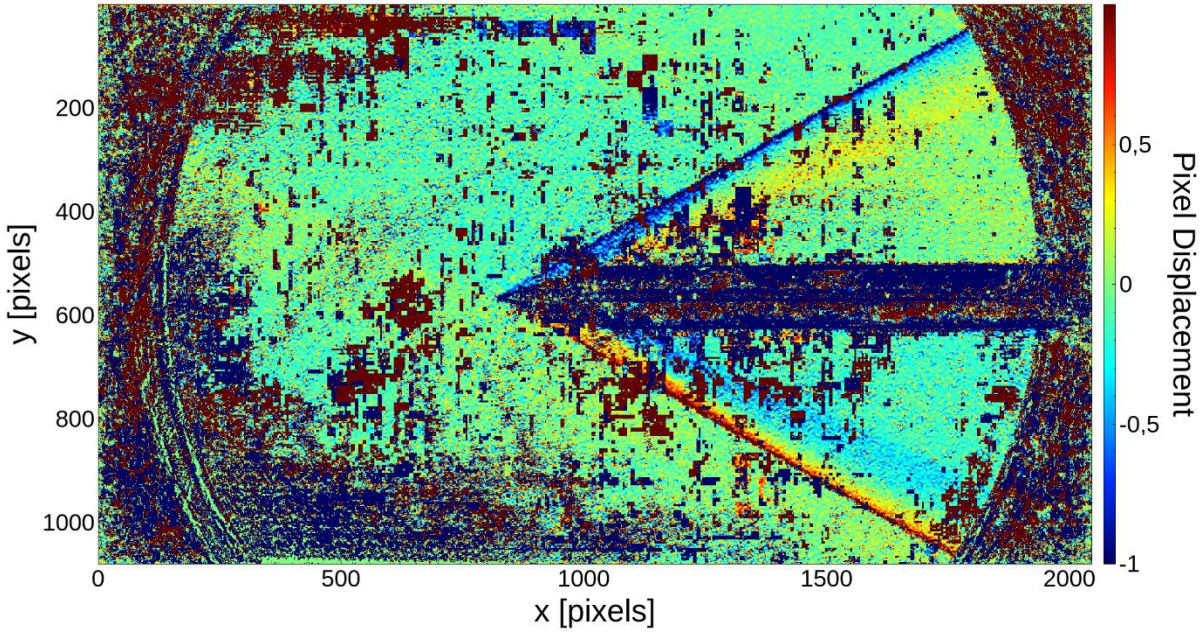


Figure 60: Lucas and Kanade algorithm, vertical pixel displacement, 9 by 9 window, 11 layers, down sample factor 0.5, wavelet background

Backgrounds

Processed speckled backgrounds showed slightly less noise than processed wavelet backgrounds. The oil present on the schlieren window was also picked up better with the wavelet backgrounds. These effects can be seen when comparing Figure 59 with Figure 60 and Figure 61 with Figure 62. At the finer scale, some regular noise can be seen in the processed wavelet background images that is not present in the processed speckled images. This can be seen comparing Figure 63 with Figure 64.

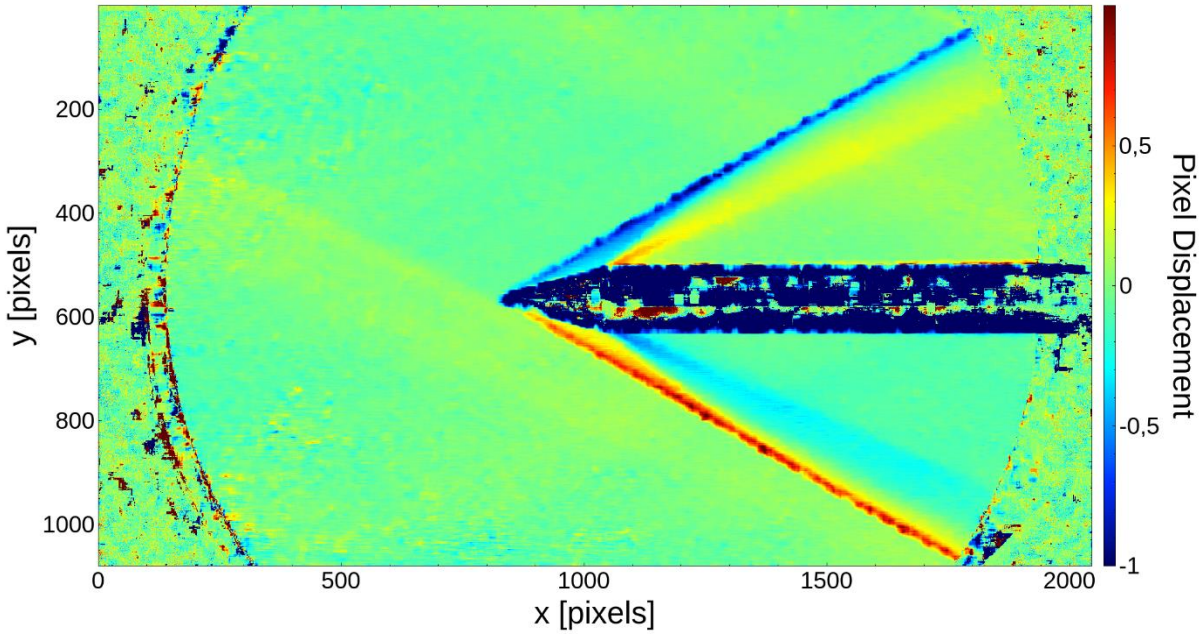


Figure 61: Lucas and Kanade algorithm, vertical pixel displacement, 17 by 17 window, single layer, speckled background

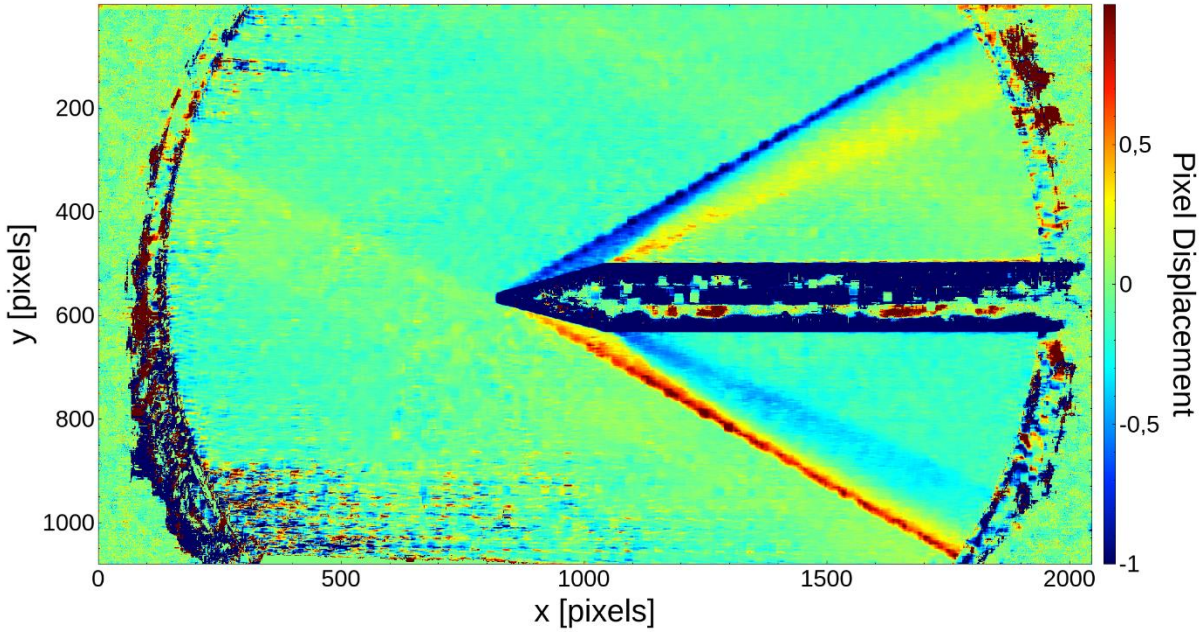


Figure 62: Lucas and Kanade algorithm, vertical pixel displacement, 17 by 17 window, single layer, wavelet background

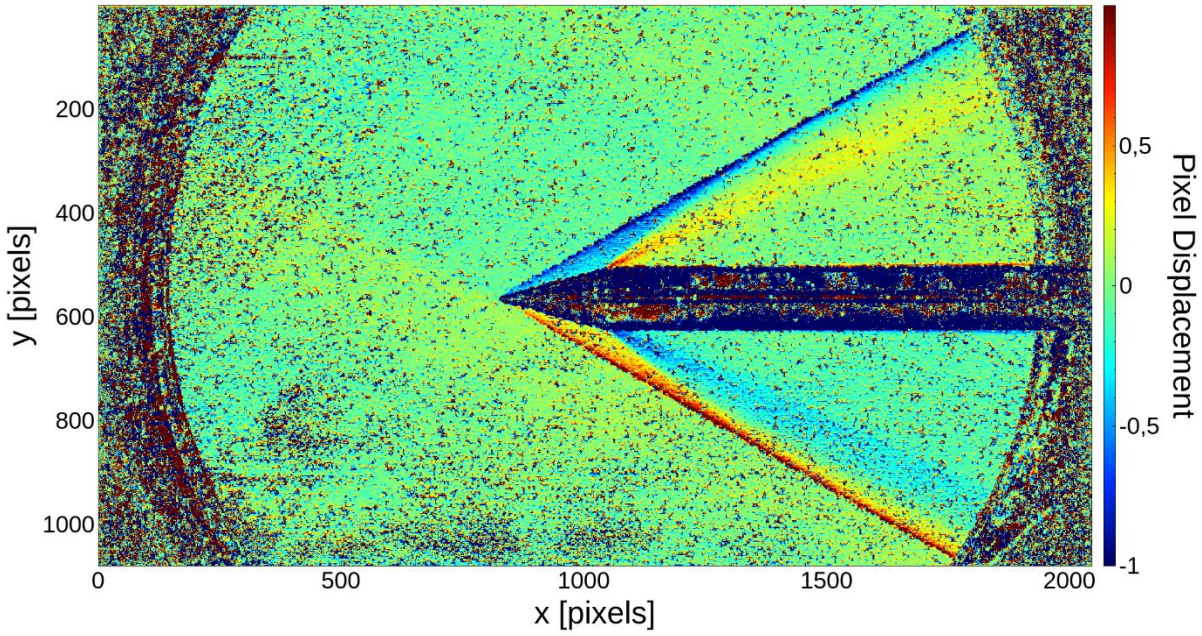


Figure 63: Lucas and Kanade algorithm, vertical pixel displacement, 5 by 5 window, single layer, speckled background

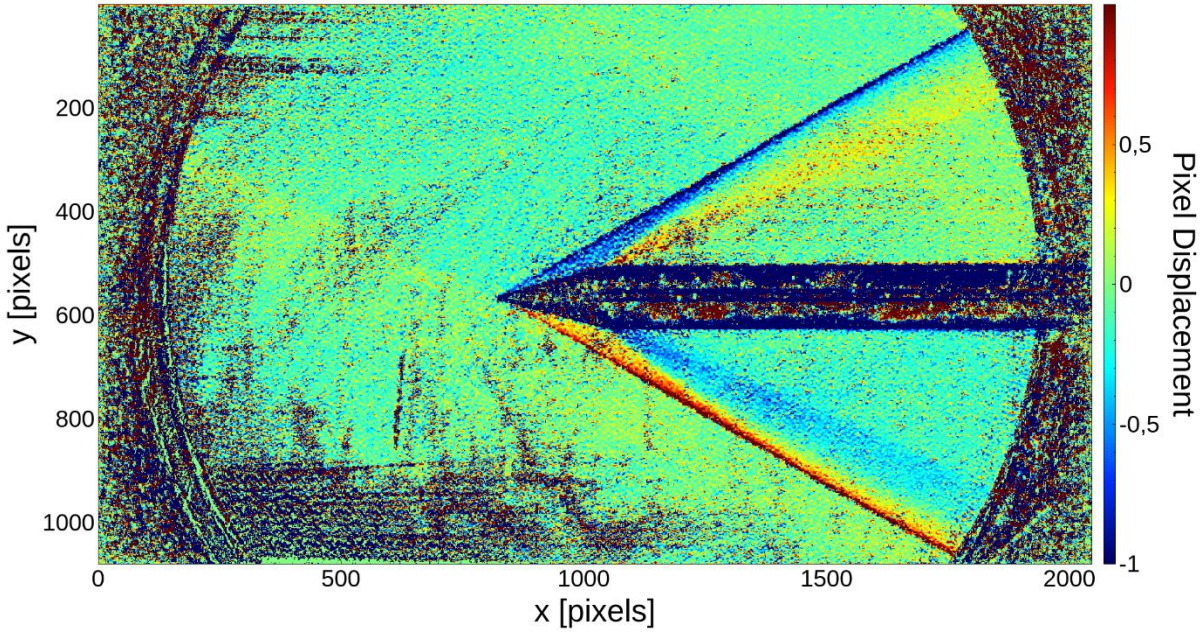


Figure 64: Lucas and Kanade algorithm, vertical pixel displacement, 5 by 5 window, single layer, wavelet background

5.1.3 Brox

The main variables that have an influence on the results from the Brox algorithm are the smoothness weighting, the gradient consistency weighting and whether a multi-resolution approach is used. If a multi-resolution approach is used, the down-sampling factor can have an influence on the results. The influence that these parameters have on the results is discussed below. Additionally, results from the two different backgrounds are compared. In general this algorithm produced BOS images with more detail and less noise than its block matching counterpart.

Smoothness weighting

Increasing the smoothness weighting decreases the noise in the processed image but at the expense of image detail. This can be seen by comparing Figure 65 with Figure 66. Figure 65 has a smoothness weighting of 0.1 and Figure 66 has a smoothness weighting of 0.4. These images are processed using a single layer approach and have a gradient consistency weighting of 50.

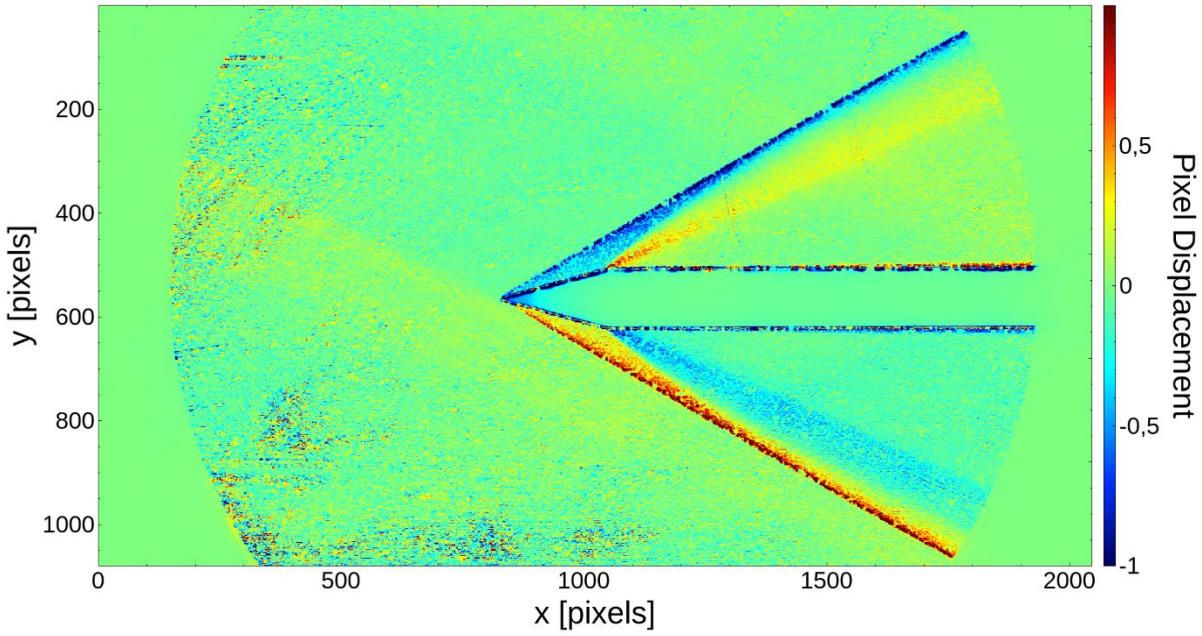


Figure 65: Brox algorithm, vertical pixel displacement, α = 0.1, γ = 50, single layer, speckled background

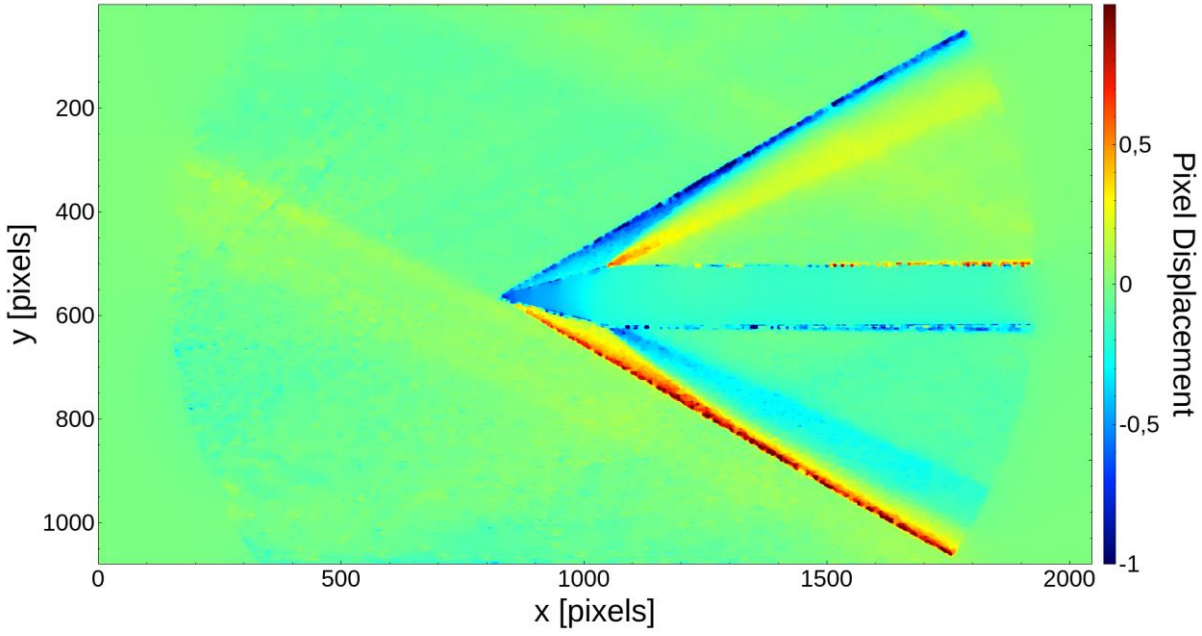


Figure 66: Brox algorithm, vertical pixel displacement, α = 0.4, γ = 50, single layer, speckled background

Gradient consistency weighting

Increasing the gradient consistency weighting improved the detail in the processed image but at the expense of additional noise. This can be seen by comparing Figure 67 and Figure 68, which have a gradient consistency of 0 and 100 respectively. The smoothness weighting was set to 0.2 and a single layer approach was used.

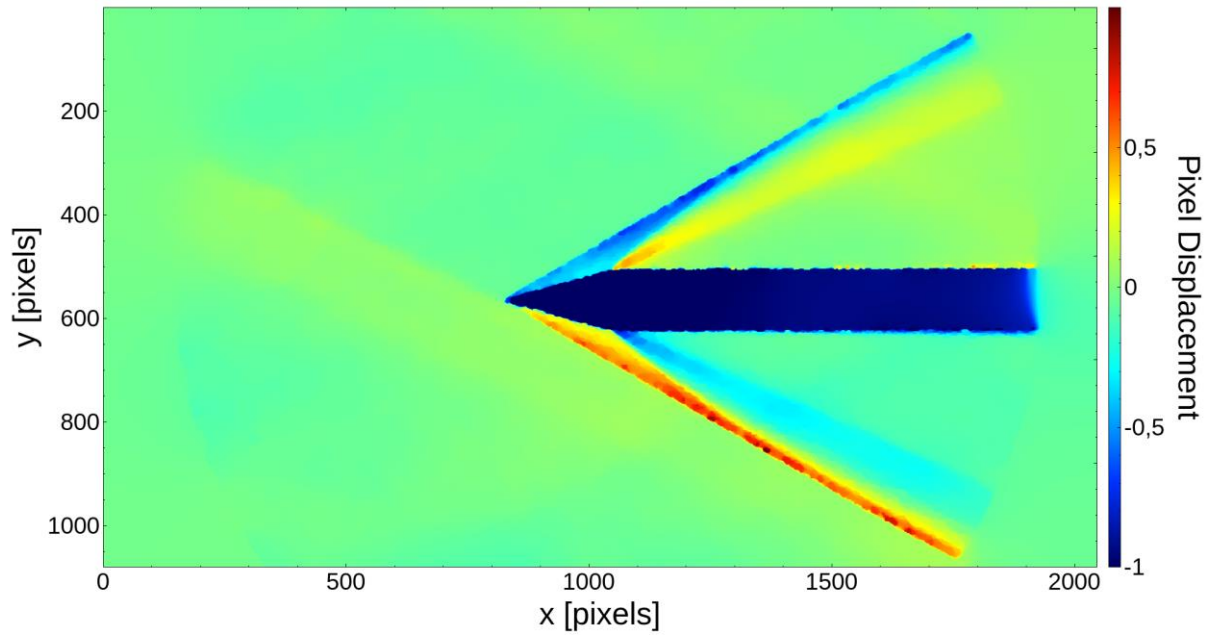


Figure 67: Brox algorithm, vertical pixel displacement, α = 0.2, γ = 0, single layer, speckled background

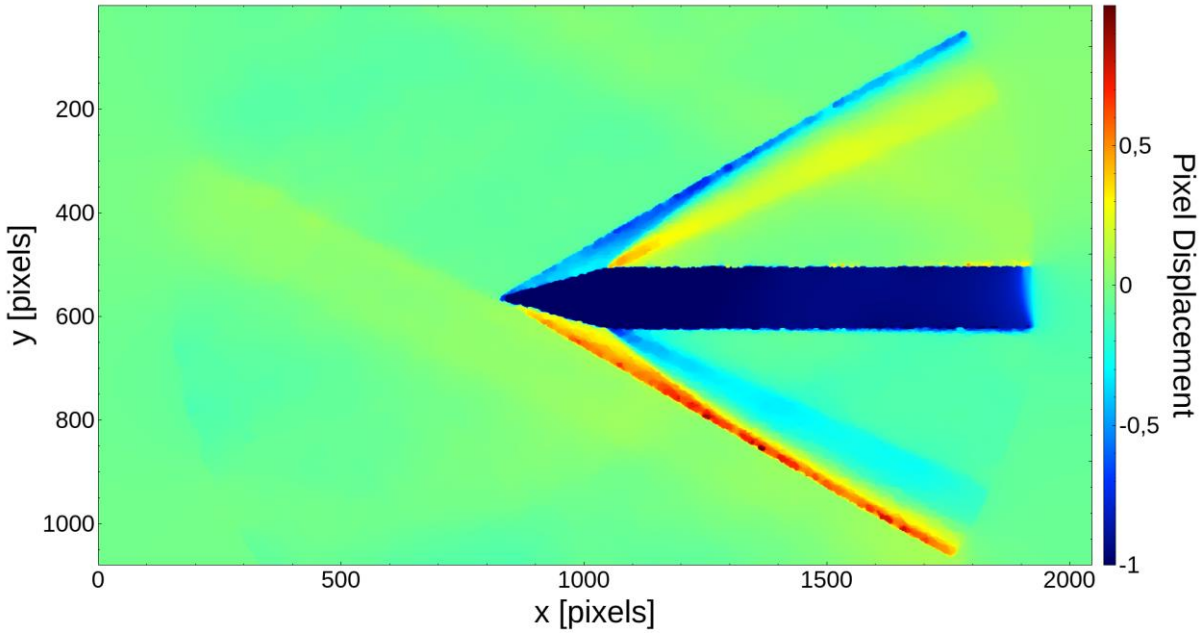


Figure 68: Brox algorithm, vertical pixel displacement, α = 0.2, γ = 100, single layer, speckled background

Layered processing

The most noticeable result of having used a multi-layered approach can be seen in the darker areas behind the shock indicating larger optical flow values. Larger artefacts are present in the multi-layered processed images, with those that used a down-sampling factor of 0.5 being more prominent.

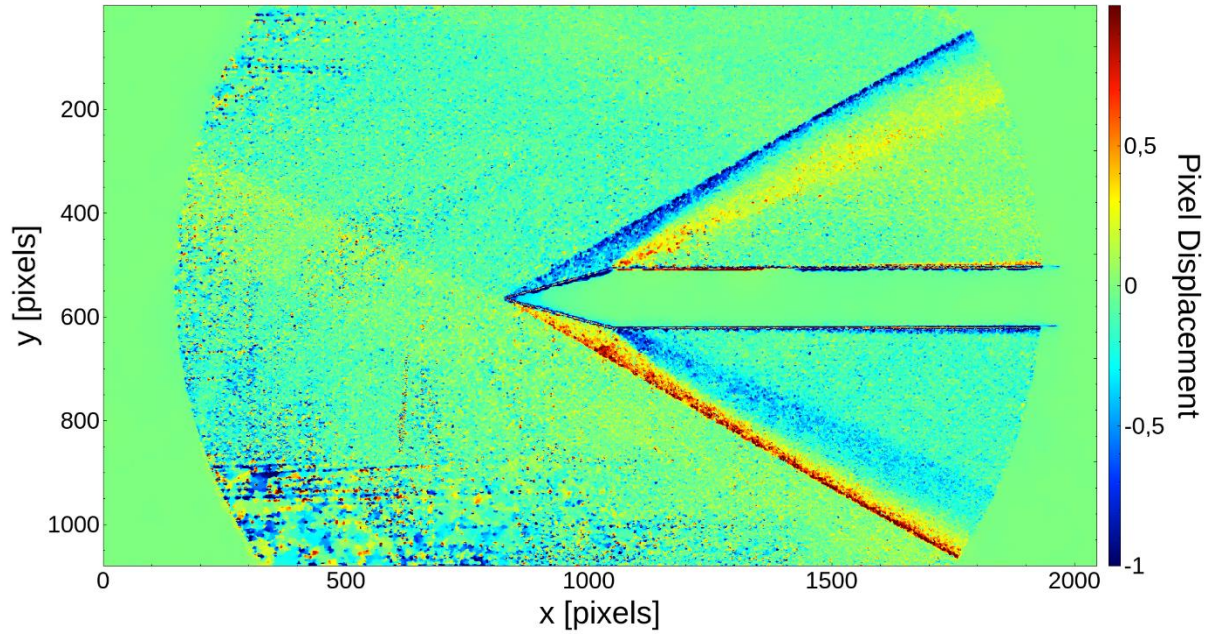


Figure 69: Brox algorithm, vertical pixel displacement, α = 0.1, γ = 100, single layer, wavelet background

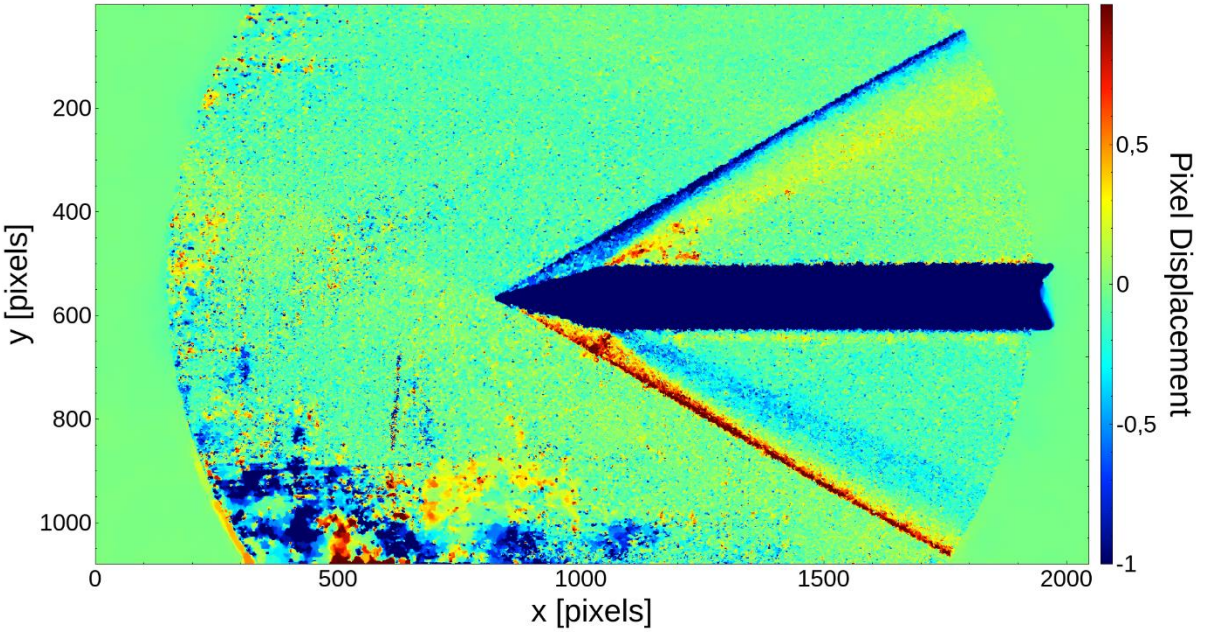


Figure 70: Brox algorithm, vertical pixel displacement, α = 0.1, γ = 100, 11 layers, down sample factor = 0.5, wavelet background

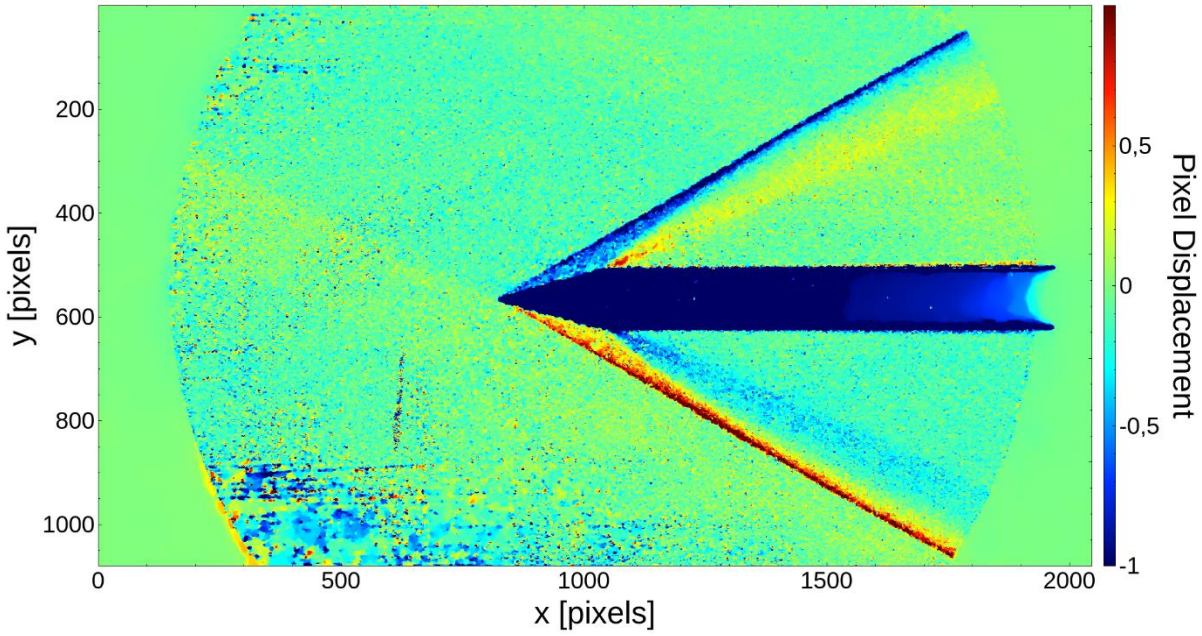


Figure 71: Brox algorithm, vertical pixel displacement, α = 0.1, γ = 100, 35 layers, down sample factor = 0.8, wavelet background

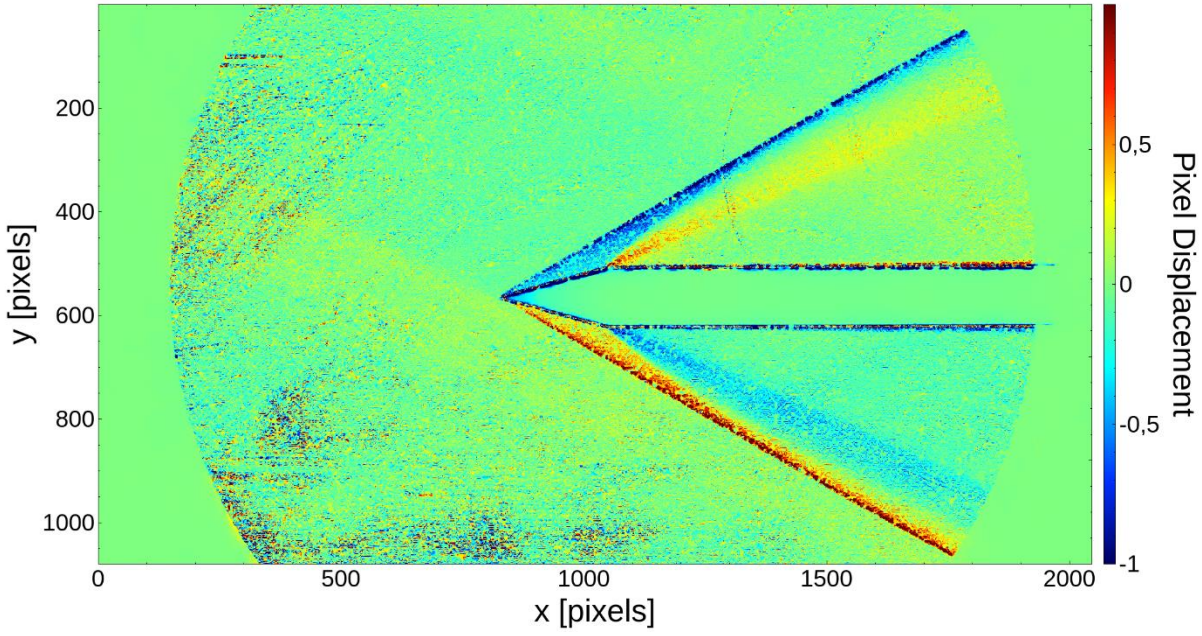


Figure 72: Brox algorithm, vertical pixel displacement, α = 0.1, γ = 100, single layer, speckled background

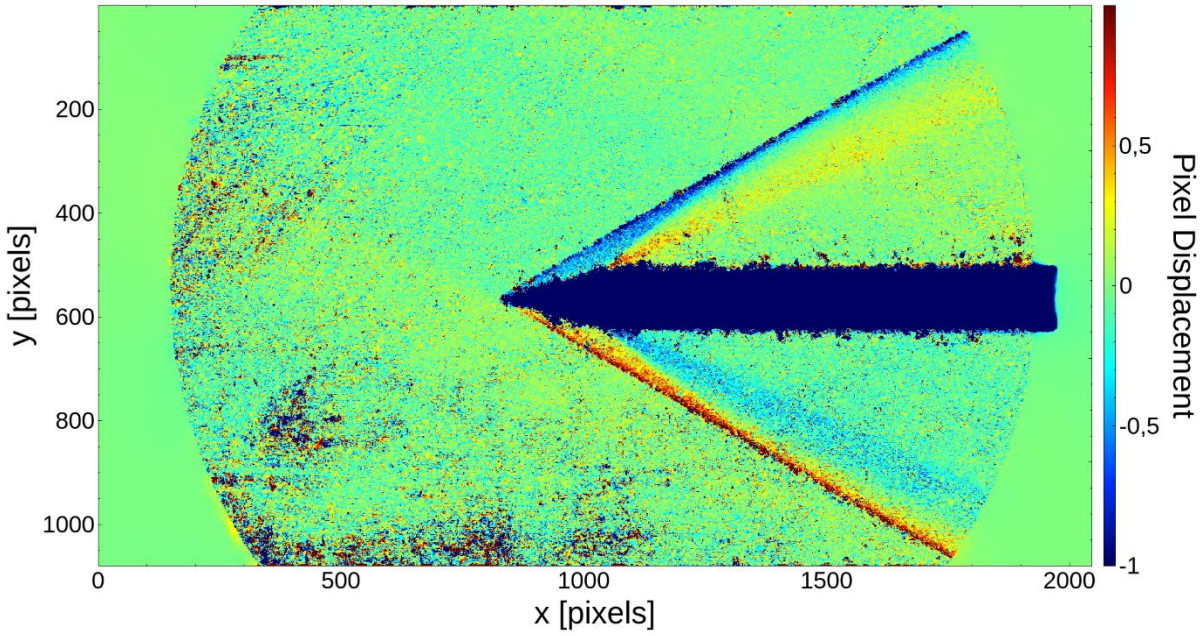


Figure 73: Brox algorithm, vertical pixel displacement, α = 0.1, γ = 100, 11 layers, down sample factor = 0.5, speckled background

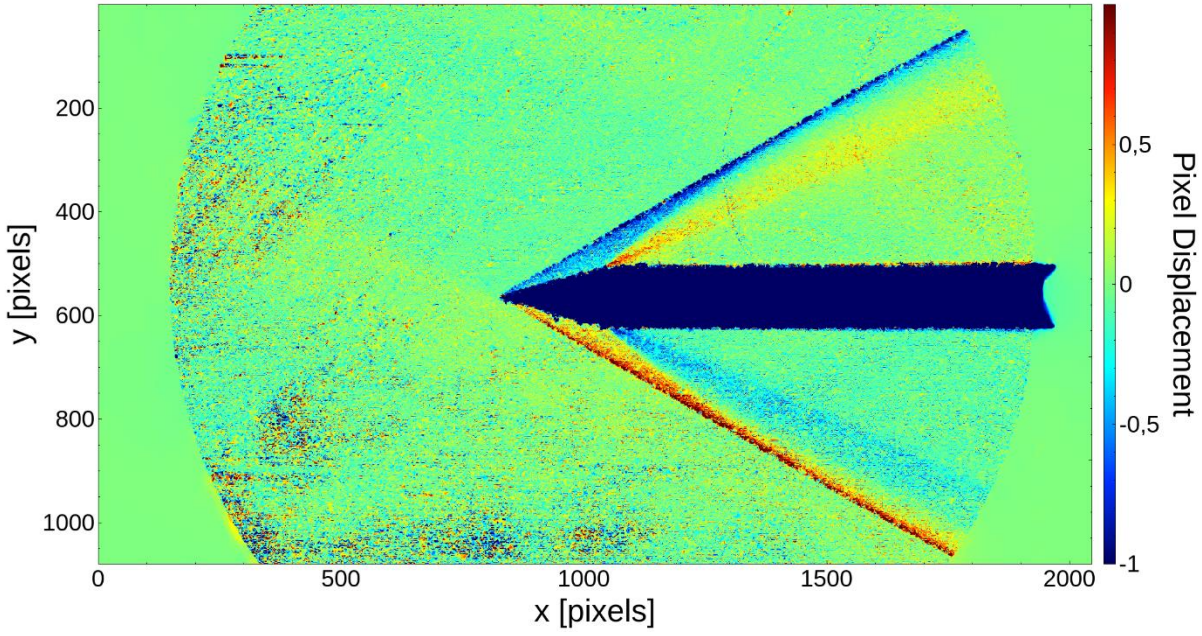


Figure 74: Brox algorithm, vertical pixel displacement, α = 0.1, γ = 100, 35 layers, down sample factor = 0.8, speckled background

Backgrounds

Consistently, for the same settings, the speckled background gave finer detail than its wavelet counterpart did. The character of the artefacts changed depending on the algorithm settings.

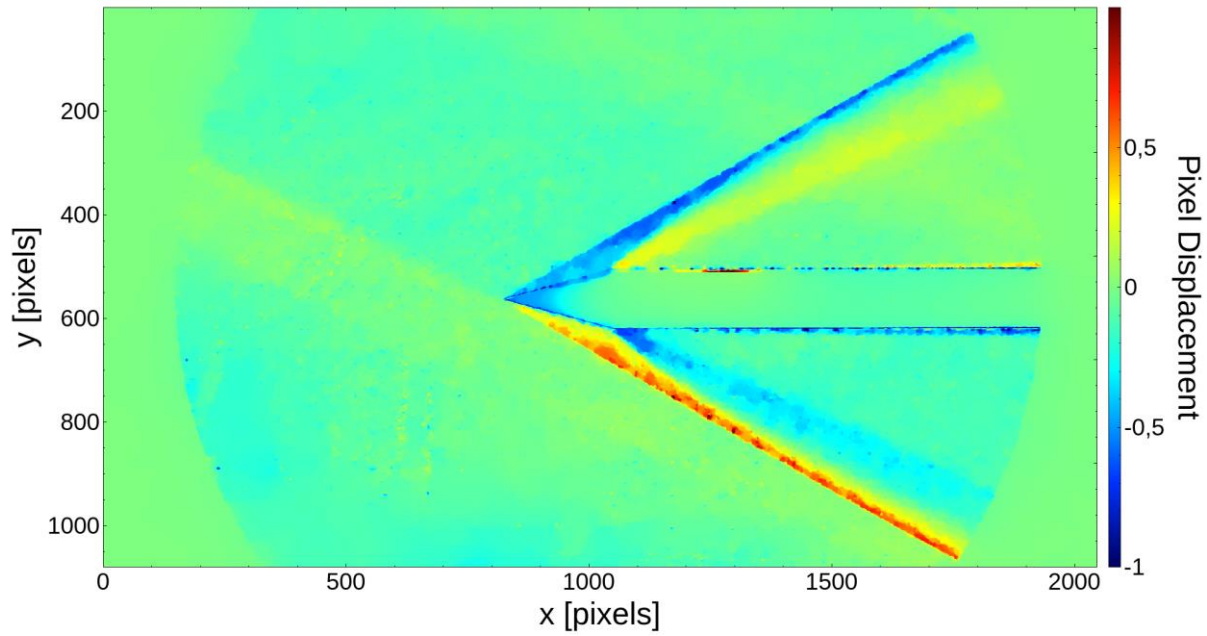


Figure 75: Brox algorithm, vertical pixel displacement, α = 0.2, γ = 50, single layer, wavelet background

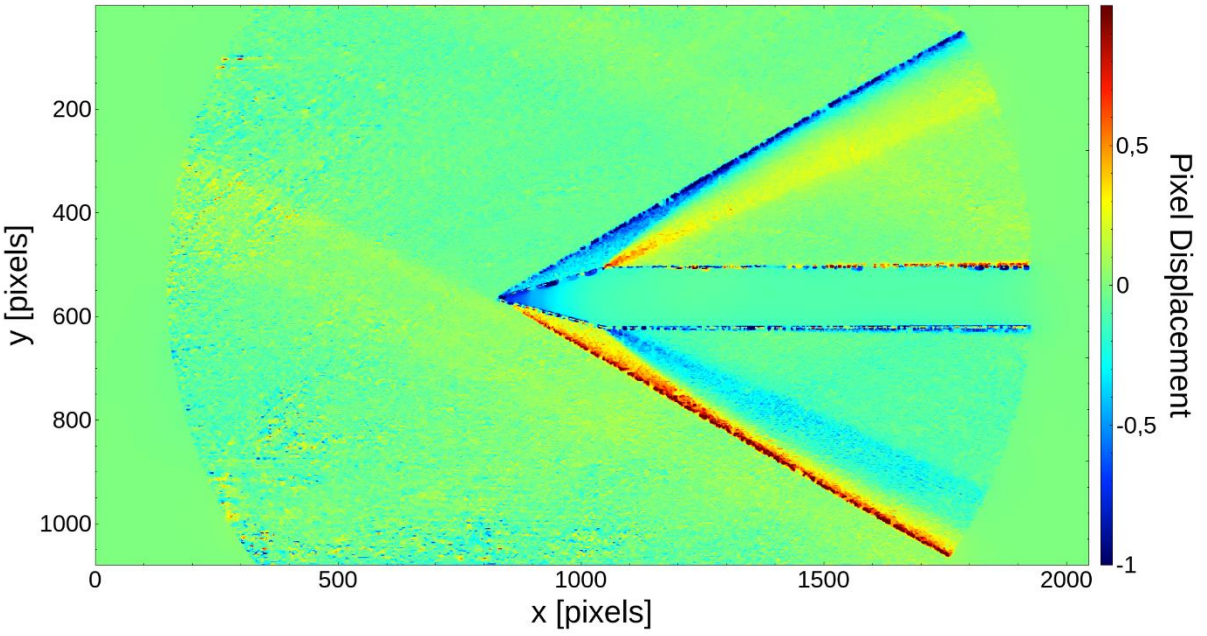


Figure 76: Brox algorithm, vertical pixel displacement, α = 0.2, γ = 50, single layer, speckled background

5.1.4 Farnebäck

The method used by Farnebäck uses a local polynomial fit to the greyscale values within the first window and a second polynomial fit of the calculated flow within the second window. The size of each of these two windows, whether single layer or multi-layer processing was used, and the down-sampling factor are all parameters that were varied. The weighting distribution used the recommended settings. The variation of the weighting settings is left for future study.

Window sizes

The chosen window sizes show trends that depend on both, so these are discussed together. In general, increasing either window size reduces noise at the expense of image detail. Increasing the greyscale fit window size; with smaller optical flow fit window size, improves the signal with little to no perceivable decrease in detail. Compare Figure 77 and Figure 78 to see this. Using the larger optical flow fit window with larger greyscale fit windows has a disproportionately larger effect on reducing detail, accompanied by a decrease in noise. Compare the difference between Figure 77 and Figure 79 with the difference between Figure 80 and Figure 81 to see this.

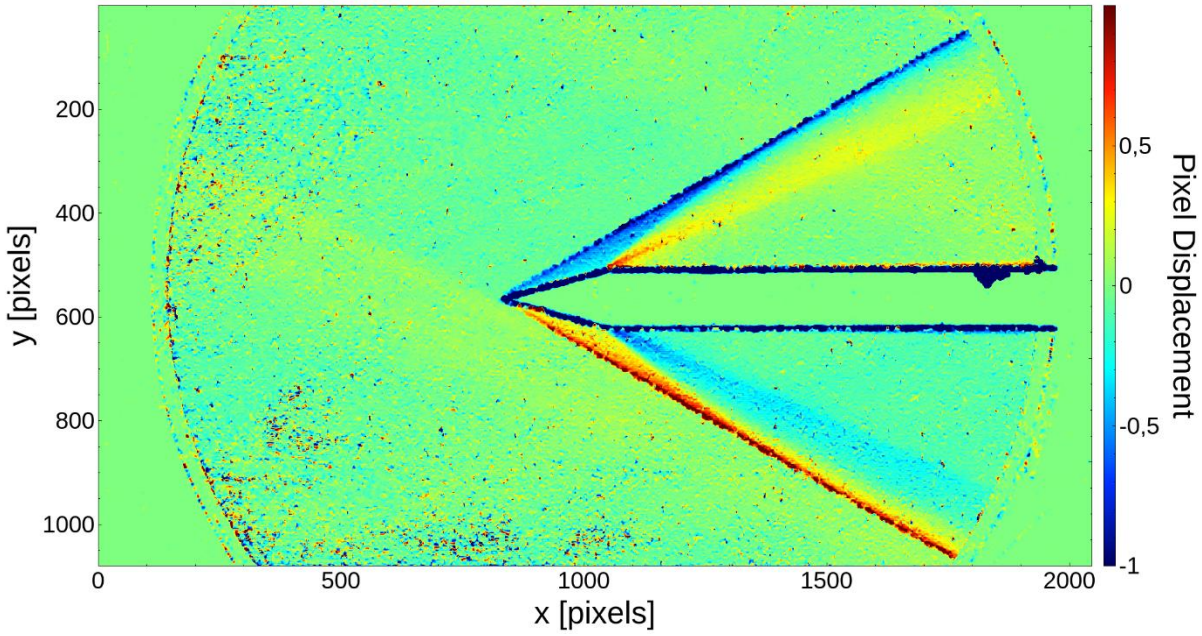


Figure 77: Farnebäck algorithm, vertical pixel displacement, 5 by 5 greyscale window, 4 by 4 optical flow window, single layer, speckled background

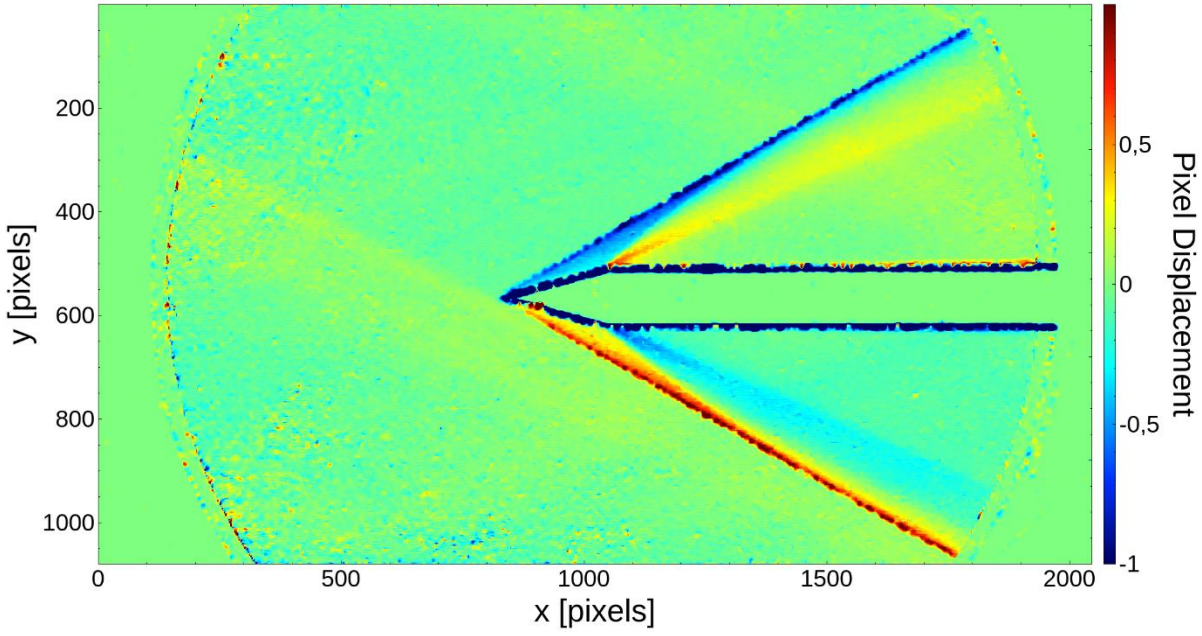


Figure 78: Farnebäck algorithm, vertical pixel displacement, 5 by 5 greyscale window, 32 by 32 optical flow window, single layer, speckled background

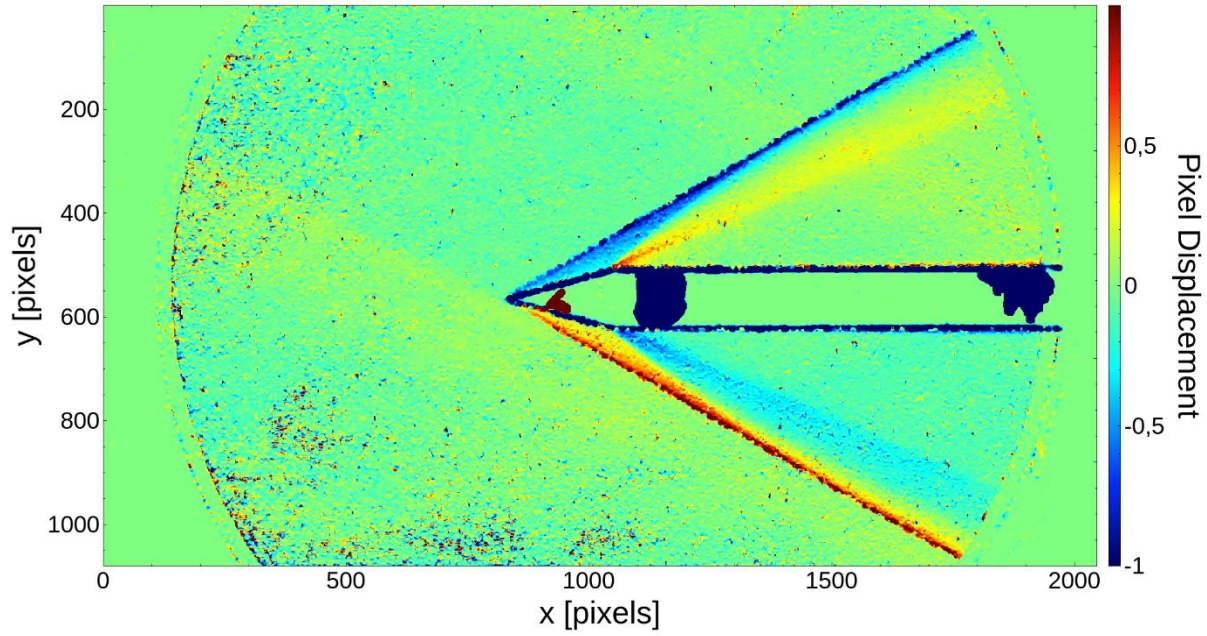


Figure 79: Farnebäck algorithm, vertical pixel displacement, 7 by 7 greyscale window, 4 by 4 optical flow window, single layer, speckled background

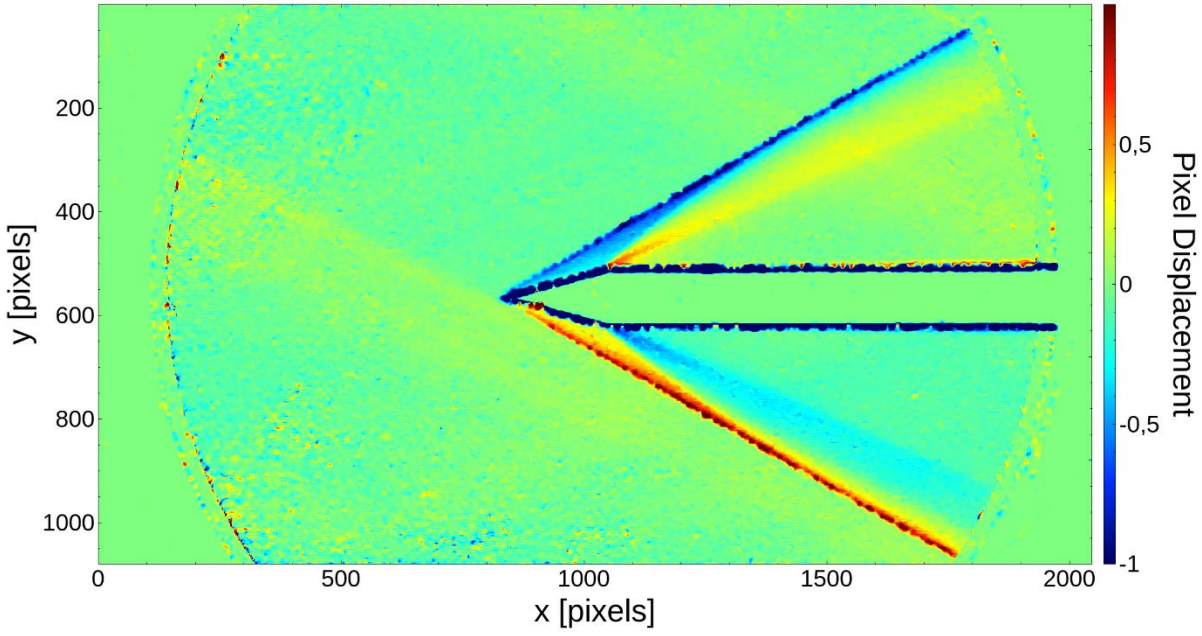


Figure 80: Farnebäck algorithm, vertical pixel displacement, 5 by 5 greyscale window, 32 by 32 optical flow window, single layer, speckled background

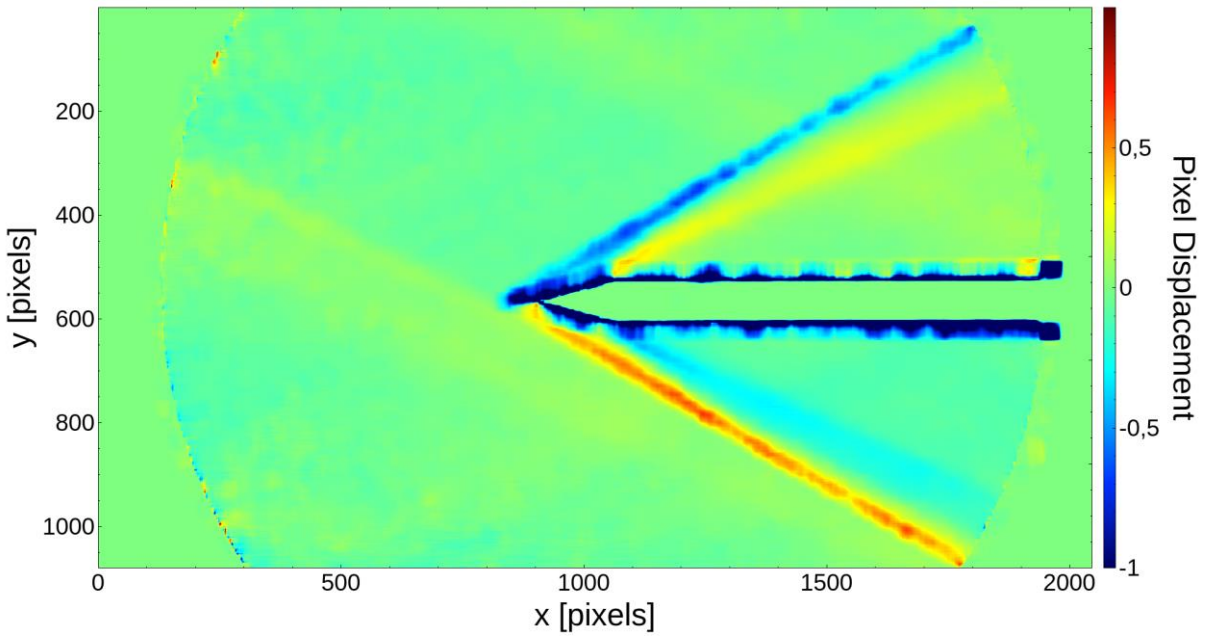


Figure 81: Farnebäck algorithm, vertical pixel displacement, 7 by 7 greyscale window, 32 by 32 optical flow window, single layer, speckled background

Layered processing

Some artefacts were evident in the flow field of the multi-layered processed images that were not present in the single layered processed images. These tended to be less severe when a down-sampling factor of 0.8 was used compared to that of 0.5.

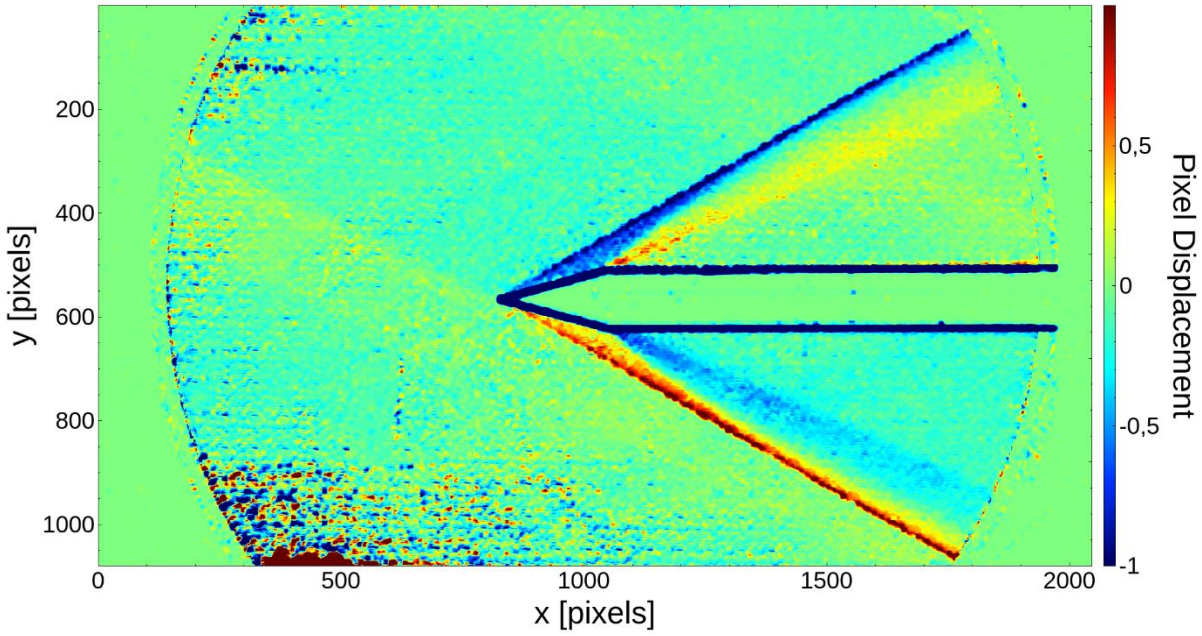


Figure 82: Farnebäck algorithm, vertical pixel displacement, 5 by 5 greyscale window, 8 by 8 optical flow window, single layer, wavelet background

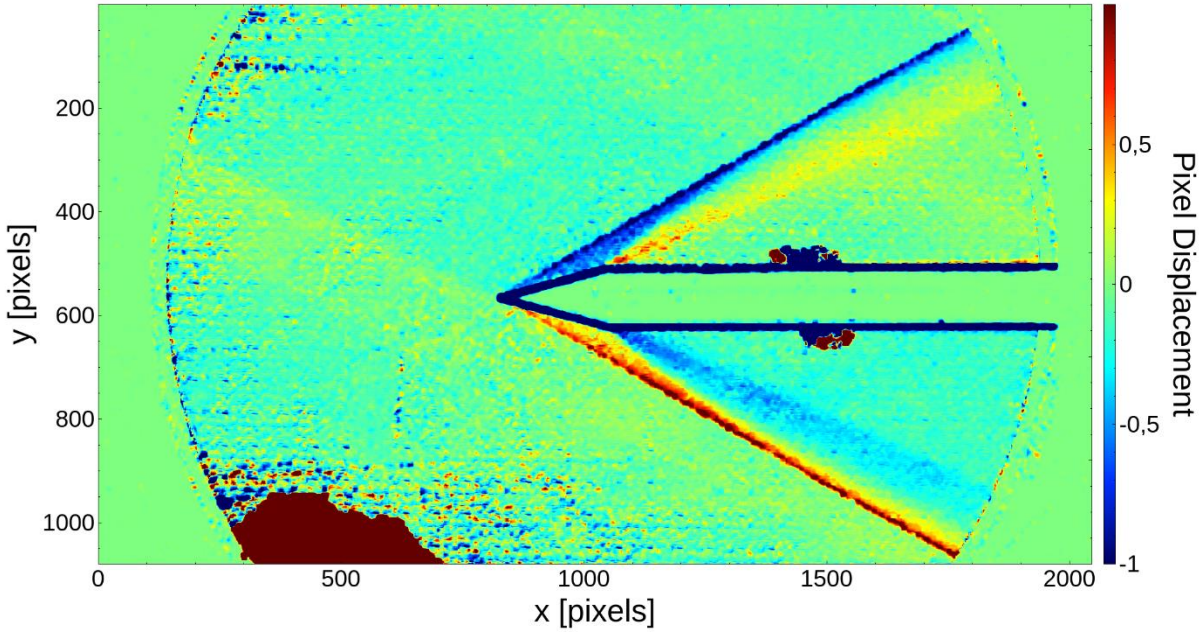


Figure 83: Farnebäck algorithm, vertical pixel displacement, 5 by 5 greyscale window, 8 by 8 optical flow window, 11 layers, down sample factor = 0.5, wavelet background

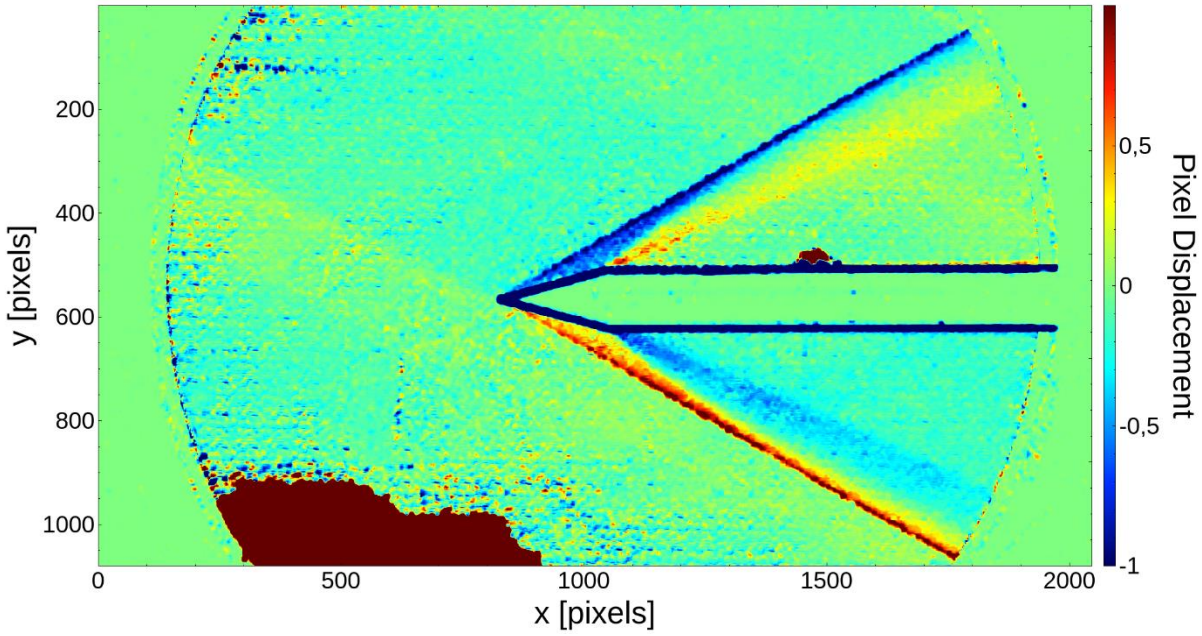


Figure 84: Farnebäck algorithm, vertical pixel displacement, 5 by 5 greyscale window, 8 by 8 optical flow window, 35 layers, down sample factor = 0.8, wavelet background

Backgrounds

For the same settings, the wavelet background gives less detail than the speckled background. Compare Figure 85 and Figure 86 to see this.

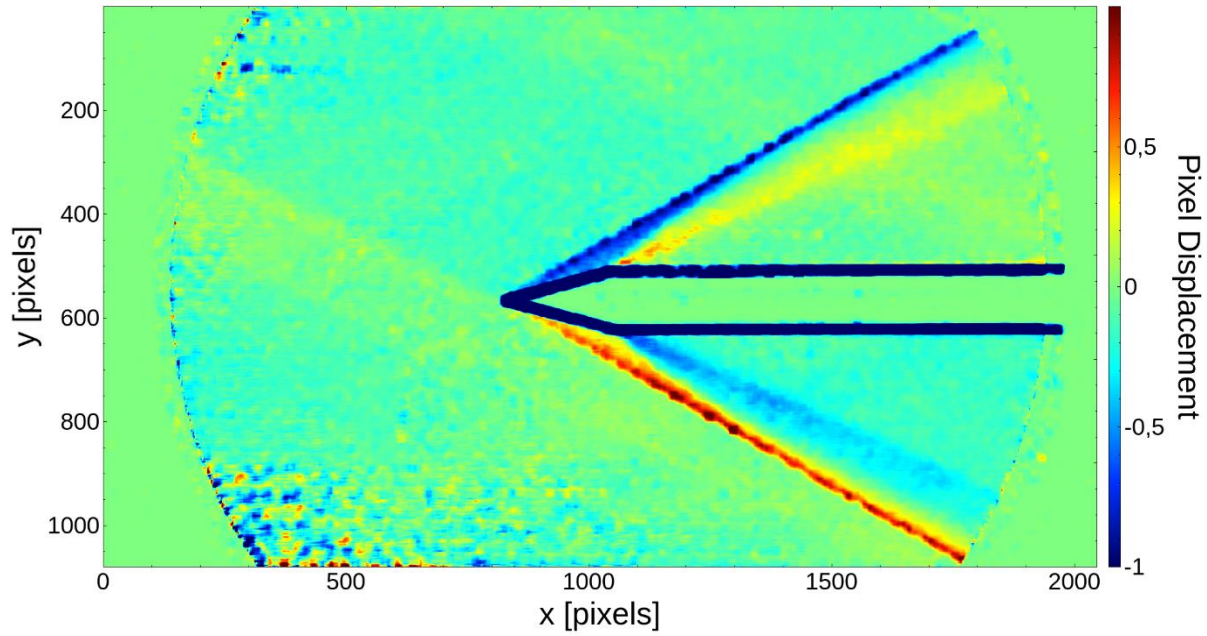


Figure 85: Farnebäck algorithm, vertical pixel displacement, 5 by 5 greyscale window, 16 by 16 optical flow window, single layer, wavelet background

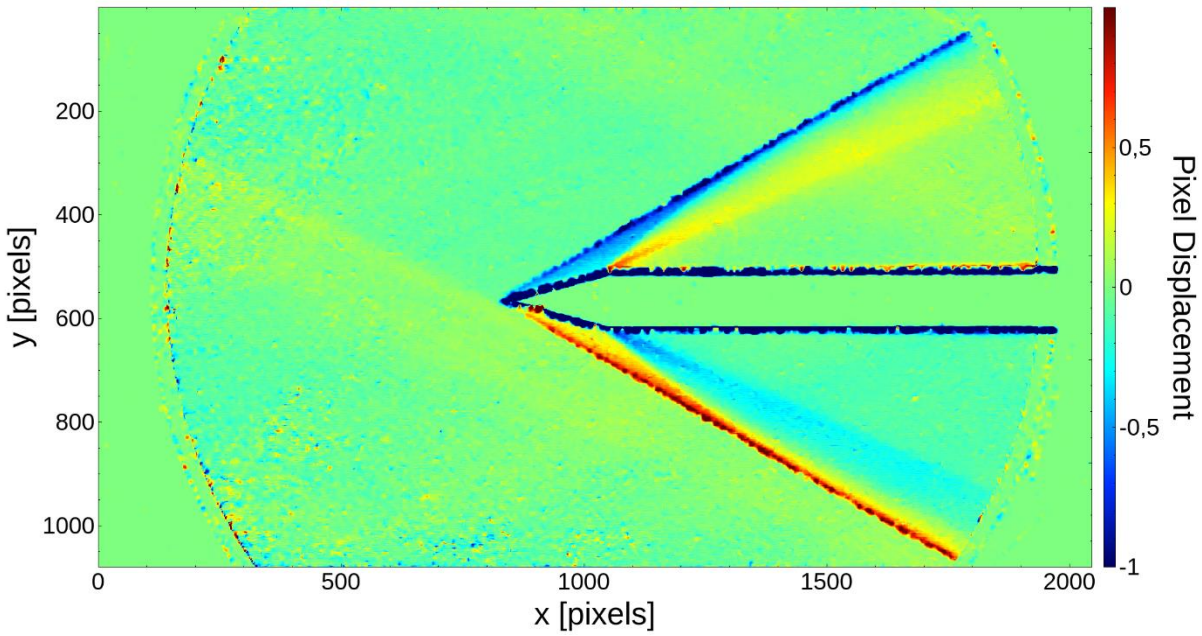


Figure 86: Farnebäck algorithm, vertical pixel displacement, 5 by 5 greyscale window, 16 by 16 optical flow window, single layer, speckled background

5.1.5 TV-L1

The parameters considered in this study for the TV-L1 algorithm were the greyscale consistency weighting, whether or not a multi-layered processing strategy was used and the down-sampling factor.

Greyscale consistency weighting

Increasing the greyscale consistency weighting improved the detail in the image at the expense of a noisier image. This can be seen by comparing Figure 87 with Figure 88.

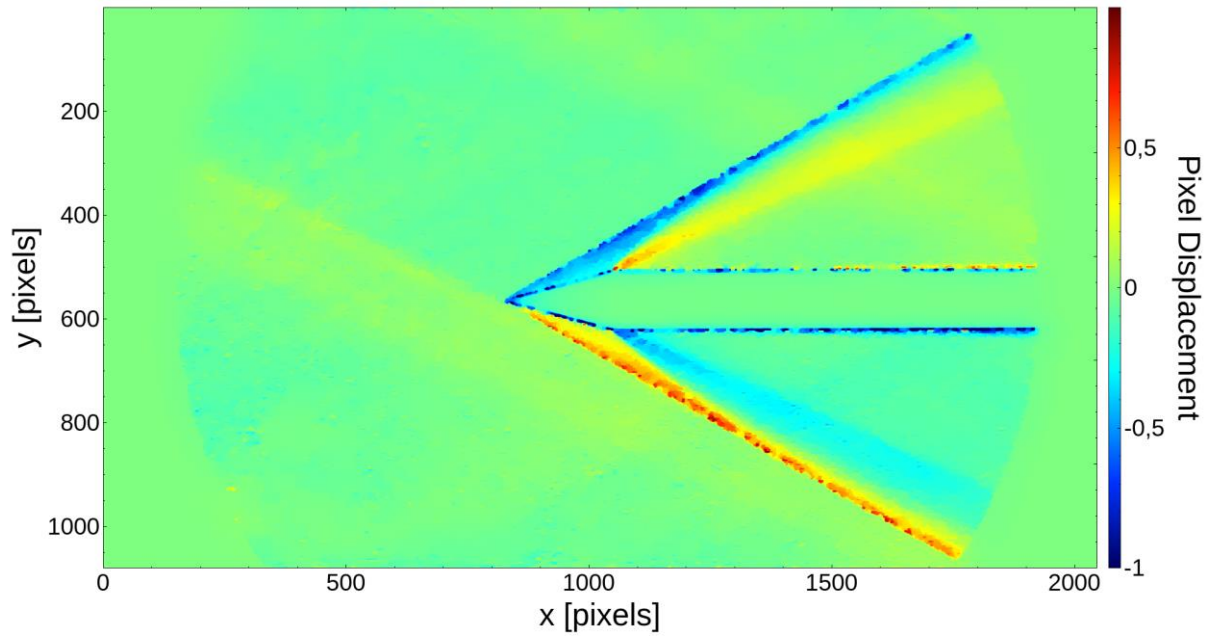


Figure 87: TV-L1 algorithm, vertical pixel displacement, λ = 0.08, single layer, speckled background

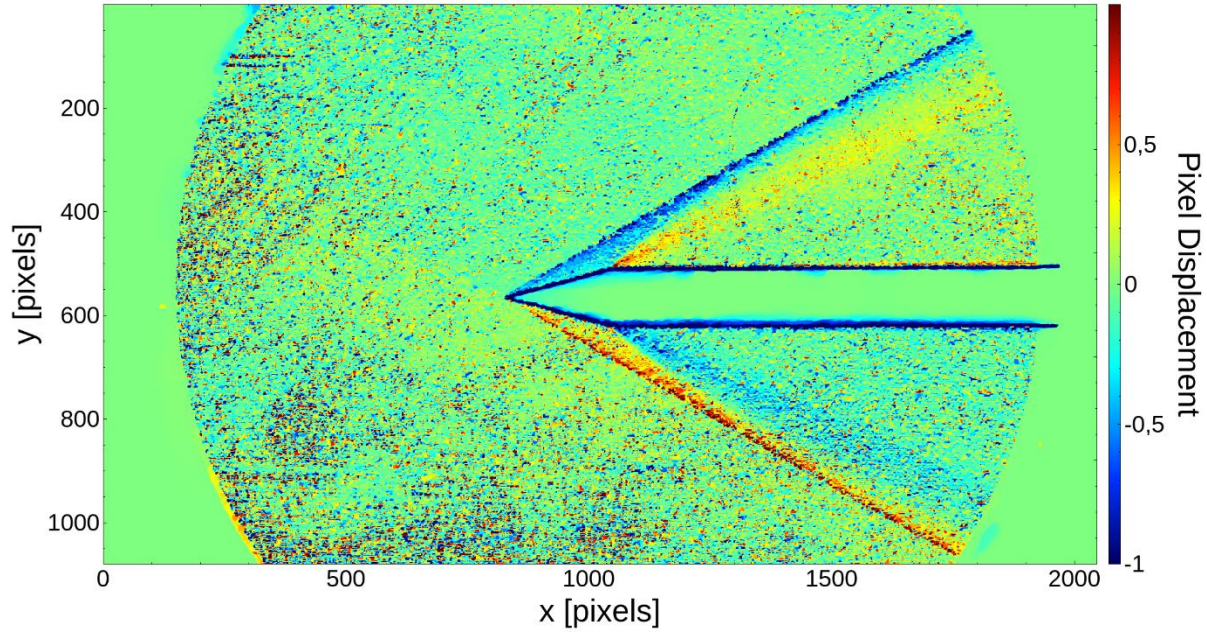


Figure 88: TV-L1 algorithm, vertical pixel displacement, λ = 0.32, single layer, speckled background

Layered processing

Using a multi-layered processing approach introduced artefacts into the flow field. These artefacts were worse for the 0.5 down-sampling factor when compared to the 0.8 down sampling factor.

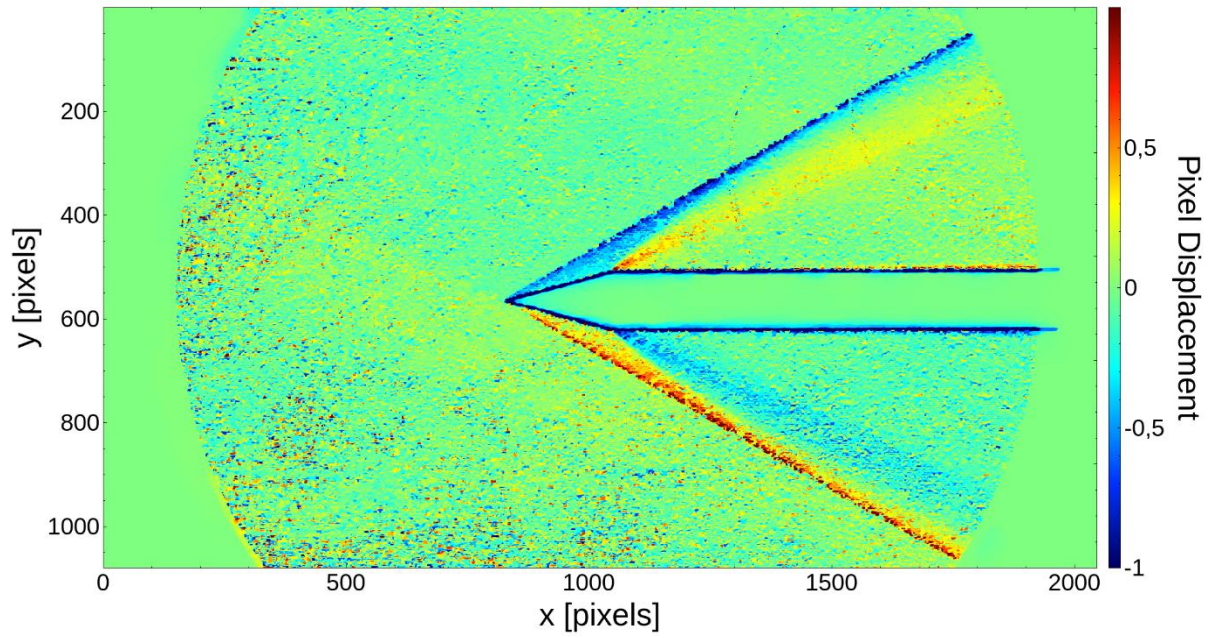


Figure 89: TV-L1 algorithm, vertical pixel displacement, λ = 0.16, single layer, speckled background

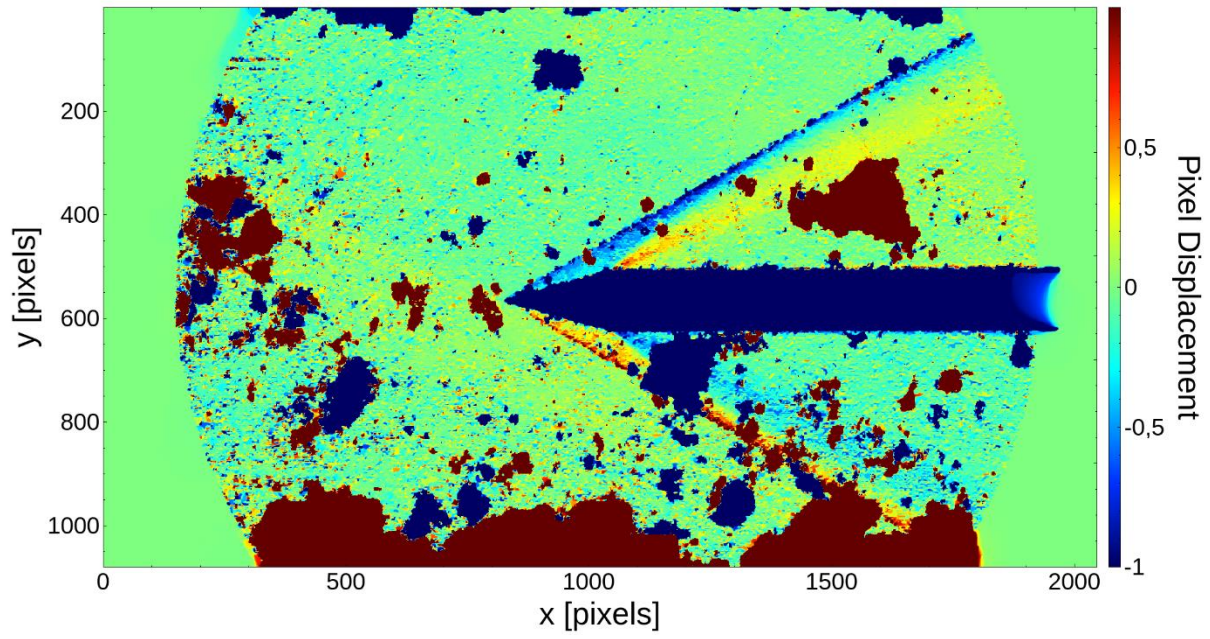


Figure 90: TV-L1 algorithm, vertical pixel displacement, λ = 0.16, 11 layers, down sample factor = 0.5, speckled background

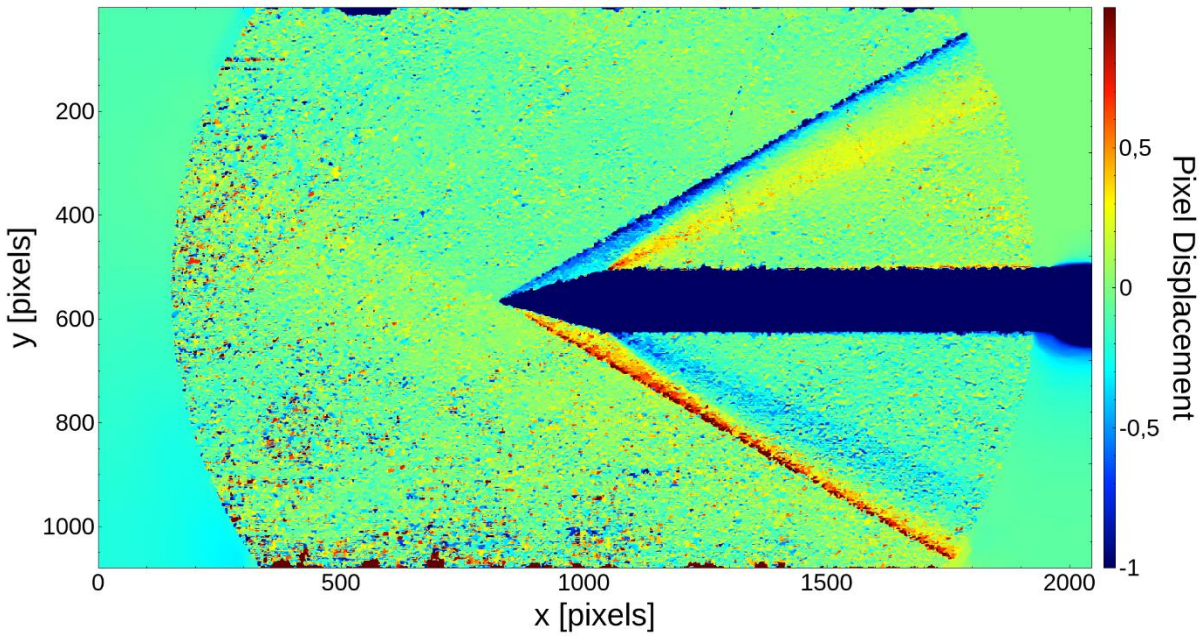


Figure 91: TV-L1 algorithm, vertical pixel displacement, λ = 0.16, 35 layers, down sample factor = 0.8, speckled background

Backgrounds

For the same settings, the wavelet background gave less detail but produced less noise than the speckled background. This can be seen by comparing Figure 92 and Figure 93.

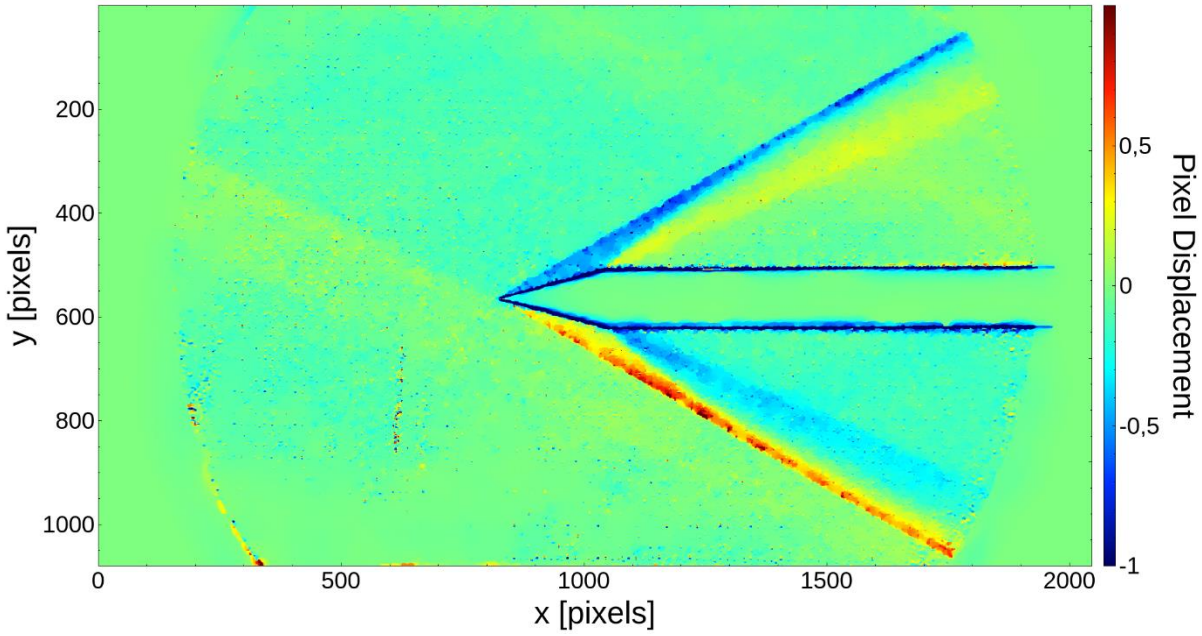


Figure 92: TV-L1 algorithm, vertical pixel displacement, λ = 0.16, single layer, wavelet background

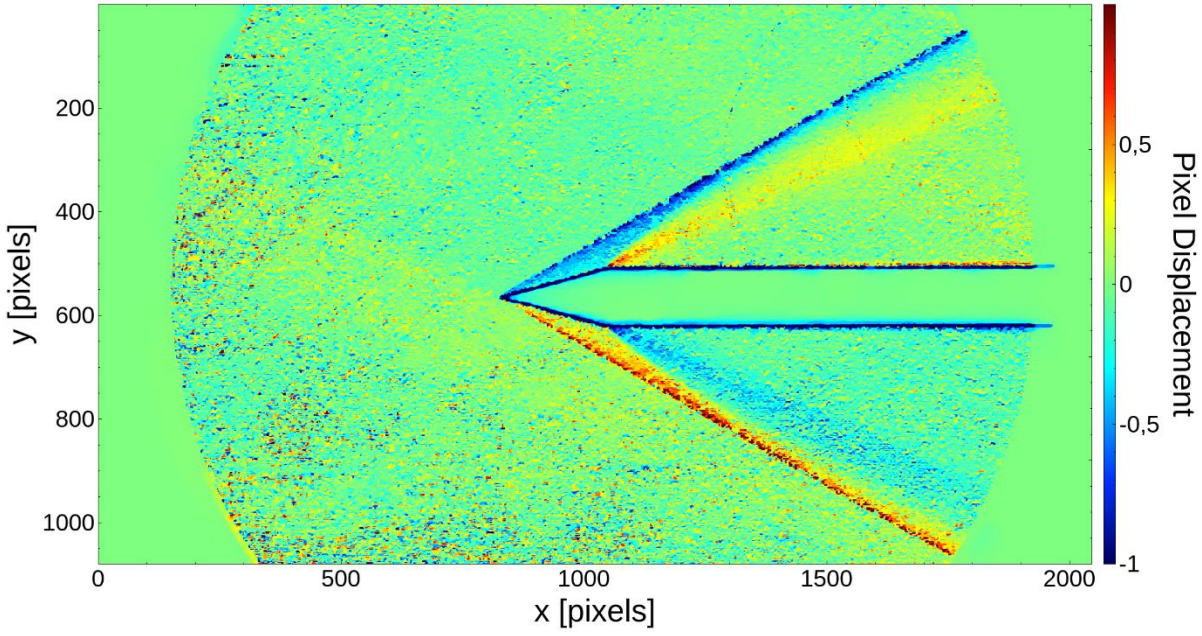


Figure 93: TV-L1 algorithm, vertical pixel displacement, λ = 0.16, single layer, wavelet background

5.2 General differences between backgrounds

In general, the wavelet backgrounds produced BOS images with less detail while speckled background produce images with more noise.

For the wavelet background processed images, systematic patterns were evident in some of the optical flow images. Because the wavelet backgrounds were created by tiling a wavelet image, it is reasonable to infer that these systematic patterns are because of the details in the background image. The aperture problem indicates that these are likely to be areas where there is little greyscale gradient information available.

5.3 The compromise between noise rejection and detail

In general, there is a compromise between noise rejection and image detail. The more one tries to remove noise from the image, the less detail is seen. This is analogous to filtering a noisy time based signal. Table 18 shows the various optical flow algorithms, their parameters, and their influence on noise rejection and detail.

Table 18: Optical flow parameters effect on noise rejection and detail

Algorithm	Increasing Parameter	Noise Rejection	Detail
Block Matching	Window Size	Increases	Decreases
Lucas and Kanade	Window Size	Increases	Decreases
Brox	Smoothness Weight	Increases	Decreases
Brox	Gradient Consistency Weight	Decreases	Increases
Farnebäck	Window Size	Increases	Decreases
Farnebäck	Window2 Size	Increases	Decreases
TVL1	Greyscale Consistency Weight	Decreases	Increases

5.4 The consequences of the aperture problem

When considering how backgrounds will perform, the aperture problem needs to be considered. Rather than casting the problem of background design as one of placing speckles on the background, one can rather cast it as one of placing edges and corners. Corners provide the most useful information regarding image movement since they locate the background element in two directions, while edges provide movement detail in one direction. Areas of constant intensity produce no information regarding movement. The problem then becomes one of producing as many high quality corners as possible without producing ambiguous results, for instance by creating a checkerboard pattern of resolution smaller than the maximum movement of the image point being tracked. Multi-resolution design strategies in conjunction with multi-resolution optical flow algorithms can also be considered but with caution as it has been shown here that multi-resolution strategies can add instability to the algorithms, creating artefacts that are not present in the single resolution algorithms.

5.5 Camera spatial resolution

Camera spatial resolution changes the resolution of both the BOS image and the optical flow measurements. A higher resolution camera means that it is possible to use a higher resolution background, which in turn produces a higher spatial resolution for the optical flow field. Additionally, with all other parameters being held constant, a smaller angular deflection of the light entering the test section will be required to move the ray's final location by one pixel. This improves the sensitivity of the BOS setup.

6 RECOMMENDATIONS FOR FUTURE WORK

6.1 Camera greyscale resolution and noise

The greyscale resolution of the camera will also improve the resolution of the sub-pixel performance of many of the algorithms because both the image intensity and image intensity gradients that are used in these algorithms are sensitive to this parameter. Further study is required to determine the exact effects.

6.2 Speckled backgrounds with gradient based algorithms

Each pixel in the camera records the number of photons hitting it during the exposure time. If the following is assumed:

- All photons (within the required frequency band) entering the pixel sensor are counted.
- The pixels are square and exactly adjacent to each other with no gap.
- The background consists of black squares and white squares whose image exactly aligns with the CCD pixel axes
- A pixel covered by a black square reads exactly 0
- A pixel covered by a white square reads exactly 1 and is not over-exposed
- The image is perfectly in focus

Then a model for the pixel reading based on the area of white versus the area of black on a particular pixel can be determined.

For a perfectly vertical edge moving laterally, the pixel reading is directly proportional to the amount of overlap that the white side of the pixel has. For example with white on the left and black on the right, the reading is given according to the following equation.

$$I = u \tag{137}$$

In this equation, I is the greyscale value and u is the per pixel lateral movement.

Similarly for a pure horizontal edge with white at the top and v as the per pixel vertical movement, the following equation holds true.

$$I = v \tag{138}$$

Interactions will occur for more complex geometries. An example is a white square moving into an otherwise black pixel from the top left. In this case, the intensity is given by the following equation.

$$I = uv \tag{139}$$

Another example is a checkerboard with the point of the four squares is moving in from the top left. Here, the following equation holds true.

$$I = uv + (1 - u)(1 - v) \tag{140}$$

$$= 2uv - u - v + 1 \tag{141}$$

In all cases, the pixel behaves as if it were a part of a smooth image.

Estimates of the derivatives from the initial image show the following.

For the vertical edge, the following is true.

$$I_i = 1 \tag{142}$$

$$I_j = 0 \tag{143}$$

For the vertical edge, the following is true.

$$I_i = 0 \tag{144}$$

$$I_j = 1 \tag{145}$$

For the square, the following is true.

$$I_i = v \tag{146}$$

$$I_j = u \tag{147}$$

$$I_{ij} = 1 \tag{148}$$

For the checkerboard pattern, the following is true.

$$I_i = 2v - 1 \tag{149}$$

$$I_j = 2u - 1 \tag{150}$$

$$I_{ij} = 1 \tag{151}$$

For the special case where either u or v are held constant while the other parameter is moved, the linearized optical flow equation gives the correct result in all cases.

For example, with the checkerboard edge, assuming pure lateral displacement after an initial vertical displacement of v_0 and u_0, as shown in the equations that follow.

$$I_1 - I_0 = 2u_1v_1 - u_1 - v_1 + 1 - 2u_0v_0 + u_0 + v_0 - 1 \tag{152}$$

$$I_1 - I_0 = 2v_0\delta u - \delta u \tag{153}$$

$$I_1 - I_0 = (2v_0 - 1)\delta u \tag{154}$$

$$I_1 - I_0 = I_i\delta u \tag{155}$$

This indicates, without delving into the details of each algorithm, that the optical flow constraint can be a reasonable approximation of the image intensity change due to sub-pixel movement with a speckled background. More research will be required to confirm and find the limitations of this hypothesis.

6.3 Application of optical flow knowledge to background design

Previously, with the exception of CGBOS [8], background design has focussed on random type backgrounds. With a knowledge of the various methods for determining optical flow, their methods and limitations better-structured backgrounds may be designed.

Some suggestions include the use of a checkerboard pattern and sparse optical flow tracking on good points to track. The reasoning for these suggestions are discussed in the sections below.

6.3.1 Checkerboard pattern

When two side-by-side squares in a speckled pattern are considered, the probability of an edge being present between the two is:

$$P(edge) = 2p(1-p) \tag{156}$$

For Monte-Carlo simulation calculating the number of edges as a fraction of the number of possible edges in thirty 1000 by 1000 speckle images, this characteristic is confirmed. The worst standard deviation obtained from these simulations was 0.00071 fractions, indicating that there is very little change in the number of edges from one random speckle background to another with the same fill ratio. Figure 94 illustrates this phenomenon.

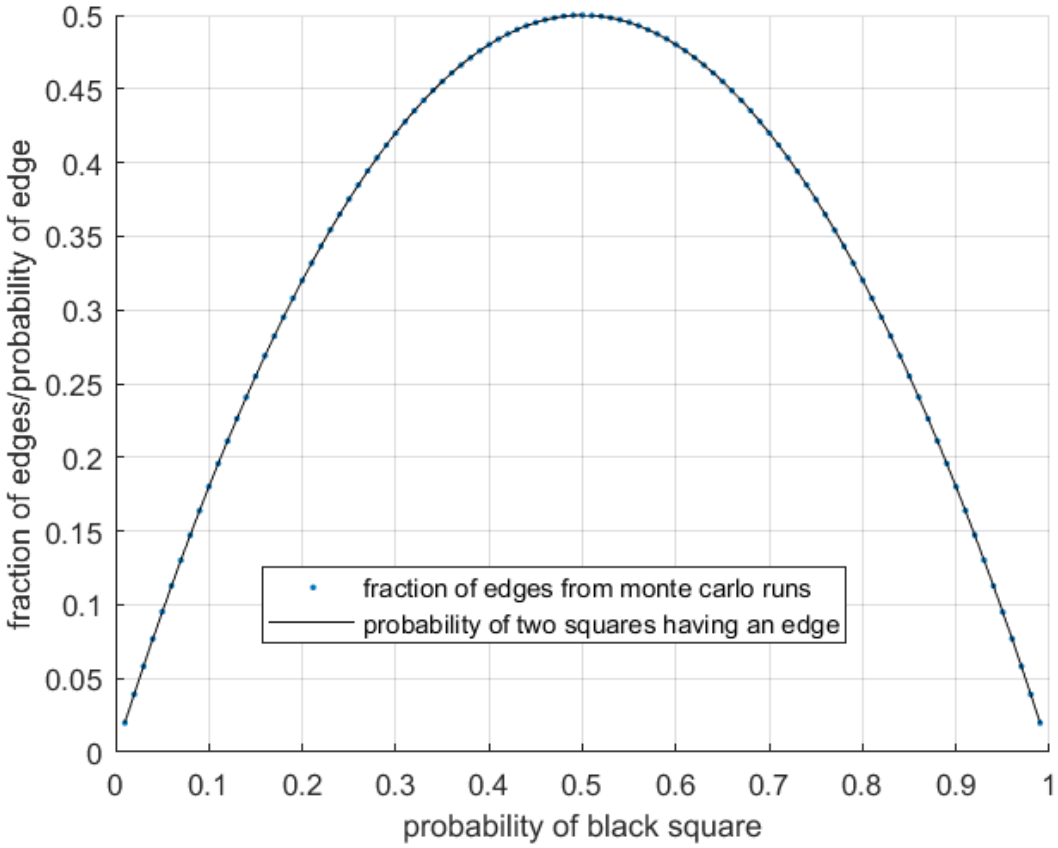


Figure 94: Variation of fraction of edges in an image with black speckle probability

There are two conclusions to draw from this finding. First, using the paradigm of maximising edges, the fill ratio that maximises the number of edges is 50 %. This value has been reported in the literature as the optimum fill ratio [17]. The second conclusion is that only half the number of possible edges are likely to be generated using this method. An improvement of up to a factor of two in the number of edges is possible with a more rigorously designed background, though aliasing may become an issue. Future research into a method for dealing with both aliasing and improving the number of edges will add value to the BOS field of study.

6.4 Colour and black and white image comparison

The major difference between colour and greyscale CCD's is the use of a Bayer mask. The Bayer mask applies a colour filter to each pixel, making it sensitive to either red, green or blue light. The pattern is shown in Figure 95.

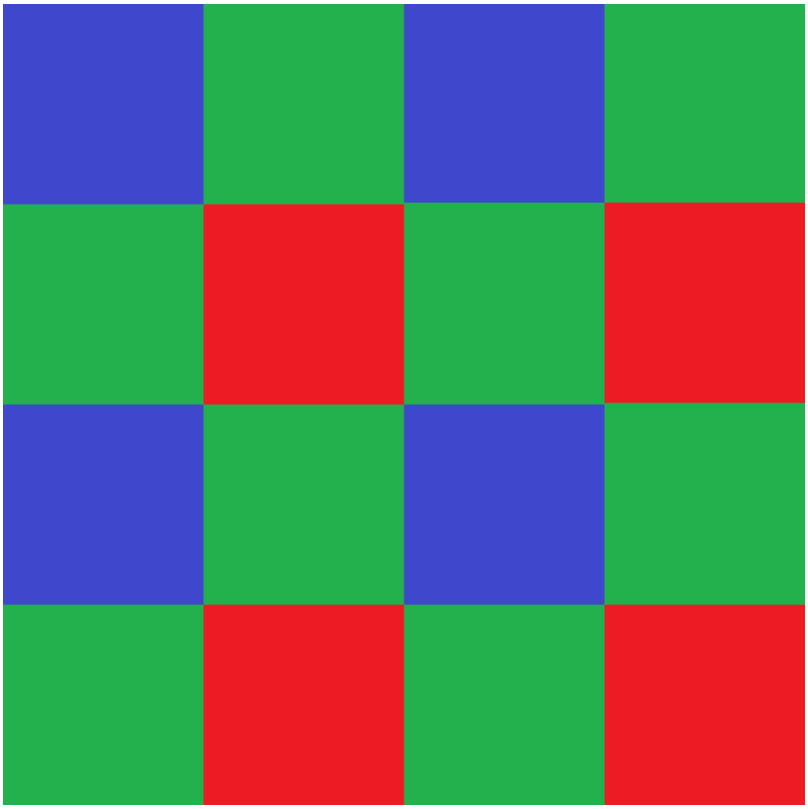


Figure 95: Illustration of a Bayer mask

It is not immediately apparent how the trade-off between the losses in resolution for each colour versus the gain in information by having two extra channels effects the outcome of a BOS measurement and may differ depending on which optical flow algorithm is used and how the results are combined. A good reference for the use of coloured speckles with a cross-correlation algorithm is that by Leopoldt *et al.* [17].

7 CONCLUSION

All of the GPU accelerated optical flow algorithms in OpenCV 3.1.0 were used to process experimentally acquired raw BOS images of flow over a cone model at a nominal Mach number of 2.5. These processed images were compared to theoretical predictions, a colour schlieren image, and a purpose-developed block-matching method that is typical of BOS systems. Each algorithm was tested with two different backgrounds, one speckled background, and one b-spline wavelet noise background. The various parameters used to tune the optical flow algorithms were also investigated.

Several insights were gained, the global insights being listed below:

- There is a general compromise between optical flow smoothness and optical flow detail. Parameter choices giving rise to more detail tend to produce more noise in the optical flow images.
- A knowledge of the aperture problem helps with the understanding of the interaction of background images and optical flow algorithms. This knowledge can also add to the design of background images for use in the BOS context.
- Multi-resolution algorithms tend to be less stable in the BOS context than their single resolution counterparts are.
- In general, the speckled background produced more detailed optical flow images with more noise than did the wavelet background.
- The newer optical-flow methods used in the OpenCV GPU accelerated toolbox generally outperformed the brute-force traditional cross-correlation block-matching technique.

In addition, some areas for future research were also identified. These include:

- The effect that camera greyscale resolution has on BOS images
- Using the knowledge of the aperture problem and multi-resolution algorithms to design better BOS backgrounds
- Researching the inclusion of colour background images with modern optical flow algorithms

The GPU accelerated optical flow algorithms in the OpenCV open source computer vision library perform well in the BOS context and show promise for future developments.

8 REFERENCES

[1] G. Meier, "Computerized Background-Oriented Schlieren," *Experiments in Fluids,* vol. 33, pp. 180-187, 2002.

[2] B. Atcheson, W. Heidrich and I. Ihrke, "An evaluation of optical flow algorithms for background oriented shclieren imaging," *Experiments in Fluids,* vol. 46, pp. 467-476, 2009.

[3] K. Hayasaka, Y. Tagawa, T. Liu and M. Kameda, "Optical-Flow-Based Background-Oriented Schlieren Technique for Measuring Laser-Induced Underwater Shock Wave," *Experiments in Fluids,* vol. 57:179, 2016.

[4] B. K. Horn and B. G. Schunk, "Determining Optical Flow," *Artificial Intelligence,* pp. 185-203, 1981.

[5] T. Brox, A. Bruhn, N. Pepenburg and J. Weickert, "High Accuracy Optical Flow Estimation Based on a Theory for Warping," in *8th European Conference on Computer Vision,* Prague, 2004.

[6] C. Zach, T. Pock and H. Bischof, "A Duality Based Approach for Realtime TV-L1 Optical Flow," in *Pattern Recognition,* Berlin, Heidelberg, 2007.

[7] F. Leopoldt, F. Sourgen and D. Klatt, "Reconstruction of the density field using the Colored Background Oriented Schlieren Technique (CBOS)," *Optics and Lasers in Engineering,* vol. 50, pp. 29-38, 2012.

[8] M. Ota, H. Kato and K. Maeno, "Improvements of Spacial Resolution of Colored Grid Background Oriented Schlieren (CGBOS) Technique by Introducing Telecentric Optical System and Reconstruction of Density Field [sic]," in *15th International Symposium on Flow Visualisation,* Minsk, 2012.

[9] "OpenCV Home Page," [Online]. Available: https://opencv.org/. [Accessed May 2018].

[10] J. D. Anderson, Jr., "10 Conical Flow," in *Modern Compressible Flow With Historical Perspective*, McGraw-Hill, 2003, pp. 363-357.

[11] J. D. Anderson, Jr., "4.3 Oblique Shock Relations," in *Modern Compressible Flow With Historical Perspective*, McGraw-Hill, 2003, pp. 133-144.

[12] J. D. Anderson, Jr., "3.5 Alternative Forms of the Energy Equation," in *Modern compressible Flow with Historical Perspective*, Mc Graw Hill, 2003, pp. 78-86.

[13] G. Settles, "2.1 Light Propogation Through Inhomogeneous Media," in *Schlieren and Shadowgraph Techniques*, Springer, 2003, pp. 25-27.

[14] B. D. Lucas and T. Kanade, "An Interative Image Registration Technique with an Application to Stereo Vision," in *IJCAI81*, 1981.

[15] G. Farnebäck, "Two-Frame Motion Estimation Based on Polynomial Expansion," in *Scandinavian Conference on Image Analysis*, Linköping, Sweden, 2003.

[16] M. Raffel, C. Willert, S. Wereley and J. Kompenhans, "Digital Spatial Correlation in PIV Evaluaion," in *Particle Image Velocimetry: A Practical Guide*, Heidelberg, Springer, 2007, pp. 132-135.

[17] F. Leopoldt, J. Simon, D. Gruppi and H. Schräfer, "Recent improvements of the Background Oriented Schlieren technique (BOS) using a colored background," in *12th International symposium on flow visualisation*, Göttingen, 2006.

[18] R. Cook and T. DeRose, "Wavelet Noise," *ACM Trans. Graph.*, vol. 24, pp. 803-811, 2005.

APPENDIX A BLOCK-MATCHING ALGORITHM DEVELOPMENT

A.1 Introduction

The most popular optical flow algorithm used in background oriented schlieren setups is the block-matching algorithm. This algorithm works by dividing the two images up into windows and comparing the correlation coefficients of the windows while translated at set displacements. The displacement that corresponds to the highest correlation is then taken to be the optical flow for that window. Some algorithms then perform an interpolation around the peak value to determine an optical flow with sub-pixel accuracy.

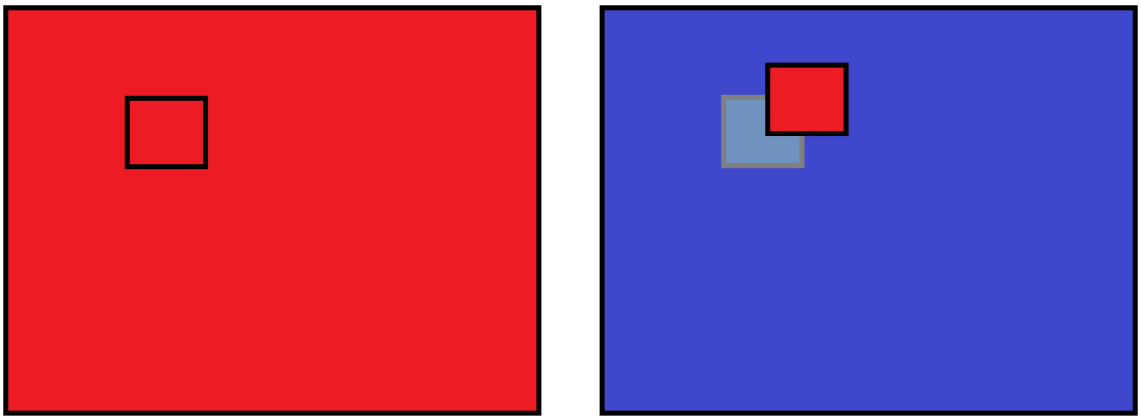


Figure 96: Image illustrating cross correlation of translated window

Given two images with intensity at pixel (x, y) denoted $I_0(x, y)$ and $I_1(x, y)$, the correlation coefficient for a window of pixel range $x_a \leq x \leq x_b$ and $y_a \leq x \leq y_b$ with I_0 having been translated by (u, v) with reference to its initial position, The correlation of the two sub-images is given by the following equation.

$$R_{(u,v)} = \sum_{x=x_z}^{x_b} \sum_{y=y_z}^{y_b} I_0(x, y)\, I_1(x + u, y + v) \tag{157}$$

The optical flow is determined to pixel accuracy by finding the following result.

$$arg \max_{u,v} \left[\sum_{x=x_z}^{x_b} \sum_{y=y_z}^{y_b} I_0(x,y)\, I_1(x+u, y+v) \right] \tag{158}$$

Once the pixel accurate optical flow (u, v) has been found, sub-pixel interpolation can be calculated using one of the following three methods.

Peak centroid

$$x_0 = \frac{(i-1)R_{(i-1,j)} + iR_{(i,j)} + (i+1)R_{(i+1,j)}}{R_{(i-1,j)} + R_{(i,j)} + R_{(i+1,j)}} \tag{159}$$

$$y_0 = \frac{(j-1)R_{(i,j-1)} + jR_{(i,j)} + (i+1)R_{(i,j+1)}}{R_{(i,j-1)} + R_{(i,j)} + R_{(i,j+1)}} \tag{160}$$

Parabolic peak fit

$$x_0 = i + \frac{R_{(i-1,j)} - R_{(i+1,j)}}{2R_{(i-1,j)} - 4R_{(i,j)} + 2R_{(i+1,j)}} \tag{161}$$

$$y_0 = j + \frac{R_{(i,j-1)} + R_{(i,j+1)}}{2R_{(i,j-1)} - 4R_{(i,j)} + 2R_{(i,j+1)}} \tag{162}$$

Gaussian peak fit

$$x_0 = i + \frac{\ln R_{(i-1,j)} - \ln R_{(i+1,j)}}{2R_{(i-1,j)} - 4R_{(i,j)} + 2R_{(i+1,j)}} \tag{163}$$

$$y_0 = j + \frac{R_{(i,j-1)} + R_{(i,j+1)}}{2\ln R_{(i,j-1)} - 4\ln R_{(i,j)} + 2\ln R_{(i,j+1)}} \tag{164}$$

A.2 Implementation

Correlations may be performed using fast Fourier transforms and weighting windows may be applied to the correlations. It was decided that a brute-force block matching algorithm be used as a reference for comparison as the FFT based algorithms have their own limitations.

No block matching optical flow algorithm exists in the OpenCV toolbox. With the popularity of the block matching method used in BOS setups, it was decided to develop one for the BOS system.

At first, it was decided to attempt to make maximum use of the parallel computing capability of the GPU used in the BOS system. Limitations in the GPU hardware forced a less aggressive approach to making use of the GPU's parallel computing ability; however, as GPU technology improves the more aggressive approach may show advantages over the current approach. The aggressive approach is outlined in the following section, as well as the reasons for seeking a less aggressive approach. The final approach that was adopted is outline in the section that follows that.

A.2.1 Parallel block matching approach

The block-matching algorithm needs to:

- For each image displacement, find correlation coefficients
    - Multiply corresponding image pixel values for each displacement
    - Sum the products of the multiplications performed in the previous step
- Find the displacement that corresponds with the maximum correlation coefficient
- Perform sub-pixel interpolation

One strategy could be to:

- Perform all multiplications at once, storing the products in memory. This would result in as many two-dimensional arrays of products as displacements being investigated.
- Each image could then have a box-filter, with dimensions equal to the window size, applied to it. This would result in two-dimensional arrays of correlation coefficients, each array corresponding to a displacement and each position in the array corresponding to a position in the original image.
- The maximum correlation coefficient for each position could be found, with its corresponding displacement.
- Sub-pixel interpolation could then be applied.

This strategy was attempted on the BOS hardware; however, it soon became apparent that the memory limitations of the GPU allowed this algorithm to be possible for only very small problem sizes.

The following sample calculation illustrates the problem.

To calculate the minimum amount of memory required to hold the results of step 1, the number of pixels per image is multiplied by the number of displacements and the size of a floating-point number, which is being used to store the value of each pixel.

Number of pixels per image = 2046 x 1080 = 2.20968E+06

Number of translations for 64 x translations and 64 y translations = 64 x 64 = 4096

Bytes per floating point number = 4

Total memory required = 2046 x 1080 x 64 x 64 x 4 = 36.2 GiB

The NVIDIA Tesla K20c GPU used in the BOS system has 5 GiB of RAM, which is insufficient to implement the algorithm outlined in this section, though not so far off that hardware in the near future may be able to implement such an algorithm.

A.2.2 Sequential correlations with parallel algorithm components

A second algorithm was designed to limit the amount of memory required while still taking advantage of the GPU's parallel computing capability.

For a particular x, y combination

- Sequentially for each u, v combination
    - Perform the necessary multiplications in parallel.
    - Use a parallel algorithm to sum the resulting products.
- Use a parallel algorithm to find the location of the maximum correlation coefficient.
- Perform sub-pixel interpolation
- Move to the next x, y combination

This algorithm was implemented on the BOS system. It has the disadvantage of having to perform the same multiplications many times, which arise from overlapping windows at different displacements. However, the algorithm works on the current system and this algorithm may be used practically to compare the block-matching algorithm to other optical flow algorithms.

A.3 Conclusion

A hybrid between the two algorithms may be possible, where the larger image is broken down into smaller segments that are processed sequentially. This is left for future study.

APPENDIX B HORN AND SCHUNK'S ALGORITHM

Horn and Schunk [4] solved the aperture problem by adding a regularisation term that minimises a weighted sum of the difference in image intensity at the position with the predicted change and the square of the optical flow gradient magnitude integrated over the whole image. The cost function, J, is shown below.

$$J = \iint \left[\left[\delta I - \left(\frac{\partial I}{\partial i} u + \frac{\partial I}{\partial j} v \right) \right] + \alpha \left[\left(\frac{\partial u}{\partial i} \right)^2 + \left(\frac{\partial u}{\partial j} \right)^2 + \left(\frac{\partial v}{\partial i} \right)^2 + \left(\frac{\partial v}{\partial j} \right)^2 \right] \right] didj \qquad (165)$$

It can be shown that J is minimised by the following set of equations by using the calculus of variations.

$$\left(\frac{\partial I}{\partial i} \right)^2 u + \frac{\partial I}{\partial i} \frac{\partial I}{\partial j} v = \alpha^2 \nabla^2 u + \frac{\partial I}{\partial i} \delta I \qquad (166)$$

$$\frac{\partial I}{\partial i} \frac{\partial I}{\partial j} u + \left(\frac{\partial I}{\partial j} \right)^2 v = \alpha^2 \nabla^2 v + \frac{\partial I}{\partial j} \delta I \qquad (167)$$

The original implementation uses an approximation to the Laplacian in the following form.

$$\nabla^2 i \approx k(\bar{\imath} - i) \qquad (168)$$

The variable $k = 3$ and $\bar{x}$ is a weighted sum of pixel values surrounding the pixel of interest, with pixels above and below the pixel of interest having weights of $\frac{1}{6}$ and pixels diagonally adjacent to the pixel of interest having weights of $\frac{1}{12}$, as indicated below.

$$\begin{matrix} \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \\ \frac{1}{6} & pixel\ of\ interest & \frac{1}{6} \\ \frac{1}{12} & \frac{1}{6} & \frac{1}{12} \end{matrix}$$

With this approximation, the solution to the problem can be written as follows.

$$\left(\alpha^2 + \left(\frac{\partial I}{\partial i}\right)^2\right)u + \frac{\partial I}{\partial i}\frac{\partial I}{\partial j}v = \alpha^2\bar{u} - \frac{\partial I}{\partial i}\delta I \tag{169}$$

$$\frac{\partial I}{\partial i}\frac{\partial I}{\partial j}u + \left(\alpha^2 + \left(\frac{\partial I}{\partial j}\right)^2\right)v = \alpha^2\bar{v} - \frac{\partial I}{\partial j}\delta I \tag{170}$$

Which gives the following solution for *w*

$$\begin{bmatrix} u \\ v \end{bmatrix} = \frac{1}{\alpha^2 + \left(\frac{\partial I}{\partial j}\right)^2 + \left(\frac{\partial I}{\partial i}\right)^2}\begin{bmatrix} \left(\alpha^2 + \left(\frac{\partial I}{\partial j}\right)^2\right)\bar{u} - \frac{\partial I}{\partial i}\frac{\partial I}{\partial j}\bar{v} - \frac{\partial I}{\partial i}\delta I \\ -\frac{\partial I}{\partial i}\frac{\partial I}{\partial j}\bar{u} + \left(\alpha^2 + \left(\frac{\partial I}{\partial i}\right)^2\right)\bar{v} - \frac{\partial I}{\partial j}\delta I \end{bmatrix} \tag{171}$$

This can be re-arranged giving the following equations.

$$u = \bar{u} - \frac{\frac{\partial I}{\partial i}\left(\frac{\partial I}{\partial i}\bar{u} + \frac{\partial I}{\partial j}\bar{v} + \delta I\right)}{\alpha^2 + \left(\frac{\partial I}{\partial j}\right)^2 + \left(\frac{\partial I}{\partial i}\right)^2} \tag{172}$$

$$v = \bar{v} - \frac{\frac{\partial I}{\partial j}\left(\frac{\partial I}{\partial i}\bar{u} + \frac{\partial I}{\partial j}\bar{v} + \delta I\right)}{\alpha^2 + \left(\frac{\partial I}{\partial j}\right)^2 + \left(\frac{\partial I}{\partial i}\right)^2} \tag{173}$$

Gauss-Seidel Iteration is then used to solve the equations numerically.

$$u^{n+1} = \bar{u}^n - \frac{\frac{\partial I}{\partial i}\left(\frac{\partial I}{\partial i}\bar{u}^n + \frac{\partial I}{\partial j}\bar{v}^n + \delta I\right)}{\alpha^2 + \left(\frac{\partial I}{\partial j}\right)^2 + \left(\frac{\partial I}{\partial i}\right)^2} \tag{174}$$

$$v^{n+1} = \bar{v}^n - \frac{\frac{\partial I}{\partial j}\left(\frac{\partial I}{\partial i}\bar{u}^n + \frac{\partial I}{\partial j}\bar{v}^n + \delta I\right)}{\alpha^2 + \left(\frac{\partial I}{\partial j}\right)^2 + \left(\frac{\partial I}{\partial i}\right)^2} \tag{175}$$

The variables required for the problem are given in the table below.

Parameter	Description
N	Number of iterations
α	Optical flow smoothness weight

APPENDIX C INDEX TO PROCESSED IMAGES

C.1 Down sampling

In all cases, the following down-sample factors correspond to the number of layers used.

Table 19: Down sampling parameters

Layers	Down Sample Factor
1	NA
11	0.5
35	0.8

C.2 Block matching results

Table 20: Block matching algorithm results

Background Type	Optical Flow Component	Window Width	Window Height	Interpolation Type	Min	Max	File
Wavelet	U	4	4	none	-4	4	/BlockMatching/Wavelet/4x4/none/BM_U_4x4_none_n4p4.bmp
Wavelet	V	4	4	none	-4	4	/BlockMatching/Wavelet/4x4/none/BM_V_4x4_none_n4p4.bmp
Wavelet	U	4	4	none	-1	1	/BlockMatching/Wavelet/4x4/none/BM_U_4x4_none_n1p1.bmp
Wavelet	V	4	4	none	-1	1	/BlockMatching/Wavelet/4x4/none/BM_V_4x4_none_n1p1.bmp
Wavelet	U	8	8	none	-4	4	/BlockMatching/Wavelet/8x8/none/BM_U_8x8_none_n4p4.bmp
Wavelet	V	8	8	none	-4	4	/BlockMatching/Wavelet/8x8/none/BM_V_8x8_none_n4p4.bmp
Wavelet	U	8	8	none	-1	1	/BlockMatching/Wavelet/8x8/none/BM_U_8x8_none_n1p1.bmp
Wavelet	V	8	8	none	-1	1	/BlockMatching/Wavelet/8x8/none/BM_V_8x8_none_n1p1.bmp
Wavelet	U	16	16	none	-4	4	/BlockMatching/Wavelet/16x16/none/BM_U_16x16_none_n4p4.bmp
Wavelet	V	16	16	none	-4	4	/BlockMatching/Wavelet/16x16/none/BM_V_16x16_none_n4p4.bmp
Wavelet	U	16	16	none	-1	1	/BlockMatching/Wavelet/16x16/none/BM_U_16x16_none_n1p1.bmp
Wavelet	V	16	16	none	-1	1	/BlockMatching/Wavelet/16x16/none/BM_V_16x16_none_n1p1.bmp
Wavelet	U	32	32	none	-4	4	/BlockMatching/Wavelet/32x32/none/BM_U_32x32_none_n4p4.bmp
Wavelet	V	32	32	none	-4	4	/BlockMatching/Wavelet/32x32/none/BM_V_32x32_none_n4p4.bmp
Wavelet	U	32	32	none	-1	1	/BlockMatching/Wavelet/32x32/none/BM_U_32x32_none_n1p1.bmp
Wavelet	V	32	32	none	-1	1	/BlockMatching/Wavelet/32x32/none/BM_V_32x32_none_n1p1.bmp
Speckle	U	4	4	none	-4	4	/BlockMatching/Speckle/4x4/none/BM_U_4x4_none_n4p4.bmp
Speckle	V	4	4	none	-4	4	/BlockMatching/Speckle/4x4/none/BM_V_4x4_none_n4p4.bmp
Speckle	U	4	4	none	-1	1	/BlockMatching/Speckle/4x4/none/BM_U_4x4_none_n1p1.bmp
Speckle	V	4	4	none	-1	1	/BlockMatching/Speckle/4x4/none/BM_V_4x4_none_n1p1.bmp

Background Type	Optical Flow Component	Window Width	Window Height	Interpolation Type	Min	Max	File
Speckle	U	8	8	none	-4	4	/BlockMatching/Speckle/8x8/none/BM_U_8x8_none_n4p4.bmp
Speckle	V	8	8	none	-4	4	/BlockMatching/Speckle/8x8/none/BM_V_8x8_none_n4p4.bmp
Speckle	U	8	8	none	-1	1	/BlockMatching/Speckle/8x8/none/BM_U_8x8_none_n1p1.bmp
Speckle	V	8	8	none	-1	1	/BlockMatching/Speckle/8x8/none/BM_V_8x8_none_n1p1.bmp
Speckle	U	16	16	none	-4	4	/BlockMatching/Speckle/16x16/none/BM_U_16x16_none_n4p4.bmp
Speckle	V	16	16	none	-4	4	/BlockMatching/Speckle/16x16/none/BM_V_16x16_none_n4p4.bmp
Speckle	U	16	16	none	-1	1	/BlockMatching/Speckle/16x16/none/BM_U_16x16_none_n1p1.bmp
Speckle	V	16	16	none	-1	1	/BlockMatching/Speckle/16x16/none/BM_V_16x16_none_n1p1.bmp
Speckle	U	32	32	none	-4	4	/BlockMatching/Speckle/32x32/none/BM_U_32x32_none_n4p4.bmp
Speckle	V	32	32	none	-4	4	/BlockMatching/Speckle/32x32/none/BM_V_32x32_none_n4p4.bmp
Speckle	U	32	32	none	-1	1	/BlockMatching/Speckle/32x32/none/BM_U_32x32_none_n1p1.bmp
Speckle	V	32	32	none	-1	1	/BlockMatching/Speckle/32x32/none/BM_V_32x32_none_n1p1.bmp
Wavelet	U	4	4	centroid	-4	4	/BlockMatching/Wavelet/4x4/centroid/BM_U_4x4_centroid_n4p4.bmp
Wavelet	V	4	4	centroid	-4	4	/BlockMatching/Wavelet/4x4/centroid/BM_V_4x4_centroid_n4p4.bmp
Wavelet	U	4	4	centroid	-1	1	/BlockMatching/Wavelet/4x4/centroid/BM_U_4x4_centroid_n1p1.bmp
Wavelet	V	4	4	centroid	-1	1	/BlockMatching/Wavelet/4x4/centroid/BM_V_4x4_centroid_n1p1.bmp
Wavelet	U	8	8	centroid	-4	4	/BlockMatching/Wavelet/8x8/centroid/BM_U_8x8_centroid_n4p4.bmp
Wavelet	V	8	8	centroid	-4	4	/BlockMatching/Wavelet/8x8/centroid/BM_V_8x8_centroid_n4p4.bmp
Wavelet	U	8	8	centroid	-1	1	/BlockMatching/Wavelet/8x8/centroid/BM_U_8x8_centroid_n1p1.bmp
Wavelet	V	8	8	centroid	-1	1	/BlockMatching/Wavelet/8x8/centroid/BM_V_8x8_centroid_n1p1.bmp
Wavelet	U	16	16	centroid	-4	4	/BlockMatching/Wavelet/16x16/centroid/BM_U_16x16_centroid_n4p4.bmp
Wavelet	V	16	16	centroid	-4	4	/BlockMatching/Wavelet/16x16/centroid/BM_V_16x16_centroid_n4p4.bmp
Wavelet	U	16	16	centroid	-1	1	/BlockMatching/Wavelet/16x16/centroid/BM_U_16x16_centroid_n1p1.bmp

Background Type	Optical Flow Component	Window Width	Window Height	Interpolation Type	Min	Max	File
Wavelet	V	16	16	centroid	-1	1	/BlockMatching/Wavelet/16x16/centroid/BM_V_16x16_centroid_n1p1.bmp
Wavelet	U	32	32	centroid	-4	4	/BlockMatching/Wavelet/32x32/centroid/BM_U_32x32_centroid_n4p4.bmp
Wavelet	V	32	32	centroid	-4	4	/BlockMatching/Wavelet/32x32/centroid/BM_V_32x32_centroid_n4p4.bmp
Wavelet	U	32	32	centroid	-1	1	/BlockMatching/Wavelet/32x32/centroid/BM_U_32x32_centroid_n1p1.bmp
Wavelet	V	32	32	centroid	-1	1	/BlockMatching/Wavelet/32x32/centroid/BM_V_32x32_centroid_n1p1.bmp
Speckle	U	4	4	centroid	-4	4	/BlockMatching/Speckle/4x4/centroid/BM_U_4x4_centroid_n4p4.bmp
Speckle	V	4	4	centroid	-4	4	/BlockMatching/Speckle/4x4/centroid/BM_V_4x4_centroid_n4p4.bmp
Speckle	U	4	4	centroid	-1	1	/BlockMatching/Speckle/4x4/centroid/BM_U_4x4_centroid_n1p1.bmp
Speckle	V	4	4	centroid	-1	1	/BlockMatching/Speckle/4x4/centroid/BM_V_4x4_centroid_n1p1.bmp
Speckle	U	8	8	centroid	-4	4	/BlockMatching/Speckle/8x8/centroid/BM_U_8x8_centroid_n4p4.bmp
Speckle	V	8	8	centroid	-4	4	/BlockMatching/Speckle/8x8/centroid/BM_V_8x8_centroid_n4p4.bmp
Speckle	U	8	8	centroid	-1	1	/BlockMatching/Speckle/8x8/centroid/BM_U_8x8_centroid_n1p1.bmp
Speckle	V	8	8	centroid	-1	1	/BlockMatching/Speckle/8x8/centroid/BM_V_8x8_centroid_n1p1.bmp
Speckle	U	16	16	centroid	-4	4	/BlockMatching/Speckle/16x16/centroid/BM_U_16x16_centroid_n4p4.bmp
Speckle	V	16	16	centroid	-4	4	/BlockMatching/Speckle/16x16/centroid/BM_V_16x16_centroid_n4p4.bmp
Speckle	U	16	16	centroid	-1	1	/BlockMatching/Speckle/16x16/centroid/BM_U_16x16_centroid_n1p1.bmp
Speckle	V	16	16	centroid	-1	1	/BlockMatching/Speckle/16x16/centroid/BM_V_16x16_centroid_n1p1.bmp
Speckle	U	32	32	centroid	-4	4	/BlockMatching/Speckle/32x32/centroid/BM_U_32x32_centroid_n4p4.bmp
Speckle	V	32	32	centroid	-4	4	/BlockMatching/Speckle/32x32/centroid/BM_V_32x32_centroid_n4p4.bmp
Speckle	U	32	32	centroid	-1	1	/BlockMatching/Speckle/32x32/centroid/BM_U_32x32_centroid_n1p1.bmp
Speckle	V	32	32	centroid	-1	1	/BlockMatching/Speckle/32x32/centroid/BM_V_32x32_centroid_n1p1.bmp
Wavelet	U	4	4	quadratic	-4	4	/BlockMatching/Wavelet/4x4/quadratic/BM_U_4x4_quadratic_n4p4.bmp
Wavelet	V	4	4	quadratic	-4	4	/BlockMatching/Wavelet/4x4/quadratic/BM_V_4x4_quadratic_n4p4.bmp

Background Type	Optical Flow Component	Window Width	Window Height	Interpolation Type	Min	Max	File
Wavelet	U	4	4	quadratic	-1	1	/BlockMatching/Wavelet/4x4/quadratic/BM_U_4x4_quadratic_n1p1.bmp
Wavelet	V	4	4	quadratic	-1	1	/BlockMatching/Wavelet/4x4/quadratic/BM_V_4x4_quadratic_n1p1.bmp
Wavelet	U	8	8	quadratic	-4	4	/BlockMatching/Wavelet/8x8/quadratic/BM_U_8x8_quadratic_n4p4.bmp
Wavelet	V	8	8	quadratic	-4	4	/BlockMatching/Wavelet/8x8/quadratic/BM_V_8x8_quadratic_n4p4.bmp
Wavelet	U	8	8	quadratic	-1	1	/BlockMatching/Wavelet/8x8/quadratic/BM_U_8x8_quadratic_n1p1.bmp
Wavelet	V	8	8	quadratic	-1	1	/BlockMatching/Wavelet/8x8/quadratic/BM_V_8x8_quadratic_n1p1.bmp
Wavelet	U	16	16	quadratic	-4	4	/BlockMatching/Wavelet/16x16/quadratic/BM_U_16x16_quadratic_n4p4.bmp
Wavelet	V	16	16	quadratic	-4	4	/BlockMatching/Wavelet/16x16/quadratic/BM_V_16x16_quadratic_n4p4.bmp
Wavelet	U	16	16	quadratic	-1	1	/BlockMatching/Wavelet/16x16/quadratic/BM_U_16x16_quadratic_n1p1.bmp
Wavelet	V	16	16	quadratic	-1	1	/BlockMatching/Wavelet/16x16/quadratic/BM_V_16x16_quadratic_n1p1.bmp
Wavelet	U	32	32	quadratic	-4	4	/BlockMatching/Wavelet/32x32/quadratic/BM_U_32x32_quadratic_n4p4.bmp
Wavelet	V	32	32	quadratic	-4	4	/BlockMatching/Wavelet/32x32/quadratic/BM_V_32x32_quadratic_n4p4.bmp
Wavelet	U	32	32	quadratic	-1	1	/BlockMatching/Wavelet/32x32/quadratic/BM_U_32x32_quadratic_n1p1.bmp
Wavelet	V	32	32	quadratic	-1	1	/BlockMatching/Wavelet/32x32/quadratic/BM_V_32x32_quadratic_n1p1.bmp
Speckle	U	4	4	quadratic	-4	4	/BlockMatching/Speckle/4x4/quadratic/BM_U_4x4_quadratic_n4p4.bmp
Speckle	V	4	4	quadratic	-4	4	/BlockMatching/Speckle/4x4/quadratic/BM_V_4x4_quadratic_n4p4.bmp
Speckle	U	4	4	quadratic	-1	1	/BlockMatching/Speckle/4x4/quadratic/BM_U_4x4_quadratic_n1p1.bmp
Speckle	V	4	4	quadratic	-1	1	/BlockMatching/Speckle/4x4/quadratic/BM_V_4x4_quadratic_n1p1.bmp
Speckle	U	8	8	quadratic	-4	4	/BlockMatching/Speckle/8x8/quadratic/BM_U_8x8_quadratic_n4p4.bmp
Speckle	V	8	8	quadratic	-4	4	/BlockMatching/Speckle/8x8/quadratic/BM_V_8x8_quadratic_n4p4.bmp
Speckle	U	8	8	quadratic	-1	1	/BlockMatching/Speckle/8x8/quadratic/BM_U_8x8_quadratic_n1p1.bmp
Speckle	V	8	8	quadratic	-1	1	/BlockMatching/Speckle/8x8/quadratic/BM_V_8x8_quadratic_n1p1.bmp
Speckle	U	16	16	quadratic	-4	4	/BlockMatching/Speckle/16x16/quadratic/BM_U_16x16_quadratic_n4p4.bmp

Background Type	Optical Flow Component	Window Width	Window Height	Interpolation Type	Min	Max	File
Speckle	V	16	16	quadratic	-4	4	/BlockMatching/Speckle/16x16/quadratic/BM_V_16x16_quadratic_n4p4.bmp
Speckle	U	16	16	quadratic	-1	1	/BlockMatching/Speckle/16x16/quadratic/BM_U_16x16_quadratic_n1p1.bmp
Speckle	V	16	16	quadratic	-1	1	/BlockMatching/Speckle/16x16/quadratic/BM_V_16x16_quadratic_n1p1.bmp
Speckle	U	32	32	quadratic	-4	4	/BlockMatching/Speckle/32x32/quadratic/BM_U_32x32_quadratic_n4p4.bmp
Speckle	V	32	32	quadratic	-4	4	/BlockMatching/Speckle/32x32/quadratic/BM_V_32x32_quadratic_n4p4.bmp
Speckle	U	32	32	quadratic	-1	1	/BlockMatching/Speckle/32x32/quadratic/BM_U_32x32_quadratic_n1p1.bmp
Speckle	V	32	32	quadratic	-1	1	/BlockMatching/Speckle/32x32/quadratic/BM_V_32x32_quadratic_n1p1.bmp
Wavelet	U	4	4	Gaussian	-4	4	/BlockMatching/Wavelet/4x4/Gaussian/BM_U_4x4_Gaussian_n4p4.bmp
Wavelet	V	4	4	Gaussian	-4	4	/BlockMatching/Wavelet/4x4/Gaussian/BM_V_4x4_Gaussian_n4p4.bmp
Wavelet	U	4	4	Gaussian	-1	1	/BlockMatching/Wavelet/4x4/Gaussian/BM_U_4x4_Gaussian_n1p1.bmp
Wavelet	V	4	4	Gaussian	-1	1	/BlockMatching/Wavelet/4x4/Gaussian/BM_V_4x4_Gaussian_n1p1.bmp
Wavelet	U	8	8	Gaussian	-4	4	/BlockMatching/Wavelet/8x8/Gaussian/BM_U_8x8_Gaussian_n4p4.bmp
Wavelet	V	8	8	Gaussian	-4	4	/BlockMatching/Wavelet/8x8/Gaussian/BM_V_8x8_Gaussian_n4p4.bmp
Wavelet	U	8	8	Gaussian	-1	1	/BlockMatching/Wavelet/8x8/Gaussian/BM_U_8x8_Gaussian_n1p1.bmp
Wavelet	V	8	8	Gaussian	-1	1	/BlockMatching/Wavelet/8x8/Gaussian/BM_V_8x8_Gaussian_n1p1.bmp
Wavelet	U	16	16	Gaussian	-4	4	/BlockMatching/Wavelet/16x16/Gaussian/BM_U_16x16_Gaussian_n4p4.bmp
Wavelet	V	16	16	Gaussian	-4	4	/BlockMatching/Wavelet/16x16/Gaussian/BM_V_16x16_Gaussian_n4p4.bmp
Wavelet	U	16	16	Gaussian	-1	1	/BlockMatching/Wavelet/16x16/Gaussian/BM_U_16x16_Gaussian_n1p1.bmp
Wavelet	V	16	16	Gaussian	-1	1	/BlockMatching/Wavelet/16x16/Gaussian/BM_V_16x16_Gaussian_n1p1.bmp

C.3 Lucas and Kanade algorithm results

Table 21: Lucas and Kanade algorithm results

Background Type	Optical Flow Component	Window Width	Window Height	Layers	Min	Max	File
Wavelet	U	5	5	1	-4	4	/LucasKanade/Wavelet/5x5/SingleRes/LK_U_5x5_1_n4p4.bmp
Wavelet	V	5	5	1	-4	4	/LucasKanade/Wavelet/5x5/SingleRes/LK_V_5x5_1_n4p4.bmp
Wavelet	U	5	5	1	-1	1	/LucasKanade/Wavelet/5x5/SingleRes/LK_U_5x5_1_n1p1.bmp
Wavelet	V	5	5	1	-1	1	/LucasKanade/Wavelet/5x5/SingleRes/LK_V_5x5_1_n1p1.bmp
Wavelet	U	9	9	1	-4	4	/LucasKanade/Wavelet/9x9/SingleRes/LK_U_9x9_1_n4p4.bmp
Wavelet	V	9	9	1	-4	4	/LucasKanade/Wavelet/9x9/SingleRes/LK_V_9x9_1_n4p4.bmp
Wavelet	U	9	9	1	-1	1	/LucasKanade/Wavelet/9x9/SingleRes/LK_U_9x9_1_n1p1.bmp
Wavelet	V	9	9	1	-1	1	/LucasKanade/Wavelet/9x9/SingleRes/LK_V_9x9_1_n1p1.bmp
Wavelet	U	17	17	1	-4	4	/LucasKanade/Wavelet/17x17/SingleRes/LK_U_17x17_1_n4p4.bmp
Wavelet	V	17	17	1	-4	4	/LucasKanade/Wavelet/17x17/SingleRes/LK_V_17x17_1_n4p4.bmp
Wavelet	U	17	17	1	-1	1	/LucasKanade/Wavelet/17x17/SingleRes/LK_U_17x17_1_n1p1.bmp
Wavelet	V	17	17	1	-1	1	/LucasKanade/Wavelet/17x17/SingleRes/LK_V_17x17_1_n1p1.bmp
Wavelet	U	33	33	1	-4	4	/LucasKanade/Wavelet/33x33/SingleRes/LK_U_33x33_1_n4p4.bmp
Wavelet	V	33	33	1	-4	4	/LucasKanade/Wavelet/33x33/SingleRes/LK_V_33x33_1_n4p4.bmp
Wavelet	U	33	33	1	-1	1	/LucasKanade/Wavelet/33x33/SingleRes/LK_U_33x33_1_n1p1.bmp
Wavelet	V	33	33	1	-1	1	/LucasKanade/Wavelet/33x33/SingleRes/LK_V_33x33_1_n1p1.bmp
Wavelet	U	5	5	11	-4	4	/LucasKanade/Wavelet/5x5/MultiRes05/LK_U_5x5_11_n4p4.bmp
Wavelet	V	5	5	11	-4	4	/LucasKanade/Wavelet/5x5/MultiRes05/LK_V_5x5_11_n4p4.bmp
Wavelet	U	5	5	11	-1	1	/LucasKanade/Wavelet/5x5/MultiRes05/LK_U_5x5_11_n1p1.bmp
Wavelet	V	5	5	11	-1	1	/LucasKanade/Wavelet/5x5/MultiRes05/LK_V_5x5_11_n1p1.bmp

Background Type	Optical Flow Component	Window Width	Window Height	Layers	Min	Max	File
Wavelet	U	9	9	11	-4	4	/LucasKanade/Wavelet/9x9/MultiRes05/LK_U_9x9_11_n4p4.bmp
Wavelet	V	9	9	11	-4	4	/LucasKanade/Wavelet/9x9/MultiRes05/LK_V_9x9_11_n4p4.bmp
Wavelet	U	9	9	11	-1	1	/LucasKanade/Wavelet/9x9/MultiRes05/LK_U_9x9_11_n1p1.bmp
Wavelet	V	9	9	11	-1	1	/LucasKanade/Wavelet/9x9/MultiRes05/LK_V_9x9_11_n1p1.bmp
Wavelet	U	17	17	11	-4	4	/LucasKanade/Wavelet/17x17/MultiRes05/LK_U_17x17_11_n4p4.bmp
Wavelet	V	17	17	11	-4	4	/LucasKanade/Wavelet/17x17/MultiRes05/LK_V_17x17_11_n4p4.bmp
Wavelet	U	17	17	11	-1	1	/LucasKanade/Wavelet/17x17/MultiRes05/LK_U_17x17_11_n1p1.bmp
Wavelet	V	17	17	11	-1	1	/LucasKanade/Wavelet/17x17/MultiRes05/LK_V_17x17_11_n1p1.bmp
Wavelet	U	33	33	11	-4	4	/LucasKanade/Wavelet/33x33/MultiRes05/LK_U_33x33_11_n4p4.bmp
Wavelet	V	33	33	11	-4	4	/LucasKanade/Wavelet/33x33/MultiRes05/LK_V_33x33_11_n4p4.bmp
Wavelet	U	33	33	11	-1	1	/LucasKanade/Wavelet/33x33/MultiRes05/LK_U_33x33_11_n1p1.bmp
Wavelet	V	33	33	11	-1	1	/LucasKanade/Wavelet/33x33/MultiRes05/LK_V_33x33_11_n1p1.bmp
Speckle	U	5	5	1	-4	4	/LucasKanade/Speckle/5x5/SingleRes/LK_U_5x5_1_n4p4.bmp
Speckle	V	5	5	1	-4	4	/LucasKanade/Speckle/5x5/SingleRes/LK_V_5x5_1_n4p4.bmp
Speckle	U	5	5	1	-1	1	/LucasKanade/Speckle/5x5/SingleRes/LK_U_5x5_1_n1p1.bmp
Speckle	V	5	5	1	-1	1	/LucasKanade/Speckle/5x5/SingleRes/LK_V_5x5_1_n1p1.bmp
Speckle	U	9	9	1	-4	4	/LucasKanade/Speckle/9x9/SingleRes/LK_U_9x9_1_n4p4.bmp
Speckle	V	9	9	1	-4	4	/LucasKanade/Speckle/9x9/SingleRes/LK_V_9x9_1_n4p4.bmp
Speckle	U	9	9	1	-1	1	/LucasKanade/Speckle/9x9/SingleRes/LK_U_9x9_1_n1p1.bmp
Speckle	V	9	9	1	-1	1	/LucasKanade/Speckle/9x9/SingleRes/LK_V_9x9_1_n1p1.bmp
Speckle	U	17	17	1	-4	4	/LucasKanade/Speckle/17x17/SingleRes/LK_U_17x17_1_n4p4.bmp
Speckle	V	17	17	1	-4	4	/LucasKanade/Speckle/17x17/SingleRes/LK_V_17x17_1_n4p4.bmp
Speckle	U	17	17	1	-1	1	/LucasKanade/Speckle/17x17/SingleRes/LK_U_17x17_1_n1p1.bmp

Background Type	Optical Flow Component	Window Width	Window Height	Layers	Min	Max	File
Speckle	V	17	17	1	-1	1	/LucasKanade/Speckle/17x17/SingleRes/LK_V_17x17_1_n1p1.bmp
Speckle	U	33	33	1	-4	4	/LucasKanade/Speckle/33x33/SingleRes/LK_U_33x33_1_n4p4.bmp
Speckle	V	33	33	1	-4	4	/LucasKanade/Speckle/33x33/SingleRes/LK_V_33x33_1_n4p4.bmp
Speckle	U	33	33	1	-1	1	/LucasKanade/Speckle/33x33/SingleRes/LK_U_33x33_1_n1p1.bmp
Speckle	V	33	33	1	-1	1	/LucasKanade/Speckle/33x33/SingleRes/LK_V_33x33_1_n1p1.bmp
Speckle	U	5	5	11	-4	4	/LucasKanade/Speckle/5x5/MultiRes05/LK_U_5x5_11_n4p4.bmp
Speckle	V	5	5	11	-4	4	/LucasKanade/Speckle/5x5/MultiRes05/LK_V_5x5_11_n4p4.bmp
Speckle	U	5	5	11	-1	1	/LucasKanade/Speckle/5x5/MultiRes05/LK_U_5x5_11_n1p1.bmp
Speckle	V	5	5	11	-1	1	/LucasKanade/Speckle/5x5/MultiRes05/LK_V_5x5_11_n1p1.bmp
Speckle	U	9	9	11	-4	4	/LucasKanade/Speckle/9x9/MultiRes05/LK_U_9x9_11_n4p4.bmp
Speckle	V	9	9	11	-4	4	/LucasKanade/Speckle/9x9/MultiRes05/LK_V_9x9_11_n4p4.bmp
Speckle	U	9	9	11	-1	1	/LucasKanade/Speckle/9x9/MultiRes05/LK_U_9x9_11_n1p1.bmp
Speckle	V	9	9	11	-1	1	/LucasKanade/Speckle/9x9/MultiRes05/LK_V_9x9_11_n1p1.bmp
Speckle	U	17	17	11	-4	4	/LucasKanade/Speckle/17x17/MultiRes05/LK_U_17x17_11_n4p4.bmp
Speckle	V	17	17	11	-4	4	/LucasKanade/Speckle/17x17/MultiRes05/LK_V_17x17_11_n4p4.bmp
Speckle	U	17	17	11	-1	1	/LucasKanade/Speckle/17x17/MultiRes05/LK_U_17x17_11_n1p1.bmp
Speckle	V	17	17	11	-1	1	/LucasKanade/Speckle/17x17/MultiRes05/LK_V_17x17_11_n1p1.bmp
Speckle	U	33	33	11	-4	4	/LucasKanade/Speckle/33x33/MultiRes05/LK_U_33x33_11_n4p4.bmp
Speckle	V	33	33	11	-4	4	/LucasKanade/Speckle/33x33/MultiRes05/LK_V_33x33_11_n4p4.bmp
Speckle	U	33	33	11	-1	1	/LucasKanade/Speckle/33x33/MultiRes05/LK_U_33x33_11_n1p1.bmp
Speckle	V	33	33	11	-1	1	/LucasKanade/Speckle/33x33/MultiRes05/LK_V_33x33_11_n1p1.bmp

C.4 Brox algorithm results

Each image was processed using:

- 50 inner loop iterations
- 50 outer loop iterations

Table 22: Brox algorithm results

Background Type	Optical Flow Component	Smoothness Weight	Gradient Consistency Weight	Layers	Min	Max	File
Wavelet	U	0.1	0	1	-4	4	/Brox/Wavelet/01_0/SingleRes/BX_U_01_0_1_n4p4.bmp
Wavelet	V	0.1	0	1	-4	4	/Brox/Wavelet/01_0/SingleRes/BX_V_01_0_1_n4p4.bmp
Wavelet	U	0.1	0	1	-1	1	/Brox/Wavelet/01_0/SingleRes/BX_U_01_0_1_n1p1.bmp
Wavelet	V	0.1	0	1	-1	1	/Brox/Wavelet/01_0/SingleRes/BX_V_01_0_1_n1p1.bmp
Wavelet	U	0.1	50	1	-4	4	/Brox/Wavelet/01_50/SingleRes/BX_U_01_50_1_n4p4.bmp
Wavelet	V	0.1	50	1	-4	4	/Brox/Wavelet/01_50/SingleRes/BX_V_01_50_1_n4p4.bmp
Wavelet	U	0.1	50	1	-1	1	/Brox/Wavelet/01_50/SingleRes/BX_U_01_50_1_n1p1.bmp
Wavelet	V	0.1	50	1	-1	1	/Brox/Wavelet/01_50/SingleRes/BX_V_01_50_1_n1p1.bmp
Wavelet	U	0.1	100	1	-4	4	/Brox/Wavelet/01_100/SingleRes/BX_U_01_100_1_n4p4.bmp
Wavelet	V	0.1	100	1	-4	4	/Brox/Wavelet/01_100/SingleRes/BX_V_01_100_1_n4p4.bmp
Wavelet	U	0.1	100	1	-1	1	/Brox/Wavelet/01_100/SingleRes/BX_U_01_100_1_n1p1.bmp
Wavelet	V	0.1	100	1	-1	1	/Brox/Wavelet/01_100/SingleRes/BX_V_01_100_1_n1p1.bmp
Wavelet	U	0.2	0	1	-4	4	/Brox/Wavelet/02_0/SingleRes/BX_U_02_0_1_n4p4.bmp
Wavelet	V	0.2	0	1	-4	4	/Brox/Wavelet/02_0/SingleRes/BX_V_02_0_1_n4p4.bmp

Background Type	Optical Flow Component	Smoothness Weight	Gradient Consistency Weight	Layers	Min	Max	File
Wavelet	U	0.2	0	1	-1	1	/Brox/Wavelet/02_0/SingleRes/BX_U_02_0_1_n1p1.bmp
Wavelet	V	0.2	0	1	-1	1	/Brox/Wavelet/02_0/SingleRes/BX_V_02_0_1_n1p1.bmp
Wavelet	U	0.2	50	1	-4	4	/Brox/Wavelet/02_50/SingleRes/BX_U_02_50_1_n4p4.bmp
Wavelet	V	0.2	50	1	-4	4	/Brox/Wavelet/02_50/SingleRes/BX_V_02_50_1_n4p4.bmp
Wavelet	U	0.2	50	1	-1	1	/Brox/Wavelet/02_50/SingleRes/BX_U_02_50_1_n1p1.bmp
Wavelet	V	0.2	50	1	-1	1	/Brox/Wavelet/02_50/SingleRes/BX_V_02_50_1_n1p1.bmp
Wavelet	U	0.2	100	1	-4	4	/Brox/Wavelet/02_100/SingleRes/BX_U_02_100_1_n4p4.bmp
Wavelet	V	0.2	100	1	-4	4	/Brox/Wavelet/02_100/SingleRes/BX_V_02_100_1_n4p4.bmp
Wavelet	U	0.2	100	1	-1	1	/Brox/Wavelet/02_100/SingleRes/BX_U_02_100_1_n1p1.bmp
Wavelet	V	0.2	100	1	-1	1	/Brox/Wavelet/02_100/SingleRes/BX_V_02_100_1_n1p1.bmp
Wavelet	U	0.4	0	1	-4	4	/Brox/Wavelet/04_0/SingleRes/BX_U_04_0_1_n4p4.bmp
Wavelet	V	0.4	0	1	-4	4	/Brox/Wavelet/04_0/SingleRes/BX_V_04_0_1_n4p4.bmp
Wavelet	U	0.4	0	1	-1	1	/Brox/Wavelet/04_0/SingleRes/BX_U_04_0_1_n1p1.bmp
Wavelet	V	0.4	0	1	-1	1	/Brox/Wavelet/04_0/SingleRes/BX_V_04_0_1_n1p1.bmp
Wavelet	U	0.4	50	1	-4	4	/Brox/Wavelet/04_50/SingleRes/BX_U_04_50_1_n4p4.bmp
Wavelet	V	0.4	50	1	-4	4	/Brox/Wavelet/04_50/SingleRes/BX_V_04_50_1_n4p4.bmp
Wavelet	U	0.4	50	1	-1	1	/Brox/Wavelet/04_50/SingleRes/BX_U_04_50_1_n1p1.bmp
Wavelet	V	0.4	50	1	-1	1	/Brox/Wavelet/04_50/SingleRes/BX_V_04_50_1_n1p1.bmp
Wavelet	U	0.4	100	1	-4	4	/Brox/Wavelet/04_100/SingleRes/BX_U_04_100_1_n4p4.bmp
Wavelet	V	0.4	100	1	-4	4	/Brox/Wavelet/04_100/SingleRes/BX_V_04_100_1_n4p4.bmp
Wavelet	U	0.4	100	1	-1	1	/Brox/Wavelet/04_100/SingleRes/BX_U_04_100_1_n1p1.bmp
Wavelet	V	0.4	100	1	-1	1	/Brox/Wavelet/04_100/SingleRes/BX_V_04_100_1_n1p1.bmp
Wavelet	U	0.1	0	11	-4	4	/Brox/Wavelet/01_0/MultiRes05/BX_U_01_0_11_n4p4.bmp

Background Type	Optical Flow Component	Smoothness Weight	Gradient Consistency Weight	Layers	Min	Max	File
Wavelet	V	0.1	0	11	-4	4	/Brox/Wavelet/01_0/MultiRes05/BX_V_01_0_11_n4p4.bmp
Wavelet	U	0.1	0	11	-1	1	/Brox/Wavelet/01_0/MultiRes05/BX_U_01_0_11_n1p1.bmp
Wavelet	V	0.1	0	11	-1	1	/Brox/Wavelet/01_0/MultiRes05/BX_V_01_0_11_n1p1.bmp
Wavelet	U	0.1	50	11	-4	4	/Brox/Wavelet/01_50/MultiRes05/BX_U_01_50_11_n4p4.bmp
Wavelet	V	0.1	50	11	-4	4	/Brox/Wavelet/01_50/MultiRes05/BX_V_01_50_11_n4p4.bmp
Wavelet	U	0.1	50	11	-1	1	/Brox/Wavelet/01_50/MultiRes05/BX_U_01_50_11_n1p1.bmp
Wavelet	V	0.1	50	11	-1	1	/Brox/Wavelet/01_50/MultiRes05/BX_V_01_50_11_n1p1.bmp
Wavelet	U	0.1	100	11	-4	4	/Brox/Wavelet/01_100/MultiRes05/BX_U_01_100_11_n4p4.bmp
Wavelet	V	0.1	100	11	-4	4	/Brox/Wavelet/01_100/MultiRes05/BX_V_01_100_11_n4p4.bmp
Wavelet	U	0.1	100	11	-1	1	/Brox/Wavelet/01_100/MultiRes05/BX_U_01_100_11_n1p1.bmp
Wavelet	V	0.1	100	11	-1	1	/Brox/Wavelet/01_100/MultiRes05/BX_V_01_100_11_n1p1.bmp
Wavelet	U	0.2	0	11	-4	4	/Brox/Wavelet/02_0/MultiRes05/BX_U_02_0_11_n4p4.bmp
Wavelet	V	0.2	0	11	-4	4	/Brox/Wavelet/02_0/MultiRes05/BX_V_02_0_11_n4p4.bmp
Wavelet	U	0.2	0	11	-1	1	/Brox/Wavelet/02_0/MultiRes05/BX_U_02_0_11_n1p1.bmp
Wavelet	V	0.2	0	11	-1	1	/Brox/Wavelet/02_0/MultiRes05/BX_V_02_0_11_n1p1.bmp
Wavelet	U	0.2	50	11	-4	4	/Brox/Wavelet/02_50/MultiRes05/BX_U_02_50_11_n4p4.bmp
Wavelet	V	0.2	50	11	-4	4	/Brox/Wavelet/02_50/MultiRes05/BX_V_02_50_11_n4p4.bmp
Wavelet	U	0.2	50	11	-1	1	/Brox/Wavelet/02_50/MultiRes05/BX_U_02_50_11_n1p1.bmp
Wavelet	V	0.2	50	11	-1	1	/Brox/Wavelet/02_50/MultiRes05/BX_V_02_50_11_n1p1.bmp
Wavelet	U	0.2	100	11	-4	4	/Brox/Wavelet/02_100/MultiRes05/BX_U_02_100_11_n4p4.bmp
Wavelet	V	0.2	100	11	-4	4	/Brox/Wavelet/02_100/MultiRes05/BX_V_02_100_11_n4p4.bmp
Wavelet	U	0.2	100	11	-1	1	/Brox/Wavelet/02_100/MultiRes05/BX_U_02_100_11_n1p1.bmp
Wavelet	V	0.2	100	11	-1	1	/Brox/Wavelet/02_100/MultiRes05/BX_V_02_100_11_n1p1.bmp

Background Type	Optical Flow Component	Smoothness Weight	Gradient Consistency Weight	Layers	Min	Max	File
Wavelet	U	0.4	0	11	-4	4	/Brox/Wavelet/04_0/MultiRes05/BX_U_04_0_11_n4p4.bmp
Wavelet	V	0.4	0	11	-4	4	/Brox/Wavelet/04_0/MultiRes05/BX_V_04_0_11_n4p4.bmp
Wavelet	U	0.4	0	11	-1	1	/Brox/Wavelet/04_0/MultiRes05/BX_U_04_0_11_n1p1.bmp
Wavelet	V	0.4	0	11	-1	1	/Brox/Wavelet/04_0/MultiRes05/BX_V_04_0_11_n1p1.bmp
Wavelet	U	0.4	50	11	-4	4	/Brox/Wavelet/04_50/MultiRes05/BX_U_04_50_11_n4p4.bmp
Wavelet	V	0.4	50	11	-4	4	/Brox/Wavelet/04_50/MultiRes05/BX_V_04_50_11_n4p4.bmp
Wavelet	U	0.4	50	11	-1	1	/Brox/Wavelet/04_50/MultiRes05/BX_U_04_50_11_n1p1.bmp
Wavelet	V	0.4	50	11	-1	1	/Brox/Wavelet/04_50/MultiRes05/BX_V_04_50_11_n1p1.bmp
Wavelet	U	0.4	100	11	-4	4	/Brox/Wavelet/04_100/MultiRes05/BX_U_04_100_11_n4p4.bmp
Wavelet	V	0.4	100	11	-4	4	/Brox/Wavelet/04_100/MultiRes05/BX_V_04_100_11_n4p4.bmp
Wavelet	U	0.4	100	11	-1	1	/Brox/Wavelet/04_100/MultiRes05/BX_U_04_100_11_n1p1.bmp
Wavelet	V	0.4	100	11	-1	1	/Brox/Wavelet/04_100/MultiRes05/BX_V_04_100_11_n1p1.bmp
Wavelet	U	0.1	0	35	-4	4	/Brox/Wavelet/01_0/MultiRes08/BX_U_01_0_35_n4p4.bmp
Wavelet	V	0.1	0	35	-4	4	/Brox/Wavelet/01_0/MultiRes08/BX_V_01_0_35_n4p4.bmp
Wavelet	U	0.1	0	35	-1	1	/Brox/Wavelet/01_0/MultiRes08/BX_U_01_0_35_n1p1.bmp
Wavelet	V	0.1	0	35	-1	1	/Brox/Wavelet/01_0/MultiRes08/BX_V_01_0_35_n1p1.bmp
Wavelet	U	0.1	50	35	-4	4	/Brox/Wavelet/01_50/MultiRes08/BX_U_01_50_35_n4p4.bmp
Wavelet	V	0.1	50	35	-4	4	/Brox/Wavelet/01_50/MultiRes08/BX_V_01_50_35_n4p4.bmp
Wavelet	U	0.1	50	35	-1	1	/Brox/Wavelet/01_50/MultiRes08/BX_U_01_50_35_n1p1.bmp
Wavelet	V	0.1	50	35	-1	1	/Brox/Wavelet/01_50/MultiRes08/BX_V_01_50_35_n1p1.bmp
Wavelet	U	0.1	100	35	-4	4	/Brox/Wavelet/01_100/MultiRes08/BX_U_01_100_35_n4p4.bmp
Wavelet	V	0.1	100	35	-4	4	/Brox/Wavelet/01_100/MultiRes08/BX_V_01_100_35_n4p4.bmp
Wavelet	U	0.1	100	35	-1	1	/Brox/Wavelet/01_100/MultiRes08/BX_U_01_100_35_n1p1.bmp

Background Type	Optical Flow Component	Smoothness Weight	Gradient Consistency Weight	Layers	Min	Max	File
Wavelet	V	0.1	100	35	-1	1	/Brox/Wavelet/01_100/MultiRes08/BX_V_01_100_35_n1p1.bmp
Wavelet	U	0.2	0	35	-4	4	/Brox/Wavelet/02_0/MultiRes08/BX_U_02_0_35_n4p4.bmp
Wavelet	V	0.2	0	35	-4	4	/Brox/Wavelet/02_0/MultiRes08/BX_V_02_0_35_n4p4.bmp
Wavelet	U	0.2	0	35	-1	1	/Brox/Wavelet/02_0/MultiRes08/BX_U_02_0_35_n1p1.bmp
Wavelet	V	0.2	0	35	-1	1	/Brox/Wavelet/02_0/MultiRes08/BX_V_02_0_35_n1p1.bmp
Wavelet	U	0.2	50	35	-4	4	/Brox/Wavelet/02_50/MultiRes08/BX_U_02_50_35_n4p4.bmp
Wavelet	V	0.2	50	35	-4	4	/Brox/Wavelet/02_50/MultiRes08/BX_V_02_50_35_n4p4.bmp
Wavelet	U	0.2	50	35	-1	1	/Brox/Wavelet/02_50/MultiRes08/BX_U_02_50_35_n1p1.bmp
Wavelet	V	0.2	50	35	-1	1	/Brox/Wavelet/02_50/MultiRes08/BX_V_02_50_35_n1p1.bmp
Wavelet	U	0.2	100	35	-4	4	/Brox/Wavelet/02_100/MultiRes08/BX_U_02_100_35_n4p4.bmp
Wavelet	V	0.2	100	35	-4	4	/Brox/Wavelet/02_100/MultiRes08/BX_V_02_100_35_n4p4.bmp
Wavelet	U	0.2	100	35	-1	1	/Brox/Wavelet/02_100/MultiRes08/BX_U_02_100_35_n1p1.bmp
Wavelet	V	0.2	100	35	-1	1	/Brox/Wavelet/02_100/MultiRes08/BX_V_02_100_35_n1p1.bmp
Wavelet	U	0.4	0	35	-4	4	/Brox/Wavelet/04_0/MultiRes08/BX_U_04_0_35_n4p4.bmp
Wavelet	V	0.4	0	35	-4	4	/Brox/Wavelet/04_0/MultiRes08/BX_V_04_0_35_n4p4.bmp
Wavelet	U	0.4	0	35	-1	1	/Brox/Wavelet/04_0/MultiRes08/BX_U_04_0_35_n1p1.bmp
Wavelet	V	0.4	0	35	-1	1	/Brox/Wavelet/04_0/MultiRes08/BX_V_04_0_35_n1p1.bmp
Wavelet	U	0.4	50	35	-4	4	/Brox/Wavelet/04_50/MultiRes08/BX_U_04_50_35_n4p4.bmp
Wavelet	V	0.4	50	35	-4	4	/Brox/Wavelet/04_50/MultiRes08/BX_V_04_50_35_n4p4.bmp
Wavelet	U	0.4	50	35	-1	1	/Brox/Wavelet/04_50/MultiRes08/BX_U_04_50_35_n1p1.bmp
Wavelet	V	0.4	50	35	-1	1	/Brox/Wavelet/04_50/MultiRes08/BX_V_04_50_35_n1p1.bmp
Wavelet	U	0.4	100	35	-4	4	/Brox/Wavelet/04_100/MultiRes08/BX_U_04_100_35_n4p4.bmp
Wavelet	V	0.4	100	35	-4	4	/Brox/Wavelet/04_100/MultiRes08/BX_V_04_100_35_n4p4.bmp

Background Type	Optical Flow Component	Smoothness Weight	Gradient Consistency Weight	Layers	Min	Max	File
Wavelet	U	0.4	100	35	-1	1	/Brox/Wavelet/04_100/MultiRes08/BX_U_04_100_35_n1p1.bmp
Wavelet	V	0.4	100	35	-1	1	/Brox/Wavelet/04_100/MultiRes08/BX_V_04_100_35_n1p1.bmp
Speckle	U	0.1	0	1	-4	4	/Brox/Speckle/01_0/SingleRes/BX_U_01_0_1_n4p4.bmp
Speckle	V	0.1	0	1	-4	4	/Brox/Speckle/01_0/SingleRes/BX_V_01_0_1_n4p4.bmp
Speckle	U	0.1	0	1	-1	1	/Brox/Speckle/01_0/SingleRes/BX_U_01_0_1_n1p1.bmp
Speckle	V	0.1	0	1	-1	1	/Brox/Speckle/01_0/SingleRes/BX_V_01_0_1_n1p1.bmp
Speckle	U	0.1	50	1	-4	4	/Brox/Speckle/01_50/SingleRes/BX_U_01_50_1_n4p4.bmp
Speckle	V	0.1	50	1	-4	4	/Brox/Speckle/01_50/SingleRes/BX_V_01_50_1_n4p4.bmp
Speckle	U	0.1	50	1	-1	1	/Brox/Speckle/01_50/SingleRes/BX_U_01_50_1_n1p1.bmp
Speckle	V	0.1	50	1	-1	1	/Brox/Speckle/01_50/SingleRes/BX_V_01_50_1_n1p1.bmp
Speckle	U	0.1	100	1	-4	4	/Brox/Speckle/01_100/SingleRes/BX_U_01_100_1_n4p4.bmp
Speckle	V	0.1	100	1	-4	4	/Brox/Speckle/01_100/SingleRes/BX_V_01_100_1_n4p4.bmp
Speckle	U	0.1	100	1	-1	1	/Brox/Speckle/01_100/SingleRes/BX_U_01_100_1_n1p1.bmp
Speckle	V	0.1	100	1	-1	1	/Brox/Speckle/01_100/SingleRes/BX_V_01_100_1_n1p1.bmp
Speckle	U	0.2	0	1	-4	4	/Brox/Speckle/02_0/SingleRes/BX_U_02_0_1_n4p4.bmp
Speckle	V	0.2	0	1	-4	4	/Brox/Speckle/02_0/SingleRes/BX_V_02_0_1_n4p4.bmp
Speckle	U	0.2	0	1	-1	1	/Brox/Speckle/02_0/SingleRes/BX_U_02_0_1_n1p1.bmp
Speckle	V	0.2	0	1	-1	1	/Brox/Speckle/02_0/SingleRes/BX_V_02_0_1_n1p1.bmp
Speckle	U	0.2	50	1	-4	4	/Brox/Speckle/02_50/SingleRes/BX_U_02_50_1_n4p4.bmp
Speckle	V	0.2	50	1	-4	4	/Brox/Speckle/02_50/SingleRes/BX_V_02_50_1_n4p4.bmp
Speckle	U	0.2	50	1	-1	1	/Brox/Speckle/02_50/SingleRes/BX_U_02_50_1_n1p1.bmp
Speckle	V	0.2	50	1	-1	1	/Brox/Speckle/02_50/SingleRes/BX_V_02_50_1_n1p1.bmp
Speckle	U	0.2	100	1	-4	4	/Brox/Speckle/02_100/SingleRes/BX_U_02_100_1_n4p4.bmp

Background Type	Optical Flow Component	Smoothness Weight	Gradient Consistency Weight	Layers	Min	Max	File
Speckle	V	0.2	100	1	-4	4	/Brox/Speckle/02_100/SingleRes/BX_V_02_100_1_n4p4.bmp
Speckle	U	0.2	100	1	-1	1	/Brox/Speckle/02_100/SingleRes/BX_U_02_100_1_n1p1.bmp
Speckle	V	0.2	100	1	-1	1	/Brox/Speckle/02_100/SingleRes/BX_V_02_100_1_n1p1.bmp
Speckle	U	0.4	0	1	-4	4	/Brox/Speckle/04_0/SingleRes/BX_U_04_0_1_n4p4.bmp
Speckle	V	0.4	0	1	-4	4	/Brox/Speckle/04_0/SingleRes/BX_V_04_0_1_n4p4.bmp
Speckle	U	0.4	0	1	-1	1	/Brox/Speckle/04_0/SingleRes/BX_U_04_0_1_n1p1.bmp
Speckle	V	0.4	0	1	-1	1	/Brox/Speckle/04_0/SingleRes/BX_V_04_0_1_n1p1.bmp
Speckle	U	0.4	50	1	-4	4	/Brox/Speckle/04_50/SingleRes/BX_U_04_50_1_n4p4.bmp
Speckle	V	0.4	50	1	-4	4	/Brox/Speckle/04_50/SingleRes/BX_V_04_50_1_n4p4.bmp
Speckle	U	0.4	50	1	-1	1	/Brox/Speckle/04_50/SingleRes/BX_U_04_50_1_n1p1.bmp
Speckle	V	0.4	50	1	-1	1	/Brox/Speckle/04_50/SingleRes/BX_V_04_50_1_n1p1.bmp
Speckle	U	0.4	100	1	-4	4	/Brox/Speckle/04_100/SingleRes/BX_U_04_100_1_n4p4.bmp
Speckle	V	0.4	100	1	-4	4	/Brox/Speckle/04_100/SingleRes/BX_V_04_100_1_n4p4.bmp
Speckle	U	0.4	100	1	-1	1	/Brox/Speckle/04_100/SingleRes/BX_U_04_100_1_n1p1.bmp
Speckle	V	0.4	100	1	-1	1	/Brox/Speckle/04_100/SingleRes/BX_V_04_100_1_n1p1.bmp
Speckle	U	0.1	0	11	-4	4	/Brox/Speckle/01_0/MultiRes05/BX_U_01_0_11_n4p4.bmp
Speckle	V	0.1	0	11	-4	4	/Brox/Speckle/01_0/MultiRes05/BX_V_01_0_11_n4p4.bmp
Speckle	U	0.1	0	11	-1	1	/Brox/Speckle/01_0/MultiRes05/BX_U_01_0_11_n1p1.bmp
Speckle	V	0.1	0	11	-1	1	/Brox/Speckle/01_0/MultiRes05/BX_V_01_0_11_n1p1.bmp
Speckle	U	0.1	50	11	-4	4	/Brox/Speckle/01_50/MultiRes05/BX_U_01_50_11_n4p4.bmp
Speckle	V	0.1	50	11	-4	4	/Brox/Speckle/01_50/MultiRes05/BX_V_01_50_11_n4p4.bmp
Speckle	U	0.1	50	11	-1	1	/Brox/Speckle/01_50/MultiRes05/BX_U_01_50_11_n1p1.bmp
Speckle	V	0.1	50	11	-1	1	/Brox/Speckle/01_50/MultiRes05/BX_V_01_50_11_n1p1.bmp

Background Type	Optical Flow Component	Smoothness Weight	Gradient Consistency Weight	Layers	Min	Max	File
Speckle	U	0.1	100	11	-4	4	/Brox/Speckle/01_100/MultiRes05/BX_U_01_100_11_n4p4.bmp
Speckle	V	0.1	100	11	-4	4	/Brox/Speckle/01_100/MultiRes05/BX_V_01_100_11_n4p4.bmp
Speckle	U	0.1	100	11	-1	1	/Brox/Speckle/01_100/MultiRes05/BX_U_01_100_11_n1p1.bmp
Speckle	V	0.1	100	11	-1	1	/Brox/Speckle/01_100/MultiRes05/BX_V_01_100_11_n1p1.bmp
Speckle	U	0.2	0	11	-4	4	/Brox/Speckle/02_0/MultiRes05/BX_U_02_0_11_n4p4.bmp
Speckle	V	0.2	0	11	-4	4	/Brox/Speckle/02_0/MultiRes05/BX_V_02_0_11_n4p4.bmp
Speckle	U	0.2	0	11	-1	1	/Brox/Speckle/02_0/MultiRes05/BX_U_02_0_11_n1p1.bmp
Speckle	V	0.2	0	11	-1	1	/Brox/Speckle/02_0/MultiRes05/BX_V_02_0_11_n1p1.bmp
Speckle	U	0.2	50	11	-4	4	/Brox/Speckle/02_50/MultiRes05/BX_U_02_50_11_n4p4.bmp
Speckle	V	0.2	50	11	-4	4	/Brox/Speckle/02_50/MultiRes05/BX_V_02_50_11_n4p4.bmp
Speckle	U	0.2	50	11	-1	1	/Brox/Speckle/02_50/MultiRes05/BX_U_02_50_11_n1p1.bmp
Speckle	V	0.2	50	11	-1	1	/Brox/Speckle/02_50/MultiRes05/BX_V_02_50_11_n1p1.bmp
Speckle	U	0.2	100	11	-4	4	/Brox/Speckle/02_100/MultiRes05/BX_U_02_100_11_n4p4.bmp
Speckle	V	0.2	100	11	-4	4	/Brox/Speckle/02_100/MultiRes05/BX_V_02_100_11_n4p4.bmp
Speckle	U	0.2	100	11	-1	1	/Brox/Speckle/02_100/MultiRes05/BX_U_02_100_11_n1p1.bmp
Speckle	V	0.2	100	11	-1	1	/Brox/Speckle/02_100/MultiRes05/BX_V_02_100_11_n1p1.bmp
Speckle	U	0.4	0	11	-4	4	/Brox/Speckle/04_0/MultiRes05/BX_U_04_0_11_n4p4.bmp
Speckle	V	0.4	0	11	-4	4	/Brox/Speckle/04_0/MultiRes05/BX_V_04_0_11_n4p4.bmp
Speckle	U	0.4	0	11	-1	1	/Brox/Speckle/04_0/MultiRes05/BX_U_04_0_11_n1p1.bmp
Speckle	V	0.4	0	11	-1	1	/Brox/Speckle/04_0/MultiRes05/BX_V_04_0_11_n1p1.bmp
Speckle	U	0.4	50	11	-4	4	/Brox/Speckle/04_50/MultiRes05/BX_U_04_50_11_n4p4.bmp
Speckle	V	0.4	50	11	-4	4	/Brox/Speckle/04_50/MultiRes05/BX_V_04_50_11_n4p4.bmp
Speckle	U	0.4	50	11	-1	1	/Brox/Speckle/04_50/MultiRes05/BX_U_04_50_11_n1p1.bmp

Background Type	Optical Flow Component	Smoothness Weight	Gradient Consistency Weight	Layers	Min	Max	File
Speckle	V	0.4	50	11	-1	1	/Brox/Speckle/04_50/MultiRes05/BX_V_04_50_11_n1p1.bmp
Speckle	U	0.4	100	11	-4	4	/Brox/Speckle/04_100/MultiRes05/BX_U_04_100_11_n4p4.bmp
Speckle	V	0.4	100	11	-4	4	/Brox/Speckle/04_100/MultiRes05/BX_V_04_100_11_n4p4.bmp
Speckle	U	0.4	100	11	-1	1	/Brox/Speckle/04_100/MultiRes05/BX_U_04_100_11_n1p1.bmp
Speckle	V	0.4	100	11	-1	1	/Brox/Speckle/04_100/MultiRes05/BX_V_04_100_11_n1p1.bmp
Speckle	U	0.1	0	35	-4	4	/Brox/Speckle/01_0/MultiRes08/BX_U_01_0_35_n4p4.bmp
Speckle	V	0.1	0	35	-4	4	/Brox/Speckle/01_0/MultiRes08/BX_V_01_0_35_n4p4.bmp
Speckle	U	0.1	0	35	-1	1	/Brox/Speckle/01_0/MultiRes08/BX_U_01_0_35_n1p1.bmp
Speckle	V	0.1	0	35	-1	1	/Brox/Speckle/01_0/MultiRes08/BX_V_01_0_35_n1p1.bmp
Speckle	U	0.1	50	35	-4	4	/Brox/Speckle/01_50/MultiRes08/BX_U_01_50_35_n4p4.bmp
Speckle	V	0.1	50	35	-4	4	/Brox/Speckle/01_50/MultiRes08/BX_V_01_50_35_n4p4.bmp
Speckle	U	0.1	50	35	-1	1	/Brox/Speckle/01_50/MultiRes08/BX_U_01_50_35_n1p1.bmp
Speckle	V	0.1	50	35	-1	1	/Brox/Speckle/01_50/MultiRes08/BX_V_01_50_35_n1p1.bmp
Speckle	U	0.1	100	35	-4	4	/Brox/Speckle/01_100/MultiRes08/BX_U_01_100_35_n4p4.bmp
Speckle	V	0.1	100	35	-4	4	/Brox/Speckle/01_100/MultiRes08/BX_V_01_100_35_n4p4.bmp
Speckle	U	0.1	100	35	-1	1	/Brox/Speckle/01_100/MultiRes08/BX_U_01_100_35_n1p1.bmp
Speckle	V	0.1	100	35	-1	1	/Brox/Speckle/01_100/MultiRes08/BX_V_01_100_35_n1p1.bmp
Speckle	U	0.2	0	35	-4	4	/Brox/Speckle/02_0/MultiRes08/BX_U_02_0_35_n4p4.bmp
Speckle	V	0.2	0	35	-4	4	/Brox/Speckle/02_0/MultiRes08/BX_V_02_0_35_n4p4.bmp
Speckle	U	0.2	0	35	-1	1	/Brox/Speckle/02_0/MultiRes08/BX_U_02_0_35_n1p1.bmp
Speckle	V	0.2	0	35	-1	1	/Brox/Speckle/02_0/MultiRes08/BX_V_02_0_35_n1p1.bmp
Speckle	U	0.2	50	35	-4	4	/Brox/Speckle/02_50/MultiRes08/BX_U_02_50_35_n4p4.bmp
Speckle	V	0.2	50	35	-4	4	/Brox/Speckle/02_50/MultiRes08/BX_V_02_50_35_n4p4.bmp

Background Type	Optical Flow Component	Smoothness Weight	Gradient Consistency Weight	Layers	Min	Max	File
Speckle	U	0.2	50	35	-1	1	/Brox/Speckle/02_50/MultiRes08/BX_U_02_50_35_n1p1.bmp
Speckle	V	0.2	50	35	-1	1	/Brox/Speckle/02_50/MultiRes08/BX_V_02_50_35_n1p1.bmp
Speckle	U	0.2	100	35	-4	4	/Brox/Speckle/02_100/MultiRes08/BX_U_02_100_35_n4p4.bmp
Speckle	V	0.2	100	35	-4	4	/Brox/Speckle/02_100/MultiRes08/BX_V_02_100_35_n4p4.bmp
Speckle	U	0.2	100	35	-1	1	/Brox/Speckle/02_100/MultiRes08/BX_U_02_100_35_n1p1.bmp
Speckle	V	0.2	100	35	-1	1	/Brox/Speckle/02_100/MultiRes08/BX_V_02_100_35_n1p1.bmp
Speckle	U	0.4	0	35	-4	4	/Brox/Speckle/04_0/MultiRes08/BX_U_04_0_35_n4p4.bmp
Speckle	V	0.4	0	35	-4	4	/Brox/Speckle/04_0/MultiRes08/BX_V_04_0_35_n4p4.bmp
Speckle	U	0.4	0	35	-1	1	/Brox/Speckle/04_0/MultiRes08/BX_U_04_0_35_n1p1.bmp
Speckle	V	0.4	0	35	-1	1	/Brox/Speckle/04_0/MultiRes08/BX_V_04_0_35_n1p1.bmp
Speckle	U	0.4	50	35	-4	4	/Brox/Speckle/04_50/MultiRes08/BX_U_04_50_35_n4p4.bmp
Speckle	V	0.4	50	35	-4	4	/Brox/Speckle/04_50/MultiRes08/BX_V_04_50_35_n4p4.bmp
Speckle	U	0.4	50	35	-1	1	/Brox/Speckle/04_50/MultiRes08/BX_U_04_50_35_n1p1.bmp
Speckle	V	0.4	50	35	-1	1	/Brox/Speckle/04_50/MultiRes08/BX_V_04_50_35_n1p1.bmp
Speckle	U	0.4	100	35	-4	4	/Brox/Speckle/04_100/MultiRes08/BX_U_04_100_35_n4p4.bmp
Speckle	V	0.4	100	35	-4	4	/Brox/Speckle/04_100/MultiRes08/BX_V_04_100_35_n4p4.bmp
Speckle	U	0.4	100	35	-1	1	/Brox/Speckle/04_100/MultiRes08/BX_U_04_100_35_n1p1.bmp
Speckle	V	0.4	100	35	-1	1	/Brox/Speckle/04_100/MultiRes08/BX_V_04_100_35_n1p1.bmp

C.5 Farnebäck Results

Each image was processed using 100 iterations

Table 23: Farnebäck algorithm results

Background Type	Optical Flow Component	Greyscale Window Size	Optical Flow Window Size	Layers	Min	Max	File
Wavelet	U	4	5	1	-4	4	/Farneback/Wavelet/4_5/SingleRes/FB_U_4_5_1_n4p4.bmp
Wavelet	V	4	5	1	-4	4	/Farneback/Wavelet/4_5/SingleRes/FB_V_4_5_1_n4p4.bmp
Wavelet	U	4	5	1	-1	1	/Farneback/Wavelet/4_5/SingleRes/FB_U_4_5_1_n1p1.bmp
Wavelet	V	4	5	1	-1	1	/Farneback/Wavelet/4_5/SingleRes/FB_V_4_5_1_n1p1.bmp
Wavelet	U	4	7	1	-4	4	/Farneback/Wavelet/4_7/SingleRes/FB_U_4_7_1_n4p4.bmp
Wavelet	V	4	7	1	-4	4	/Farneback/Wavelet/4_7/SingleRes/FB_V_4_7_1_n4p4.bmp
Wavelet	U	4	7	1	-1	1	/Farneback/Wavelet/4_7/SingleRes/FB_U_4_7_1_n1p1.bmp
Wavelet	V	4	7	1	-1	1	/Farneback/Wavelet/4_7/SingleRes/FB_V_4_7_1_n1p1.bmp
Wavelet	U	8	5	1	-4	4	/Farneback/Wavelet/8_5/SingleRes/FB_U_8_5_1_n4p4.bmp
Wavelet	V	8	5	1	-4	4	/Farneback/Wavelet/8_5/SingleRes/FB_V_8_5_1_n4p4.bmp
Wavelet	U	8	5	1	-1	1	/Farneback/Wavelet/8_5/SingleRes/FB_U_8_5_1_n1p1.bmp
Wavelet	V	8	5	1	-1	1	/Farneback/Wavelet/8_5/SingleRes/FB_V_8_5_1_n1p1.bmp
Wavelet	U	8	7	1	-4	4	/Farneback/Wavelet/8_7/SingleRes/FB_U_8_7_1_n4p4.bmp
Wavelet	V	8	7	1	-4	4	/Farneback/Wavelet/8_7/SingleRes/FB_V_8_7_1_n4p4.bmp
Wavelet	U	8	7	1	-1	1	/Farneback/Wavelet/8_7/SingleRes/FB_U_8_7_1_n1p1.bmp
Wavelet	V	8	7	1	-1	1	/Farneback/Wavelet/8_7/SingleRes/FB_V_8_7_1_n1p1.bmp
Wavelet	U	16	5	1	-4	4	/Farneback/Wavelet/16_5/SingleRes/FB_U_16_5_1_n4p4.bmp

Background Type	Optical Flow Component	Greyscale Window Size	Optical Flow Window Size	Layers	Min	Max	File
Wavelet	V	16	5	1	-4	4	/Farneback/Wavelet/16_5/SingleRes/FB_V_16_5_1_n4p4.bmp
Wavelet	U	16	5	1	-1	1	/Farneback/Wavelet/16_5/SingleRes/FB_U_16_5_1_n1p1.bmp
Wavelet	V	16	5	1	-1	1	/Farneback/Wavelet/16_5/SingleRes/FB_V_16_5_1_n1p1.bmp
Wavelet	U	16	7	1	-4	4	/Farneback/Wavelet/16_7/SingleRes/FB_U_16_7_1_n4p4.bmp
Wavelet	V	16	7	1	-4	4	/Farneback/Wavelet/16_7/SingleRes/FB_V_16_7_1_n4p4.bmp
Wavelet	U	16	7	1	-1	1	/Farneback/Wavelet/16_7/SingleRes/FB_U_16_7_1_n1p1.bmp
Wavelet	V	16	7	1	-1	1	/Farneback/Wavelet/16_7/SingleRes/FB_V_16_7_1_n1p1.bmp
Wavelet	U	32	5	1	-4	4	/Farneback/Wavelet/32_5/SingleRes/FB_U_32_5_1_n4p4.bmp
Wavelet	V	32	5	1	-4	4	/Farneback/Wavelet/32_5/SingleRes/FB_V_32_5_1_n4p4.bmp
Wavelet	U	32	5	1	-1	1	/Farneback/Wavelet/32_5/SingleRes/FB_U_32_5_1_n1p1.bmp
Wavelet	V	32	5	1	-1	1	/Farneback/Wavelet/32_5/SingleRes/FB_V_32_5_1_n1p1.bmp
Wavelet	U	32	7	1	-4	4	/Farneback/Wavelet/32_7/SingleRes/FB_U_32_7_1_n4p4.bmp
Wavelet	V	32	7	1	-4	4	/Farneback/Wavelet/32_7/SingleRes/FB_V_32_7_1_n4p4.bmp
Wavelet	U	32	7	1	-1	1	/Farneback/Wavelet/32_7/SingleRes/FB_U_32_7_1_n1p1.bmp
Wavelet	V	32	7	1	-1	1	/Farneback/Wavelet/32_7/SingleRes/FB_V_32_7_1_n1p1.bmp
Wavelet	U	4	5	11	-4	4	/Farneback/Wavelet/4_5/MultiRes05/FB_U_4_5_11_n4p4.bmp
Wavelet	V	4	5	11	-4	4	/Farneback/Wavelet/4_5/MultiRes05/FB_V_4_5_11_n4p4.bmp
Wavelet	U	4	5	11	-1	1	/Farneback/Wavelet/4_5/MultiRes05/FB_U_4_5_11_n1p1.bmp
Wavelet	V	4	5	11	-1	1	/Farneback/Wavelet/4_5/MultiRes05/FB_V_4_5_11_n1p1.bmp
Wavelet	U	4	7	11	-4	4	/Farneback/Wavelet/4_7/MultiRes05/FB_U_4_7_11_n4p4.bmp
Wavelet	V	4	7	11	-4	4	/Farneback/Wavelet/4_7/MultiRes05/FB_V_4_7_11_n4p4.bmp
Wavelet	U	4	7	11	-1	1	/Farneback/Wavelet/4_7/MultiRes05/FB_U_4_7_11_n1p1.bmp
Wavelet	V	4	7	11	-1	1	/Farneback/Wavelet/4_7/MultiRes05/FB_V_4_7_11_n1p1.bmp

Background Type	Optical Flow Component	Greyscale Window Size	Optical Flow Window Size	Layers	Min	Max	File
Wavelet	U	8	5	11	-4	4	/Farneback/Wavelet/8_5/MultiRes05/FB_U_8_5_11_n4p4.bmp
Wavelet	V	8	5	11	-4	4	/Farneback/Wavelet/8_5/MultiRes05/FB_V_8_5_11_n4p4.bmp
Wavelet	U	8	5	11	-1	1	/Farneback/Wavelet/8_5/MultiRes05/FB_U_8_5_11_n1p1.bmp
Wavelet	V	8	5	11	-1	1	/Farneback/Wavelet/8_5/MultiRes05/FB_V_8_5_11_n1p1.bmp
Wavelet	U	8	7	11	-4	4	/Farneback/Wavelet/8_7/MultiRes05/FB_U_8_7_11_n4p4.bmp
Wavelet	V	8	7	11	-4	4	/Farneback/Wavelet/8_7/MultiRes05/FB_V_8_7_11_n4p4.bmp
Wavelet	U	8	7	11	-1	1	/Farneback/Wavelet/8_7/MultiRes05/FB_U_8_7_11_n1p1.bmp
Wavelet	V	8	7	11	-1	1	/Farneback/Wavelet/8_7/MultiRes05/FB_V_8_7_11_n1p1.bmp
Wavelet	U	16	5	11	-4	4	/Farneback/Wavelet/16_5/MultiRes05/FB_U_16_5_11_n4p4.bmp
Wavelet	V	16	5	11	-4	4	/Farneback/Wavelet/16_5/MultiRes05/FB_V_16_5_11_n4p4.bmp
Wavelet	U	16	5	11	-1	1	/Farneback/Wavelet/16_5/MultiRes05/FB_U_16_5_11_n1p1.bmp
Wavelet	V	16	5	11	-1	1	/Farneback/Wavelet/16_5/MultiRes05/FB_V_16_5_11_n1p1.bmp
Wavelet	U	16	7	11	-4	4	/Farneback/Wavelet/16_7/MultiRes05/FB_U_16_7_11_n4p4.bmp
Wavelet	V	16	7	11	-4	4	/Farneback/Wavelet/16_7/MultiRes05/FB_V_16_7_11_n4p4.bmp
Wavelet	U	16	7	11	-1	1	/Farneback/Wavelet/16_7/MultiRes05/FB_U_16_7_11_n1p1.bmp
Wavelet	V	16	7	11	-1	1	/Farneback/Wavelet/16_7/MultiRes05/FB_V_16_7_11_n1p1.bmp
Wavelet	U	32	5	11	-4	4	/Farneback/Wavelet/32_5/MultiRes05/FB_U_32_5_11_n4p4.bmp
Wavelet	V	32	5	11	-4	4	/Farneback/Wavelet/32_5/MultiRes05/FB_V_32_5_11_n4p4.bmp
Wavelet	U	32	5	11	-1	1	/Farneback/Wavelet/32_5/MultiRes05/FB_U_32_5_11_n1p1.bmp
Wavelet	V	32	5	11	-1	1	/Farneback/Wavelet/32_5/MultiRes05/FB_V_32_5_11_n1p1.bmp
Wavelet	U	32	7	11	-4	4	/Farneback/Wavelet/32_7/MultiRes05/FB_U_32_7_11_n4p4.bmp
Wavelet	V	32	7	11	-4	4	/Farneback/Wavelet/32_7/MultiRes05/FB_V_32_7_11_n4p4.bmp
Wavelet	U	32	7	11	-1	1	/Farneback/Wavelet/32_7/MultiRes05/FB_U_32_7_11_n1p1.bmp

Background Type	Optical Flow Component	Greyscale Window Size	Optical Flow Window Size	Layers	Min	Max	File
Wavelet	V	32	7	11	-1	1	/Farneback/Wavelet/32_7/MultiRes05/FB_V_32_7_11_n1p1.bmp
Wavelet	U	4	5	35	-4	4	/Farneback/Wavelet/4_5/MultiRes08/FB_U_4_5_35_n4p4.bmp
Wavelet	V	4	5	35	-4	4	/Farneback/Wavelet/4_5/MultiRes08/FB_V_4_5_35_n4p4.bmp
Wavelet	U	4	5	35	-1	1	/Farneback/Wavelet/4_5/MultiRes08/FB_U_4_5_35_n1p1.bmp
Wavelet	V	4	5	35	-1	1	/Farneback/Wavelet/4_5/MultiRes08/FB_V_4_5_35_n1p1.bmp
Wavelet	U	4	7	35	-4	4	/Farneback/Wavelet/4_7/MultiRes08/FB_U_4_7_35_n4p4.bmp
Wavelet	V	4	7	35	-4	4	/Farneback/Wavelet/4_7/MultiRes08/FB_V_4_7_35_n4p4.bmp
Wavelet	U	4	7	35	-1	1	/Farneback/Wavelet/4_7/MultiRes08/FB_U_4_7_35_n1p1.bmp
Wavelet	V	4	7	35	-1	1	/Farneback/Wavelet/4_7/MultiRes08/FB_V_4_7_35_n1p1.bmp
Wavelet	U	8	5	35	-4	4	/Farneback/Wavelet/8_5/MultiRes08/FB_U_8_5_35_n4p4.bmp
Wavelet	V	8	5	35	-4	4	/Farneback/Wavelet/8_5/MultiRes08/FB_V_8_5_35_n4p4.bmp
Wavelet	U	8	5	35	-1	1	/Farneback/Wavelet/8_5/MultiRes08/FB_U_8_5_35_n1p1.bmp
Wavelet	V	8	5	35	-1	1	/Farneback/Wavelet/8_5/MultiRes08/FB_V_8_5_35_n1p1.bmp
Wavelet	U	8	7	35	-4	4	/Farneback/Wavelet/8_7/MultiRes08/FB_U_8_7_35_n4p4.bmp
Wavelet	V	8	7	35	-4	4	/Farneback/Wavelet/8_7/MultiRes08/FB_V_8_7_35_n4p4.bmp
Wavelet	U	8	7	35	-1	1	/Farneback/Wavelet/8_7/MultiRes08/FB_U_8_7_35_n1p1.bmp
Wavelet	V	8	7	35	-1	1	/Farneback/Wavelet/8_7/MultiRes08/FB_V_8_7_35_n1p1.bmp
Wavelet	U	16	5	35	-4	4	/Farneback/Wavelet/16_5/MultiRes08/FB_U_16_5_35_n4p4.bmp
Wavelet	V	16	5	35	-4	4	/Farneback/Wavelet/16_5/MultiRes08/FB_V_16_5_35_n4p4.bmp
Wavelet	U	16	5	35	-1	1	/Farneback/Wavelet/16_5/MultiRes08/FB_U_16_5_35_n1p1.bmp
Wavelet	V	16	5	35	-1	1	/Farneback/Wavelet/16_5/MultiRes08/FB_V_16_5_35_n1p1.bmp
Wavelet	U	16	7	35	-4	4	/Farneback/Wavelet/16_7/MultiRes08/FB_U_16_7_35_n4p4.bmp
Wavelet	V	16	7	35	-4	4	/Farneback/Wavelet/16_7/MultiRes08/FB_V_16_7_35_n4p4.bmp

Background Type	Optical Flow Component	Greyscale Window Size	Optical Flow Window Size	Layers	Min	Max	File
Wavelet	U	16	7	35	-1	1	/Farneback/Wavelet/16_7/MultiRes08/FB_U_16_7_35_n1p1.bmp
Wavelet	V	16	7	35	-1	1	/Farneback/Wavelet/16_7/MultiRes08/FB_V_16_7_35_n1p1.bmp
Wavelet	U	32	5	35	-4	4	/Farneback/Wavelet/32_5/MultiRes08/FB_U_32_5_35_n4p4.bmp
Wavelet	V	32	5	35	-4	4	/Farneback/Wavelet/32_5/MultiRes08/FB_V_32_5_35_n4p4.bmp
Wavelet	U	32	5	35	-1	1	/Farneback/Wavelet/32_5/MultiRes08/FB_U_32_5_35_n1p1.bmp
Wavelet	V	32	5	35	-1	1	/Farneback/Wavelet/32_5/MultiRes08/FB_V_32_5_35_n1p1.bmp
Wavelet	U	32	7	35	-4	4	/Farneback/Wavelet/32_7/MultiRes08/FB_U_32_7_35_n4p4.bmp
Wavelet	V	32	7	35	-4	4	/Farneback/Wavelet/32_7/MultiRes08/FB_V_32_7_35_n4p4.bmp
Wavelet	U	32	7	35	-1	1	/Farneback/Wavelet/32_7/MultiRes08/FB_U_32_7_35_n1p1.bmp
Wavelet	V	32	7	35	-1	1	/Farneback/Wavelet/32_7/MultiRes08/FB_V_32_7_35_n1p1.bmp
Speckle	U	4	5	1	-4	4	/Farneback/Speckle/4_5/SingleRes/FB_U_4_5_1_n4p4.bmp
Speckle	V	4	5	1	-4	4	/Farneback/Speckle/4_5/SingleRes/FB_V_4_5_1_n4p4.bmp
Speckle	U	4	5	1	-1	1	/Farneback/Speckle/4_5/SingleRes/FB_U_4_5_1_n1p1.bmp
Speckle	V	4	5	1	-1	1	/Farneback/Speckle/4_5/SingleRes/FB_V_4_5_1_n1p1.bmp
Speckle	U	4	7	1	-4	4	/Farneback/Speckle/4_7/SingleRes/FB_U_4_7_1_n4p4.bmp
Speckle	V	4	7	1	-4	4	/Farneback/Speckle/4_7/SingleRes/FB_V_4_7_1_n4p4.bmp
Speckle	U	4	7	1	-1	1	/Farneback/Speckle/4_7/SingleRes/FB_U_4_7_1_n1p1.bmp
Speckle	V	4	7	1	-1	1	/Farneback/Speckle/4_7/SingleRes/FB_V_4_7_1_n1p1.bmp
Speckle	U	8	5	1	-4	4	/Farneback/Speckle/8_5/SingleRes/FB_U_8_5_1_n4p4.bmp
Speckle	V	8	5	1	-4	4	/Farneback/Speckle/8_5/SingleRes/FB_V_8_5_1_n4p4.bmp
Speckle	U	8	5	1	-1	1	/Farneback/Speckle/8_5/SingleRes/FB_U_8_5_1_n1p1.bmp
Speckle	V	8	5	1	-1	1	/Farneback/Speckle/8_5/SingleRes/FB_V_8_5_1_n1p1.bmp
Speckle	U	8	7	1	-4	4	/Farneback/Speckle/8_7/SingleRes/FB_U_8_7_1_n4p4.bmp

Background Type	Optical Flow Component	Greyscale Window Size	Optical Flow Window Size	Layers	Min	Max	File
Speckle	V	8	7	1	-4	4	/Farneback/Speckle/8_7/SingleRes/FB_V_8_7_1_n4p4.bmp
Speckle	U	8	7	1	-1	1	/Farneback/Speckle/8_7/SingleRes/FB_U_8_7_1_n1p1.bmp
Speckle	V	8	7	1	-1	1	/Farneback/Speckle/8_7/SingleRes/FB_V_8_7_1_n1p1.bmp
Speckle	U	16	5	1	-4	4	/Farneback/Speckle/16_5/SingleRes/FB_U_16_5_1_n4p4.bmp
Speckle	V	16	5	1	-4	4	/Farneback/Speckle/16_5/SingleRes/FB_V_16_5_1_n4p4.bmp
Speckle	U	16	5	1	-1	1	/Farneback/Speckle/16_5/SingleRes/FB_U_16_5_1_n1p1.bmp
Speckle	V	16	5	1	-1	1	/Farneback/Speckle/16_5/SingleRes/FB_V_16_5_1_n1p1.bmp
Speckle	U	16	7	1	-4	4	/Farneback/Speckle/16_7/SingleRes/FB_U_16_7_1_n4p4.bmp
Speckle	V	16	7	1	-4	4	/Farneback/Speckle/16_7/SingleRes/FB_V_16_7_1_n4p4.bmp
Speckle	U	16	7	1	-1	1	/Farneback/Speckle/16_7/SingleRes/FB_U_16_7_1_n1p1.bmp
Speckle	V	16	7	1	-1	1	/Farneback/Speckle/16_7/SingleRes/FB_V_16_7_1_n1p1.bmp
Speckle	U	32	5	1	-4	4	/Farneback/Speckle/32_5/SingleRes/FB_U_32_5_1_n4p4.bmp
Speckle	V	32	5	1	-4	4	/Farneback/Speckle/32_5/SingleRes/FB_V_32_5_1_n4p4.bmp
Speckle	U	32	5	1	-1	1	/Farneback/Speckle/32_5/SingleRes/FB_U_32_5_1_n1p1.bmp
Speckle	V	32	5	1	-1	1	/Farneback/Speckle/32_5/SingleRes/FB_V_32_5_1_n1p1.bmp
Speckle	U	32	7	1	-4	4	/Farneback/Speckle/32_7/SingleRes/FB_U_32_7_1_n4p4.bmp
Speckle	V	32	7	1	-4	4	/Farneback/Speckle/32_7/SingleRes/FB_V_32_7_1_n4p4.bmp
Speckle	U	32	7	1	-1	1	/Farneback/Speckle/32_7/SingleRes/FB_U_32_7_1_n1p1.bmp
Speckle	V	32	7	1	-1	1	/Farneback/Speckle/32_7/SingleRes/FB_V_32_7_1_n1p1.bmp
Speckle	U	4	5	11	-4	4	/Farneback/Speckle/4_5/MultiRes05/FB_U_4_5_11_n4p4.bmp
Speckle	V	4	5	11	-4	4	/Farneback/Speckle/4_5/MultiRes05/FB_V_4_5_11_n4p4.bmp
Speckle	U	4	5	11	-1	1	/Farneback/Speckle/4_5/MultiRes05/FB_U_4_5_11_n1p1.bmp
Speckle	V	4	5	11	-1	1	/Farneback/Speckle/4_5/MultiRes05/FB_V_4_5_11_n1p1.bmp

Background Type	Optical Flow Component	Greyscale Window Size	Optical Flow Window Size	Layers	Min	Max	File
Speckle	U	4	7	11	-4	4	/Farneback/Speckle/4_7/MultiRes05/FB_U_4_7_11_n4p4.bmp
Speckle	V	4	7	11	-4	4	/Farneback/Speckle/4_7/MultiRes05/FB_V_4_7_11_n4p4.bmp
Speckle	U	4	7	11	-1	1	/Farneback/Speckle/4_7/MultiRes05/FB_U_4_7_11_n1p1.bmp
Speckle	V	4	7	11	-1	1	/Farneback/Speckle/4_7/MultiRes05/FB_V_4_7_11_n1p1.bmp
Speckle	U	8	5	11	-4	4	/Farneback/Speckle/8_5/MultiRes05/FB_U_8_5_11_n4p4.bmp
Speckle	V	8	5	11	-4	4	/Farneback/Speckle/8_5/MultiRes05/FB_V_8_5_11_n4p4.bmp
Speckle	U	8	5	11	-1	1	/Farneback/Speckle/8_5/MultiRes05/FB_U_8_5_11_n1p1.bmp
Speckle	V	8	5	11	-1	1	/Farneback/Speckle/8_5/MultiRes05/FB_V_8_5_11_n1p1.bmp
Speckle	U	8	7	11	-4	4	/Farneback/Speckle/8_7/MultiRes05/FB_U_8_7_11_n4p4.bmp
Speckle	V	8	7	11	-4	4	/Farneback/Speckle/8_7/MultiRes05/FB_V_8_7_11_n4p4.bmp
Speckle	U	8	7	11	-1	1	/Farneback/Speckle/8_7/MultiRes05/FB_U_8_7_11_n1p1.bmp
Speckle	V	8	7	11	-1	1	/Farneback/Speckle/8_7/MultiRes05/FB_V_8_7_11_n1p1.bmp
Speckle	U	16	5	11	-4	4	/Farneback/Speckle/16_5/MultiRes05/FB_U_16_5_11_n4p4.bmp
Speckle	V	16	5	11	-4	4	/Farneback/Speckle/16_5/MultiRes05/FB_V_16_5_11_n4p4.bmp
Speckle	U	16	5	11	-1	1	/Farneback/Speckle/16_5/MultiRes05/FB_U_16_5_11_n1p1.bmp
Speckle	V	16	5	11	-1	1	/Farneback/Speckle/16_5/MultiRes05/FB_V_16_5_11_n1p1.bmp
Speckle	U	16	7	11	-4	4	/Farneback/Speckle/16_7/MultiRes05/FB_U_16_7_11_n4p4.bmp
Speckle	V	16	7	11	-4	4	/Farneback/Speckle/16_7/MultiRes05/FB_V_16_7_11_n4p4.bmp
Speckle	U	16	7	11	-1	1	/Farneback/Speckle/16_7/MultiRes05/FB_U_16_7_11_n1p1.bmp
Speckle	V	16	7	11	-1	1	/Farneback/Speckle/16_7/MultiRes05/FB_V_16_7_11_n1p1.bmp
Speckle	U	32	5	11	-4	4	/Farneback/Speckle/32_5/MultiRes05/FB_U_32_5_11_n4p4.bmp
Speckle	V	32	5	11	-4	4	/Farneback/Speckle/32_5/MultiRes05/FB_V_32_5_11_n4p4.bmp
Speckle	U	32	5	11	-1	1	/Farneback/Speckle/32_5/MultiRes05/FB_U_32_5_11_n1p1.bmp

Background Type	Optical Flow Component	Greyscale Window Size	Optical Flow Window Size	Layers	Min	Max	File
Speckle	V	32	5	11	-1	1	/Farneback/Speckle/32_5/MultiRes05/FB_V_32_5_11_n1p1.bmp
Speckle	U	32	7	11	-4	4	/Farneback/Speckle/32_7/MultiRes05/FB_U_32_7_11_n4p4.bmp
Speckle	V	32	7	11	-4	4	/Farneback/Speckle/32_7/MultiRes05/FB_V_32_7_11_n4p4.bmp
Speckle	U	32	7	11	-1	1	/Farneback/Speckle/32_7/MultiRes05/FB_U_32_7_11_n1p1.bmp
Speckle	V	32	7	11	-1	1	/Farneback/Speckle/32_7/MultiRes05/FB_V_32_7_11_n1p1.bmp
Speckle	U	4	5	35	-4	4	/Farneback/Speckle/4_5/MultiRes08/FB_U_4_5_35_n4p4.bmp
Speckle	V	4	5	35	-4	4	/Farneback/Speckle/4_5/MultiRes08/FB_V_4_5_35_n4p4.bmp
Speckle	U	4	5	35	-1	1	/Farneback/Speckle/4_5/MultiRes08/FB_U_4_5_35_n1p1.bmp
Speckle	V	4	5	35	-1	1	/Farneback/Speckle/4_5/MultiRes08/FB_V_4_5_35_n1p1.bmp
Speckle	U	4	7	35	-4	4	/Farneback/Speckle/4_7/MultiRes08/FB_U_4_7_35_n4p4.bmp
Speckle	V	4	7	35	-4	4	/Farneback/Speckle/4_7/MultiRes08/FB_V_4_7_35_n4p4.bmp
Speckle	U	4	7	35	-1	1	/Farneback/Speckle/4_7/MultiRes08/FB_U_4_7_35_n1p1.bmp
Speckle	V	4	7	35	-1	1	/Farneback/Speckle/4_7/MultiRes08/FB_V_4_7_35_n1p1.bmp
Speckle	U	8	5	35	-4	4	/Farneback/Speckle/8_5/MultiRes08/FB_U_8_5_35_n4p4.bmp
Speckle	V	8	5	35	-4	4	/Farneback/Speckle/8_5/MultiRes08/FB_V_8_5_35_n4p4.bmp
Speckle	U	8	5	35	-1	1	/Farneback/Speckle/8_5/MultiRes08/FB_U_8_5_35_n1p1.bmp
Speckle	V	8	5	35	-1	1	/Farneback/Speckle/8_5/MultiRes08/FB_V_8_5_35_n1p1.bmp
Speckle	U	8	7	35	-4	4	/Farneback/Speckle/8_7/MultiRes08/FB_U_8_7_35_n4p4.bmp
Speckle	V	8	7	35	-4	4	/Farneback/Speckle/8_7/MultiRes08/FB_V_8_7_35_n4p4.bmp
Speckle	U	8	7	35	-1	1	/Farneback/Speckle/8_7/MultiRes08/FB_U_8_7_35_n1p1.bmp
Speckle	V	8	7	35	-1	1	/Farneback/Speckle/8_7/MultiRes08/FB_V_8_7_35_n1p1.bmp
Speckle	U	16	5	35	-4	4	/Farneback/Speckle/16_5/MultiRes08/FB_U_16_5_35_n4p4.bmp
Speckle	V	16	5	35	-4	4	/Farneback/Speckle/16_5/MultiRes08/FB_V_16_5_35_n4p4.bmp

Background Type	Optical Flow Component	Greyscale Window Size	Optical Flow Window Size	Layers	Min	Max	File
Speckle	U	16	5	35	-1	1	/Farneback/Speckle/16_5/MultiRes08/FB_U_16_5_35_n1p1.bmp
Speckle	V	16	5	35	-1	1	/Farneback/Speckle/16_5/MultiRes08/FB_V_16_5_35_n1p1.bmp
Speckle	U	16	7	35	-4	4	/Farneback/Speckle/16_7/MultiRes08/FB_U_16_7_35_n4p4.bmp
Speckle	V	16	7	35	-4	4	/Farneback/Speckle/16_7/MultiRes08/FB_V_16_7_35_n4p4.bmp
Speckle	U	16	7	35	-1	1	/Farneback/Speckle/16_7/MultiRes08/FB_U_16_7_35_n1p1.bmp
Speckle	V	16	7	35	-1	1	/Farneback/Speckle/16_7/MultiRes08/FB_V_16_7_35_n1p1.bmp
Speckle	U	32	5	35	-4	4	/Farneback/Speckle/32_5/MultiRes08/FB_U_32_5_35_n4p4.bmp
Speckle	V	32	5	35	-4	4	/Farneback/Speckle/32_5/MultiRes08/FB_V_32_5_35_n4p4.bmp
Speckle	U	32	5	35	-1	1	/Farneback/Speckle/32_5/MultiRes08/FB_U_32_5_35_n1p1.bmp
Speckle	V	32	5	35	-1	1	/Farneback/Speckle/32_5/MultiRes08/FB_V_32_5_35_n1p1.bmp
Speckle	U	32	7	35	-4	4	/Farneback/Speckle/32_7/MultiRes08/FB_U_32_7_35_n4p4.bmp
Speckle	V	32	7	35	-4	4	/Farneback/Speckle/32_7/MultiRes08/FB_V_32_7_35_n4p4.bmp
Speckle	U	32	7	35	-1	1	/Farneback/Speckle/32_7/MultiRes08/FB_U_32_7_35_n1p1.bmp
Speckle	V	32	7	35	-1	1	/Farneback/Speckle/32_7/MultiRes08/FB_V_32_7_35_n1p1.bmp

C.6 TV-L1 Results

Each iteration was processed using the following parameters:

- ϵ=0.01
- γ=0
- 999 Iterations
- 99 warps
- τ=0.12
- θ=0.01

Table 24: TV-L1 algorithm results

Background Type	Optical Flow Component	Greyscale Consistency Weight	Layers	Min	Max	File
Wavelet	U	0.08	1	-4	4	/TVL1/Wavelet/008/SingleRes/TV_U_008_1_n4p4.bmp
Wavelet	V	0.08	1	-4	4	/TVL1/Wavelet/008/SingleRes/TV_V_008_1_n4p4.bmp
Wavelet	U	0.08	1	-1	1	/TVL1/Wavelet/008/SingleRes/TV_U_008_1_n1p1.bmp
Wavelet	V	0.08	1	-1	1	/TVL1/Wavelet/008/SingleRes/TV_V_008_1_n1p1.bmp
Wavelet	U	0.16	1	-4	4	/TVL1/Wavelet/016/SingleRes/TV_U_016_1_n4p4.bmp
Wavelet	V	0.16	1	-4	4	/TVL1/Wavelet/016/SingleRes/TV_V_016_1_n4p4.bmp
Wavelet	U	0.16	1	-1	1	/TVL1/Wavelet/016/SingleRes/TV_U_016_1_n1p1.bmp
Wavelet	V	0.16	1	-1	1	/TVL1/Wavelet/016/SingleRes/TV_V_016_1_n1p1.bmp

Background Type	Optical Flow Component	Greyscale Consistency Weight	Layers	Min	Max	File
Wavelet	U	0.32	1	-4	4	/TVL1/Wavelet/032/SingleRes/TV_U_032_1_n4p4.bmp
Wavelet	V	0.32	1	-4	4	/TVL1/Wavelet/032/SingleRes/TV_V_032_1_n4p4.bmp
Wavelet	U	0.32	1	-1	1	/TVL1/Wavelet/032/SingleRes/TV_U_032_1_n1p1.bmp
Wavelet	V	0.32	1	-1	1	/TVL1/Wavelet/032/SingleRes/TV_V_032_1_n1p1.bmp
Wavelet	U	0.08	11	-4	4	/TVL1/Wavelet/008/MultiRes05/TV_U_008_11_n4p4.bmp
Wavelet	V	0.08	11	-4	4	/TVL1/Wavelet/008/MultiRes05/TV_V_008_11_n4p4.bmp
Wavelet	U	0.08	11	-1	1	/TVL1/Wavelet/008/MultiRes05/TV_U_008_11_n1p1.bmp
Wavelet	V	0.08	11	-1	1	/TVL1/Wavelet/008/MultiRes05/TV_V_008_11_n1p1.bmp
Wavelet	U	0.16	11	-4	4	/TVL1/Wavelet/016/MultiRes05/TV_U_016_11_n4p4.bmp
Wavelet	V	0.16	11	-4	4	/TVL1/Wavelet/016/MultiRes05/TV_V_016_11_n4p4.bmp
Wavelet	U	0.16	11	-1	1	/TVL1/Wavelet/016/MultiRes05/TV_U_016_11_n1p1.bmp
Wavelet	V	0.16	11	-1	1	/TVL1/Wavelet/016/MultiRes05/TV_V_016_11_n1p1.bmp
Wavelet	U	0.32	11	-4	4	/TVL1/Wavelet/032/MultiRes05/TV_U_032_11_n4p4.bmp
Wavelet	V	0.32	11	-4	4	/TVL1/Wavelet/032/MultiRes05/TV_V_032_11_n4p4.bmp
Wavelet	U	0.32	11	-1	1	/TVL1/Wavelet/032/MultiRes05/TV_U_032_11_n1p1.bmp
Wavelet	V	0.32	11	-1	1	/TVL1/Wavelet/032/MultiRes05/TV_V_032_11_n1p1.bmp
Wavelet	U	0.08	35	-4	4	/TVL1/Wavelet/008/MultiRes08/TV_U_008_35_n4p4.bmp
Wavelet	V	0.08	35	-4	4	/TVL1/Wavelet/008/MultiRes08/TV_V_008_35_n4p4.bmp
Wavelet	U	0.08	35	-1	1	/TVL1/Wavelet/008/MultiRes08/TV_U_008_35_n1p1.bmp
Wavelet	V	0.08	35	-1	1	/TVL1/Wavelet/008/MultiRes08/TV_V_008_35_n1p1.bmp
Wavelet	U	0.16	35	-4	4	/TVL1/Wavelet/016/MultiRes08/TV_U_016_35_n4p4.bmp
Wavelet	V	0.16	35	-4	4	/TVL1/Wavelet/016/MultiRes08/TV_V_016_35_n4p4.bmp

Background Type	Optical Flow Component	Greyscale Consistency Weight	Layers	Min	Max	File
Wavelet	U	0.16	35	-1	1	/TVL1/Wavelet/016/MultiRes08/TV_U_016_35_n1p1.bmp
Wavelet	V	0.16	35	-1	1	/TVL1/Wavelet/016/MultiRes08/TV_V_016_35_n1p1.bmp
Wavelet	U	0.32	35	-4	4	/TVL1/Wavelet/032/MultiRes08/TV_U_032_35_n4p4.bmp
Wavelet	V	0.32	35	-4	4	/TVL1/Wavelet/032/MultiRes08/TV_V_032_35_n4p4.bmp
Wavelet	U	0.32	35	-1	1	/TVL1/Wavelet/032/MultiRes08/TV_U_032_35_n1p1.bmp
Wavelet	V	0.32	35	-1	1	/TVL1/Wavelet/032/MultiRes08/TV_V_032_35_n1p1.bmp
Speckle	U	0.08	1	-4	4	/TVL1/Speckle/008/SingleRes/TV_U_008_1_n4p4.bmp
Speckle	V	0.08	1	-4	4	/TVL1/Speckle/008/SingleRes/TV_V_008_1_n4p4.bmp
Speckle	U	0.08	1	-1	1	/TVL1/Speckle/008/SingleRes/TV_U_008_1_n1p1.bmp
Speckle	V	0.08	1	-1	1	/TVL1/Speckle/008/SingleRes/TV_V_008_1_n1p1.bmp
Speckle	U	0.16	1	-4	4	/TVL1/Speckle/016/SingleRes/TV_U_016_1_n4p4.bmp
Speckle	V	0.16	1	-4	4	/TVL1/Speckle/016/SingleRes/TV_V_016_1_n4p4.bmp
Speckle	U	0.16	1	-1	1	/TVL1/Speckle/016/SingleRes/TV_U_016_1_n1p1.bmp
Speckle	V	0.16	1	-1	1	/TVL1/Speckle/016/SingleRes/TV_V_016_1_n1p1.bmp
Speckle	U	0.32	1	-4	4	/TVL1/Speckle/032/SingleRes/TV_U_032_1_n4p4.bmp
Speckle	V	0.32	1	-4	4	/TVL1/Speckle/032/SingleRes/TV_V_032_1_n4p4.bmp
Speckle	U	0.32	1	-1	1	/TVL1/Speckle/032/SingleRes/TV_U_032_1_n1p1.bmp
Speckle	V	0.32	1	-1	1	/TVL1/Speckle/032/SingleRes/TV_V_032_1_n1p1.bmp
Speckle	U	0.08	11	-4	4	/TVL1/Speckle/008/MultiRes05/TV_U_008_11_n4p4.bmp
Speckle	V	0.08	11	-4	4	/TVL1/Speckle/008/MultiRes05/TV_V_008_11_n4p4.bmp
Speckle	U	0.08	11	-1	1	/TVL1/Speckle/008/MultiRes05/TV_U_008_11_n1p1.bmp
Speckle	V	0.08	11	-1	1	/TVL1/Speckle/008/MultiRes05/TV_V_008_11_n1p1.bmp

Background Type	Optical Flow Component	Greyscale Consistency Weight	Layers	Min	Max	File
Speckle	U	0.16	11	-4	4	/TVL1/Speckle/016/MultiRes05/TV_U_016_11_n4p4.bmp
Speckle	V	0.16	11	-4	4	/TVL1/Speckle/016/MultiRes05/TV_V_016_11_n4p4.bmp
Speckle	U	0.16	11	-1	1	/TVL1/Speckle/016/MultiRes05/TV_U_016_11_n1p1.bmp
Speckle	V	0.16	11	-1	1	/TVL1/Speckle/016/MultiRes05/TV_V_016_11_n1p1.bmp
Speckle	U	0.32	11	-4	4	/TVL1/Speckle/032/MultiRes05/TV_U_032_11_n4p4.bmp
Speckle	V	0.32	11	-4	4	/TVL1/Speckle/032/MultiRes05/TV_V_032_11_n4p4.bmp
Speckle	U	0.32	11	-1	1	/TVL1/Speckle/032/MultiRes05/TV_U_032_11_n1p1.bmp
Speckle	V	0.32	11	-1	1	/TVL1/Speckle/032/MultiRes05/TV_V_032_11_n1p1.bmp
Speckle	U	0.08	35	-4	4	/TVL1/Speckle/008/MultiRes08/TV_U_008_35_n4p4.bmp
Speckle	V	0.08	35	-4	4	/TVL1/Speckle/008/MultiRes08/TV_V_008_35_n4p4.bmp
Speckle	U	0.08	35	-1	1	/TVL1/Speckle/008/MultiRes08/TV_U_008_35_n1p1.bmp
Speckle	V	0.08	35	-1	1	/TVL1/Speckle/008/MultiRes08/TV_V_008_35_n1p1.bmp
Speckle	U	0.16	35	-4	4	/TVL1/Speckle/016/MultiRes08/TV_U_016_35_n4p4.bmp
Speckle	V	0.16	35	-4	4	/TVL1/Speckle/016/MultiRes08/TV_V_016_35_n4p4.bmp
Speckle	U	0.16	35	-1	1	/TVL1/Speckle/016/MultiRes08/TV_U_016_35_n1p1.bmp
Speckle	V	0.16	35	-1	1	/TVL1/Speckle/016/MultiRes08/TV_V_016_35_n1p1.bmp
Speckle	U	0.32	35	-4	4	/TVL1/Speckle/032/MultiRes08/TV_U_032_35_n4p4.bmp
Speckle	V	0.32	35	-4	4	/TVL1/Speckle/032/MultiRes08/TV_V_032_35_n4p4.bmp
Speckle	U	0.32	35	-1	1	/TVL1/Speckle/032/MultiRes08/TV_U_032_35_n1p1.bmp
Speckle	V	0.32	35	-1	1	/TVL1/Speckle/032/MultiRes08/TV_V_032_35_n1p1.bmp