

A METRIC COMPILATION ANALYSIS OF TERRESTRIAL
ATMOSPHERIC TURBULENCE SUPPRESSION ALGORITHMS FOR USE
IN LONG RANGE DIGITAL VIDEO SURVEILLANCE

by

BRYN WALTERS

A dissertation regarding algorithmic atmospheric turbulence suppression for use in long range video surveillance, and the comparison of those algorithms as part of the requirements for the degree

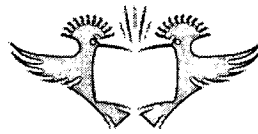
MASTER OF ENGINEERING

in

ELECTRICAL AND ELECTRONIC ENGINEERING SCIENCES

at the

UNIVERSITY OF JOHANNESBURG



SUPERVISOR: DR. WA CLARKE

July 2008

ACKNOWLEDGEMENTS

This research project would not have been possible without the support of many people. The author wishes to express his gratitude to his supervisor, Dr. Willem A Clarke who was abundantly helpful and offered invaluable assistance, support and guidance. Thanks also go to Prof. Jan-Harm C Pretorius, Prof. André Booysen, and Dr. Andre De Jaeger for their support. The author also wishes to thank Mrs. Lucia Pelsier for her efficiency, organisation, and enthusiasm.

Deepest gratitude is also due the CSIR (Council for Scientific and Industrial Research) for their financial aid during 2007 and the CSIR DPSS (Defence, Peace, Safety, and Security) group for their guidance, interest and support. The financial assistance of the National Research Foundation (NRF) towards this research is also acknowledged. Opinions expressed and conclusions arrived at, are those of the author and are not necessarily to be attributed to the NRF.

Special thanks also to all his graduate friends, especially Philip Robinson for sharing of literature and invaluable assistance with the experiments.

The author would also like to convey thanks to the Faculty of Engineering and the Built Environment for providing the equipment necessary for the imaging system.

Lastly but not least, the author wishes to express gratitude to his loving family who have always provided support, enthusiasm and encouragement.

SUMMARY

Atmospheric turbulence (also referred to as optical or heat Scintillation, or heat shimmer) is a particular problem encountered in video surveillance, especially over distances where the target object focused on is over 1km in the distance. Images obtained from video surveillance are commonly required to be of a high quality for object identification and classification. Atmospheric turbulence causes degradation in the image quality through the blurring and a warping of the image, making object identification difficult. Algorithms have and still are being developed to suppress the image turbulence in digital video footage and enhance detail. There is a lack of reliable comparisons among algorithms to provide research direction, methods for identification of the best algorithms for particular applications, identification of useful image processing techniques and a full understanding of the problem.

This need and lack of comparisons among the algorithms and atmospheric turbulence degraded videos is identified through the problem identification chapter.

A literature study is undertaken in which the source of atmospheric turbulence and models are identified, image processing techniques discussed, filtering of electromagnetic waves reviewed, a review of some equipment, and a discussion of metrics. This is followed by the presentation of a number of atmospheric turbulence suppression algorithms developed by other authors.

After a discussion of the algorithm implementations, the experimental design is described for algorithm image quality and performance investigation as well as the effect of optical filters. Experimental results are presented and discussed which provide repeatable results pertaining to the algorithms' image quality and processing requirements. The results allowed identification of the algorithms' strengths and weaknesses, how they compare, and their suitability for real and post processing environments. Efficient performing software components were also able to be identified, particularly Illuminance-Reflectance adjustment. The experiments and results provide a solution to this atmospheric turbulence comparison problem.

Table of Contents

ACKNOWLEDGEMENTS.....	I
SUMMARY.....	II
1 CHAPTER 1: PROBLEM IDENTIFICATION.....	1-1
1.1 INTRODUCTION AND BACKGROUND.....	1-1
1.2 PURPOSE OF THE PROJECT	1-2
1.3 SCOPE OF THE PROJECT	1-2
1.4 RESEARCH METHODOLOGY	1-3
1.5 DOCUMENT OVERVIEW	1-5
2 CHAPTER 2: LITERATURE STUDY	2-1
2.1 INTRODUCTION AND OVERVIEW	2-1
2.2 PHYSICS SURROUNDING ATMOSPHERIC TURBULENCE	2-1
2.3 ATMOSPHERIC TURBULENCE MODELLING	2-4
2.4 DIGITAL IMAGE CONSTRUCTION.....	2-8
2.5 BASIC GREY-LEVEL IMAGE PROCESSING USING HISTOGRAM	2-9
2.6 INTERPOLATION	2-14
2.6.1 <i>Nearest-Neighbour Interpolation</i>	2-14
2.6.2 <i>Bilinear Interpolation</i>	2-15
2.7 FILTERS.....	2-15
2.7.1 <i>Windowing</i>	2-16
2.7.2 <i>Moving average filter</i>	2-17
2.7.3 <i>Low-Pass filtering</i>	2-18
2.7.4 <i>High-Pass Filtering</i>	2-18
2.7.5 <i>Image restoration filtering methods</i>	2-19
2.8 DISCRETE FOURIER TRANSFORM VS. FAST FOURIER TRANSFORM	2-19
2.9 GAUSSIAN AND LAPLACIAN PYRAMIDS	2-20
2.10 FILTERING ELECTROMAGNETIC WAVES.....	2-23
2.10.1 <i>Electromagnetic waves</i>	2-23
2.10.2 <i>Polarization of electromagnetic waves</i>	2-25
2.10.3 <i>The electromagnetic spectrum</i>	2-27
2.10.4 <i>Refraction of electromagnetic waves</i>	2-28
2.10.5 <i>Advantages and disadvantages of colour filtering electromagnetic waves</i>	2-29
2.10.6 <i>Wratten numbers</i>	2-30
2.10.7 <i>Optical filters</i>	2-30
2.11 ALTERNATIVE APPROACHES.....	2-32



2.12 EQUIPMENT 2-34

2.12.1 Telescopes and lenses..... 2-36

2.12.1.1 Telescope Accessories 2-39

2.12.2 Cameras..... 2-40

2.13 COMPARISON AND TESTING METHODS 2-43

2.14 CONCLUSION..... 2-47

3 CHAPTER 3: ALGORITHM OVERVIEW..... 3-1

3.1 INTRODUCTION AND OVERVIEW 3-1

3.2 DFT ANALYSIS METHOD:..... 3-1

3.3 NEURAL NETWORK METHOD: 3-4

3.4 TIME-SEQUENCE REGISTRATION METHOD: 3-5

3.5 ADAPTIVE CONTROL GRID METHOD: 3-6

3.5.1 Parameter estimation procedure:..... 3-8

3.5.2 Segmentation: 3-9

3.5.3 Compensating for turbulence: 3-9

3.6 IMAGE REGISTRATION AND FUSION METHOD:..... 3-10

3.7 SPECKLE INTERFEROMETRY METHOD (TRIPLE CORRELATION THEORY) 3-12

3.7.1 Recursive Fourier phase estimation from bi-spectrum..... 3-15

3.7.2 Two dimensional signal Fourier phase reconstruction: Fourier series approach..... 3-18

3.8 INDEPENDENT COMPONENT ANALYSIS (ICA) METHOD..... 3-20

3.9 HOMOMORPHIC FILTERING AND POWER SPECTRUM BASED METHOD 3-21

3.10 CONCLUSION..... 3-24

4 CHAPTER 4: ALGORITHMS DETAIL DESIGN..... 4-1

4.1 INTRODUCTION AND OVERVIEW 4-1

4.2 SPECKLE MASKING ALGORITHM..... 4-1

4.2.1 Fourier Modulus Calculation..... 4-3

4.2.2 Fourier Phase Calculation..... 4-3

4.2.3 Additional processing steps 4-6

4.3 IMAGE REGISTRATION AND LAPLACIAN PYRAMID IMAGE FUSION..... 4-7

4.4 HOMOMORPHIC FILTERING AND POWER SPECTRUM RESTORATION 4-8

4.5 WIENER FILTERING APPROACH..... 4-11

4.6 OWN ALGORITHM DEVELOPMENTS 4-12

4.6.1 Wiener Filtering using Laplacian Operator..... 4-13

4.6.2 Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression 4-13

4.6.3 Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener Filter.. 4-16

4.7 CONCLUSION..... 4-17

5 CHAPTER 5: EXPERIMENTAL DESIGN..... 5-1

5.1 INTRODUCTION AND OVERVIEW 5-1



5.2 IMAGING SYSTEM..... 5-1

5.3 IMAGING SYSTEM EQUIPMENT SETUP 5-3

5.4 OVERVIEW OF EXPERIMENTS 5-3

5.5 EXPERIMENT 1: ALGORITHM PERFORMANCE EXPERIMENT..... 5-5

5.5.1 Aim..... 5-5

5.5.2 Equipment..... 5-5

5.5.3 Method..... 5-5

5.5.4 Experiment Relevance 5-6

5.5.5 Expected Outcome..... 5-6

5.5.6 Uncertainty Control..... 5-6

5.6 EXPERIMENT 2: ALGORITHM SHARPNESS EXPERIMENT 5-7

5.6.1 Aim..... 5-7

5.6.2 Background - Laplacian Operator 5-7

5.6.3 Equipment..... 5-8

5.6.4 Method..... 5-8

5.6.5 Experiment Relevance 5-8

5.6.6 Expected Outcome..... 5-9

5.6.7 Uncertainty Control..... 5-9

5.7 EXPERIMENT 3: ALGORITHM ABERRATION EXPERIMENT..... 5-10

5.7.1 Aim..... 5-10

5.7.2 Equipment..... 5-10

5.7.3 Method..... 5-10

5.7.4 Experiment Relevance 5-11

5.7.5 Expected Outcome..... 5-12

5.7.6 Uncertainty Control..... 5-12

5.8 EXPERIMENT 4: ALGORITHM MTF EXPERIMENT 5-13

5.8.1 Aim..... 5-13

5.8.2 Background..... 5-13

5.8.3 Equipment..... 5-13

5.8.4 Method..... 5-14

5.8.5 Experiment Relevance 5-15

5.8.6 Expected Outcome..... 5-16

5.8.7 Uncertainty Control..... 5-16

5.9 EXPERIMENT 5: ALGORITHM MSE AND PSNR EXPERIMENT 5-17

5.9.1 Aim..... 5-17

5.9.2 Background..... 5-17

5.9.3 Equipment..... 5-18

5.9.4 Method..... 5-18

5.9.5 Experiment Relevance 5-19

5.9.6 Expected Outcome..... 5-19



5.9.7 *Uncertainty Control*..... 5-20

5.10 EXPERIMENT 6: WIENER FILTERING LAMBDA SELECTION EXPERIMENT 5-20

5.10.1 *Aim*..... 5-20

5.10.2 *Equipment*..... 5-20

5.10.3 *Method*..... 5-20

5.10.4 *Experiment Relevance* 5-21

5.10.5 *Expected Outcome* 5-21

5.10.6 *Uncertainty Control*..... 5-22

5.11 EXPERIMENT 7: FILTERING RECEIVED LIGHT EXPERIMENT 5-22

5.11.1 *Aim*..... 5-22

5.11.2 *Equipment*..... 5-22

5.11.3 *Method*..... 5-23

5.11.4 *Experiment Relevance* 5-23

5.11.5 *Expected Outcome* 5-24

5.11.6 *Uncertainty Control*..... 5-24

5.12 CONCLUSION..... 5-24

6 CHAPTER 6: RESULTS..... 6-1

6.1 IMPLEMENTATION ISSUES 6-1

6.1.1 *Speckle Masking Issues* 6-1

6.1.2 *Illuminance Reflectance Issues*..... 6-2

6.1.3 *Full Histogram Stretch*..... 6-2

6.1.4 *Kurtosis Minimization* 6-3

6.1.5 *General Programming Precautions* 6-3

6.2 EXPERIMENTAL ISSUES 6-4

6.2.1 *Providing for Comparability between Experiments* 6-4

6.2.2 *Atmospheric Haze Degrading Seeing Conditions*..... 6-4

6.2.3 *High Level of Atmospheric Turbulence*..... 6-5

6.2.4 *Effects of Wind and Finding a Suitable Site*..... 6-6

6.2.5 *Creating Reference Image for MSE*..... 6-6

6.2.6 *Maintaining Short Exposure Times* 6-7

6.2.7 *Separate Video Footage Captured for Frequency Targets*..... 6-7

6.3 SYSTEM PLATFORM 6-7

6.3.1 *Hardware Specifications* 6-8

6.3.2 *Software Specifications*..... 6-8

6.4 ATMOSPHERIC CONDITIONS RECORD..... 6-9

6.5 EXPERIMENT 1: ALGORITHM PERFORMANCE EXPERIMENT RESULTS..... 6-9

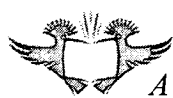
6.5.1 *Concluding remarks* 6-12

6.6 EXPERIMENT 2: ALGORITHM SHARPNESS EXPERIMENT RESULTS 6-13

6.6.1 *Concluding Remarks*..... 6-16

6.7 EXPERIMENT 3: ALGORITHM ABERRATION EXPERIMENT RESULTS 6-16

6.7.1	<i>Concluding Remarks</i>	6-19
6.8	EXPERIMENT 4: ALGORITHM MTF EXPERIMENT RESULTS	6-20
6.8.1	<i>Concluding Remarks</i>	6-24
6.9	EXPERIMENT 5: ALGORITHM MSE AND PSNR EXPERIMENT RESULTS	6-25
6.9.1	<i>Concluding Remarks</i>	6-27
6.10	EXPERIMENT 6: WIENER FILTERING LAMBDA SELECTION EXPERIMENT RESULTS	6-27
6.10.1	<i>Concluding Remarks</i>	6-29
6.11	EXPERIMENT 7: FILTERING RECEIVED LIGHT EXPERIMENT RESULTS	6-30
6.11.1	<i>Concluding Remarks</i>	6-33
6.12	CONCLUSION.....	6-33
7	CHAPTER 7: EXAMPLE IMAGES AND RESULTS DISCUSSION	7-1
7.1	INTRODUCTION	7-1
7.2	EXAMPLE IMAGES	7-1
7.3	SUMMARY AND LINKING OF RESULTS.....	7-5
7.3.1	<i>Image Registration and Laplacian Pyramid Image Fusion</i>	7-5
7.3.2	<i>Homomorphic Filtering and Power Spectrum based method</i>	7-6
7.3.3	<i>Wiener Filtering using Laplacian Operator</i>	7-6
7.3.4	<i>Wiener Filtering using Selected Lambda for Multiple Image Frames</i>	7-6
7.3.5	<i>Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression</i>	7-7
7.3.6	<i>Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener Filtering</i>	7-7
7.3.7	<i>Speckle Masking</i>	7-8
7.3.8	<i>MSE and PSNR</i>	7-8
7.3.9	<i>Imaging System and Optical Filters</i>	7-9
7.3.10	<i>Real time and post processing applications</i>	7-9
7.4	CONCLUSION.....	7-10
8	CHAPTER 8: CONCLUSION	8-1
8.1	OVERVIEW OF WORK	8-1
8.2	SUCCESSFUL PROBLEM SOLUTION	8-3
8.3	RELEVANCE OF EXPERIMENTS AND RESULTS.....	8-4
8.4	ISSUES ENCOUNTERED	8-5
8.5	FUTURE WORK.....	8-6
8.6	PUBLICATIONS AND PRESENTATIONS	8-7
9	REFERENCES	9-1
10	ADDENDUM A	10-1
	SPECKLE MASKING ALGORITHM CODE	10-1
10.1	'MAKEHANNING' FUNCTION	10-1
10.2	'IMAGEREGISTRATION' FUNCTION	10-1
10.3	'BSPECALC' FUNCTION	10-2



10.4	'PHASESCALC' FUNCTION	10-5
10.5	MAIN FILE (RUN)	10-7
11	ADDENDUM B	11-1
IMAGE REGISTRATION AND LAPLACIAN PYRAMID IMAGE FUSION ALGORITHM CODE. 11-1		
11.1	'IMAGEFUSION' FUNCTION	11-1
11.2	'GREDUCE' FUNCTION	11-3
11.3	'GEXPAND' FUNCTION	11-3
11.4	'IMGENERGY' FUNCTION	11-4
11.5	'IMGMATCH' FUNCTION	11-5
11.6	'IMGWEIGHTS' FUNCTION	11-5
11.7	MAIN FILE (RUN)	11-6
12	ADDENDUM C	12-1
HOMOMORPHIC FILTERING AND POWERS SPECTRUM RESTORATION BASED ALGORITHM CODE 12-1		
12.1	'RATIOAVG' FUNCTION	12-1
12.2	'CHOOSEHIGHFREQIMG' FUNCTION	12-1
12.3	'HFREQNUM' FUNCTION	12-2
12.4	'PADIMAGE' FUNCTION	12-3
12.5	'HOMOMORPHICFILT' FUNCTION	12-4
12.6	MAIN FILE (RUN)	12-5
13	ADDENDUM D	13-1
WIENER FILTERING USING KURTOSIS MINIMIZATION ALGORITHM CODE 13-1		
13.1	'RGB2INTENSITY' FUNCTION	13-1
13.2	'RATIOAVG' FUNCTION	13-1
13.3	'ATMOSOTF' FUNCTION	13-1
13.4	'WIENEROPT' FUNCTION	13-2
13.5	MAIN FILE (RUN)	13-2
14	ADDENDUM E	14-1
WIENER FILTERING USING LAPLACIAN OPERATOR ALGORITHM CODE 14-1		
14.1	'RGB2INTENSITY' FUNCTION	14-1
14.2	'RATIOAVG' FUNCTION	14-1
14.3	'ATMOSOTF' FUNCTION	14-1
14.4	'CUMLAP' FUNCTION	14-2
14.5	'WIENEROPT' FUNCTION	14-2
14.6	MAIN FILE (RUN)	14-3
15	ADDENDUM F	15-1

DYNAMIC ILLUMINANCE-REFLECTANCE ATMOSPHERIC TURBULENCE SUPPRESSION	
ALGORITHM CODE.....	15-1
15.1 'RGB2INTENSITY' FUNCTION.....	15-1
15.2 'RATIOAVG' FUNCTION.....	15-1
15.3 'WIS' FUNCTION	15-1
15.4 MAIN FILE (RUN)	15-2
16 ADDENDUM G.....	16-1
DYNAMIC ILLUMINANCE-REFLECTANCE ATMOSPHERIC TURBULENCE SUPPRESSION	
WITH WIENER FILTERING ALGORITHM CODE.....	
16.1 'RGB2INTENSITY' FUNCTION.....	16-1
16.2 'RATIOAVG' FUNCTION.....	16-1
16.3 'ATMOSOTF' FUNCTION	16-1
16.4 'CUMLAP' FUNCTION	16-2
16.5 'WIENEROPT' FUNCTION	16-2
16.6 'WIS' FUNCTION	16-3
16.7 MAIN FILE (RUN)	16-3
17 ADDENDUM H.....	17-1
EXPERIMENT 1: ALGORITHM FPS PERFORMANCE ADDITIONAL GRAPHS AND TABLES..	
17-1	
17.1 IMAGE REGISTRATION AND LAPLACIAN PYRAMID IMAGE FUSION.....	17-1
17.2 HOMOMORPHIC FILTERING AND POWER SPECTRUM BASED METHOD	17-1
17.3 WIENER FILTERING USING KURTOSIS MINIMIZATION	17-2
17.4 WIENER FILTERING USING LAPLACIAN OPERATOR	17-3
17.5 DYNAMIC ILLUMINANCE-REFLECTANCE ATMOSPHERIC TURBULENCE SUPPRESSION	17-4
17.6 DYNAMIC ILLUMINANCE-REFLECTANCE ATMOSPHERIC TURBULENCE SUPPRESSION WITH WIENER FILTERING	17-5
17.7 SPECKLE MASKING	17-7
18 ADDENDUM I	18-1
EXPERIMENT 2: ALGORITHM FPS PERFORMANCE ADDITIONAL GRAPHS	
18-1	
18.1 UNPROCESSED VIDEO.....	18-1
18.2 IMAGE REGISTRATION AND LAPLACIAN PYRAMID IMAGE FUSION.....	18-2
18.3 HOMOMORPHIC FILTERING AND POWER SPECTRUM BASED METHOD	18-3
18.4 WIENER FILTERING USING LAPLACIAN OPERATOR	18-4
18.5 DYNAMIC ILLUMINANCE-REFLECTANCE ATMOSPHERIC TURBULENCE SUPPRESSION	18-5
18.6 DYNAMIC ILLUMINANCE-REFLECTANCE ATMOSPHERIC TURBULENCE SUPPRESSION WITH WIENER FILTERING	18-6
19 ADDENDUM J.....	19-1

EXPERIMENT 3: ALGORITHM ABERRATION ADDITIONAL TABLES.....	19-1
19.1 MEASUREMENT RESULTS.....	19-1
20 ADDENDUM K.....	20-1
EXPERIMENT 4: ALGORITHM MTF RESULT ADDITIONAL TABLES.....	20-1
21 ADDENDUM L.....	21-1
EXPERIMENT 6: WIENER FILTERING LAMBDA SELECTION ADDITIONAL GRAPHS AND TABLES.....	21-1
21.1 ALGORITHM PERFORMANCE RESULTS	21-1
21.1.1 <i>Wiener Filtering User Selected Lambda Algorithm</i>	21-1
21.1.2 <i>Wiener Filtering using Laplacian Operator</i>	21-2
21.2 IMAGE SHARPNESS RESULTS.....	21-3
21.3 MTF RESULTS	21-4
22 ADDENDUM M.....	22-1
EXPERIMENT 7: FILTERING RECEIVED LIGHT ADDITIONAL GRAPHS AND TABLES.....	22-1
22.1 IMAGE SHARPNESS RESULTS ACROSS 50 FRAMES	22-1
22.2 MTF RESULTS	22-3



Table of Figures

<i>FIGURE 1-1: RESEARCH METHODOLOGY BLOCK DIAGRAM.</i>	<i>1-4</i>
<i>FIGURE 2-1: REFRACTED LIGHT WAVES THROUGH DIFFERENT DENSITIES.....</i>	<i>2-2</i>
<i>FIGURE 2-2: EFFECT OF TURBULENT ATMOSPHERE ON A POINT OF LIGHT (ILLUSTRATION FROM [8]).....</i>	<i>2-3</i>
<i>FIGURE 2-3: EXAMPLE OF PHOTOMETRIC AND GEOMETRIC DISTORTIONS.....</i>	<i>2-5</i>
<i>FIGURE 2-4: EXAMPLE OF A POINT SPREAD FUNCTION.....</i>	<i>2-7</i>
<i>FIGURE 2-5: CCD CAPTURING LIGHT TRAVELLING THROUGH TURBULENT ATMOSPHERE FROM A POINT OBJECT.....</i>	<i>2-7</i>
<i>FIGURE 2-6: CONSTRUCTION OF A DIGITAL VIDEO.....</i>	<i>2-9</i>
<i>FIGURE 2-7: CENTRED DISTRIBUTION HISTOGRAM EXAMPLE.</i>	<i>2-10</i>
<i>FIGURE 2-8: LEFT BIASED DISTRIBUTION HISTOGRAM EXAMPLE.</i>	<i>2-11</i>
<i>FIGURE 2-9: RIGHT BIASED DISTRIBUTION HISTOGRAM EXAMPLE.</i>	<i>2-11</i>
<i>FIGURE 2-10: FULL DYNAMIC RANGED HISTOGRAM EXAMPLE.</i>	<i>2-12</i>
<i>FIGURE 2-11: EXPANDED HISTOGRAM EXAMPLE.....</i>	<i>2-13</i>
<i>FIGURE 2-12: IMAGE NEGATIVE HISTOGRAM EXAMPLE.</i>	<i>2-13</i>
<i>FIGURE 2-13: EXAMPLES OF DIFFERENT WINDOW SHAPES.....</i>	<i>2-17</i>
<i>FIGURE 2-14: GAUSSIAN PYRAMID FOR LEVEL 0 TO 3.....</i>	<i>2-21</i>
<i>FIGURE 2-15: EXPANDED LAPLACIAN PYRAMID CREATED FROM EXPANDED GAUSSIAN PYRAMID.....</i>	<i>2-22</i>

FIGURE 2-16: ELECTROMAGNETIC WAVE.	2-24
FIGURE 2-17: ELECTROMAGNETIC WAVE POLARIZATION BY REFLECTION.	2-26
FIGURE 2-18: POLARIZATION BY SCATTERING.	2-27
FIGURE 2-19: ELECTROMAGNETIC SPECTRUM.	2-28
FIGURE 2-20: REFRACTION OF WHITE LIGHT.	2-28
FIGURE 2-21: COLOURED OPTICAL FILTER EFFECTS.	2-31
FIGURE 2-22: ALTERNATIVE APPROACH USING ADAPTIVE PHASE CORRECTING DEVICE (LC SLM).	2-33
FIGURE 2-23: ALTERNATIVE APPROACH USING ADAPTIVE PHASE CORRECTING DEVICE (MEMS MIRROR).	2-33
FIGURE 2-24: COMPARISONS OF TELESCOPE DESIGNS.	2-35
FIGURE 2-25: THE QUESTAR MFL 90 MULTI FOCAL LENGTH LENS.	2-36
FIGURE 2-26: THE CELESTRON NEXSTAR 8 SE TELESCOPE.	2-38
FIGURE 2-27: THE MEADE ETX 125PE TELESCOPE.	2-38
FIGURE 2-28: THE MEADE 8" LX90GPS TELESCOPE.	2-39
FIGURE 2-29: ARECONT VISION AV3100 SURVEILLANCE CAMERA.	2-41
FIGURE 2-30: PIXELINK PL-A741 INDUSTRIAL AND MACHINE VISION CAMERA.	2-41
FIGURE 2-31: QIMAGING RETIGA 1300 CCD CAMERA.	2-42
FIGURE 2-32: FREQUENCY MODULATION TEST SETUP.	2-44
FIGURE 2-33: ILLUSTRATION OF RMSE DOWNFALL.	2-46
FIGURE 3-1: OPTICAL FLOW FIELD REPRESENTATION.	3-7
FIGURE 3-2: IMAGE FUSION COMPARISON.	3-11
FIGURE 3-3: FLOWCHART OF CONCEPTUAL SOLUTION TO OBTAINING A NEAR DIFFRACTION-LIMITED IMAGE.	3-13
FIGURE 3-4: BLOCK DIAGRAM OF PROCESSING STEPS FOR BISPECTRAL SPECKLE IMAGING.	3-14
FIGURE 3-5: THE HEXAGONAL STRUCTURE OF THE BI-SPECTRUM.	3-16
FIGURE 3-6: THE HOMOMORPHIC FILTERING AND POWER SPECTRUM BASED METHOD BLOCK DIAGRAM.	3-22
FIGURE 3-7: CONVOLUTIONAL MASK.	3-22
FIGURE 4-1: SPECKLE MASKING SIMPLIFIED FLOW DIAGRAM.	4-2
FIGURE 4-2: SPECKLE MASKING PSEUDO CODE SNIPPET.	4-5
FIGURE 4-3: IMAGE FRAME SEGMENTED INTO FOUR OVERLAPPING SEGMENTS. SEGMENT 1 IS HIGHLIGHTED.	4-6
FIGURE 4-4: 64X64 HAMMING WINDOW FOR MULTIPLICATION WITH A 64X64 SIZED SEGMENT DURING RECONSTRUCTION.	4-7
FIGURE 4-5: BLOCK DIAGRAM OF IMAGE REGISTRATION AND LAPLACIAN PYRAMID IMAGE FUSION ALGORITHM.	4-8
FIGURE 4-6: BLOCK DIAGRAM OF HOMOMORPHIC FILTERING AND POWER SPECTRUM RESTORATION BASED ALGORITHM.	4-9
FIGURE 4-7: CONVOLUTIONAL MASK.	4-10
FIGURE 4-8: BLOCK DIAGRAM OF DYNAMIC ILLUMINANCE-REFLECTANCE ATMOSPHERIC TURBULENCE SUPPRESSION ALGORITHM.	4-15
FIGURE 4-9: DYNAMIC ILLUMINANCE-REFLECTANCE ATMOSPHERIC TURBULENCE SUPPRESSION WITH WIENER FILTERING.	4-16
FIGURE 5-1: IMAGING SYSTEM.	5-2
FIGURE 5-2: OVERVIEW OF EXPERIMENTS.	5-4

FIGURE 5-3: CHECKERBOARD TARGET EXAMPLE.....	5-10
FIGURE 5-4: SPATIAL FREQUENCY CHART EXAMPLE.	5-14
FIGURE 5-5: EXPECTED MTF CURVE.	5-16
FIGURE 6-1: SPECKLE MASKING SUB IMAGE SIZE COMPARISON PERFORMANCE GRAPH.....	6-10
FIGURE 6-2: ALGORITHM COMPARISON FPS PERFORMANCE GRAPH.....	6-10
FIGURE 6-3: IMAGE REGISTRATION AND LAPLACIAN PYRAMID IMAGE FUSION FPS PERFORMANCE GRAPH.....	6-11
FIGURE 6-4: AVERAGE IMAGE SHARPNESS COMPARISON, HIGHEST TO LOWEST.	6-15
FIGURE 6-5: VIDEO ABERRATION RESULTS – HORIZONTAL DISPLACEMENT.	6-17
FIGURE 6-6: VIDEO ABERRATION RESULTS – VERTICAL DISPLACEMENT.	6-18
FIGURE 6-7: IMAGE REGISTRATION AND LAPLACIAN PYRAMID IMAGE FUSION ALGORITHM MTF CURVE.....	6-21
FIGURE 6-8: HOMOMORPHIC FILTERING AND POWER SPECTRUM BASED METHOD MTF CURVE.	6-21
FIGURE 6-9: WIENER FILTERING USING LAPLACIAN OPERATOR ALGORITHM MTF CURVE.	6-22
FIGURE 6-10: DYNAMIC ILLUMINANCE-REFLECTANCE METHOD MTF CURVE.	6-22
FIGURE 6-11: DYNAMIC ILLUMINANCE-REFLECTANCE WITH WIENER FILTERING METHOD MTF CURVE.....	6-23
FIGURE 6-12: SPECKLE MASKING ALGORITHM MTF CURVE.	6-23
FIGURE 6-13: WIENER FILTERING FPS PERFORMANCE COMPARISON GRAPH.....	6-28
FIGURE 6-14: WIENER FILTERING MTF COMPARISON.....	6-29
FIGURE 6-15: IMAGE SHARPNESS RESULTS OF UNFILTERED AND FILTERED VIDEO.	6-31
FIGURE 6-16: MTF RESULTS OF UNFILTERED AND FILTERED VIDEO.	6-31
FIGURE 6-17: MSE RESULTS OF UNFILTERED AND FILTERED VIDEO (LOWER IS BETTER).	6-32
FIGURE 6-18: PSNR RESULTS OF UNFILTERED AND FILTERED VIDEO (HIGHER IS BETTER).	6-33
FIGURE 7-1: UNPROCESSED VIDEO FRAME.....	7-2
FIGURE 7-2: IMAGE REGISTRATION AND LAPLACIAN PYRAMID IMAGE FUSION.	7-2
FIGURE 7-3: HOMOMORPHIC FILTERING AND POWER SPECTRUM BASED METHOD.	7-3
FIGURE 7-4: WIENER FILTERING USING LAPLACIAN OPERATOR.....	7-3
FIGURE 7-5: DYNAMIC ILLUMINANCE-REFLECTANCE ATMOSPHERIC TURBULENCE SUPPRESSION.....	7-4
FIGURE 7-6: DYNAMIC ILLUMINANCE-REFLECTANCE ATMOSPHERIC TURBULENCE SUPPRESSION WITH WIENER FILTERING.....	7-4
FIGURE 7-7: A) AN UNPROCESSED VIDEO FRAME. B) SPECKLE MASKING AFTER PROCESSING 100 FRAMES.....	7-5
FIGURE 17-1: DYNAMIC HOMOMORPHIC FILTERING AND POWER SPECTRUM FPS PERFORMANCE GRAPH.....	17-2
FIGURE 17-2: WIENER FILTERING USING KURTOSIS MINIMIZATION FPS PERFORMANCE GRAPH.....	17-3
FIGURE 17-3: WIENER FILTERING USING LAPLACIAN OPERATOR FPS PERFORMANCE GRAPH.....	17-4
FIGURE 17-4: DYNAMIC ILLUMINANCE-REFLECTANCE ADJUSTMENT FPS PERFORMANCE GRAPH.....	17-5
FIGURE 17-5: DYNAMIC ILLUMINANCE-REFLECTANCE WITH WIENER FILTERING FPS PERFORMANCE GRAPH.....	17-6
FIGURE 18-1: UNPROCESSED VIDEO - IMAGE SHARPNESS VS. FRAME NUMBER GRAPH.	18-1
FIGURE 18-2: IMAGE REGISTRATION AND FUSION - IMAGE SHARPNESS VS. FRAME NUMBER GRAPH.	18-2
FIGURE 18-3: HOMOMORPHIC FILTERING AND POWER SPECTRUM METHOD – IMAGE SHARPNESS VS. FRAME NUMBER GRAPH.....	18-3
FIGURE 18-4: WIENER FILTERING USING LAPLACIAN OPERATOR - IMAGE SHARPNESS VS. FRAME NUMBER GRAPH. .	18-4

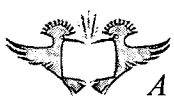


FIGURE 18-5: DYNAMIC ILLUMINANCE-REFLECTANCE ATMOSPHERIC TURBULENCE SUPPRESSION - IMAGE SHARPNESS VS. FRAME NUMBER GRAPH..... 18-5

FIGURE 18-6: DYNAMIC ILLUMINANCE-REFLECTANCE ATMOSPHERIC TURBULENCE SUPPRESSION WITH WIENER FILTERING - IMAGE SHARPNESS VS. FRAME NUMBER GRAPH. 18-6

FIGURE 21-1: WIENER FILTERING WITHOUT SHARPNESS MEASUREMENT A SELECTION FPS PERFORMANCE GRAPH. 21-1

FIGURE 21-2: WIENER FILTERING USING LAPLACIAN OPERATOR FPS PERFORMANCE GRAPH..... 21-2

FIGURE 21-3: WIENER FILTERING WITHOUT SHARPNESS MEASUREMENT A SELECTION - SHARPNESS MEASUREMENT GRAPH..... 21-3

FIGURE 21-4: WIENER FILTERING USING LAPLACIAN OPERATOR – SHARPNESS MEASUREMENT GRAPH 21-3

FIGURE 21-5: WIENER FILTERING WITHOUT SHARPNESS MEASUREMENT A SELECTION – MTF GRAPH..... 21-4

FIGURE 21-6: WIENER FILTERING USING LAPLACIAN OPERATOR – MTF GRAPH..... 21-4

FIGURE 22-1: UNFILTERED VIDEO - SHARPNESS RESULT. 22-1

FIGURE 22-2: COLOUR FILTERED VIDEO - SHARPNESS RESULT. 22-2

FIGURE 22-3: POLARISED FILTERED VIDEO - SHARPNESS RESULT. 22-2

FIGURE 22-4: UNFILTERED VIDEO - MTF CURVE. 22-3

FIGURE 22-5: RED #24 COLOUR FILTERED VIDEO - MTF CURVE. 22-4

FIGURE 22-6: POLARISED VIDEO - MTF CURVE. 22-5

List of Tables



TABLE 2-1: DFT COMPUTATIONS VS. FFT COMPUTATIONS..... 2-20

TABLE 2-2: THE QUESTAR MFL 90 MULTI FOCAL LENGTH LENS FOCAL LENGTHS AND RATIOS..... 2-37

TABLE 3-1: RECURSIVE PHASE RECOVERY USING BI-SPECTRUM PHASE INFORMATION..... 3-17

TABLE 6-1: AVERAGE IMAGE SHARPNESS COMPARISON, HIGHEST TO LOWEST..... 6-14

TABLE 6-2: VIDEO ABERRATION RESULTS – HORIZONTAL DISPLACEMENT. 6-17

TABLE 6-3: VIDEO ABERRATION RESULTS – VERTICAL DISPLACEMENT. 6-18

TABLE 6-4: ALGORITHM MSE AND PSNR EXPERIMENT - MSE EXPERIMENT RESULTS..... 6-26

TABLE 6-5: ALGORITHM MSE AND PSNR EXPERIMENT - PSNR EXPERIMENT RESULTS. 6-26

TABLE 6-6: IMAGE SHARPNESS RESULTS FOR UNFILTERED AND FILTERED VIDEO..... 6-30

TABLE 6-7: MSE AND PSNR RESULTS FOR UNFILTERED AND FILTERED VIDEO. 6-32

TABLE 17-1: IMAGE REGISTRATION AND LAPLACIAN PYRAMID IMAGE FUSION FPS PERFORMANCE 17-1

TABLE 17-2: HOMOMORPHIC FILTERING AND POWER SPECTRUM METHOD FPS PERFORMANCE..... 17-2

TABLE 17-3: WIENER FILTERING USING KURTOSIS MINIMIZATION FPS PERFORMANCE..... 17-3

TABLE 17-4: WIENER FILTERING USING LAPLACIAN OPERATOR FPS PERFORMANCE 17-4

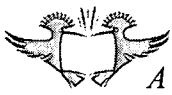
TABLE 17-5: DYNAMIC ILLUMINANCE-REFLECTANCE ADJUSTMENT FPS PERFORMANCE 17-5

TABLE 17-6: DYNAMIC ILLUMINANCE-REFLECTANCE WITH WIENER FILTERING FPS PERFORMANCE 17-6

TABLE 17-7: SPECKLE MASKING WITH SUB IMAGE SIZE OF 128X128, 64X64, AND 32X32 PERFORMANCE 17-7

TABLE 19-1: UNPROCESSED VIDEO ABERRATION RESULT. 19-1

TABLE 19-2: IMAGE REGISTRATION AND LAPLACIAN PYRAMID IMAGE FUSION ABERRATION RESULT.	19-2
TABLE 19-3: HOMOMORPHIC FILTERING AND POWER SPECTRUM BASED METHOD ABERRATION RESULT.	19-2
TABLE 19-4: WIENER FILTERING USING LAPLACIAN OPERATOR ABERRATION RESULT.	19-2
TABLE 19-5: DYNAMIC ILLUMINANCE-REFLECTANCE ATMOSPHERIC TURBULENCE SUPPRESSION ABERRATION RESULT.	19-3
TABLE 19-6: DYNAMIC ILLUMINANCE-REFLECTANCE ATMOSPHERIC TURBULENCE SUPPRESSION WITH WIENER FILTERING ABERRATION RESULT.	19-3
TABLE 20-1: UNPROCESSED VIDEO MTF MEASUREMENTS.....	20-1
TABLE 20-2: IMAGE REGISTRATION AND LAPLACIAN PYRAMID IMAGE FUSION ALGORITHM MTF MEASUREMENTS.	20-2
TABLE 20-3: HOMOMORPHIC FILTERING AND POWER SPECTRUM BASED METHOD MTF MEASUREMENTS.	20-2
TABLE 20-4: WIENER FILTERING USING LAPLACIAN OPERATOR ALGORITHM MTF MEASUREMENTS.....	20-3
TABLE 20-5: DYNAMIC ILLUMINANCE-REFLECTANCE METHOD MTF MEASUREMENTS.....	20-3
TABLE 20-6: DYNAMIC ILLUMINANCE-REFLECTANCE WITH WIENER FILTERING METHOD MTF MEASUREMENTS.	20-4
TABLE 20-7: SPECKLE MASKING ALGORITHM MTF MEASUREMENTS.....	20-4
TABLE 21-1: WIENER FILTERING WITHOUT SHARPNESS MEASUREMENT A SELECTION FPS PERFORMANCE.....	21-1
TABLE 21-2: WIENER FILTERING USING LAPLACIAN OPERATOR FPS PERFORMANCE.....	21-2
TABLE 22-1: UNFILTERED VIDEO - MTF MEASUREMENTS.	22-3
TABLE 22-2: RED #24 COLOUR FILTERED VIDEO - MTF MEASUREMENTS.....	22-4
TABLE 22-3: POLARISED VIDEO - MTF MEASUREMENTS.....	22-5



List of Symbols

nr_1	Index of Refraction of first material
nr_2	Index of Refraction of second material
θ_1	Angle of incident light to normal
θ_2	Angle of refracted light to normal
Δt	Some change in time
P_S	Short exposure laser spot
P_C	Deflection distance of laser spot
P_L	Long exposure laser spot
i_d	Degraded image
h	Point Spread Function (PSF)
i_i	Ideal image
I_d	Fourier transform of degraded image
H	Optical Transfer Function (OTF)
I_i, O	Fourier transform of ideal image
n_s	Additive noise
N_s	Fourier transform of additive noise
m	Row index
n	Column index
t	Time index
$H_{hist}(\cdot)$	Histogram function
g_L	Grey level index
G	Maximum grey level
J	Number of occurrences of an intensity level

List of Symbols (continued)

M	Total number of rows
N	Total number of columns
n	Vector defining position in an image at coordinates (m, n)
$f()$	Some image
$g()$	Some other image after processing
P	Multiplicative scaling factor
L	Additive offset
n_1	First coordinate used in Nearest Neighbour Interpolation
n_2	Second coordinate used in Nearest Neighbour Interpolation
$n_{10} \dots n_{23}$	Eight coordinates defining four points, $f(n_{10}, n_{20})$, $f(n_{11}, n_{21})$, $f(n_{12}, n_{22})$, and $f(n_{13}, n_{23})$ used in Bilinear Interpolation
$A_0 \dots A_3$	Weights calculated and used in Bilinear Interpolation
k	Filter's row coordinate /index
l	Filter's column coordinate /index
$h_{ker}(\cdot)$	Filter kernel in spatial domain
$H_{ker}(\cdot)$	Filter kernel in frequency domain
u	First dimension frequency coordinate
v	Second dimension frequency coordinate
u, v, p, q	Frequency vectors, where $u = (u_1, u_2)$, $v = (v_1, v_2)$, $p = (p_x, p_y)$, $q = (q_x, q_y)$,
p	Specifies filter order
Ω_c	Filter cut off frequency
$w(\cdot, \cdot)$	Filter mask
E	Electric field

List of Symbols (continued)

B	Magnetic field
I_{max}	Maximum intensity value
I_{min}	Minimum intensity value
ω	Angular frequency
Φ	Spatial frequency vector (u,v)
A	Linear equation matrix
x	Vector of unknowns
b	Solution set
\mathcal{F}	Fourier transform
\mathcal{F}^{-1}	Inverse Fourier transform
d_1	Horizontal component
d_2	Vertical component
R	Spatial region
α, β	Parameters estimated and used in Control Grid Interpolation $\alpha = [\alpha_1, \alpha_2, \alpha_2, \alpha_3]$, $\beta = [\beta_1, \beta_2, \beta_2, \beta_3]$
$\theta[\mathbf{n}]$	$\theta[\mathbf{n}] = [1, m, n, mn]^T$
$I_{B,n}(\mathbf{u}, \mathbf{v})$	Four dimensional bi-spectrum
$\psi(\mathbf{u}, \mathbf{v})$	Fourier phase of Four dimensional bi-spectrum $I_{B,n}(\mathbf{u}, \mathbf{v})$
$\mathbf{T}_{p,q}^{(3)}$	Two dimensional bi-spectrum shortened form
\mathbf{T}_p	Fourier transform of a one dimensional signal. Shortened form of $\mathbf{T}(p)$
\mathbf{T}_q	Fourier transform of a one dimensional signal. Shortened form of $\mathbf{T}(q)$
\mathbf{T}_{-p-q}	Fourier transform of a one dimensional signal. Shortened form of $\mathbf{T}(-(q + p))$
$\beta_{p,q}$	Bi-spectrum Fourier phase of bi-spectrum $\mathbf{T}_{p,q}^{(3)}$

List of Symbols (continued)

ϕ_p	Fourier phase of T_p
ϕ_q	Fourier phase of T_q
$-\phi_{p+q}$	Fourier phase of T_{-p-q}
b_0, \dots, b_4	Filter coefficients
$H_R(u, v)$	Power spectrum restoration filter
$H_D(u, v)$	Degradation transfer function
$W_F(u, v)$	Ideal image power spectrum
$W_i(u, v)$	Image power spectrum
$W_N(u, v)$	Noise power spectrum
λ	Scalar value used in OTF model. Value controls severity (shape) of OTF
$E(.)$	Expected value operation
μ	Mean
x	Some signal
σ	Standard deviation
$L(m, n)$	Illuminance component
$R(m, n)$	Reflectance component
W	Maximum Luminance in a low frequency white area
B	Maximum Luminance in a low frequency black area

1 CHAPTER 1: PROBLEM IDENTIFICATION

1.1 INTRODUCTION AND BACKGROUND

Video images and their use have become increasingly popular due to their availability, increasing quality and decreasing costs [1]. Video surveillance makes use of this technology to classify, and identify targets. Where the captured images are processed for object identification, low quality images can be a cause for concern [1, 2, 3]. Low quality can extend to some or all of the following properties: blur, low resolution, noise, low image stability, a small dynamic range of image [1]. Video enhancement attempts to tackle the low quality image problem by making use of one or more algorithms to convert the low quality image to one which is of a high quality [3, 4].

Video surveillance equipment can be used in military applications to view objects from distances of 100 m to 10 km [4]. The light waves travelling from the object to the surveillance camera or equipment, may pass through turbulent atmosphere on a warm day due to the rising of heat and convection currents [1, 5]. The light rays travelling through this turbulent atmosphere are refracted and leads to a degraded image with distortion [1, 4, 5]. The degradation is normally at its peak in the middle of the day and the degradation effect is referred to as heat shimmer, optical/heat scintillation, atmospheric turbulence or “seeing”.

There are two types of scintillation, terrestrial and astronomical. The terrestrial form of scintillation is the more complex of the two and is seen when viewing objects on or close to the earth’s surface; astronomical scintillation is seen when viewing an astronomical body, and this scintillation problem has been present in astronomy for decades (seen as the “twinkling” of stars) [5]. Both scintillations are due to atmospheric disturbances in the earth’s atmosphere. Within the astronomy field, some telescopes, specifically larger ones within observatories, make use of adaptive optics in an attempt to counteract the effects the turbulent atmosphere has on captured images of astronomy bodies.

With the increase of technology, processors are becoming faster, cheaper and smaller, making image enhancement and restoration techniques using software running on these

processors a tangible choice. While there are a few algorithms and different methods which may attempt to address the turbulent atmosphere imaging problem, there is very little or no comparisons between these methods in regards to their performance and efficiency.

1.2 PURPOSE OF THE PROJECT

While a few algorithms attempt to address the atmospheric turbulence problem in order to enhance images and extend the range of surveillance equipment, there is no clear distinction in regards to which is a better choice, or leading method under certain conditions. This study attempts to try and begin bridging the gap that exists in the comparisons between various atmospheric turbulence suppression algorithms, both in terms of quality of turbulence suppressed video and processing requirements.

This study will address the problem of suitable metrics for image quality and performance requirement measurements, through a compilation of metrics used in experiments. Metrics and experiments which may be implemented to produce repeatable and comparative results are important for future algorithm analysis.

By making use of this analysis one may be in a position to compare the algorithms and provide a direction in which atmospheric turbulence suppression has shown greater success. The side by side comparisons may also inspire newer or combination of techniques to be used in addressing this imaging problem. The results from the metrics applied to the algorithm implementations allows one to better judge and make choices as to which atmospheric turbulence suppression methods are better suited to real time processing and which provide the best enhancement in post processing.

1.3 SCOPE OF THE PROJECT

The broad range of atmospheric turbulence will be kept in context with regards to long range video surveillance. This context consists of the use of digital video equipment which may be used for capturing images of objects between distances of 100m and 10km from the video recorder. The focus will be on “terrestrial” atmospheric turbulence.

To narrow the scope, the study will focus on the obvious initial case for investigation, where the captured video images consist of a stationary scene which may contain small local object movement. Video footage will be captured of an object between 1km and 2km from the camera depending on atmospheric turbulence levels during summer. Specifically software post processing restoration of images will be of focus, with the analysis or comparison of a chosen few algorithms in order to ensure timely completion of this study.

With a desire for a comparison of algorithms, there is an obvious need for a metric or compilation of metrics. A metric compilation will therefore be constructed which will be implemented within experiments for algorithm analysis. The experimentation will be within the context of atmospheric turbulence degradation.

1.4 RESEARCH METHODOLOGY

A previous understanding of the atmospheric turbulence problem was created from previous studies (see [6]). The prior understanding has helped in the formulation of a research methodology. A simple block diagram is given in figure 1-1 to illustrate the basic methodology.

A literature study will be undertaken which provides an understanding and illustration of different methods which attempt to address the imaging problem. The study will also provide insight into different ways with which the methods and algorithms may be compared. Image processing techniques which may be used, as well as smaller pertinent details that may have affect on this project may also be included in this literature study. From the literature study, methods shall be selected and implemented using a compilation of metrics, allowing for an experimental investigation into the algorithms' image quality capabilities and processing needs.

Test digital video footage will be obtained using a camera coupled to a telescope or spotting scope. Long range surveillance over a distance between 1 km and 2 km in length will be used for obtaining test video footage and footage used with experiments.

The same video footages will be processed by the algorithms (through post processing), multiple experiments may analyse the same video and from the results, the different algorithm methods may be compared and an insight into the performance, image quality and other atmospheric turbulence capabilities formed. Various forms of the algorithms may also be tested by tweaking the algorithms through parameter adjustments or forming hybrid solutions.

From the results, conclusions may be drawn as to the direction in which the problem is better treated as well as the applicability of algorithms in different circumstances (such as real time processing vs. post processing).

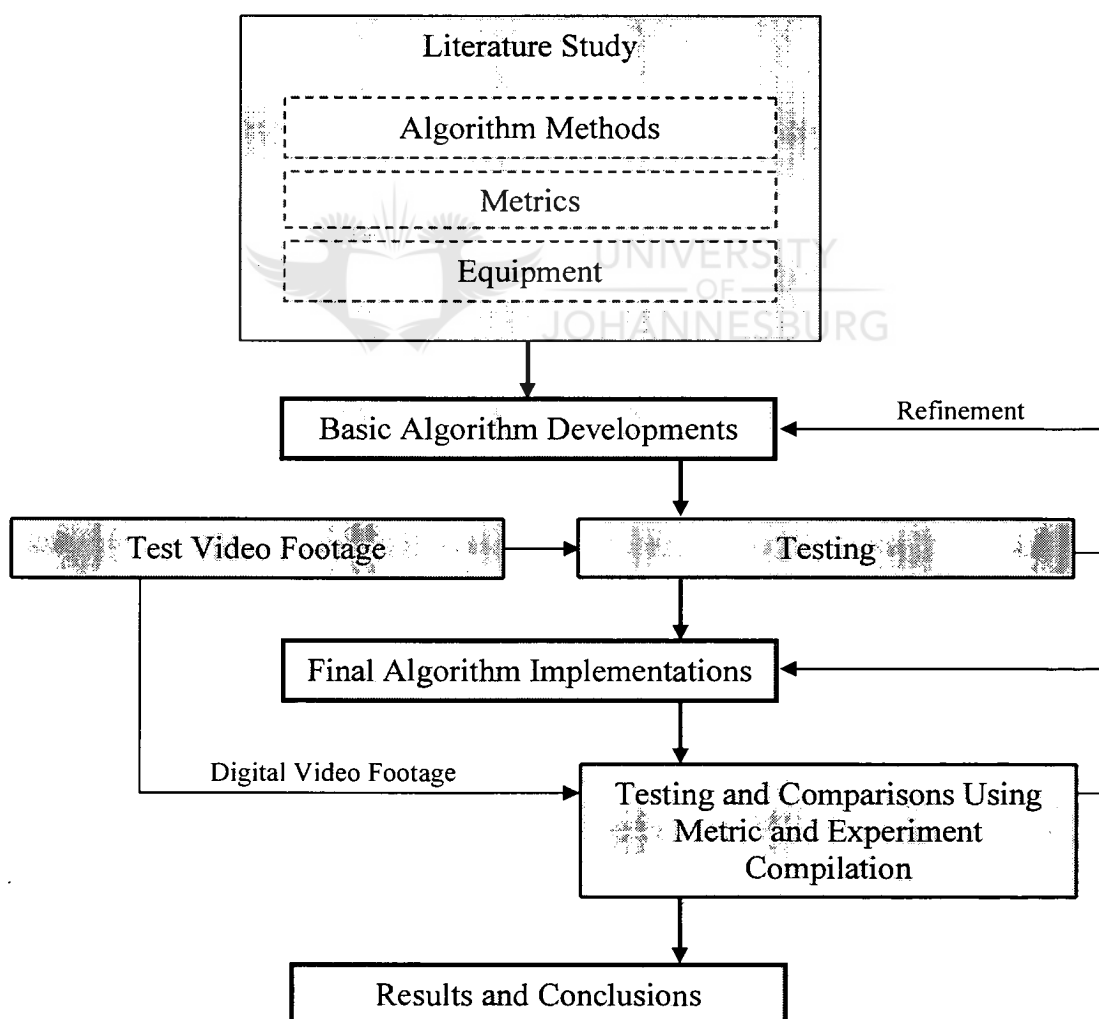


Figure 1-1: Research Methodology block diagram.

1.5 DOCUMENT OVERVIEW

Chapter 1 has outlined the projects purpose, scope and objectives, and presented a research methodology to address the problem.

A literature study is reported on in chapter 2 which provides some background as to the source of atmospheric turbulence, discusses image processing mathematical models used in representing atmospheric turbulence, reviews some image processing techniques, contains an overview of filtering electromagnetic waves, discussion of some equipment, and presents some suitable metrics.

The literature study is followed by a chapter which details the findings of considered algorithms for use in atmospheric turbulence by other authors. Algorithms which are to be analysed through the experiment compilation are discussed in detail in chapter 4 pertaining to their construction and implementation.

Experimental design is important in order to obtain repeatable and reliable results that may be used in analysis of this atmospheric turbulence problem, chapter 5 presents this. Results for the experiments listed in the Experimental Design chapter are presented and discussed in chapter 6 and includes implementation and experimental issues encountered.

Chapter 7 presents some processed image frame examples for both the readers' interest and to aid in a final discussion and comparison of the interpreted results as a whole. This study then finishes in chapter 8 with a final conclusion to the work that was undertaken.

2 CHAPTER 2: LITERATURE STUDY

2.1 INTRODUCTION AND OVERVIEW

This chapter discusses some of the various methods and techniques used in image processing. Many of these methods and techniques are used in the atmospheric turbulence suppression algorithms. The literature study starts with an investigation of the physics surrounding the atmospheric turbulence and the modelling of it which various authors employ. Some image processing techniques and those relevant to atmospheric turbulence suppression algorithms are discussed, followed by a discussion of electromagnetic wave filtering and preconditioning of received light. The literature study continues with a review of some of the relevant equipment used in capturing the turbulent video footage and some metrics that may be useful when designing the experiments. Other content which may be relevant to this project and the problem of terrestrial atmospheric turbulence suppression is also discussed.

2.2 PHYSICS SURROUNDING ATMOSPHERIC TURBULENCE

To be able to effectively approach a problem, one would like to understand the problem itself. In this section the atmospheric turbulence is discussed and how it has an effect on distorting images whose light waves have passed through this turbulent atmosphere. The relevant physics of the problem is explained with the help of Serway et al in [7].

To begin with we will first consider the thermodynamics of the problem. Suppose we have a hot surface surrounded by cooler air. The cooler air comes into contact with the hot surface due to convectional air currents. The air is then heated causing an excitation of the air molecules which now have a higher kinetic energy and the molecules become further separated from one another.

This heated portion of air has molecules separated a greater distance from one another than the cooler air and by definition is less dense than the cooler air. This results in the heated air to rise and cooler air to sweep under and take its place (convectional currents).

The convectional currents sweeping downwards to fill in the volume of the rising air may disturb the rising air column and produce non uniformity. There is thus non-uniformities in the density of the air.

Serway et al. [7] explains the consequences in the path of light when travelling through different densities of matter. A light ray is refracted when the ray travels through two elements of different densities. The light ray bends towards the normal between two volumes when it travels from a less dense volume to one which is denser. It bends away travelling from a denser to a less dense volume.

The relationship of the refraction of light between two densities can be expressed through Snell's law of refraction:

$$nr_1 \sin \theta_1 = nr_2 \sin \theta_2, \quad (2.1)$$

where nr_1 and nr_2 is the index of refraction and θ_1 and θ_2 is the angle between the light ray and the normal between the two densities for density 1 and 2 respectively.

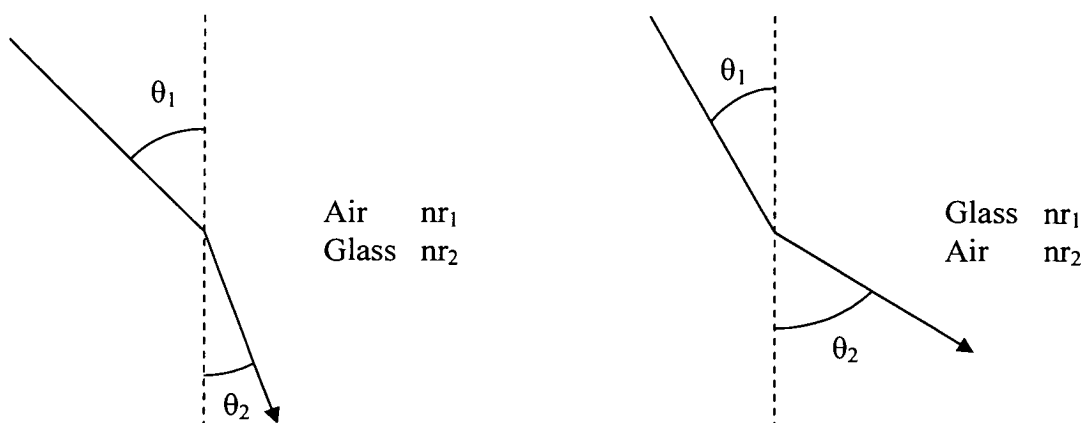


Figure 2-1: Refracted light waves through different densities

As the light rays travel through the rising columns of air the light waves are refracted. If the rising columns of air have surface normals which are parallel to one another, then Snell's law tells us that the light waves entering the column of air will be parallel to the light waves

exiting the column of air. However, due to disturbances introduced by the convective currents and the stochastic nature of the rising column of air, the normals are not parallel resulting in light waves exiting the column of air to not be parallel to those entering it.

Cohen et al. in [8] discusses how this refractive index of air is dependent on the light wavelength, atmospheric pressure and temperature, but because the pressure variations are small, the temperature differences are the main factor in refractive index changes. Cohen et al. goes on to explain that the convection currents break the atmosphere into different inhomogeneities as discussed above, but both large scale inhomogeneities and smaller scale ones exist. The large scale inhomogeneities are primarily due to the sun's rays heating up differing portions of the earth's surface (e.g. a tarred road and a grassy field). The refractive index fluctuations are referred to as turbulent eddies [8].

An example used by Fante in [9] and illustrated in [8] is shown in figure 2-2 for completeness and is used in the explanation of the effects these eddies have on a laser spot of light.

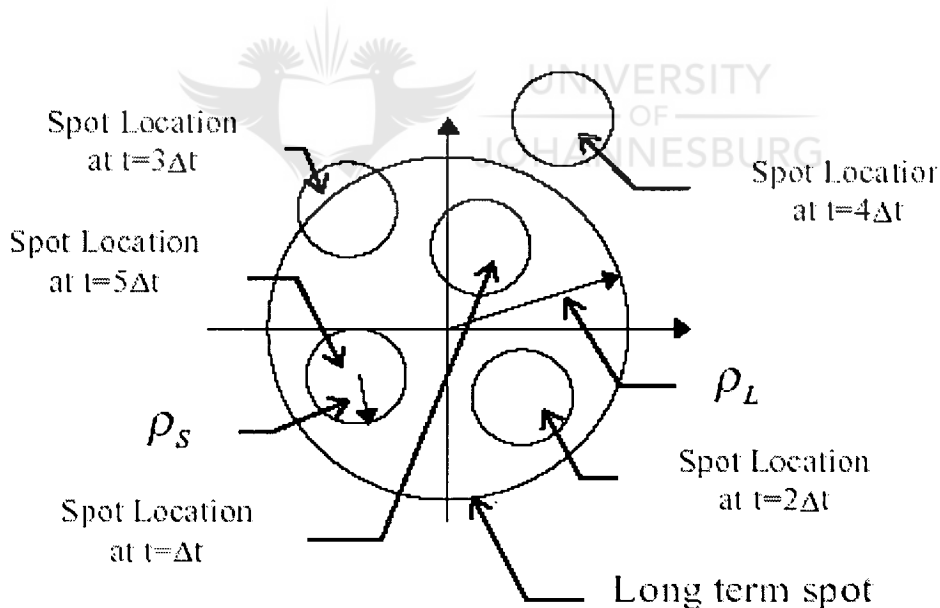


Figure 2-2: Effect of turbulent atmosphere on a point of light (illustration from [8])

If one considers a laser beam which passes through turbulent atmosphere onto a piece of paper, Figure 2-2 is an example of what one may observe of the point laser of light on the paper for different times at Δt , $2\Delta t$, $3\Delta t$, $4\Delta t$ and $5\Delta t$. As explained by Cohen et al. within a short exposure picture one would notice a broadening of the laser spot (ρ_S) which is deflected

by a distance (P_c). He discusses that the deflection to a new geometric location is primarily due to larger eddies while the summation of smaller eddies accounts for the broadening of the laser spot. If a longer exposure time is taken, one would see a much wider spot with the radius P_L . Due to the quasi-periodic nature of the turbulent atmosphere as discussed by various authors [1, 4, 8, 10, 11], by averaging over the estimated period of the geometric variance (or over many periods) one can find the true location of the point of light. In the example history shown in figure 2-2 this true centre location of the laser point of light, as if no turbulent atmosphere were present, is the centre of the long exposure spot of light with radius P_L .

Due to the atmospheric turbulence, the wavefront of light which a camera receives is tilted, producing the shifts in the image [12]. The atmospheric turbulence also scatters, and absorbs some of the light due to molecules in the air known as aerosols [12, 13].

2.3 ATMOSPHERIC TURBULENCE MODELLING

Various parameters may cause image distortion such as camera shake due to wind or turbulent atmosphere causing geometric distortions, lens optics, defocusing, aerosols present in the air, blurring and turbulent atmosphere causing photometric distortions. The geometric distortions relate how captured objects in a degraded video footage do not appear in their geometrically correct locations in a video frame in relation to the real object itself (e.g. warping of the object due to the turbulent atmosphere between the real object and the video capturing device). The photometric distortions account for the object appearing blurred in a video footage. Atmospheric turbulence can be the cause of both geometric distortion and photometric distortion [1] but not necessarily the only cause, there may be other contributing factors such as wind causing camera shake or the camera lens genially being out of focus. An example of geometric and photometric distortions is given in Figure 2-3.



Original image



Photometric distortion



Geometric distortion



Photometric and Geometric distortion

Figure 2-3: Example of Photometric and Geometric distortions

There are a few authors that use models which try to model how the degradation is taking place on the ideal image and producing one distorted and blurred by the atmospheric turbulence. Some of the authors relate the degradation of the image by the turbulent atmosphere as the convolution of the ideal undistorted image with a Point Spread Function (PSF) [2] [3].

$$\mathbf{i}_d = \mathbf{h} * \mathbf{i}_i, \quad (2.2)$$

where \mathbf{i}_d is the degraded image, \mathbf{h} the PSF, \mathbf{i}_i the ideal image, and $*$ showing convolution. With \mathbf{i}_i and \mathbf{h} unknown the problem is ill defined. In the frequency domain the convolution becomes multiplication and the problem can be viewed as a blind deconvolution problem.

$$\mathbf{I}_d = \mathbf{H}\mathbf{I}_i, \quad (2.3)$$

with \mathbf{I}_d , \mathbf{H} and \mathbf{I}_i the corresponding frequency domain counterparts of those in (2.2). \mathbf{H} is known as the Optical Transfer Function (OTF).

A slightly more accurate model is one in which additive noise (\mathbf{n}_S) is included so (2.2) and (2.3) respectively become that shown in (2.4) and (2.5).

$$\mathbf{i}_d = \mathbf{h} * \mathbf{i}_i + \mathbf{n}_S, \quad (2.4)$$

$$\mathbf{I}_d = \mathbf{H}\mathbf{I}_i + \mathbf{N}_S, \quad (2.5)$$

where \mathbf{N}_S is the Fourier transform of \mathbf{n}_S .

Other authors recognise the geometric distortion apart from the photometric distortion created by the PSF and model the system slightly differently [1, 10], while other authors view the problem purely from an image enhancement viewpoint in which the video needs to be stabilised and geometric distortions removed from an image and the need for deblurring of the image:

$$g(m, n, t) = D[x(m, n, t) * h(m, n, t), t] + n_s(m, n, t), \quad (2.6)$$

with $g(m, n, t)$ the degraded image frame, $x(m, n, t)$ the ideal undistorted image frame, $h(m, n, t)$ the PSF, $n_s(m, n, t)$ additive noise and $D[., t]$ a time dependant function which causes the geometric distortions. (2.6) is presented by Li et al. in [10] and shows some similarities with (2.4), the only difference is $D[., t]$ is included for the geometric distortions. To simplify the model the methods often neglect the additive noise, $n_s(m, n, t)$.

An example of a PSF and how the CCD of a digital camera may capture a point of light travelling through turbulent atmosphere is shown in figure 2-4 and figure 2-5.

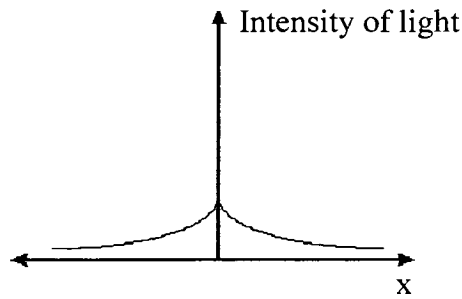


Figure 2-4: Example of a Point Spread Function
(Taken from [18])

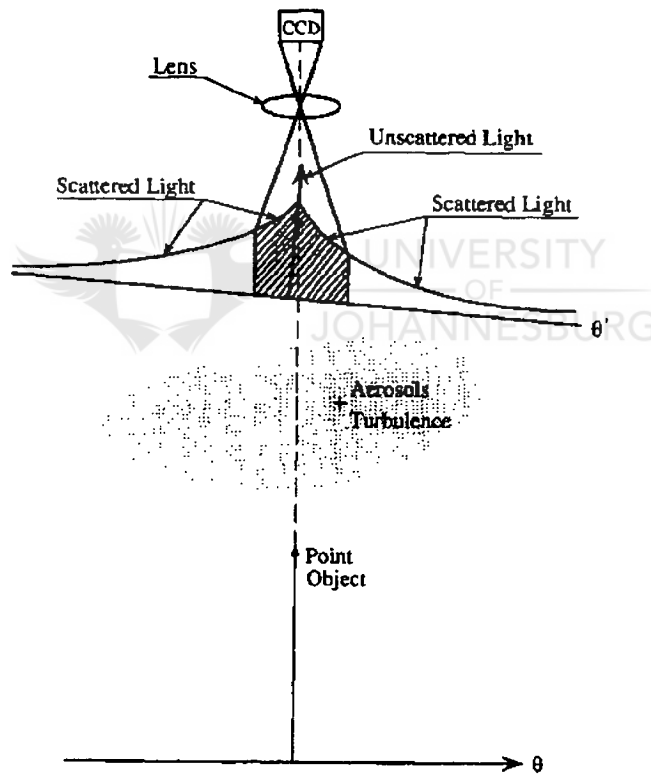


Figure 2-5: CCD capturing light travelling through turbulent atmosphere from a point object
(Taken from [12])

2.4 DIGITAL IMAGE CONSTRUCTION

To be able to implement various algorithms for use with digital video images, and to understand which mathematical functions to use and how they should be utilised, we need to consider the surrounding construction of a digital video.

We will broadly consider the construction around a digital image.

As discussed in [20] a digital image can be viewed as a two dimensional array with its indexes starting at the top left hand corner. Each index (x,y) of the array contains a value called an intensity value which is normally between 0 and 1 (real) or between 0 and 255 (integer).

A grey scale image (or sometimes referred to as a grey image) is an image as described above where the intensity values correspond to different levels of grey. The highest intensity value is white and the lowest intensity black. Lower intensity values are darker than higher intensity values.

A colour image has multiple two dimensional arrays for a single image. If we consider the colour space RGB (**R**ed **G**reen **B**lue), we have three two dimensional arrays containing intensity values, a two dimensional array to signify intensities of red, one for green and the last for blue. There are also other colour spaces such as HSV (**H**ue **S**aturation **V**ariance) but we will not look into these colour spaces as the RGB colour space in this context is the easiest to work with.

The index location in a digital image is known as a pixel. A 320 x 240 digital image is an image with a length of 320 and a height of 240 and contains $320 \times 240 = 76800$ pixels. The 320 x 240 is known as the resolution of an image. The resolution is almost always a power of 2 in order to help processing done by a computer.

In digital video, the single grey or colour image is known as an image frame. With the digital video footage its construction can be viewed as containing an extra dimension, that of the frame number. A digital video is simply one containing a succession of image frames, and video footage nowadays commonly contains a rate of 25 frames per second (FPS) although

this number is dependent on the video capturing device used (PAL-25 fps, NTSC-30 fps). The number of frames per second a video capturing device obtains need not however be the same amount set by the PAL or NTSC standards of 25 and 30 fps.

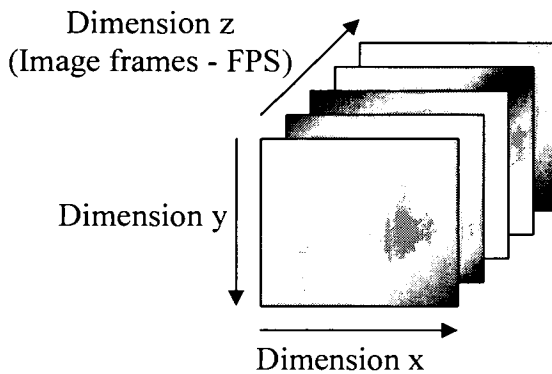


Figure 2-6: Construction of a digital video

2.5 BASIC GREY-LEVEL IMAGE PROCESSING USING HISTOGRAM

Some basic image processing may be the manipulation of the image Histogram. The Histogram is an important tool in image processing and is simply a function and graph showing the counted pixel intensity values of an image. The Histogram gives no information as to the position of intensity values but just the frequency of occurrence of the intensity values. For a good dynamic range and detail presented in an image, the whole length of the Histogram should be well used. [20]

The following is sourced from [20].

The Histogram ($H_{hist}(\cdot)$) may be defined for an image $f()$ with J occurrences of an intensity level (grey level) g as:

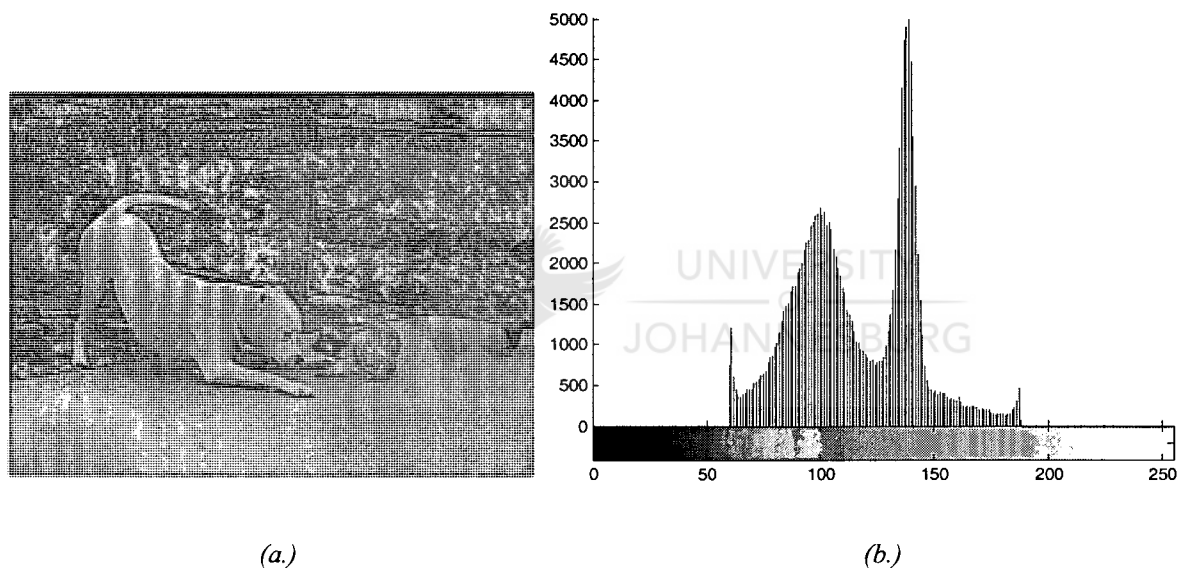
$$H_{hist} g = J . \quad (2.7)$$

The grey level used henceforth is from 0 to 255. Figures 2-7, 2-8, 2-9 and 2-10 give some example images and their corresponding Histogram, from which the effects may be seen.

The average optical density (AOD) is a measure of the average grey level and can be computed using the histogram by:

$$\text{AOD}(f()) = \frac{1}{NM} \sum_{g=0}^{G-1} gH_{hist}(g), \quad (2.8)$$

where the dimensions of the image is $M \times N$, $H_{hist}(g)$ the histogram for image $f()$ for the grey level g .



*Figure 2-7: Centred distribution Histogram example.
(a) is the image of the centred distributed histogram shown in (b)*

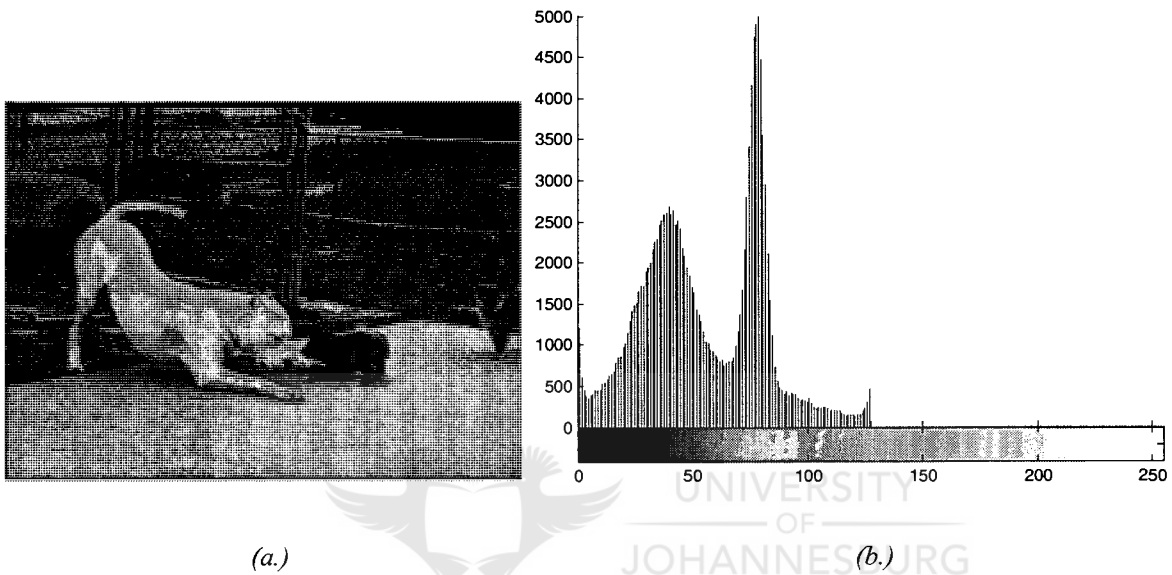
If an image is constrained to a small set of grey levels then it will not provide as much contrast as one in which all the grey levels are used (Figure 2-7 vs. Figure 2-10). If an image is under exposed the grey levels present will predominately be the lower levels as illustrated in Figure 2-8. Figure 2-9 illustrates an image constrained to the higher levels.

To correct and adjust the histogram point operations may be used. A point operation is a function which is identically applied to each pixel in an image.

The linear point operation can be defined as:

$$g(n) = Pf(n) + L, \quad (2.9)$$

where $f(n)$ is the image at position n , $g(n)$ the new image, P a multiplicative scaling factor and L an additive offset.

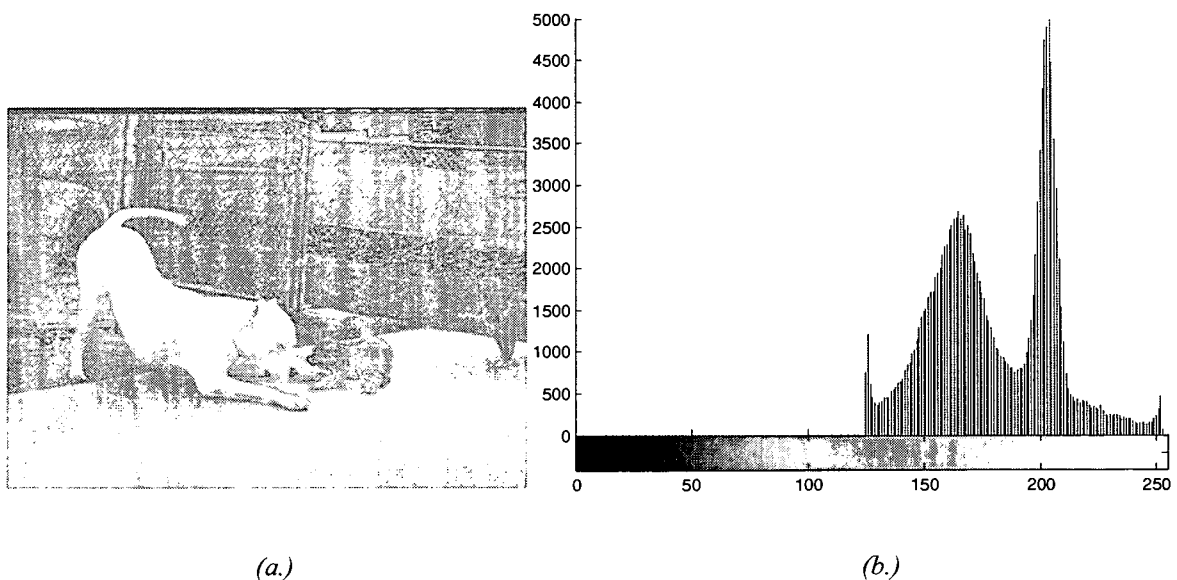


(a.)

(b.)

Figure 2-8: Left biased distribution Histogram example.

(a) is the image of the left biased distribution histogram shown in (b)



(a.)

(b.)

Figure 2-9: Right biased distribution Histogram example.

(a) is the image of the right biased distribution histogram shown in (b)

If the image in figure 2-7 is $f(..)$, then a negative L can transform the image to that Figure 2-8 and a positive L to that shown in Figure 2-9. The scaling term P will enable a contraction or expansion across grey levels if it is less than or greater than 1. One must be careful to not set P or L to high as saturation will result and there may be overflow errors or there will be clipping (maximum is 255 and values will be clipped to this level unless the number of bits used for the grey levelling is increased), the same problem is with lower values which may result in underflow conditions. A combination of a positive L and a less than 1 multiplicative value, P , can transform the image from that in Figure 2-10 to the image in Figure 2-7. Going in the reverse direction with a greater than 1 P and negative L (from Figure 2-7 to Figure 2-11) there is a improvement from Figure 2-7 but the histogram differs to that of Figure 2-10 due to the grey levels being discrete and resulting round-off errors.

In short the additive offset can transform the image to a given average brightness level, and making use of the AOD we can better compare images by setting their AODs to the same level using the offset. The multiplicative scaling term allows compression or stretching of the histogram (also known as a contrast stretch), with a stretched histogram having an image which has better visibility of detail than an image with a compressed histogram. The stretching of a histogram to span all grey levels is known as a full scale histogram stretch or FSHS.

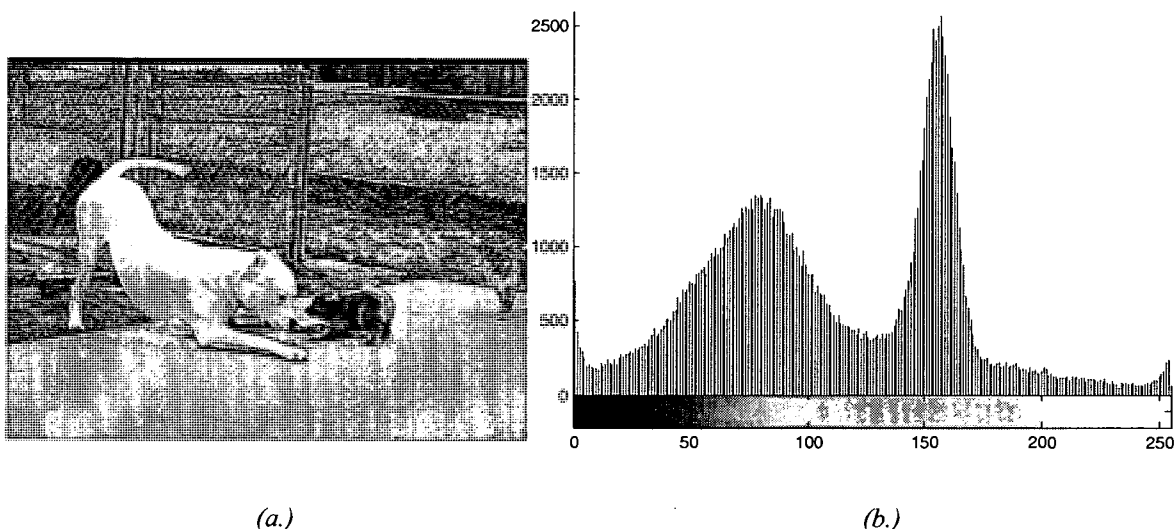
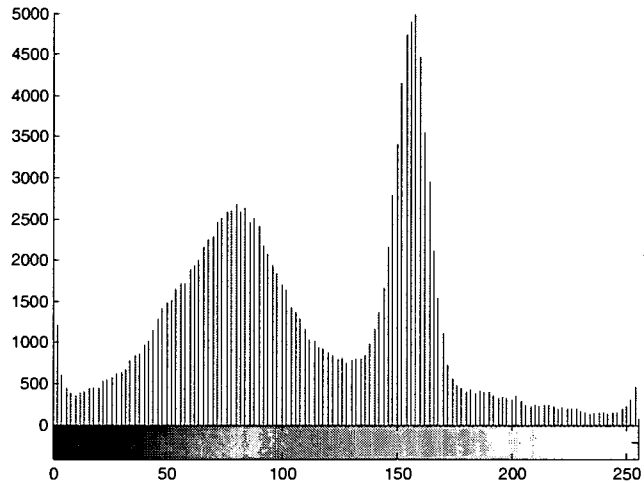


Figure 2-10: Full dynamic ranged Histogram example.
(a) is the image of the full dynamic ranged histogram shown in (b)



(a.)

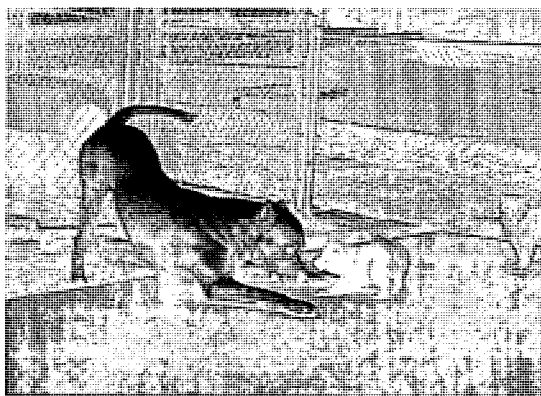


(b.)

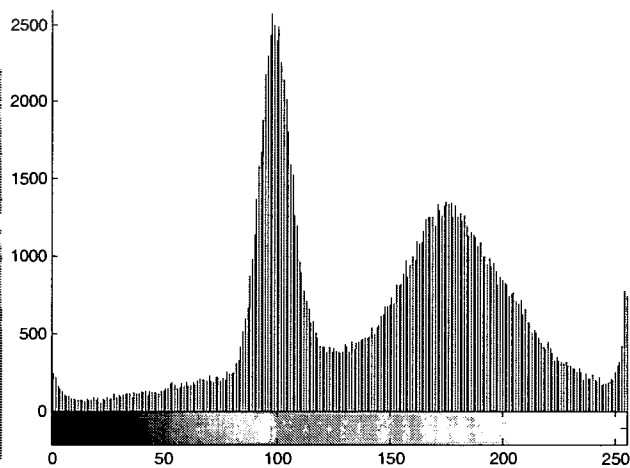
Figure 2-11: Expanded Histogram example.

(a) is the image of the expanded histogram shown in (b)

To make an image negative a scaling term of $P=-1$ and $L=G-1$ is used, where G is the maximum grey level. Figure 2-12 shows the negative image of Figure 2-10.



(a.)



(b.)

Figure 2-12: Image Negative Histogram example.

(a) is the negative image of the histogram shown in (b)

Up to now linear point operations have been discussed. There are also nonlinear point operations such as the logarithmic point operation and the important Histogram equalization (or sometimes known as histogram flattening).

The logarithmic operation is a conjoined operation with the FSHS and can be defined as:

$$g(\mathbf{n}) = \text{FSHS} \log[1 + f(\mathbf{n})] , \quad (2.10)$$

where FSHS is the full scale histogram stretch, $f(\mathbf{n})$ the original image, $g(\mathbf{n})$ the resultant image and \mathbf{n} the position. The additive unity is there to avoid the logarithm of 0. The logarithmic point operation works by suppressing brighter objects from which the FSHS is then applied with a final result of bringing forward detail from dimmer objects.

2.6 INTERPOLATION

Geometric image operations generally require the spatial mapping of the coordinates of one image to define a new image. However, digital images have discrete coordinates and the mapping might not fall on the discrete lattice structure. Interpolation is then necessary to take non-integer coordinates to integer values for expression in the required discrete row-column format [20]. In the following discussions we consider the coordinate function $\mathbf{a}(\mathbf{n})=[a_1(n_1, n_2), a_2(n_1, n_2)]$ which is usually continuous (e.g. analogue images). The need for interpolation by the above definition then arises through $a_1(n_1, n_2)$ and $a_2(n_1, n_2)$ coordinates not being integer values.

2.6.1 Nearest-Neighbour Interpolation

Nearest neighbour interpolation simply maps the geometric coordinates to the nearest integer coordinates of f , such that:

$$g(\mathbf{n}) = f\{\text{INT}[a_1(n_1, n_2) + 0.5], \text{INT}[a_2(n_1, n_2) + 0.5]\}, \quad (2.11)$$

where $\text{INT}[R]$ (as used in [20]) specifies the nearest integer value which is less than or equal to R . f is the function which is being interpolated using rounded integer coordinates and $g(\mathbf{n})$ the interpolated function or image.

2.6.2 Bilinear Interpolation

Bilinear Interpolation produces a smoother interpolation than the nearest neighbour approach (which results in jagged edges and a blocky effect) [20].

The bilinear interpolation is defined by Bovik in [20] as follows: Given four image coordinates of $f[\mathbf{a}(\mathbf{n})]$ (which could be nearest neighbours), $f(n_{10}, n_{20})$, $f(n_{11}, n_{21})$, $f(n_{12}, n_{22})$, and $f(n_{13}, n_{23})$, then the image $g(\mathbf{n})$ is computed through:

$$g(n_1, n_2) = A_0 + A_1 n_1 + A_2 n_2 + A_3 n_1 n_2, \quad (2.12)$$

where the weights A_0 , A_1 , A_2 and A_3 are found using:

$$\begin{bmatrix} A_0 \\ A_1 \\ A_2 \\ A_3 \end{bmatrix} = \begin{bmatrix} 1 & n_{10} & n_{20} & n_{10}n_{20} \\ 1 & n_{11} & n_{21} & n_{11}n_{21} \\ 1 & n_{12} & n_{22} & n_{12}n_{22} \\ 1 & n_{13} & n_{23} & n_{13}n_{23} \end{bmatrix}^{-1} \begin{bmatrix} f(n_{10}, n_{20}) \\ f(n_{11}, n_{21}) \\ f(n_{12}, n_{22}) \\ f(n_{13}, n_{23}) \end{bmatrix}. \quad (2.13)$$

In some cases the $a_1(n_1, n_2)$ and $a_2(n_1, n_2)$ mapping coordinates may not fall within the pixel ranges. In these cases, it is usual for a nominal value to be used such as $g(\mathbf{n})=0$. [20]

2.7 FILTERS

Filtering and the method of filtering has been around for decades and has become a very broad topic. This section specifically focuses on and touches on some of the aspects of two dimensional filters used in conjunction with discrete time and space signals (i.e. digital images and video).

Digital filtering is one of the most important processes in DSP (Digital Signal Processing) [27]. The filtering operation for an important class of filters as stated by Ifecher et al. in [27] is defined as:

$$y(n) = \sum_{k=0}^{N-1} h_{ker}(k) f(n-k), \quad (2.14)$$

where $h_{ker}(k)$, $k=0, \dots, N-1$ are the coefficients of the filter, $f(n)$ the filtered signal and $y(n)$ the output. For the two dimensional case the filtering would be expressed as:

$$y(m, n) = \sum_{k=0}^{M-1} \sum_{l=0}^{N-1} h_{ker}(k, l) f(m-k, n-l), \quad (2.15)$$

where $h_{ker}(k, l)$, $k=0, \dots, M-1$; $l=0, \dots, N-1$ are the filter coefficients and $f(m, n)$ the two dimensional input signal and $y(m, n)$ the two dimensional output signal.

In general filtering is convolution of the impulse response of the filter with the signal. In the frequency domain the convolution becomes multiplication and is the multiplication of the filters transfer function with the spectrum of the signal.

Digital filters are generally grouped into two categories, IIR (infinite impulse response) and FIR (finite impulse response) filters. FIR filters are a common choice for image processing as they can have an exact linear phase response and thus introduce no phase distortion. FIR filters on the other hand require more coefficients than an IIR filter for a sharp cutoff frequency. This greater number relates to the requirement of more processing time and storage for coefficients. [27]

2.7.1 Windowing

Windowing is a method used within image processing and filtering specifically whereby pixel intensity values are collected around a location according to a rule. The rule set by the window for example may be a half of one grey level plus a third of

another grey level. The ruling layout is known as the window and may be of any shape (most commonly a square or circular type shape). [20]

Figure 2-13 shows some examples of the different shapes a window might have. The shaded area is the position the window is presently at and the position for which the grey ruling is used and the calculated result stored. For a sliding window the window moves to each pixel location and uses the ruling with the surrounding grey levels to calculate the value for that point.

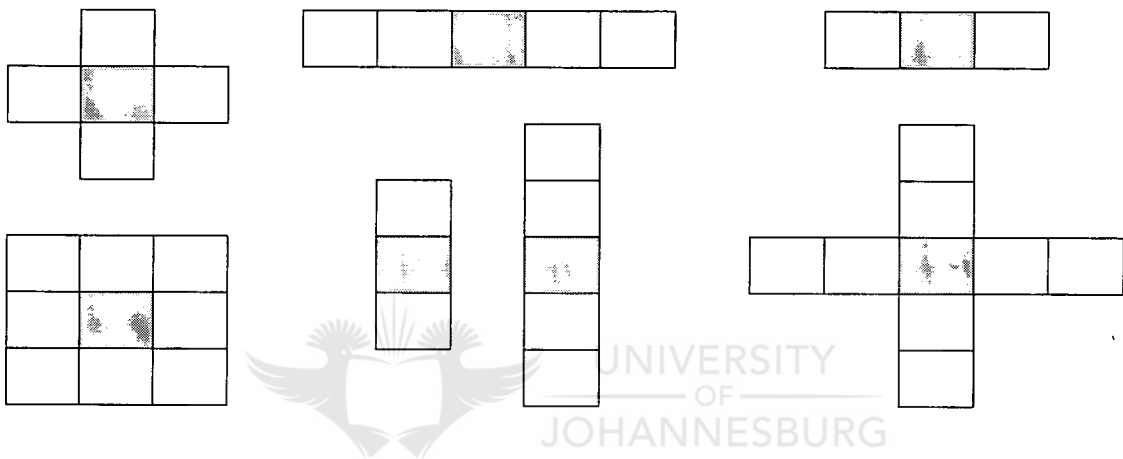


Figure 2-13: Examples of different window shapes

The shaded area is the position the window is presently at, around which the grey level ruling is used using the window and the calculated result stored

2.7.2 Moving average filter

The moving average filter simply collects intensity values using the geometric shape of a window and averaging the sample. This process provides for a smoothing effect and suppresses noise. [20]

2.7.3 Low-Pass filtering

An alternative to the average filter is the Low pass filter. With this filter design the frequency response may be forced to zero at the cut-off frequencies, resulting in an ideal low pass filter. However the impulse response of the filter in the ideal case may extend to infinity and thus needs to be approximated for practical use. There are various approximations that may be made, such as just truncating the impulse response or tapering it using a hamming window to reduce the Gibbs phenomenon which one encounters with other signals. [20]

The Butterworth filter is one approximation (out of many) of the ideal low pass filter and has the following transfer function:

$$H_{\text{ker}}(u, v) = \frac{1}{1 + \left(\frac{\sqrt{u^2 + v^2}}{\Omega_c} \right)^{2p}}, \quad (2.16)$$

where Ω_c is the filter cut-off frequency and p specifies the order of the filter. A larger p provides for a greater precision.

The Butterworth filter's impulse response also extends to infinity and so is spatially truncated in order to be implemented. [20]

2.7.4 High-Pass Filtering

High-Pass filtering as one may know is filtering out the lower frequencies of a signal. In spatial filtering of an image it has an effect opposite to Low-Pass filtering in which edges of objects become more distinct rather than smoothed. Unfortunately high pass filtering has the tendency of introducing high frequency noise. [20] [27]



2.7.5 Image restoration filtering methods

There are a variety of filtering techniques in the realm of image processing and restoration which tries to address the effects of the PSF in (2.2) such as direct filtering, Wiener filtering, geometric mean filtering, constrained least squares, Lucy Richardson and others. [20]

The direct filtering method simply uses the inverse of the OTF (Optical transfer function), however the inverse does not necessarily exist and thus the reasoning for the other techniques. The Wiener filter is a popular choice but while it is able to suppress noise it only removes a small amount of blur. The biggest hurdle of these techniques is that they require knowledge of the PSF, something which in many applications is unknown.

Blind deconvolution is a filtering method where the PSF is unknown.

Due to this ill-posed nature of image restoration, regularization theory is often used and most restoration methods fall under this framework. Regularization can be divided into two categories: regularization in functional spaces (using norms), and control of dimensionality. Regularization Theory accepts that a unique, true solution from imperfect data is not possible and a model is created which defines a class of acceptable solutions among which one of them must be found. This allows a reasonable solution to be obtained. [4] [6] [40]

2.8 DISCRETE FOURIER TRANSFORM VS. FAST FOURIER TRANSFORM

When one wants to transfer a signal to the frequency domain, whether the signal is time varying or spatial varying and the spatial frequency is required, there are two options that may be considered. When considering the Discrete Fourier Transform (DFT) and the Fast Fourier Transform (FFT), the FFT is more computationally efficient especially for signals with a larger amount of data points. Considering Table 3.3 in chapter 3 of [27] which is re-

illustrated below, it is obviously apparent that we would much rather perform an FFT over a DFT even if the image size is double that performed by the DFT for larger values of K.

To utilize the FFT with oversampling from size M (time domain) to N (frequency domain), we can zero pad the right hand side and bottom of our image of size M in order to make it size N. If this zero padded image is processed using the FFT we obtain the same result as that obtained using the DFT and direct oversampling from M to N.

K	DFT		FFT	
	Number of complex multiplications	Number of Complex additions	Number of complex multiplications	Number of Complex additions
64	4 096	4 032	192	384
128	16 384	16 256	448	896
256	65 536	65 280	1 024	2 048
512	262 144	261 632	2 304	4 608
1024	1 048 576	1 047 552	5 120	10 240
2048	4 194 304	4 192 256	11 264	22 528
4096	16 777 216	16 773 120	24 576	49 152

Table 2-1: DFT computations vs. FFT computations

2.9 GAUSSIAN AND LAPLACIAN PYRAMIDS

Gaussian and Laplacian pyramids are commonly used in video and image compression techniques (as well as wavelet techniques which are similar).

As discussed by Burt et al. in [41], the Laplacian pyramid can be constructed from the Gaussian pyramid. The Gaussian Pyramid is constructed from level 0 to N_{level} as follows:

- The initial image resides at level 0
- The image at level $n_{\text{level}} + 1$, is a low passed filtered image of level n_{level} and down sampled by a factor of 2.

Burt et al. [41] gives the equation for creation of a Gaussian pyramid as follows:

$$G_{level}(m, n) = \sum_{k=-2}^2 \sum_{l=-2}^2 w(k, l).G_{level-1}(2m+k, 2n+l), \quad (2.17)$$

where w is a 5 by 5 low pass filter mask.

Burt et al. [41] stipulates from three constraints that the filter weights should be the following, where ‘ a ’ is chosen to be between 0.4 and 0.6:

$$w(0) = a \qquad w(-1) = w(1) = 0.25 \qquad w(-2) = w(2) = 0.25 - a/2$$

A method in expanding a level so it is the same size as the level below it is described by:

$$G_{level\ expanded}(m, n) = 4 \sum_{k=-2}^2 \sum_{l=-2}^2 w(k, l).G_{level}[(m+k)/2, (n+l)/2]. \quad (2.18)$$

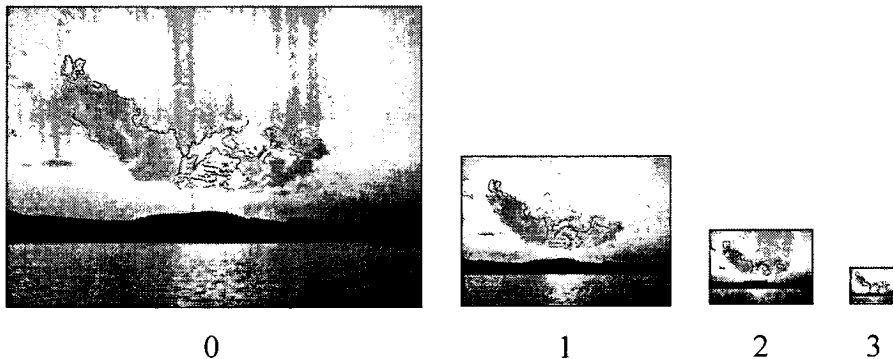


Figure 2-14: Gaussian Pyramid for level 0 to 3

The Laplacian Pyramid is constructed by taking the Gaussian image at level n_{level} and subtracted the expanded Gaussian image at level $n_{level} + 1$. The last level image of the Laplacian pyramid is the last level image of the Gaussian pyramid.

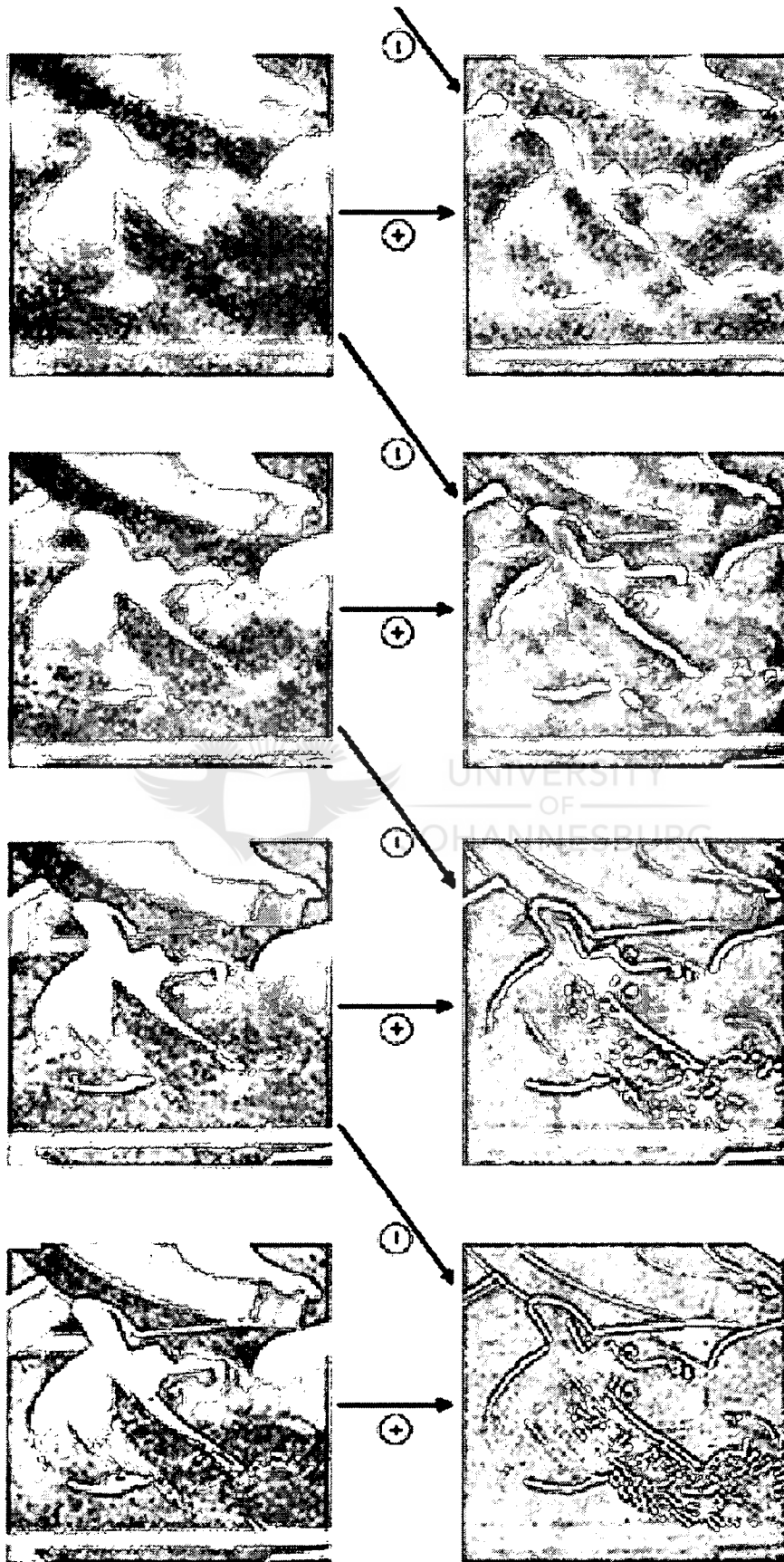


Figure 2-15: Expanded Laplacian Pyramid created from expanded Gaussian Pyramid

Image compression uses Laplacian pyramids because the original image (Gaussian pyramid level 0) can be created by expanding the image at the Laplacian pyramid level $n_{\text{level}} + 1$ using the Gaussian expand technique discussed above, and adding the expanded image of level $n_{\text{level}} + 1$ to level n_{level} . The process continues till level 1 is expanded and added to level 0. The resulting image is the original image at level 0 in the Gaussian pyramid.

The following figure demonstrates how the Laplacian pyramid is constructed from the Gaussian pyramid. Notice how the levels of the Laplacian pyramid contains the high frequency data which was filtered out of the Gaussian pyramid levels. The lowest level of the Laplacian pyramid shows the edges of the image. This figure was extracted from [41].

2.10 FILTERING ELECTROMAGNETIC WAVES

Image processing is made easier and can have greater success when surrounding conditions such as lighting is set to levels which image processing algorithms are developed for. This is true for many other circumstances other than just image processing and is a logical assessment. When conditions sway from those which are ideal, the system and in our discussion, the image processing algorithms need to be robust to handle the change.

This section provides an overview of electromagnetic waves and the filtering of such waves for the suppression of atmospheric turbulence effects, or image contrast enhancement before the light travelling from the object of interest is recorded by the CCD of the digital camera. One may say that this section discusses some *pre-processing* optical filtering techniques in an attempt to enhance atmospheric turbulence degraded images and set better conditions for the image processing algorithms.

2.10.1 Electromagnetic waves

Electromagnetic waves consist of an oscillating electric field perpendicular to the wave propagation (direction electromagnetic wave is travelling), and an oscillating magnetic field perpendicular to the electric field and wave propagation. The magnetic field and electric field both have the same frequency. [42]

Figure 2-16 illustrates an electromagnetic wave created from an oscillating charge along the y-axis. The electric field (\mathbf{E}) resides in the plane of the y-axis and the perpendicular magnetic field (\mathbf{B}) resides in the plane of z-axis with the propagation direction of the electromagnetic wave along the x-axis. The oscillating charge in the figure is referred to as the wave source. Since all directions of vibration may be possible for a wave source, the resulting electromagnetic wave is a combination (superposition) of waves from the vibrations in different directions. [51]

In a vacuum the propagation speed of the electromagnetic wave is 299 792 458 m/s. (the speed of light). In fact as discussed in the next section, light and colour as we generally know it, is a frequency band of the electromagnetic spectrum.

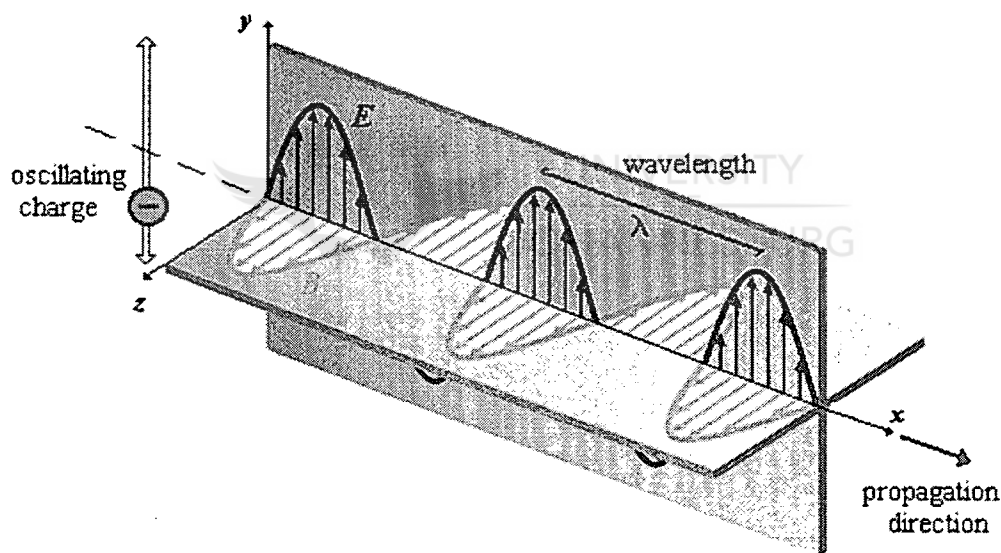


Figure 2-16: Electromagnetic wave.

(Illustration from [43])

The electromagnetic waves carry energy and that energy is absorbed by objects that it encounters. Energy from higher frequency electromagnetic waves are more readily absorbed by objects than lower frequency waves. [42] [52]

Electromagnetic waves are also referred to as electromagnetic radiation.

2.10.2 Polarization of electromagnetic waves

As discussed in section 2.10.1, an electromagnetic wave is a resultant of a superposition of electromagnetic waves of a vibrating wave source. This electromagnetic wave in the context of light is referred to as an unpolarized light beam. [51]

The direction of polarization for an individual wave is classified as the direction of oscillation of the electric field. An unpolarized light beam has many directions of polarizations. An electromagnetic wave is termed to be linearly polarized, plane polarized, or just polarized if it has only one direction of oscillation of the electric field at all points. [51]

Light can be polarized through the selective absorption of electromagnetic wave directions, reflection, double refraction, or scattering. The selective absorption by a material is the most common in polarizing light. [51]

An incident beam can be polarised by reflection when the condition:

$$\frac{nr_2}{nr_1} = (\sin \theta_1) / (\sin \theta_2) = \tan \theta_1, \quad (2.19)$$

where θ_1 is the angle between the incident light and the normal and θ_2 is the angle between the refracted light beam and the normal. An angle of 90 degrees separates the reflected and refracted beam as illustrated in figure 2-17. The reflected beam is completely polarised, as is the refracted. [51]

If the surface off of which the reflected beam originates is horizontal, then the reflected beam will be horizontally polarized and the refracted beam vertically polarized. This polarization is commonly found in nature with light reflecting off water. Polarized sunglasses remove this reflected glare by only allowing vertically polarized light through.

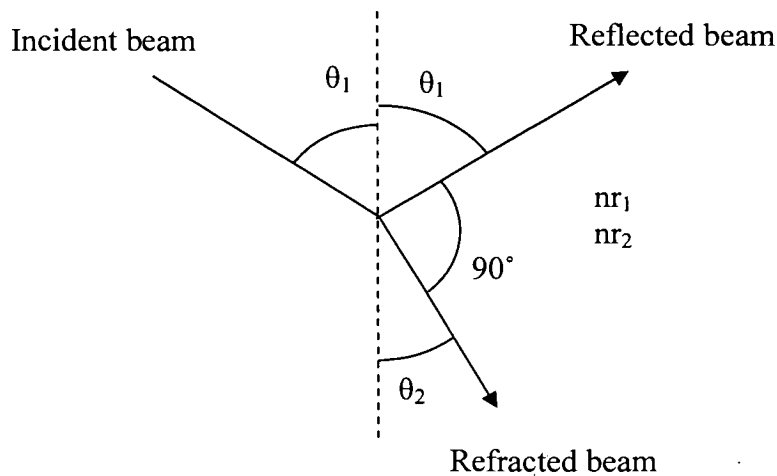


Figure 2-17: Electromagnetic wave polarization by reflection.

Double refraction polarization is a result from light entering substances such as a quartz or calcite crystal. Such substances are characterised by two indices of refraction and are termed as double-refracting or birefringent. When light enters these substances it is separated into two plane, polarized light waves which travel at differing velocities through the substance. [51]

Polarization by scattering is a phenomenon which is observed every day. The scattering of light by gas molecules was mentioned in earlier, and now proceeds with a deeper discussion in the context of polarization.

Polarization by scattering is a result of a substance's electrons reradiating a portion of incident light, with the angle of incident light perpendicular to the observer of the reradiated light. This scattering effect accounts for the observation of blue light when looking directly up at the sky. Gas molecules that make up air reradiate the incident light to the observer on the ground. The polarization by scattering is illustrated in figure 2-18.

Light of various wavelengths λ which is incident on gas molecules with a diameter d (such as Oxygen and Nitrogen) with $d \ll \lambda$, reradiates light waves and results in the

phenomenon of blue sky and red coloured sky during sunsets. The phenomenon is due from the fact that the relative intensity of the scattered light varies as $1/\lambda^4$, and shorter wavelengths of light (such as blue light) is scattered more efficiently than longer wavelengths (such as red light). [51]

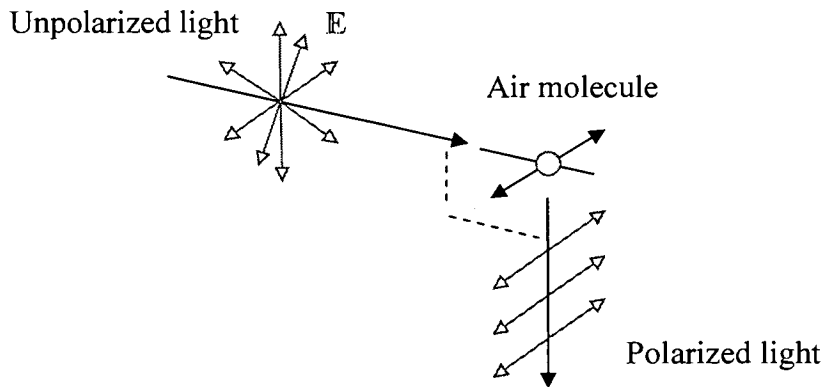


Figure 2-18: Polarization by scattering.

2.10.3 The electromagnetic spectrum



As one may expect, the electromagnetic wave may have various frequencies and is determined by the frequency at which the charge (which sets up the electric and magnetic field) oscillates. Frequencies can range from as little as a few Hz through the kHz, MHz, GHz, THz (Terahertz), PHz (Petahertz) and EHz (Exahertz) ranges.

The electromagnetic spectrum shows the possible frequency range of electromagnetic waves with wavelengths of thousands of kilometres down to those smaller than the width of an atom. The visible light spectrum which our eyes can detect forms only a small part of the electromagnetic spectrum.

The visible light spectrum is bordered by ultraviolet light at the high frequency side and infrared light at the low frequency side (these reside outside of the visible light spectrum). In order from highest frequency to lowest frequency we have blue, green and red coloured light.

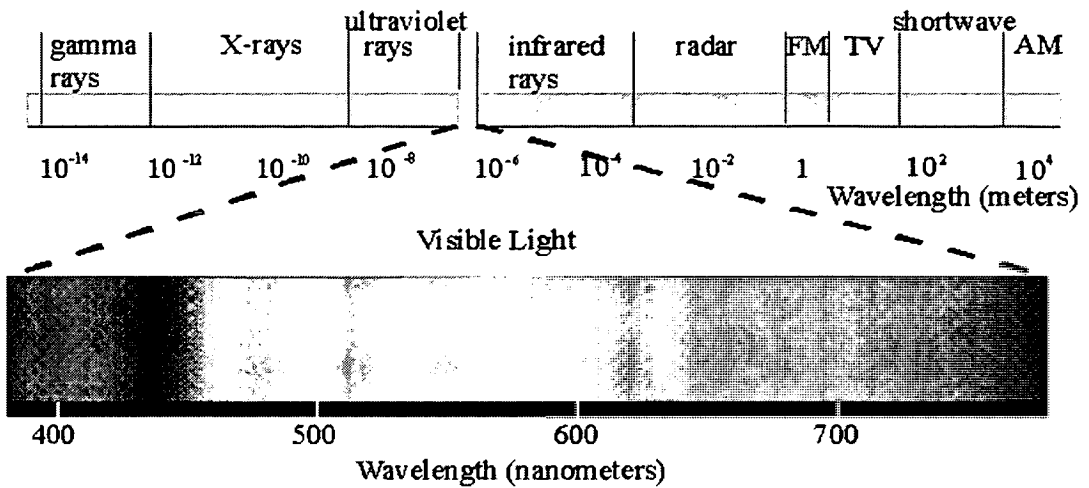


Figure 2-19: Electromagnetic spectrum.

2.10.4 Refraction of electromagnetic waves

Electromagnetic waves are refracted when passing from one density to another. A typical example is the glass prism and a beam of white light entering the prism as illustrated in Figure 2-20.

The white light is composed of all colours of the visible light spectrum. When the light enters and passes through the prism, the electromagnetic waves are refracted at different angles. Red light being refracted the least and blue and violet light are refracted by the largest amount.

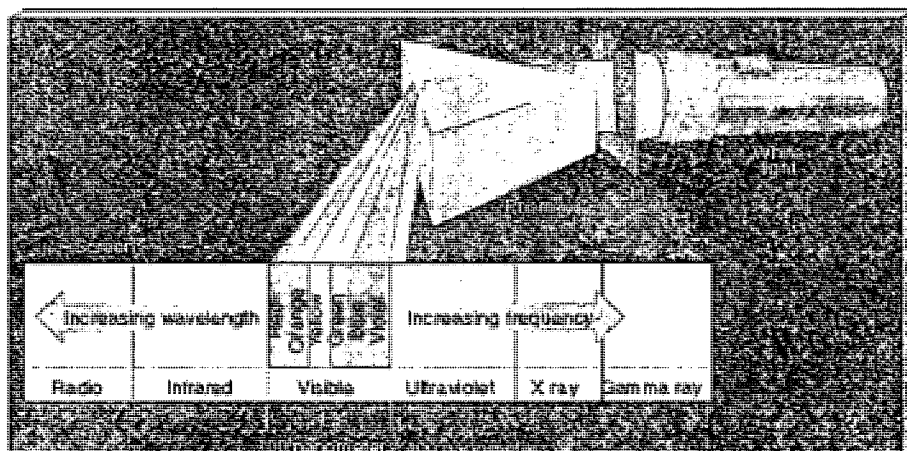


Figure 2-20: Refraction of white light.

(Illustration from [44])

From this observation, higher frequency electromagnetic waves are refracted a larger amount than the smaller frequencies. In [46], Okayama et al. found that higher frequency light was refracted and dispersed to a greater degree than the lower frequency light when travelling through atmospheric turbulence as well. This coincides with the physics theory that the atmospheric turbulence consists of differing density mixtures of air outlined in section 2.2.

2.10.5 Advantages and disadvantages of colour filtering electromagnetic waves

By colour filtering electromagnetic waves, a portion of the frequency is being removed. In terms of a colour image, contrast may be improved by allowing only a certain frequency band of light to be captured. For example using a red coloured filter so blue objects appear dark.

With higher frequency light being refracted to a higher degree than lower frequencies, it may be advantageous to use a coloured filter and allow only the lower frequencies to be captured. In the presence of atmospheric turbulence, a blurry image is captured due to the refraction of light through the atmosphere. Removing some of the overly refracted light may remove some of the blurriness associated with atmospheric turbulence.

The use of filters however, results in information content being removed, and thus that frequency information is lost. The use of filters may also require the use of longer exposure times when capturing images due to filtering out some light intensity. These filters which may require a longer exposure time normally have a filter factor. The filter factor indicates the multiple in which the exposure time will need to be increased. An example is a filter which has a filter factor specified as 3, which may cause an exposure time increase from 1/60 seconds (without filter) to 1/20 seconds (with filter). [47]

A filter may also degrade image quality through the scattering of light (filter has its own PSF). Degradation of image quality may also occur because of dust on the filter and reflections of light off the filter. [47]

2.10.6 Wratten numbers

Optical filters are labelled according a system known as Wratten numbers. Each optical filter is given a number code known as a Wratten number which identifies with the set of frequencies of light it filters.

An example is a filter with a Wratten number of 2A. This filter has a pale yellow colour and used in absorbing ultraviolet radiation. Another filter with a Wratten number of 21 has an orange colour and may be used as a contrast filter for blue and green light absorption.

2.10.7 Optical filters

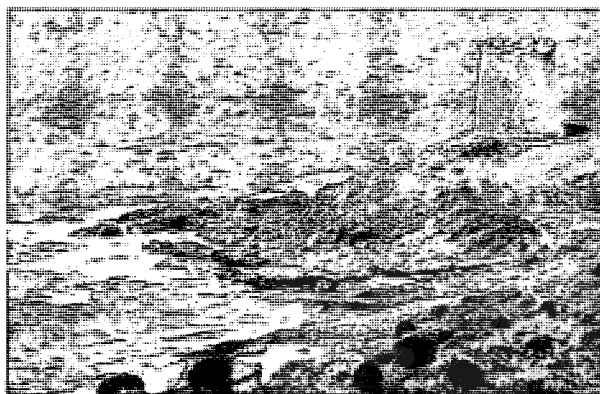


For contrast enhancement, a coloured filter which is orange or red may be used. Possibilities of coloured filters is a Wratten number of 21 or for more intense contrast a Wratten number of 22, 23 or 24 (red coloured filter). An optical filter named by its colour or appearance (e.g. red) is the colour that the filter allows through and not that which it blocks.

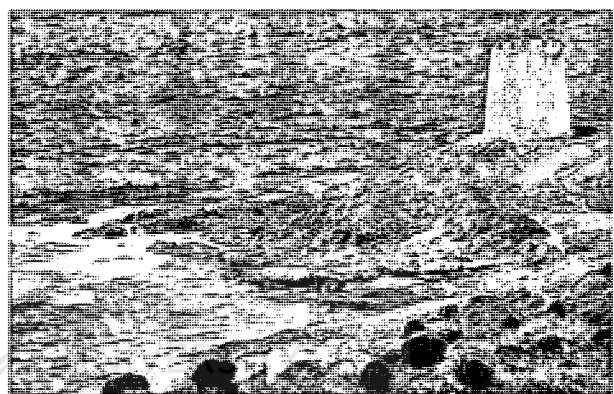
Filters with a Wratten number of 25, 26 or 29 absorb much of the visible light and are generally used for infrared photography.

In some cases, the use of contrast enhancement and colour filters is dependent on the colour of the subject of interest and the colour surrounding the subject (in these cases the image is sometimes recorded in black and white, thus resulting in this context the wording of 'contrast enhancement') [45]. An example is a picture of a red hot air balloon against a blue sky. Instead of converting to black and white straight away, a red filter could be used first to filter out the blue of the sky resulting in a brighter

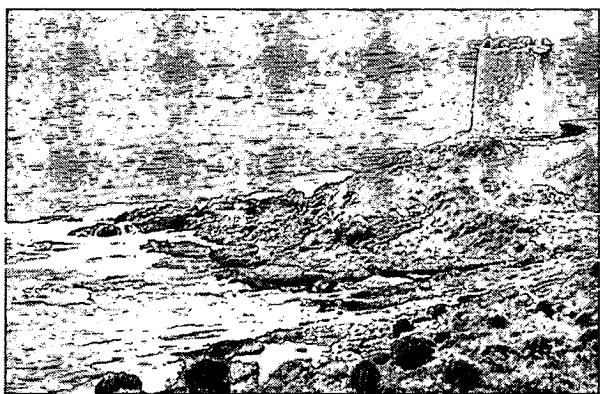
white balloon against a dark sky. If we used a blue filter instead, then the red of the balloon would be filtered out, and will result in a white sky and dark balloon. However as discussed, the use of a blue filter (for example in the red balloon and blue sky) may result in the loss of detail in the hot air balloon if the light from that object has little or no blue component. For this context of contrast enhancement, the coloured filter used is dependent on the subject and environment surrounding the subject. The selection of the best coloured filter is chosen in general to be complimentary to the colour of the region in which contrast is desired to be enhanced before converting to black and white [45].



Unfiltered colour image



Red filter applied



Green filter applied



Blue filter applied

Figure 2-21: Coloured optical filter effects.

(Illustration from [45])

A polarization filter filters light having an oscillating electric field of a specific direction and is primarily used to reduce glare from reflecting sunlight such as that off water. Such polarization filters which are used in photography make use of polarization by absorption.

Filters can be purchased from Celestron, Meade and other telescope and telescope accessory suppliers which fit between the eyepiece and the eyepiece holder (ocular) of the telescope. Filters are available for a range of eyepiece sizes.

2.11 ALTERNATIVE APPROACHES

Within the Astronomy field, Atmospheric turbulence also causes a problem by creating temporally shifting blurs. Some larger telescopes address the problem through the use of deformable mirrors in adaptive optics as Cohen et al. discusses in [59]. Ardeberg et al. in [60] explains the different optics used with larger telescopes. He describes passive, active, live and adaptive optics, where live optics can correct for high frequency errors due to wind, and adaptive optics consists of one or more deformable mirrors which are able to correct atmospheric turbulence errors.

Figure 2-22 shows an example of the use of adaptive optics. In this example an LC SLM (Liquid-Crystal Spatial Light Modulator) is used to correct for the phase distorted wavefront and controlled via a VLSI (Very Large Scale Integration) controller. Figure 2-23 illustrates the application of a deformable mirror to correct for the phase distortion; these adaptive mirrors are made up of an array of smaller deformable mirrors, such as a MEMS (Micro-Electro-Mechanical Systems) mirror, and may vary in size between different manufacturers.

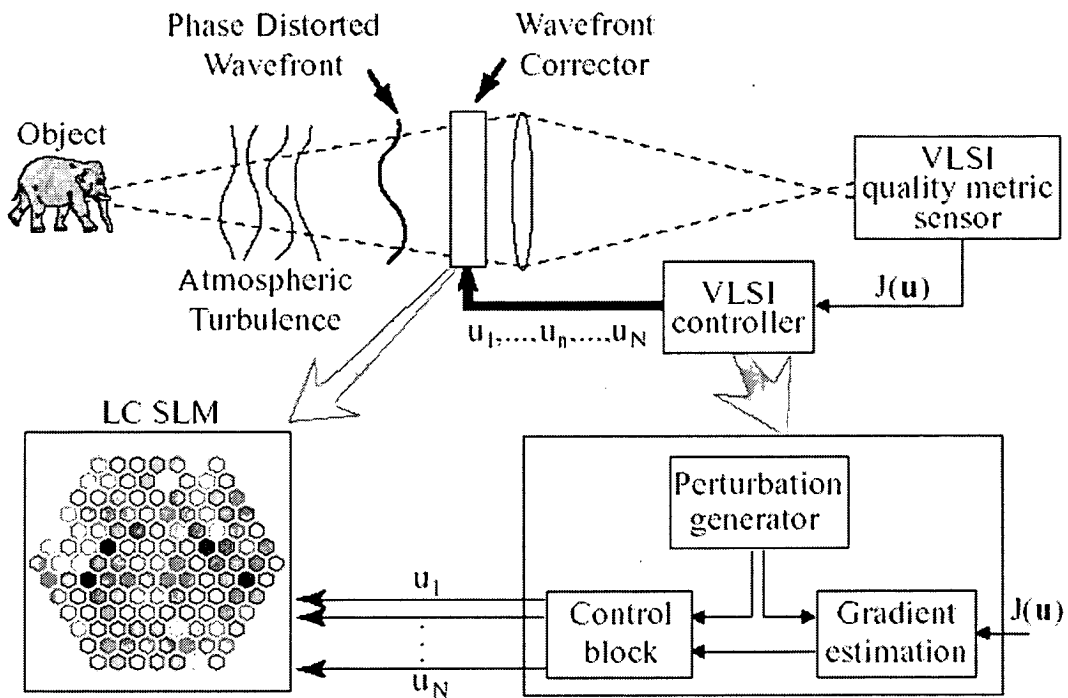


Figure 2-22: Alternative approach using adaptive phase correcting device (LC SLM).

(Image from [59])

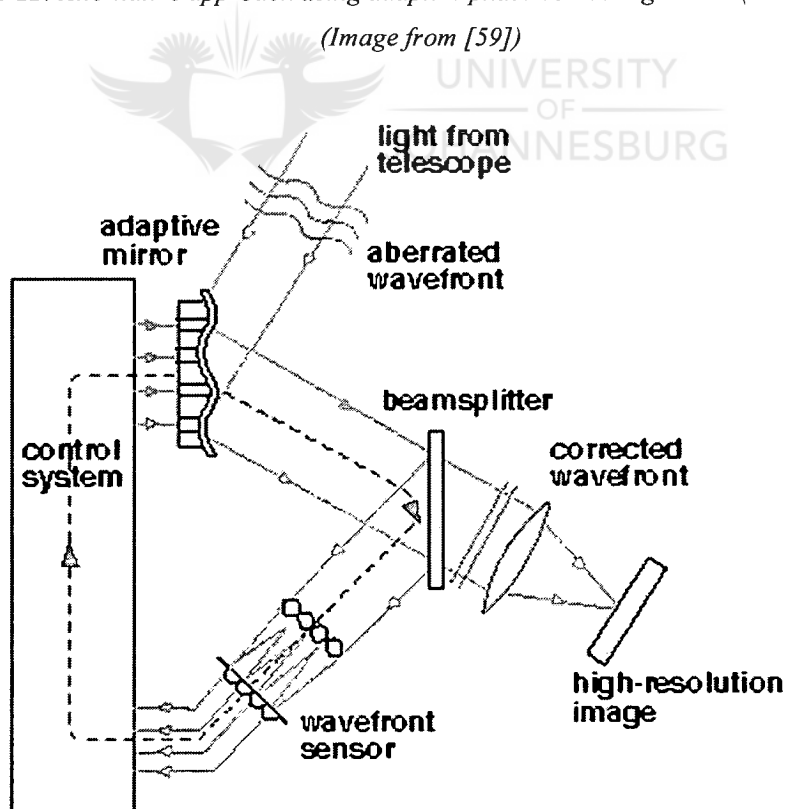


Figure 2-23: Alternative approach using adaptive phase correcting device (MEMS mirror).

(Image from [61])

The development of a greater sophistication in wavefront sensors, adaptive optics capabilities, and smaller deformable mirrors such as the MEMS devices, resulting in two to three fold gains in resolution, has resulted in an increased interest within research groups [61].

"While a large portion of our efforts are focused on astronomy and defence research, a growing portion of development is now aimed at serving different areas such as ophthalmology, laser-based telecom, optical metrology, and confocal microscopy," says Hapet Berberian, industry consultant at AOA. [61]

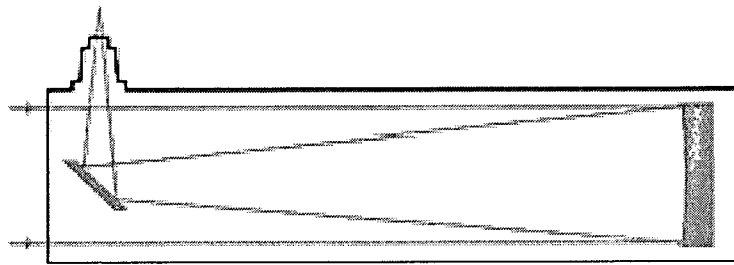
2.12 EQUIPMENT

The equipment used for capturing digital video footage containing distortions caused by atmospheric turbulence is an important topic. Without the right equipment the captured video may be inadequate and this will affect the testing of the algorithms and the discovery of their capabilities.

In some previous and current studies [86] [62] the equipment used for capturing the atmospheric turbulence on video consisted of a Telescope coupled to a digital video camera with a sufficiently low exposure time of 10 ms or less to "freeze" the atmospheric turbulence. The low exposure time is required so as to minimize the blurring caused by the atmospheric turbulence.

Most telescopes have a 1.25" standard eyepiece diameter size (American design) with Japanese designed eyepieces having a 0.965" standard. With the American design, the interchangeable 1.25" eyepiece slides into the telescopes 1.25" eyepiece tube. The magnification can be changed by replacing the eyepiece with one of a different focal length. The magnification that is created can be computed by taking the telescopes focal length and dividing that by the eyepiece value (e.g. a 2000mm focal length for the telescope using a 10mm valued eyepiece will give a 200x optical magnification). The focal length of a telescope can be calculated from its aperture and focal ratio by multiplying the two (e.g. the focal length of an 8 inch (200mm objective mirror or lens) clear aperture telescope with a

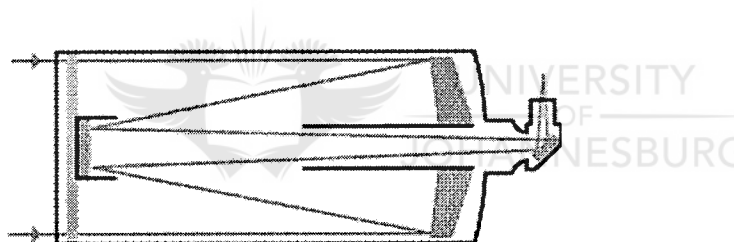
f/10 focal ratio is 2000mm). The greater the focal ratio (or sometimes referred to as the f number) the lesser the amount of light there is that reaches the focal plane (eyepiece), measuring brightness per unit area of the image. [63]



(a) Newtonian Reflector



(b) Refractor



(c) Catadioptrics

Figure 2-24: Comparisons of telescope designs.

The green line represents the travelling path of light, purple objects represent mirrors and blue objects represent lenses. (Images from [64])

There are various types of telescope designs as shown in Figure 2-24. Newtonian reflectors are generally inexpensive but are large in diameter, are an open ended tube design, sensitive to being bumped and require collimation of the optics now and then (realignment of optics). The Newtonian designs however can offer low focal ratios and collect a lot of light.

Refractors are much more expensive than Newtonian reflectors with a smaller but longer tube. The refractors can offer excellent views and are robust without the need for collimation.



Catadioptric designs are generally priced between the Newtonian reflector and refractor telescopes. The Catadioptrics are fairly robust and require collimation only occasionally, are a closed tube design like the refractors, but are broader in diameter and shorter. They have higher focal ratios but offer greater portability. Two common Catadioptrics are Schmidt-Cassegrains and Maksutov-Cassegrains.

2.12.1 Telescopes and lenses

In [65] they give the specifications of a multi-focal telescopic lens which can be attached to the “c” thread of a surveillance video camera. This telescopic lens, QUESTAR MFL 90 (see Figure 2-25), allows the user to select different focal lengths thus allowing for the selection of different magnifications. The different focal lengths which can be selected range between 480mm and 3100mm and is shown in table 2-2. It has an approximate aperture of 3.5”, the unit is 35cm in length, 5.4kg in weight (excludes any mount) and weather resistant. [65]



Figure 2-25: The QUESTAR MFL 90 Multi Focal Length Lens.

(Taken from [65])

Focal Length (mm)	Focal Ratio
480	f/5.4
660	f/7.4
820	f/9.2
955	f/10.7
1280	f/14.6
2230	f/25.0
3100	f/34.8

Table 2-2: The QUESTAR MFL 90 Multi Focal Length Lens focal lengths and ratios.
(Taken from [65])

The telescope used in [86] is a Celestron 8" Schmidt-Cassegrain.

The Celestron Nexstar 8 SE (Figure 2-26) is a high quality 8" Schmidt-Cassegrain, with a 2032mm focal length with a focal ratio of f/10. The telescope is just less than 11 Kg with a 4 Kg motorised mount (able to operate on 8 AA batteries) [66]. The smaller 6" sibling, the Nexstar 6 SE has a 1500mm focal length with a focal ratio of f/10. The Nexstar uses the 1.25" eyepiece standard.

The Celestron Nexstar 8 SE, and the smaller Celestron Nexstar 6 SE at the time of this writing are priced around R19 000 – R20 000 and R13 000 respectively.

Meade offer a 5" aperture Maksutov-Cassegrain telescope, the ETX 125PE (see Figure 2-27). The ETX 125PE has a focal length of 1900mm and a focal ratio of f/15. The higher focal ratio means that the image will be dimmer than that of the Celestron Nexstar. The ETX series also comes equipped with a motorised mount. The telescope weighs just less than 7 Kg with an 11.4 Kg shipping weight. [67]

At the time of this writing, the Meade ETX 125PE cost around R15 000.

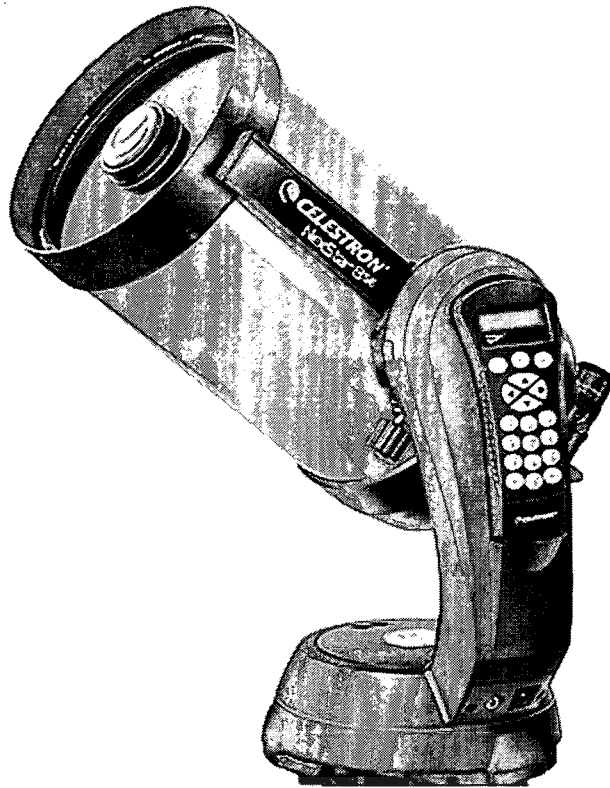


Figure 2-26: The Celestron Nexstar 8 SE telescope.

(Image from [66])

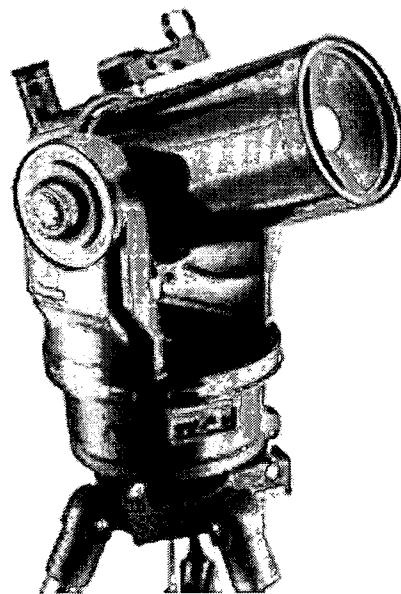


Figure 2-27: The Meade ETX 125PE telescope.

(Image from [67])

Meade also offers an 8" Schmidt-Cassegrain, the 8" LX90 GPS (see figure 2-28). The 8" LX90GPS has a focal length of 2000mm with a focal ratio of f/10. The telescope also has a motorised mount and additionally GPS capabilities to determine its true location and time. The telescope weighs an approximate 15 Kg and the motorised mount 8.5 Kg. The motorised mount can use 8 C sized batteries. [68]

At the time of this writing the Meade 8" LX90 GPS cost around R27 000.



Figure 2-28: The Meade 8" LX90GPS telescope.
(Image from [68])

2.12.1.1 Telescope Accessories

One can purchase adapters for the telescope that fit into the 1.25" ocular and allows a T-thread camera to be attached. There are also adapters available such as that sold by Astrovid which allows the connection of a C or CS thread video camera. In both cases the adapters replace the 1.25" eyepiece.

Barlow lenses are sold by many brands such as Meade, Celestron, Zhumell and others. A Barlow lens fits between the 1.25" ocular of the telescope and the 1.25" eyepiece or other adapter. The 2x Barlow lens increases the magnification by two while a 3x increases the magnification by three. There are also Barlow lenses which have both the 2x and 3x capabilities which are selectable by the user.

2.12.2 Cameras

There are many different types of digital video cameras on the market. Camcorders are obviously the most common in general usage and have become quite good in their quality. Surveillance cameras and modified Web cameras seem to be the more popular choice when it comes to coupling a video camera to a telescope [69].

A video surveillance camera which is used within the University of Johannesburg Electrical Engineering department is the Arecont Vision AV3100 (Figure 2-29). The AV3100 is a 3 mega pixel camera which uses C threaded lenses. It has a 1/2" CCD and is capable of 20 fps at a resolution of 1920x1080. The Arecont AV3100 uses Ethernet over which the motion jpg format video is transmitted. The camera weighs 450g without the lens. [70]

PixeLINK offer a 1.3 megapixel camera, the PL-A741 (see Figure 2-30), with a 2/3" optics format and c-mount (lens not included). The camera is monochrome with a FireWire interface and capable of 27 fps in a 1280x1024 resolution, 33fps at 1000x1000, 107 fps at a VGA resolution of 640x480, and 8000 fps at 427x8. All the resolutions can be defined for a selectable region of interest. Optionally supplied is a developer's kit which consists of the API (application programming interface) and GUI (graphical user interface). The camera allows the user to control the shutter (rolling or synchronous), exposure time (0.04ms to 1s), gain and brightness (black level), frame rate (from 2 fps upwards), region of interest, 8bit or 10bit pixel format, and others. The dimensions of the camera are 35mm x 50mm x 100mm and weighs at 160g (without lens). [71]

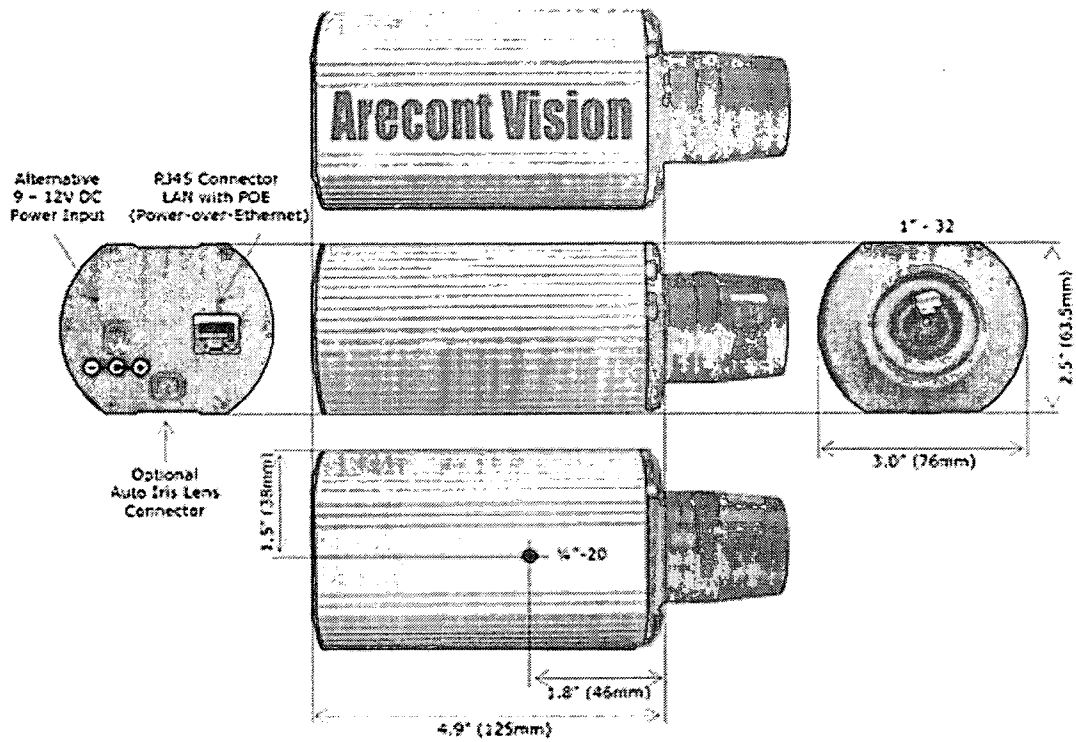


Figure 2-29: Arecont Vision AV3100 Surveillance Camera.

(Image from [70])

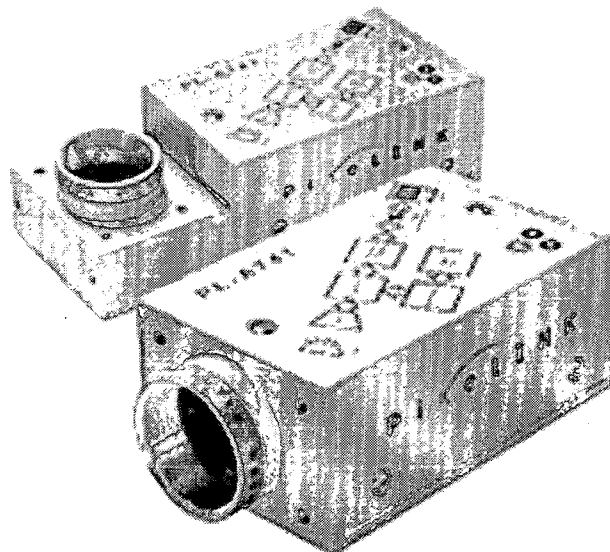


Figure 2-30: PixeLINK PL-A741 industrial and machine vision camera.

(Image from [71])

At time of this writing a quotation was received which priced the PL-A741 at about R15 900. One minor problem which may arise within portability (such as using a laptop for image capturing or processing) is that the camera uses FireWire and not many laptops have FireWire ports (other than Apple) without having to use a PCMCIA FireWire card.

An alternative to the PixeLINK PL-A741 may be the usb 2.0 interfaced PL-B741U with similar specifications to the PL-A741 but slightly slower frame rates with 24FPS at 1280x1024, 30 fps at 1000x1000, 96 at 640x480 and up to 7200 fps at 724x8.

For a colour version equivalent to the PL-A741 there is the PL-A742.

PixeLINK Capture OEM is a software utility which PixeLINK provide for free download off their website and acts as a configuration utility which provides access to all the cameras features.

In [86], a Qimaging Retiga 1300 camera is used which is a c-mount camera with a 2/3" inch optical format. The camera has a specific application to be used with microscopes. It has a 12bit CCD and capable of 1300x1300 resolution images at 10 fps. The camera has a FireWire interface and at the time of this writing the Qimaging Retiga 1300 had an estimated cost of \$6950.00 (direct conversion – R50 000).

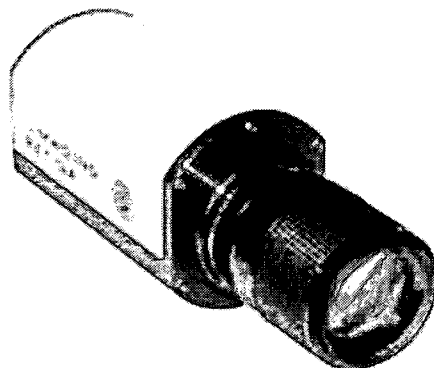


Figure 2-31: Qimaging Retiga 1300 CCD camera.

2.13 COMPARISON AND TESTING METHODS

The testing and comparisons of algorithms and their resultant image quality can be difficult with MSE (mean square error) and PSNR (peak signal to noise ratio) tests possibly giving conflicting results to subjective testing results [72]. The testing becomes a greater problem when considering that one may be dealing with multiple images and the testing and comparison of a video as a whole.

In a previous study [6], the aberration produced by a video sequence may be measured by the method of maintaining maximum pixel intensities across the image sequence. The method uses a checkerboard pattern object with equally sized black and white blocks as the object viewed through the turbulent atmosphere. The maximum pixel intensities are recorded from the image sequence to produce the single image containing maximum values. From this grayscale image the white blocks will appear fatter than the black blocks and if one plotted a profile across the pattern, peaks will be broader than the valleys allowing for the measurement of the aberration.

In [86] Carrano uses a method which measures the modulation of different spatial frequencies. This method allows the determination of how visible edges of objects will be with higher modulations representing sharper edges and higher frequencies representing finer mesh objects with steeper edge gradients.

In the test, various greyscale versions of sinusoids of differing frequencies of interest are printed and viewed through turbulent atmosphere. The turbulent atmosphere blurs the image and reduces higher frequency content. The modulations of the frequencies are measured as:

$$\text{Modulation} = \left(\frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}} \right), \quad (2.20)$$

where I_{\max} is the maximum sinusoid intensity value and I_{\min} the minimum intensity value of a specific frequency.

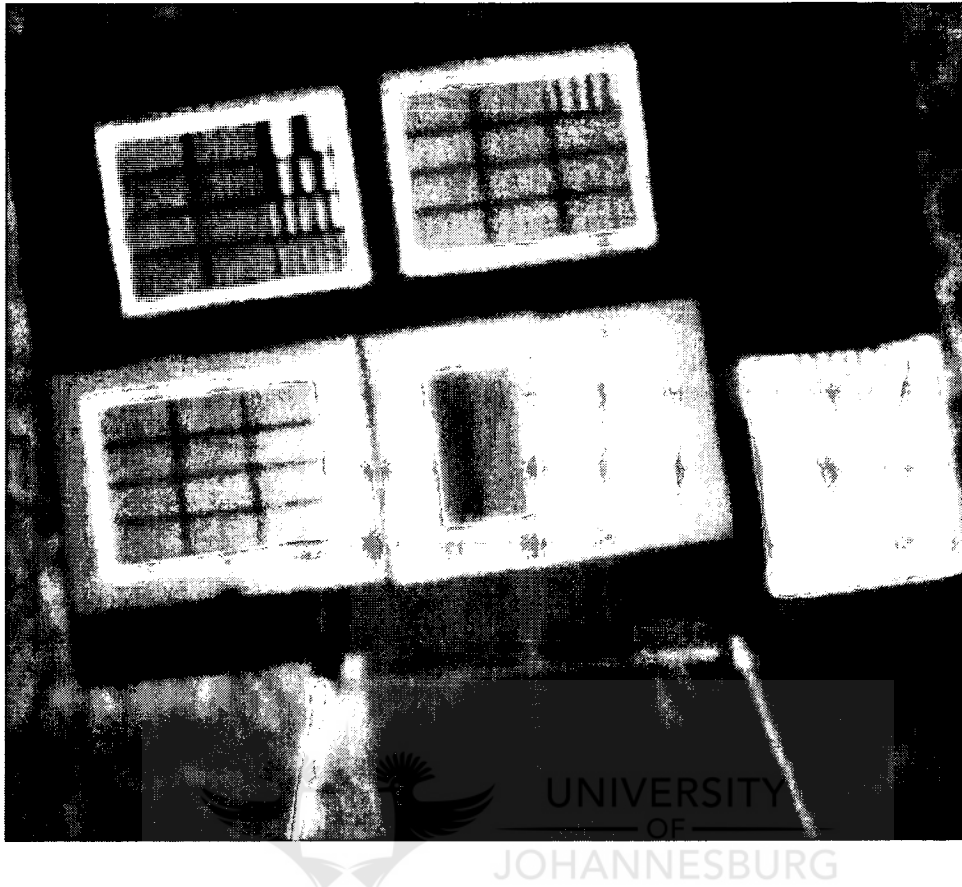


Figure 2-32: Frequency Modulation test setup.

(From [86])

Kopriva et al. in [92] propose the use of the Laplacian operator as a metric for determining image sharpness. The image sharpening metric is used to correspond a sharper image to one which contains a smaller amount of atmospheric turbulence effects.

The Laplacian operator approximates the linear second order derivative of an image $f(m,n)$ in the row and column directions (m and n),

$$S(m,n) = \nabla^2 f(m,n) \equiv \frac{\partial^2 f(m,n)}{\partial m^2} + \frac{\partial^2 f(m,n)}{\partial n^2}, \quad (2.21)$$

with the image sharpening metric defined as:

$$S_4 = \frac{1}{MN} \sum_m \sum_n |S(m, n)|, \quad (2.22)$$

where the image is of size $M \times N$.

A metric which is commonly used together with data and image compression is the MSE (mean square error). A higher number from a MSE test indicates more errors between the test data and base data. The MSE is defined as:

$$\text{MSE}(\mathbf{i}_d) = \frac{1}{MN} \sum_{m,n} (\mathbf{i}_d - \mathbf{i}_i)^2, \quad (2.23)$$

where, \mathbf{i}_d is the estimate or degraded image of the undistorted image, \mathbf{i}_i , of size $M \times N$. The RMSE (root mean square error) is simply the square root of the MSE.

As stated in [72] and [73], MSE or RMSE results can be contradictory to subjective results. In [73], Teo et al. states that “It has long been accepted that MSE (or RMSE) is inaccurate in predicting perceived distortion”. An example is shown in Figure 2-33 to illustrate this discussion. In the figure, the top two images were created by adding different types of distortions to the original. The resulting RMSE values are shown below the two images.

If one were to use the RMSE value as a selection for the image with least distortion then the first image with the RMSE value of 8.5 would be chosen over the image with the RMSE value of 9. This suggests that the first image is of better quality than the second.

The PSNR (peak signal to noise ratio) of an 8 bit grey level image is defined as:

$$\text{PSNR}(\mathbf{i}_d) = 20 \log_{10} \left(\frac{255}{\text{RMSE}(\mathbf{i}_d)} \right), \quad (2.24)$$

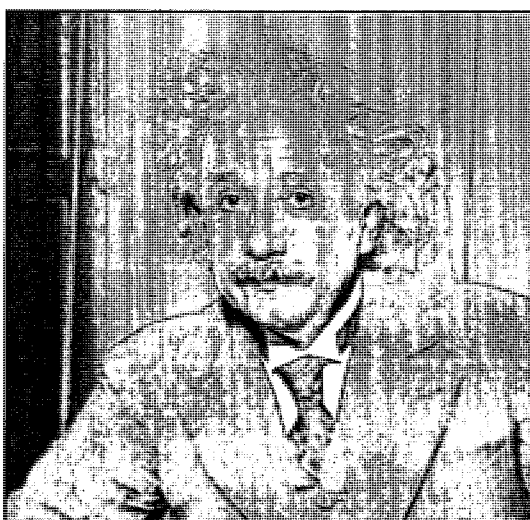
and being based off the RMSE, it faces the same problem as that illustrated in Figure 2-33.



RMSE = 8.5



RMSE = 9.0



Original Image

*Figure 2-33: Illustration of RMSE downfall.
(Illustration from [73])*



2.14 CONCLUSION

In this chapter the relevant literature study was undertaken which touches on some important methods used in image processing, identifies some atmospheric turbulence suppression methods, some applicable equipment that may be used in the project study and experimentation and testing, as well as some metrics which are applicable to atmospheric turbulence degraded video.

The next chapter presents a number of algorithms that have been developed by other authors to address the imaging through atmospheric turbulence problem.



3 CHAPTER 3: ALGORITHM OVERVIEW

3.1 INTRODUCTION AND OVERVIEW

In this chapter, various atmospheric turbulence suppression algorithms which have been developed and documented by other authors over the past years are reviewed. These algorithms attempt to suppress the distortion created within images by turbulent atmosphere describe in the literature study.

Some of these algorithms are selected for implementation so that they may undergo analysis and experimental testing, against a new atmospheric suppression algorithm proposed by the author.

3.2 DFT ANALYSIS METHOD:

In [2], Frieden proposes a linear solution to the problem of imaging through atmospheric turbulence which seems simplistic in its approach although it could be restrictive to applicable image sizes.

The method below is summarized from [2].

The method starts with a look at the M point DFT (Discrete Fourier Transformation) and takes into consideration that the image can be over sampled rather than at just the Nyquist frequency (twice the maximum frequency):

$$I(\omega) = \sum_{m=0}^{M-1} \Delta x \cdot i(m\Delta x) e^{-j\omega m\Delta x}, \quad -\Omega_c \leq \omega \leq \Omega_c, \quad (3.1)$$

where $\Delta x = \pi / \Omega_c$ and Ω_c the spectrum cut-off frequency.

The image spectrums are modelled in the following way:

$$\mathbf{I}_d^{(i)} = \mathbf{H}_n^{(i)} \mathbf{I}_n, i = 1,2; n = 0,1,2,\dots,N-1. \quad (3.2)$$

\mathbf{I}_n is the observed image spectrum, \mathbf{I}_n the object spectrum and \mathbf{H}_n the random optical transfer function and N the total number of samples (frequency domain samples).

By using the above equation and dividing the one spectrum by the other, the Object spectrum O_n cancels and results in the ratio below:

$$\mathbf{D}_n \equiv \frac{\mathbf{I}_d^{(1)}}{\mathbf{I}_d^{(2)}} = \frac{\mathbf{H}_n^{(1)} \mathbf{I}_n}{\mathbf{H}_n^{(2)} \mathbf{I}_n} = \frac{\mathbf{H}_n^{(1)}}{\mathbf{H}_n^{(2)}}, n=0, 1, \dots, N-1. \quad (3.3)$$

Representing $\mathbf{H}_n^{(i)}$ ($i = 1,2$) in their DFT form we have:

$$\mathbf{D}_n = \frac{\sum_{m=0}^{M-1} \mathbf{h}_m^{(1)} e^{-2\pi j m n / N}}{\sum_{m=0}^{M-1} \mathbf{h}_m^{(2)} e^{-2\pi j m n / N}}, n = 0, 1, \dots, N-1. \quad (3.4)$$

The \mathbf{h}_m^1 and \mathbf{h}_m^2 functions are the PSF (Point Spread Functions) relating to \mathbf{H}_n^1 and \mathbf{H}_n^2 respectively.

Recapping, \mathbf{D}_n is found from the ratio of the frequency domain representation of the two distorted images and can be represented as a ratio with the two unknowns \mathbf{h}_m^1 and \mathbf{h}_m^2 which we want to find. \mathbf{D}_n can also be represented as the following since it is a complex number:

$$\mathbf{D}_n = \mathbf{M}_n e^{j\omega}, n = 0, 1, 2, \dots, N-1, \text{ where } \omega = 2\pi\Phi_n/N. \quad (3.5)$$

\mathbf{M}_n is the magnitude of \mathbf{D}_n and ω_n the phase angle. The $2\pi/N$ factor in ω_n is simply there for convenience so it can be factored out later but is not necessary.

The DFT analysis continues and breaks the equation into a set of two homogeneous equations, one for the real part and one for the complex part.

$$\begin{aligned} \text{Re part : } & \sum_m \mathbf{h}_m^{(1)} \cos\left(\frac{2\pi mn}{N}\right) \\ & - \sum_m \mathbf{M}_n \mathbf{h}_m^{(2)} \cos\left[\frac{2\pi}{N}(\varphi_n - mn)\right] = 0, \end{aligned} \quad (3.6)$$

$$\begin{aligned} \text{Im part : } & \sum_m \mathbf{h}_m^{(1)} \sin\left(\frac{2\pi mn}{N}\right) \\ & + \sum_m \mathbf{M}_n \mathbf{h}_m^{(2)} \sin\left[\frac{2\pi}{N}(\varphi_n - mn)\right] = 0. \end{aligned} \quad (3.7)$$

In order to obtain a better estimate of each PSF in their ratio, the homogeneous equations are made inhomogeneous, obtaining:

$$\begin{aligned} & \sum_{m \neq [M/2]} \mathbf{h}_m^{(1)} \cos\left(\frac{2\pi mn}{N}\right) - \sum_m \mathbf{M}_n \mathbf{h}_m^{(2)} \cos\left[\frac{2\pi}{N}(\varphi_n - mn)\right] \\ & = -\cos\left(\frac{2\pi[M/2]n}{N}\right), \text{ for } n = 0, 1, \dots, N-1, \end{aligned} \quad (3.8)$$

$$\begin{aligned} & \sum_{m \neq [M/2]} \mathbf{h}_m^{(1)} \sin\left(\frac{2\pi mn}{N}\right) + \sum_m \mathbf{M}_n \mathbf{h}_m^{(2)} \sin\left[\frac{2\pi}{N}(\varphi_n - mn)\right] \\ & = -\sin\left(\frac{2\pi[M/2]n}{N}\right), \text{ for } n = 0, 1, \dots, N-1. \end{aligned} \quad (3.9)$$

If you use $N = 2M$ equations, then the first $M - 1$ and last $M + 1$ equations are linearly independent and may be used to form the solution set. The method above applies to a one dimensional image ($1 \times M$).

For a two dimensional image, Frieden [2] states that for a $M \times M$ image, an $N \times N$ matrix allowed for a unique solution if $N = 1.5M$, which relates to a 50% over sampling. To transform the above method and equations for use in two dimensions, the summations need to be performed twice for the x and y directions. Also due to the frequency transform the cosine

and sine terms must contain $2\pi mn/N + 2\pi kp/N$ where m is summation in x direction with n and k is summation in y direction with p . The image must also be zero padded to ensure $M \times M$ so $m, k = 1, 2, 3, \dots, M$. The selected term of $S_{mk}^{(1)}$ resides at $m, k, = M/2$ like before and is excluded in the summation on the left hand side of the equation and forms the solution on the right hand side, but all other terms with m or k equal to $M/2$ must be included in the summation (m and k are not both equal to $M/2$).

From this adaptation we may form a set of linear equations (**A**) with unknowns(**x**) and a solution set (**b**).

$$\mathbf{Ax} = \mathbf{b}. \quad (3.10)$$

With linear analysis and the matrix [**A|b**] we can reduce the matrix to row echelon form and find the solutions to the unknowns. With the unknowns now found they can be used with their associated distorted image to inverse filter the distorted image and obtain the approximate image without atmospheric turbulent effects present.

This method requires two short exposure images to be available and is computationally restricted to the increasing matrix sizes with increasing image sizes (especially N which is 50% larger than M when dealing with two dimensional images). With the two dimensional implementation, the number of linear equations is $2N^2 - 1$ and the number of unknowns is $2M^2 - 1$, where $N = 1.5M$.

3.3 NEURAL NETWORK METHOD:

In [18] they make use of neural networks based on the Monte Carlo model to try and estimate the atmospheric PSF. The method is used to try and reduce the computing bottleneck that other methods and algorithms encounter. The immediate problem with the method is that training data is required to 'teach' the algorithm to estimate the PSF with a fair accuracy.

The model following is summarized from [18].

The model is a multilayer feed forward network which contains one hidden layer. The input data to the model consists of five trial parameters accompanying the image data (the image data consists of 2000 1-meter bins). The five accompanying trial parameters are:

I1 = Atmosphere type (Integer, range = {1,2,3})

I2 = Wavelength (Integer, range = {1,2,3,...,9})

I3 = aerosol model (Integer, range = {1,2,...,6}) , unused in their current implementation

I4 = aerosol optical depth (real, range (0.0, 1.0))

I5 = scale height (real, range (0.0, 10.0))

The neural network makes use of both conjunctive and coarse coding to provide for a sufficient amount of differentiation and generalization. The coarse coding allows the algorithm to perform more efficiently.

The training file contained 1400 patterns and test data of 400 patterns. The training of the network stopped when the correct percentage was a little higher than 95%. The test data produced a result of a 90% correct PSF estimation. Cong states that this high percentage indicates that the network can be used reliably to estimate the PSF of atmospheric environments that it has never even seen before.

The network unfortunately (as with most neural networks) requires a large training set, and the above network requires the input of various trial parameters.

3.4 TIME-SEQUENCE REGISTRATION METHOD:

In [5] a method is proposed and implemented using a hierarchically decreasing window registration to shift image sequences, and estimate the atmospheric PSF.

The method in [5] is summarized below.

The method uses images obtained with a shutter speed of less than 10ms to 'freeze' the atmospheric turbulence in the image.

An initial prototype image is formed ($f_0(m,n)$) which is just the average of sample images, creating an image which seems blurred.

Registration is performed on the images to create x and y shift maps corresponding to the prototype image $f_0(m,n)$. The registration procedure works by using two region windows of decreasing size. In the windows a phase correlation between the windowed pixels is performed. The co-ordinates where the phase correlation is the maximum represent the shift between the region window patterns. The region window is corrected by applying the opposite shift that was identified and the test repeated until the shift is close to zero. The window size is then decreased and focused on a subset of pixels to correct their region shifts.

The x and y shift maps created from the registration can be inversed and used with their original images to correct them to their true geometry. These images are then summed to create a second prototype that is corrected for motion ($f_2(m,n)$), there may however still be a small amount of blur. This small amount of blur can be further reduced by use of a global deconvolution if the residual PSF can be estimated.

Thorpe et al state that the prototype image ($f_2(m,n)$) can then be used as the new prototype image for the image registration instead of $f_0(m,n)$ in order to produce a sharper image. For an optimal result, they say that only two iterations of this registration technique are necessary.

Although not necessary for use with the algorithm, the PSF ($h(m,n)$) that caused the restored image to become as blurry as the original prototype image is:

$$h(x, y) = \mathcal{F}^{-1} \{ \mathcal{F} \{ f_0(x, y) \} / \mathcal{F} \{ f_2(x, y) \} \} \quad (3.11)$$

3.5 ADAPTIVE CONTROL GRID METHOD:

This method suppresses atmospheric turbulence effects in images by mapping the motion of the image from one image to the other using vectors and then using this information, it tries to correct the image to suppress the motion.

The method below is summarised from [76]:

The first part of the procedure utilises bilinear interpolation to increase the resolution of the image by a factor of four. Frakes et al state that the reason for this is to allow for sub-pixel accuracy. Spatial filtering is then performed to increase the high frequency information in each frame using an inverse Laplacian emphasised filter.

Control Grid Interpolation (CGI) is then performed to obtain a set of vectors that describe the movement of the image pixels from the one image to another. The motion field is obtained by segmenting the image into a set of contiguous squares whose corners are control points. The motion vectors between the control points are derived using the bilinear interpolation method.

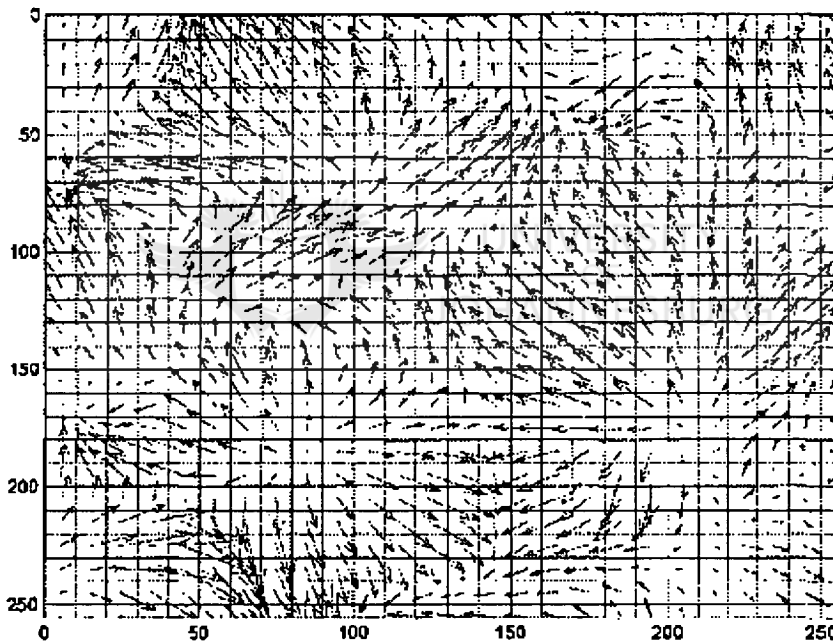


Figure 3-1: Optical flow field representation.

(Taken from [76])

CGI differs from the conventional block matching method commonly used in most motion detection techniques and is attractive due to its ability to represent complex non-translational motion. This image restoration method used by Frakes et al, uses a high resolution CGI algorithm and embedded optical flow equations for calculation of the control points. Figure 3-1 shows a representation of the optical flow vectors obtained using the CGI algorithm.

3.5.1 Parameter estimation procedure:

(3.12), (3.13) and (3.14) describe the relationship between the pixels of image $I_0[\mathbf{n}]$ and $I_1[\mathbf{n}]$, where $d_1[\mathbf{n}]$ is the horizontal component and $d_2[\mathbf{n}]$ the vertical component of the displacement at the spatial location (m,n) . For each segmented block α and β are estimated iteratively.

The bilinear parameters in each region R are found by minimizing (3.15). Using a Taylor series expansion (3.15) is approximated as (3.16).

$$I_1[\mathbf{n}] = I_0[m + d_1[\mathbf{n}], n + d_2[\mathbf{n}]] \quad (3.12)$$

$$d_1[\mathbf{n}] = \alpha_1 + \alpha_2 m + \alpha_2 n + \alpha_3 mn = \alpha^T \theta[\mathbf{n}] \quad (3.13)$$

$$d_2[\mathbf{n}] = \beta_1 + \beta_2 m + \beta_2 n + \beta_3 mn = \beta^T \theta[\mathbf{n}] \quad (3.14)$$

$$\sum_{\mathbf{n} \in R} I_0[\mathbf{n}] - I_1[m + \alpha^T \theta[\mathbf{n}], n + \beta^T \theta[\mathbf{n}]]^2 \quad (3.15)$$

$$\sum_{\mathbf{n} \in R} \left(I_0[\mathbf{n}] - I_1[\mathbf{n}] - \frac{\partial I_1[\mathbf{n}]}{\partial m} \alpha^T \theta[\mathbf{n}] - \frac{\partial I_1[\mathbf{n}]}{\partial n} \beta^T \theta[\mathbf{n}] \right)^2 \quad (3.16)$$

By solving the inverse of (3.16), the desired model parameters are found. The parameter estimation can be improved by changing the Taylor series approximation location from (m, n) to the new location $(m + \alpha^T \theta[\mathbf{n}], n + \beta^T \theta[\mathbf{n}])$ and re-estimating the parameters. Frakes et al found that the estimation usually converged in less than five iterations.



3.5.2 Segmentation:

The image is segmented into rectangular regions using a quadtree. In the beginning of the parameter estimation procedure the image is segmented into $M \times N$ equally sized blocks and each block contains a one leaf quadtree. Eight motion parameters are estimated for each leaf. A single leaf of each quadtree is split into for more leafs on each pass of the estimation procedure. The leaf that is chosen to be split is the one containing the parameters which minimises (3.15) the best.

3.5.3 Compensating for turbulence:

The motion vector fields provide us with the information of how the image flow is changing from the one image to the next. However because atmospheric turbulence is continuous, the motion vector field between images cannot be used to shift the image to its geometrically correct position. However since atmospheric turbulence is quasi-periodic, the turbulent motion vectors should sum to zero over a single period. By summing the motion vector fields over an estimated period, a base image frame can be obtained. By comparing the vector fields to this base frame the turbulent motion can be compensated resulting in suppressed atmospheric turbulence video footage.

Local object motion within an image can also be accounted for and left unchanged by making use of thresholding on the motion vectors. The threshold value which is dependent on the amount of atmospheric turbulence present can be determined by averaging the magnitudes of motion vectors. Any vector whose magnitude is greater by a specified percentage of the threshold is flagged as real motion (flags contained in a matrix with dimensions equal to the matrix of motion vectors). When the image is compensated for turbulence, the matrix of flags is consulted to see whether a pixel has undergone real motion. If the pixel has undergone real motion it is left unchanged. The advantage is allowing for real object motion in images while still correcting for atmospheric turbulence in the rest of the image.

3.6 IMAGE REGISTRATION AND FUSION METHOD:

This algorithm is constructed with two existing techniques, Image registration whereby geometric distortions are reduced and image fusion which corrects photometric distortion. Image registration techniques have been developed based on a pyramid structure which is very accurate and robust for both global and local motion estimation [1].

In [1], Zhoa et al describe the image fusion as three steps:

- Laplacian pyramid decomposition of images from level 0 (the original images) to level N;
- Weight computation based on salient pattern at each level; and
- Image reconstruction from level N to 0 using weights at each level to combine source pyramids.

The method below is summarised from [1]:

Step 1: Obtain an initial image.

Step 2: Perform video stabilization using frame to frame motion compensation and image warping using initial image.

Step 3: Enhance current frame using image fusion.

The number of neighbour frames used for image fusion was 10 due to the demand of fast video processing, however a larger number of neighbours may be chosen and is dependent on the atmospheric turbulence and object motion.

Step 2 is simply the image registration.

The selection of an initial image is important though as the second step aligns the frames geometrically with this initial frame. It is therefore important that this initial frame is chosen such that it contains the smallest amount of atmospheric turbulence possible. Ideally we want an initial frame that is unaltered by atmospheric turbulence.

Zhoa et al [1] suggested the following approaches to address the initial image frame selection problem:

- Compute the **first** initial frame: 1. manual selection of frame; or 2. determine a frame where the scintillation is zero and warp this back to in initial frame using object matching techniques; or 3. remove objects from all the frames and compute the most likely shape using Bayesian rule from the mixture of object motion and atmospheric turbulence.
- Compute the **current** initial frame: Warp a current frame to a previous initial frame, referring to option 2 above.



Figure 3-2: Image Fusion comparison.

From left to Right: Averaging of 10 frames, fusion of 10 frames using a clear initial image, and fusion of 10 frames using a blurred initial image (first row shows original images and the second row an enlarged area of interest.) (Taken from [1])

Zhoa et al determined that one would not care if they had an initial frame with little or no geometric distortion but photometric distortion. Although the image may be blurred (photometric distortion) the image fusion step takes information from all the frames in the group. Zhoa et al show that there is no remarkable difference when using a clear initial image or a blurred one (see Figure 3-2).

3.7 SPECKLE INTERFEROMETRY METHOD (TRIPLE CORRELATION THEORY)

Speckle Interferometry is a post processing technique which has been used in Astronomy for years, such as in retrieving Binary star information. Its basis was first established by Labeyrie in 1970 [80].

Speckle Interferometry refers to techniques which retrieve diffraction-limited (or near diffraction-limited) image information in the presence of atmospheric turbulence by making use of multiple speckle images [81]. A diffraction Limited image is the ideal image which would be observed through a telescope without the presence of any atmospheric turbulence. Speckle Images are images that are captured in a sufficiently short exposure time (typically less than 10ms) that freeze the atmospheric turbulence in place (sometimes referred to as specklegrams). The word ‘speckle’ in ‘speckle image’ and ‘specklegram’ comes from the way in which a single star (point source of light) is seen to be broken up into speckles due to the atmospheric turbulence [81] [82]

Labeyrie [80] proposed and demonstrated the use of a second order correlation (power spectrum) but it only allowed for the determination of the Fourier magnitude (modulus) and not the Fourier phase [83]. In 1977 Weigelt proposed the use of a triple correlation and its Fourier transform known as the bi-spectrum to retrieve phase information. This is discussed in detail in [84] and [85].

As Hoffmann explains in [81], when “Speckle Interferometry” began it referred to the second order correlation with “speckle imaging” or “speckle masking” referring to the triple order correlation technique such as in [84] which extends on the second order correlation to retrieve additional image information for a more accurate reconstruction (Fourier phase retrieval). Hoffman further explains that for convenience, the classical interferometry method and the newer speckle imaging are thought of as a part of a collection of techniques to which the term “Speckle Interferometry” is broadly applied.

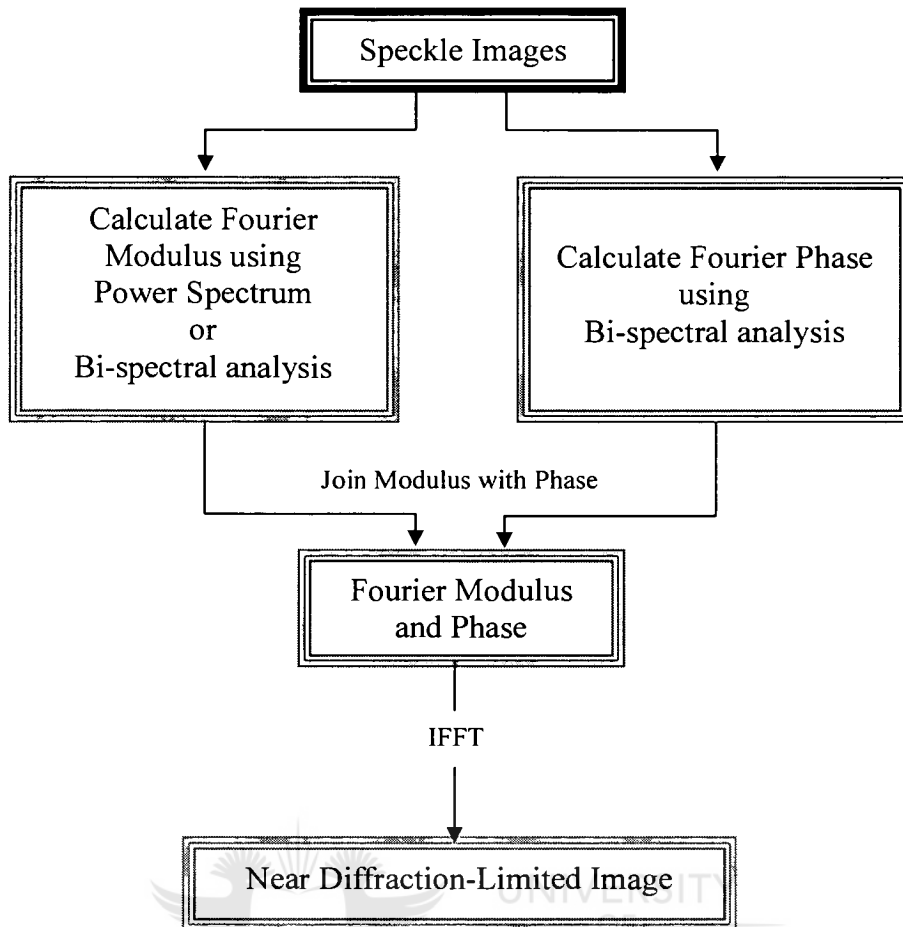


Figure 3-3: Flowchart of conceptual solution to obtaining a near diffraction-limited image.

In a simple form the theory is to separately calculate the Fourier modulus and the Fourier phase of an image (spatial Frequency), and then use the calculated magnitude and phase with the inverse Fourier transform to retrieve the image which is near diffraction-limited (see Figure 3-3). The Fourier modulus may be calculated either by the Power Spectrum (second order autocorrelation) or Bispectrum (Fourier transform of third order autocorrelation). The Fourier phase is calculated using the Bispectrum.

In [86] Carrano proposes a method using speckle imaging to remove atmospheric turbulence from images captured over horizontal paths (i.e. terrestrial atmospheric turbulence). The method described below is from [86].

The method follows the broader conceptual solution shown in Figure 3-3. For calculation of the Fourier modulus the power spectrum is used while the bi-spectrum is used for the Fourier phase estimation. Figure 3-4 re-illustrates the method in [86].

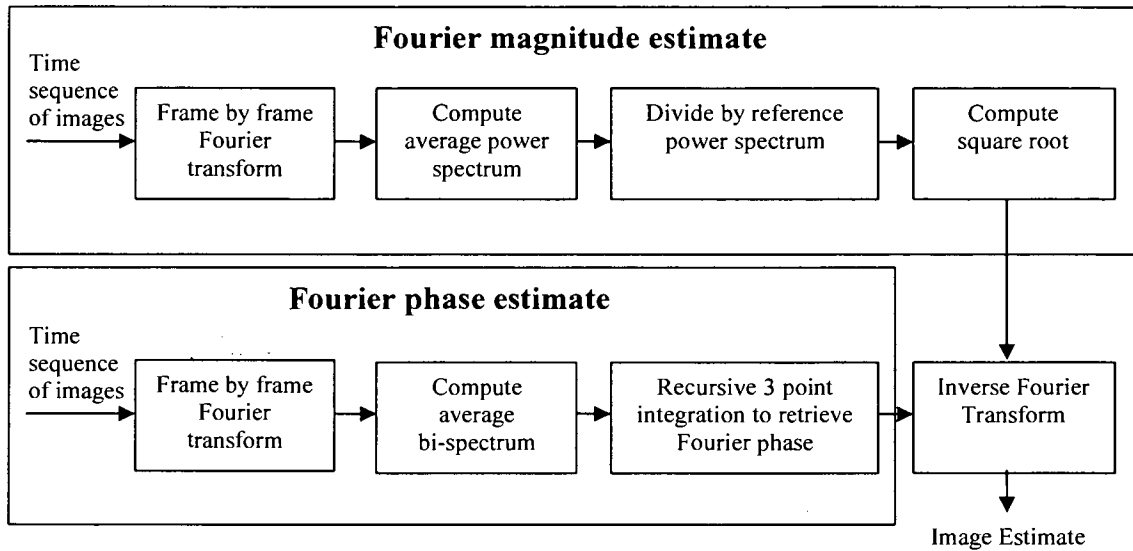


Figure 3-4: Block diagram of processing steps for Bispectral speckle imaging.

(Block diagram from [86].)

For the calculation of the Fourier magnitude, the power spectrum is calculated for multiple short exposure images and the average power spectrum is then calculated.

Using the model of (2.3) the Object (ideal undistorted image) Fourier modulus can be estimated through:

$$|O(\mathbf{u})|_{estimate} = \left[\frac{\langle |I_d(\mathbf{u})|^2 \rangle}{\langle |H(\mathbf{u})|^2 \rangle} \right]^{1/2}, \quad (3.17)$$

where $|O(\mathbf{u})|$ is the modulus of the object (ideal image viewed with no atmospheric turbulence), $\langle |I_d(\mathbf{u})|^2 \rangle$ the calculated average power spectrum, and $\langle |H(\mathbf{u})|^2 \rangle$ the point reference power spectrum. The point reference power spectrum can be obtained by capturing speckle images of a point source (such as a star in astronomy), but since this is obviously not possible or feasibly applicable with terrestrial surveillance, a model is used. The model used is the Labeyrie-Korff transfer function, with r_0 (the atmospheric coherence length) chosen

and used in the function. It is stated that trying values for r_0 between 1cm and 4cm in 0.5cm increments will almost always yield one or more acceptable images [86].

The bi-spectrum is defined in the spatial frequency space as:

$$I_{B,n}(\mathbf{u}, \mathbf{v}) \equiv I_n(\mathbf{u})I_n(\mathbf{v})I_n(-\mathbf{u} - \mathbf{v}), \quad (3.18)$$

where \mathbf{u} and \mathbf{v} are spatial frequency vectors. The bi-spectrum of each speckle image is calculated, from which the average bi-spectrum is calculated and used for the Fourier phase retrieval. In [84] a method is devised and used to recursively find the Fourier phase according to a 3-point integration. Conceptually the ruling for the integration and how it is performed is through the following formulae:

$$\arg | O(\mathbf{u} + \mathbf{v}) | = \arg | O(\mathbf{u}) | + \arg | O(\mathbf{v}) | - \arg | \langle I_{B,n}(\mathbf{u}, \mathbf{v}) \rangle |, \quad (3.19)$$

where $\arg|\dots|$ refers to the phase. For the phase calculation, redundant information in the bi-spectrum is used and an average Fourier phase calculated in order to suppress noise and increase the SNR (signal to noise ratio). [85]

For improvement in the produced image quality the above method is applied to overlapping portions of the image (tiles), which are then stitched back together once the reconstruction of the Fourier modulus and phase has taken place to form the final image. To avoid ringing effects due to the overlap, a Hanning window apodization is used.

3.7.1 Recursive Fourier phase estimation from bi-spectrum

Bartelt et al. in [85] proposed a method for the Fourier phase estimation through a recursive integration technique. The following text explains the method.

If one considers the discrete bi-spectrum:

$$T_{p,q}^{(3)} = T_p \cdot T_q \cdot T_{-p-q}, \quad (3.20)$$

where $T_{p,q}^{(3)}$ is the bi-spectrum and T_p , T_q , T_{-p-q} are sampled Fourier transforms of the signal.

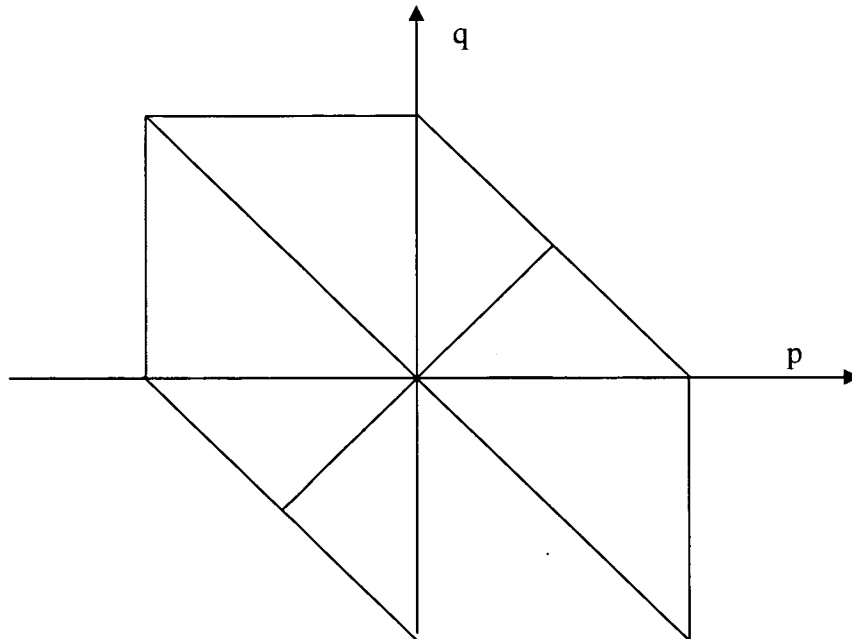


Figure 3-5: The Hexagonal structure of the bi-spectrum.

For a real signal the shaded octant is all that is needed to completely determine the rest of the bi-spectrum due to the symmetrical properties.

Using the bi-spectrums modulus and phase representation we find the phase relationship:

$$\beta_{p,q} = \varphi_p + \varphi_q - \varphi_{p+q}, \quad (3.21)$$

where $\beta_{p,q}$, φ_p , φ_q , and $-\varphi_{p+q}$ are the Fourier phases of $T_{p,q}^{(3)}$, T_p , T_q , and T_{-p-q} respectively. From the relationship it can be seen that there exist combinations of p and q that will yield the same Fourier phase angle φ .

p	q	$\varphi_{p+q} = \varphi_p + \varphi_q - \beta_{p,q}$
1	0	$\varphi_1 = \varphi_1 + \varphi_0 - \beta_{1,0}$
1	1	$\varphi_2 = \varphi_1 + \varphi_1 - \beta_{1,1}$
2	0	$\varphi_2 = \varphi_2 + \varphi_0 - \beta_{2,0}$
2	1	$\varphi_3 = \varphi_2 + \varphi_1 - \beta_{2,1}$
2	2	$\varphi_4 = \varphi_2 + \varphi_2 - \beta_{2,2}$
3	0	$\varphi_3 = \varphi_3 + \varphi_0 - \beta_{3,0}$
3	1	$\varphi_4 = \varphi_3 + \varphi_1 - \beta_{3,1}$
3	2	$\varphi_5 = \varphi_3 + \varphi_2 - \beta_{3,2}$
3	3	$\varphi_6 = \varphi_3 + \varphi_3 - \beta_{3,3}$
4	0	$\varphi_4 = \varphi_4 + \varphi_0 - \beta_{4,0}$
4	1	$\varphi_5 = \varphi_4 + \varphi_1 - \beta_{4,1}$
4	2	$\varphi_6 = \varphi_4 + \varphi_2 - \beta_{4,2}$
\vdots	\vdots	\vdots
n	0	$\varphi_n = \varphi_n + \varphi_0 - \beta_{n,0}$
n	1	$\varphi_{n+1} = \varphi_n + \varphi_1 - \beta_{n,1}$
\vdots	\vdots	\vdots
n	n	$\varphi_{n+n} = \varphi_n + \varphi_n - \beta_{n,n}$

Table 3-1: Recursive phase recovery using bi-spectrum phase information.

Due to the symmetry of the bi-spectrum values of p and q which are used to find φ are restricted. The bi-spectrum data resides within a hexagonal region and for a real signal, only one octant of the bi-spectrum is necessary to completely determine it due to symmetry relations with the other octants. [85].

For the determination of the Fourier phase using the recursive technique Bartelt et al. [85] states that φ_0 and φ_1 can be arbitrarily set to 0. Table 3-1 shows the relation between p and q and the unknown φ phases. Phases can be averaged where p and q yield the same φ .

From Table 3-1 we can see how we can use the redundant information in the bi-spectrum to get a phase average (e.g. averaging to find φ_4 using $p=2, q=2$ and $p=3, q=1$). In using the averaging though, Bartelt et al. [85] states that the exponential factors should be averaged ($\exp(i\varphi_p)$) rather than just summing up φ_p , because the φ_p are only determined up to modulo 2π .

3.7.2 Two dimensional signal Fourier phase reconstruction: Fourier series approach

This method is summarized from [87] in which the Fourier phase is reconstructed using a Fourier series approach and the bi-spectrum for a two dimensional signal.

For the following denotation of the bi-spectrum:

$$I_{B,n}(\mathbf{u}, \mathbf{v}) \equiv I_n(\mathbf{u})I_n(\mathbf{v})I_n^*(\mathbf{u} + \mathbf{v}), \quad (3.22)$$

where I_n denotes the Fourier transform of the two dimensional signal $i(\mathbf{n})$, with $\mathbf{n} = [m,n]^T$ and $\mathbf{u} = [u_1, u_2]^T$. $I_{B,n}(\mathbf{u}, \mathbf{v})$ is the bi-spectrum using spatial frequency vectors \mathbf{u} and \mathbf{v} and $*$ denotes the conjugate.

Using the notation $I_{B,n}(\mathbf{u}, \mathbf{v}) = |I_{B,n}(\mathbf{u}, \mathbf{v})| \exp[i\psi(\mathbf{u})]$ and for the Fourier transform of the two dimensional signal, $I(\mathbf{u}) = |I(\mathbf{u})| \exp[i\Phi(\mathbf{u})]$, with evaluation along the line $\mathbf{u} = \mathbf{v}$, the Fourier phases are written as:

$$\phi(\mathbf{u}) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} a(m, n) \sin(u_1 m + u_2 n) + \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} a(m, -n) \sin(u_1 m - u_2 n), \quad (3.23)$$

$$\psi(\mathbf{u}) = \sum_{m=0}^{\infty} \sum_{n=0}^{\infty} \hat{a}(m, n) \sin(u_1 m + u_2 n) + \sum_{m=1}^{\infty} \sum_{n=1}^{\infty} \hat{a}(m, -n) \sin(u_1 m - u_2 n). \quad (3.24)$$

To find $\Phi(\mathbf{u})$ its coefficients, $a(m,n)$, are determined from the $\psi(\mathbf{u})$ coefficients, $\hat{a}(m,n)$ using the following algorithm described by diante et al. [87].

(1) If either m or n is odd, then

$$a(m,n) = \frac{1}{2} \hat{a}(m,n) . \quad (3.25)$$

(2) If either $(m/2)$ or $(n/2)$ is odd, then

$$a(m,n) = \frac{1}{2} \hat{a}(m,n) + \frac{1}{4} \hat{a}\left(\frac{m}{2}, \frac{n}{2}\right) . \quad (3.26)$$

(3) If both $(m/2)$ and $(n/2)$ are even, then

$$a(m,n) = \frac{1}{2} \hat{a}(m,n) + \frac{1}{2} a\left(\frac{m}{2}, \frac{n}{2}\right) . \quad (3.27)$$

First steps (1) and (2) are carried out and then step (3) which uses even spaced information from $a(m,n)$.

Given $N \times N$ samples, $I_{B,n}(\mathbf{u}, \mathbf{u})$, of the bi-spectrum for $\mathbf{u} = [(2\pi/N)k, (2\pi/N)\boldsymbol{\ell}]^T$, $k, \boldsymbol{\ell} = 0, 1, \dots, N-1$, $\hat{a}(m,n)$ is defined as:

$$\hat{a}(m,n) \approx -\frac{2}{N^2} \text{Im}\{\text{DFT}[\psi(k,l)]\} . \quad (3.28)$$

The Fourier phase is then reconstructed as:

$$\hat{\phi}(\mathbf{u}) = \sum_{(k,l)} a(k,l) \sin(\mathbf{u}_1 k + \mathbf{u}_2 l) . \quad (3.29)$$

3.8 INDEPENDENT COMPONENT ANALYSIS (ICA) METHOD

Independent Component Analysis in its general definition and usage uses multiple inputs from sensors for breaking a signal into component signals. The application of ICA may be extended into various signal processing fields such as environmental noise cancellation, whereby the environmental noise is identified as a separate component of the original signal. Another example is the separation of two voices with the use of two microphones. Two mixed voice signals are captured with the two microphones from which the voices are attempted to be separated using the ICA method. [89]

Thus in a possible simpler view, ICA attempts the separation of an observed signal into the fundamental signals which were mixed, through the use of multiple observatory sensors.

The theory below is discussed and presented in [89], [90] and [91] as well as some methods in separating the component sources.

Considering the following model in which one observes n linear mixtures x_1, x_2, \dots, x_n of n components:

$$x_j = a_{j1}s_1 + a_{j2}s_2 + \dots + a_{jn}s_n, \text{ for all } j, \quad (3.30)$$

where x_j is the j^{th} observed mixture, a_{ji} are the parameters which determine the mix and s_i the i^{th} original component signal (source).

The mixing model is then expressed as:

$$\mathbf{x} = \mathbf{A}\mathbf{s}, \quad (3.31)$$

where \mathbf{x} and \mathbf{s} are the column vectors of the observed mixed signals and component signals respectively. Matrix \mathbf{A} determines the mixing of the signals. The model in (3.31) is referred to as the independent component analysis or ICA model [89].

The idea is the estimation of matrix \mathbf{A} in (3.31), from which the inverse of matrix \mathbf{A} is computed and used with \mathbf{x} to find the component signals in the vector \mathbf{s} .

As discussed in [89] the starting point of the ICA is the assumption that the component signals s_i are statistically independent and it is a requirement that they have non-Gaussian distributions. In [92], Kopriva et al. make the statement that the source signals (s_j) be statistically independent and non-Gaussian (*except possibly one*).

Kopriva et al. in [92] make use of ICA in order to try and separate the atmospheric distortions and the original image scene (image scene viewed if no distortions were present) using an image sequence as the vector set for the mixed signal observations (x_i) where each image frame data in a video is treated as information from a sensor. The approach differs slightly from normal ICA approaches, in that instead of sensors and their data being separated by space, they are separated by time. A fourth order cumulant-based ICA algorithm is used, JADE (joint approximate diagonalization of the eigen-matrices), together with a minimization of the squares of the fourth order cross-cumulants of the observed image sequence (for statistical independence achievement of estimated sources). Video frames which are used with the ICA data model need to be different enough so measurements are independent, and so the Kullback-Leiber divergence is incorporated and used to measure the mutual information of the images, ensuring a selection of linearly independent measurements (images). The algorithm is tested under weak atmospheric turbulence conditions and compared to a straight frame averaging technique using the Laplacian operator as a metric. [92]

3.9 HOMOMORPHIC FILTERING AND POWER SPECTRUM BASED METHOD

The Homomorphic filtering and power spectrum based method is described in detail in [6] and [28]. In essence the method uses the spectrum of a frame average and the spectrum from high pass Homomorphic filtering, to form a power spectrum filter for the frame averaged output (see Figure 3-6).

High frequency frame selection is performed by convolving the images with the convolutional mask shown in Figure 3-7. Grey-level threshold is then performed using the

intensity value of 63 (8 bit). The number of values which exceed the threshold are counted and the image frame which has the highest count is selected as the image which is used with the Homomorphic filtering.

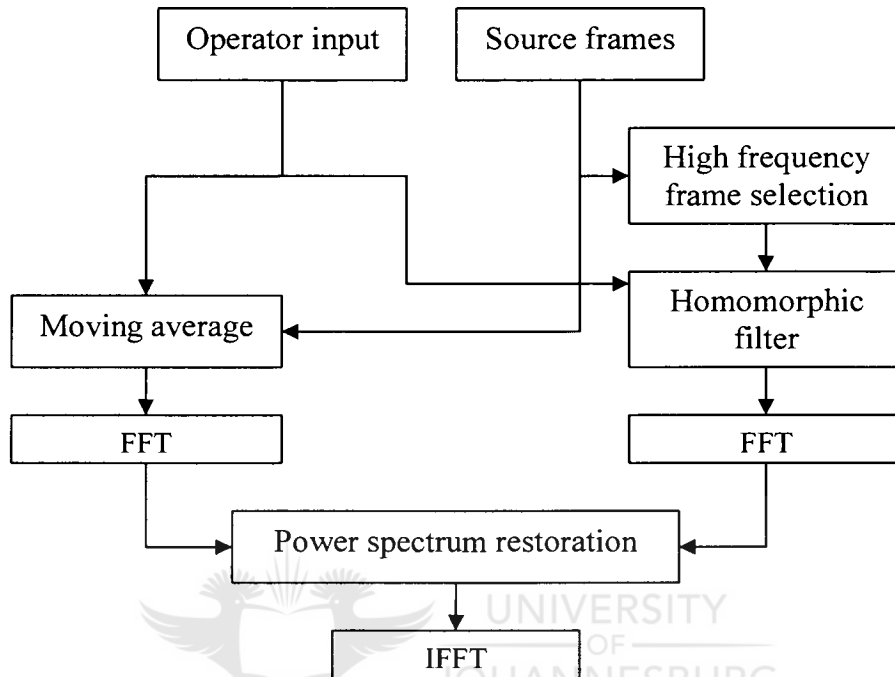


Figure 3-6: The Homomorphic filtering and Power spectrum based method block diagram.

0	-1	0
-1	4	-1
0	-1	0

Figure 3-7: Convolutional mask.

The process and functional block following the high frequency frame selection is the Homomorphic filtering. The filter used is a direct form fourth order FIR high pass filter with the discrete time domain transfer function:

$$H_{\text{ker}}(u) = \sum_{k=0}^4 b_k u^{-k}, \text{ with filter coefficients } b_0, \dots, b_4. \quad (3.32)$$

The operator is able to specify which filter coefficients the filter is to use, allowing for different cut-on frequencies to be selected. The two dimensional filtering is obtained by first applying the filter in the x direction and then filtering in the y direction.

Homomorphic filtering is structured by taking the natural logarithm of the image intensity values, transferring to the frequency domain, multiplication with the filter's frequency response, transference back to time domain and then finally outputting the exponential of the values. To reduce computational complexity this process can be performed in the time domain and convolution with the filters impulse response.

The power spectrum restoration filter is specified in [28] to have the following transfer function:

$$H_R(u, v) = \sqrt{\frac{W_F(u, v)}{|H_D(u, v)|^2 W_F(u, v) + W_N(u, v)}}, \quad (3.33)$$

where W_F is the ideal image power spectrum, W_N is the noise power spectrum, and H_D is the degradation transfer function which is unknown.

To approximate the above transfer function, the power spectrum from the Homomorphic filtered image is used in the numerator and the power spectrum of the image from the averaging is used as the denominator. Note that the square root does not fall away.

The power spectrum filter that is created is then multiplied with the FFT (Fast Fourier Transform) output and the IFFT (Inverse Fast Fourier Transform) performed to recovery the turbulence suppressed image.



3.10 CONCLUSION

This chapter presented various algorithms that have been created and documented over the past years by other authors.

The following chapter is Algorithms Detail Design, in which algorithms that will be implemented and used for atmospheric turbulence suppression and a comparison to a new algorithm suggested by this author are presented and described in detail specific to their implementation.



4 CHAPTER 4: ALGORITHMS DETAIL DESIGN

4.1 INTRODUCTION AND OVERVIEW

In this chapter the various algorithms that are to be implemented and compared are described in greater detail with regards to their construction and implementation. The algorithms described in this chapter will be tested and compared to one another using numerous experiments.

For additional contrast enhancement of the processed video, a full scale histogram stretch may be performed on the image frames as described in the literature study chapter.

Section 4.6 contains algorithms composed by this author, most notably the Illuminance-Reflectance based algorithms which are based using a slightly modified method described in [93].

For Matlab code of the algorithms and their supporting functions, the reader is referred to Addendums A to G.

4.2 SPECKLE MASKING ALGORITHM

Speckle Masking is a method used within the astronomy field in an attempt to suppress atmospheric turbulence effects which degrade an image of an astronomical object and provide an image which is near diffraction limited as described in chapter 3. This method is used extensively to retrieve object information within solar images (images of the sun and its 'surface').

The method in its simplest explanation is to separately determine the Fourier modulus using the averaged Power Spectrum of image frames and determine the Fourier phase using a triple integration technique with the averaged bi-spectrum. The Fourier modulus and phase forms

the spectrum of the object which is near diffraction limited and all that is left to do to recover the object image is an IFFT. This simplistic explanation is illustrated in Figure 4-1.

Describing the method in more detail, the Fourier modulus and Fourier phase computations shall be described separately below.

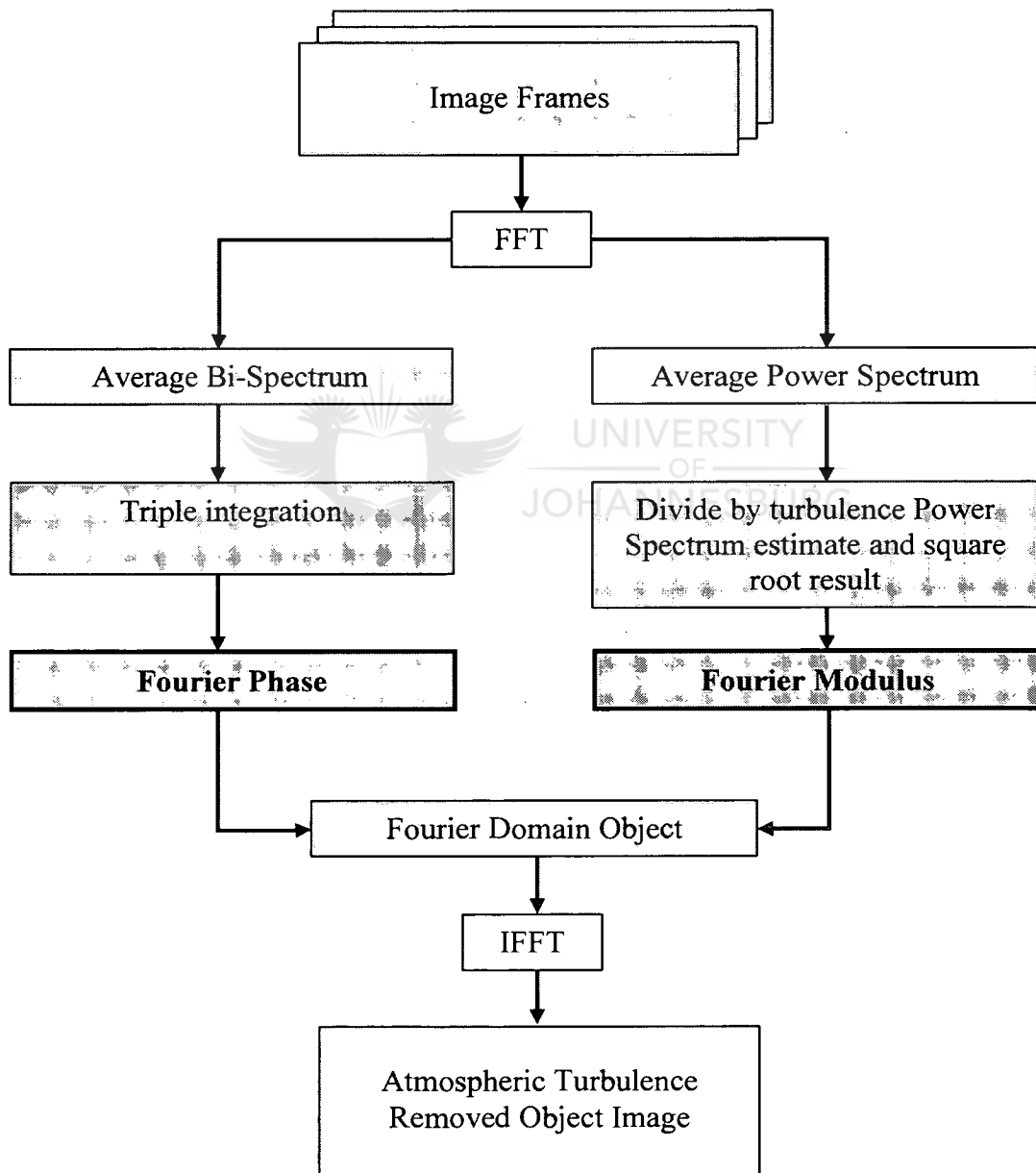


Figure 4-1: Speckle Masking simplified flow diagram.

4.2.1 Fourier Modulus Calculation

The Fourier modulus calculation (see right hand branch of Figure 4-1) is performed by averaging the Power Spectrum of multiple image frames and then dividing this Power Spectrum by a reference Power Spectrum. The reference Power Spectrum is the squared Fourier modulus of the atmospheric turbulence PSF. Since we cannot find the reference Power Spectrum without a point source such as a star, a reference is computed using the model developed by Korff [96].

Once divided by the reference Power Spectrum, the corrected Power Spectrum elements are square rooted to yield the Fourier modulus.

4.2.2 Fourier Phase Calculation

The Fourier phase calculation begins by finding the averaged Bi-Spectrum of multiple image frames. The Bi-Spectrum, $I_{B,n}$, of an image frame, I_n , is defined in the spatial frequency space as:

$$I_{B,n}(\mathbf{u}, \mathbf{v}) \equiv I_n(\mathbf{u})I_n(\mathbf{v})I_n(-\mathbf{u} - \mathbf{v}), \quad (4.1)$$

where \mathbf{u} and \mathbf{v} are spatial frequency vectors.

The Bi-Spectrum of a two dimensional object is itself four dimensional and contains plenty of redundant information as discussed already in chapter 3. To reduce computational complexity, only relevant Bi-Spectrum values are calculated that are needed in this methods phase recovery step. In order to keep to standardised symbols used when working with bi-spectrums the spatial frequency vectors $\mathbf{p}=(p_x, p_y)$ and $\mathbf{q}=(q_x, q_y)$ will be used instead of the vectors $\mathbf{u}=(u_1, u_2)$ and $\mathbf{v}=(v_1, v_2)$.

Phase recovery proceeds based on the following equation:

$$\psi(p_x + q_x, p_y + q_y) = \varphi(p_x, p_y) + \varphi(q_x, q_y) - \beta(p_x, p_y, q_x, q_y), \quad (4.2)$$

where $\psi(\mathbf{p}+\mathbf{q})$, $\psi(\mathbf{p})$, and $\psi(\mathbf{q})$ are the Fourier phases of a two dimensional signal and $\beta(\mathbf{p},\mathbf{q})$ is the Bi-Spectrum phase at location (\mathbf{p},\mathbf{q}) .

$\psi(0,1)$ and $\psi(1,0)$ cannot be found through the above equation and are the initial phases from which all higher frequency phases are found. $\psi(0,1)$ and $\psi(1,0)$ are simply set equal to the averaged image frame's respective Fourier phases.

By using different values for \mathbf{p} and \mathbf{q} we can see that we can find a specific phase multiple times while keeping within a single octant of the Bi-Spectrum. This information is used in this phase recovery method, so that calculated values at a specific phase location may be averaged to improve the signal to noise ratio of the reconstructed image.

The Fourier phases are calculated in the following way given that image frames are size $N \times N$:

Find $\psi(0,2)$, $\psi(0,3)$, $\psi(0,4)$, ..., $\psi(0,N-1)$ using equation (4.2).

Find $\psi(2,0)$, $\psi(3,0)$, $\psi(4,0)$, ..., $\psi(N-1,0)$ using equation (4.2).

Find $\psi(1,1)$ using $\psi(0,1)$ and $\psi(1,0)$ in equation (4.2).

Find $\psi(1,2)$, $\psi(1,3)$, $\psi(1,4)$, ..., $\psi(1,N-1)$ using equation (4.2).

Find $\psi(2,1)$, $\psi(3,1)$, $\psi(4,1)$, ..., $\psi(N-1,1)$ using equation (4.2).

The above have only one possible occurrence value when considering a single octant of the Bi-Spectrum. For calculation of the rest of the Fourier phases, the code snippet shown in Figure 4-2 is used where the variable 'numtoavg' set by the user is the number of calculated phase values to average over for SNR improvement.

```

for k = 2:N-1
    for l = 2:N-k
        qx = floor(k/2)
        qy = floor(l/2)
        occurs = qx + qy - 1    % this is the maximum number
                                of values that can be
                                averaged
        if occurs > numtoavg then
            occurs = numtoavg
        endif

        for num = 1:occurs
            px = k - qx
            py = k - qy

            temp = expphases(qx, qy) .* expphases(px, py) .*
                    exp(-i*(BSpecAngle(k, l, num)))
            expphases(k, l) = expphases(k, l) + temp/occurs

            if qx > qy then    % Decrease greater vector
                                component of q
                qx = qx - 1
            else
                qy = qy - 1
            endif
        endfor
    endfor % l loop end
endfor    % k loop end

phases = angle(expphases)

```

Figure 4-2: Speckle Masking Pseudo code snippet

In the algorithm code of Figure 4-2, `angle(.)`, is a function which returns the phase of an object. The averaging used to find the phase at position (k,l) is performed using the exponential in order to remove ambiguities of $+\pi$ which can occur if averaged directly [97]. `BSpecAngle(k,l,num)` is the Bi-Spectrum angle calculated for finding the phase at (k,l) with $num = 1 \dots numtoavg$.

The calculation of the Bi-Spectrum follows the same algorithm formulae as discussed above except that instead of the phase calculation portion (temp and exp phases calculation in Figure 4-2), the Bi-Spectrum is calculated using:

$$\text{BSpecAngle}(k,l,num) = \text{angle}((\mathbf{I}(p_x,p_y) * \mathbf{I}(q_x,q_y) * \text{conj}(\mathbf{I}(k,l))), \quad (4.3)$$

where \mathbf{I} is the Fourier transform of the image frame and the $\text{conj}(\cdot)$ function returns the conjugate of an element. Bi-Spectrum elements are also calculated for use when finding $\psi(0,2)$, $\psi(0,3)$, $\psi(1,2)$, $\psi(1,1)$, $\psi(2,3)$... etc that have only one phase value occurrence in the Bi-Spectrum octant.

4.2.3 Additional processing steps

Now that the Fourier modulus and phase reconstructions have been described, additional processing steps which are undertaken when performing the reconstruction shall be discussed.

In order to overcome anisoplanatism (the spatial variation of the PSF across an image) effects in reconstruction of the image as discussed by Pehlemann et al. [97], the image frames are divided into smaller overlapping segments which are processed and then recombined using a Hamming window at the end of the algorithm. Figure 4-3 illustrates the segmentation and Figure 4-4 the hamming window.

1	1,2	2
1,3	1,2,3,4	2,4
3	3,4	4

Figure 4-3: Image frame segmented into four overlapping segments. Segment 1 is highlighted.

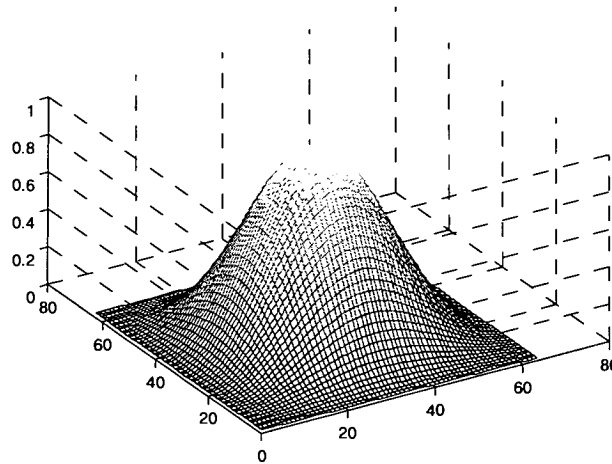


Figure 4-4: 64x64 Hamming window for multiplication with a 64x64 sized segment during reconstruction.

Before performing the Speckle Masking algorithm on a set of segments, basic registration is performed within a small region surrounding the segment to remove small translational motion. The reference image used for the registration is the spatially corresponding segment in the first image frame.

4.3 IMAGE REGISTRATION AND LAPLACIAN PYRAMID IMAGE FUSION

Laplacian Pyramid Image Fusion follows the theory laid out by Zhou et al. [1] and as part of undergraduate work in [6]. Figure 4-5 gives a simple Block diagram for the algorithm.

The method has three main steps. The first step is obtaining a reference image frame, which is simply the frame averaged image as calculated using equation (4.4). Image Registration is then performed on each image frame within a sliding window set so that object detail match up to those in the reference frame. The last step performs Laplacian Pyramid Image Fusion using the multiple registered image frames.

These three steps are explained in detail in chapter 4 of [6] as part of the undergraduate study.

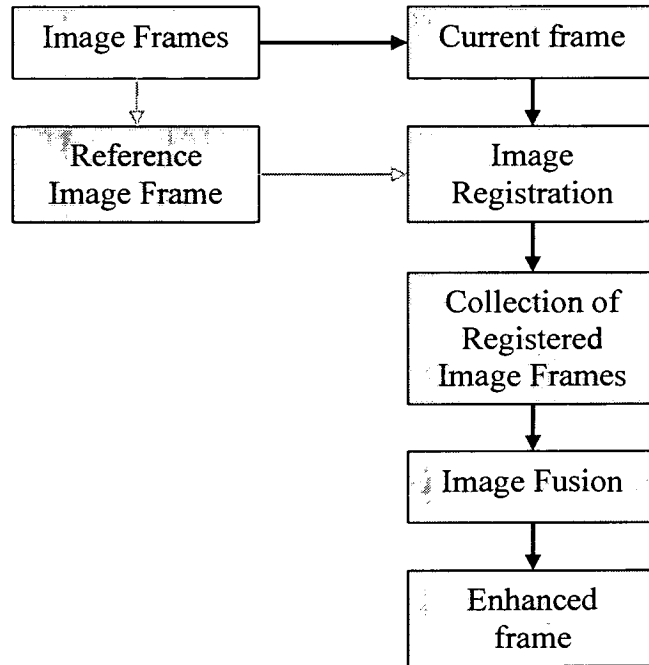


Figure 4-5: Block diagram of Image Registration and Laplacian Pyramid Image Fusion Algorithm.

4.4 HOMOMORPHIC FILTERING AND POWER SPECTRUM RESTORATION

The Homomorphic Filtering and Power Spectrum Restoration based method originally described by Thales Advanced Engineering (now called Protoclea Advanced Image Engineering) [28], treats the problem in two parallel processing steps as shown in Figure 4-6.

The Ratio frame average block calculates the frame average using the following equation to suppress geometric distortions and provide Fourier Phase information during the Power Spectrum Restoration Step:

$$g^{(t)}(m, n) = \alpha \cdot f(m, n) + (1 - \alpha)g^{(t-1)}(m, n), \quad (4.4)$$

where $g^{(t)}(m, n)$ is the resulting image frame ratio average, $f(m, n)$ the current image frame, $g^{(t-1)}(m, n)$ the frame average calculate on the previous iteration and alpha a scalar between 0 and 1 which is specified by the user and controls the weighting between the current frame and ratio average.

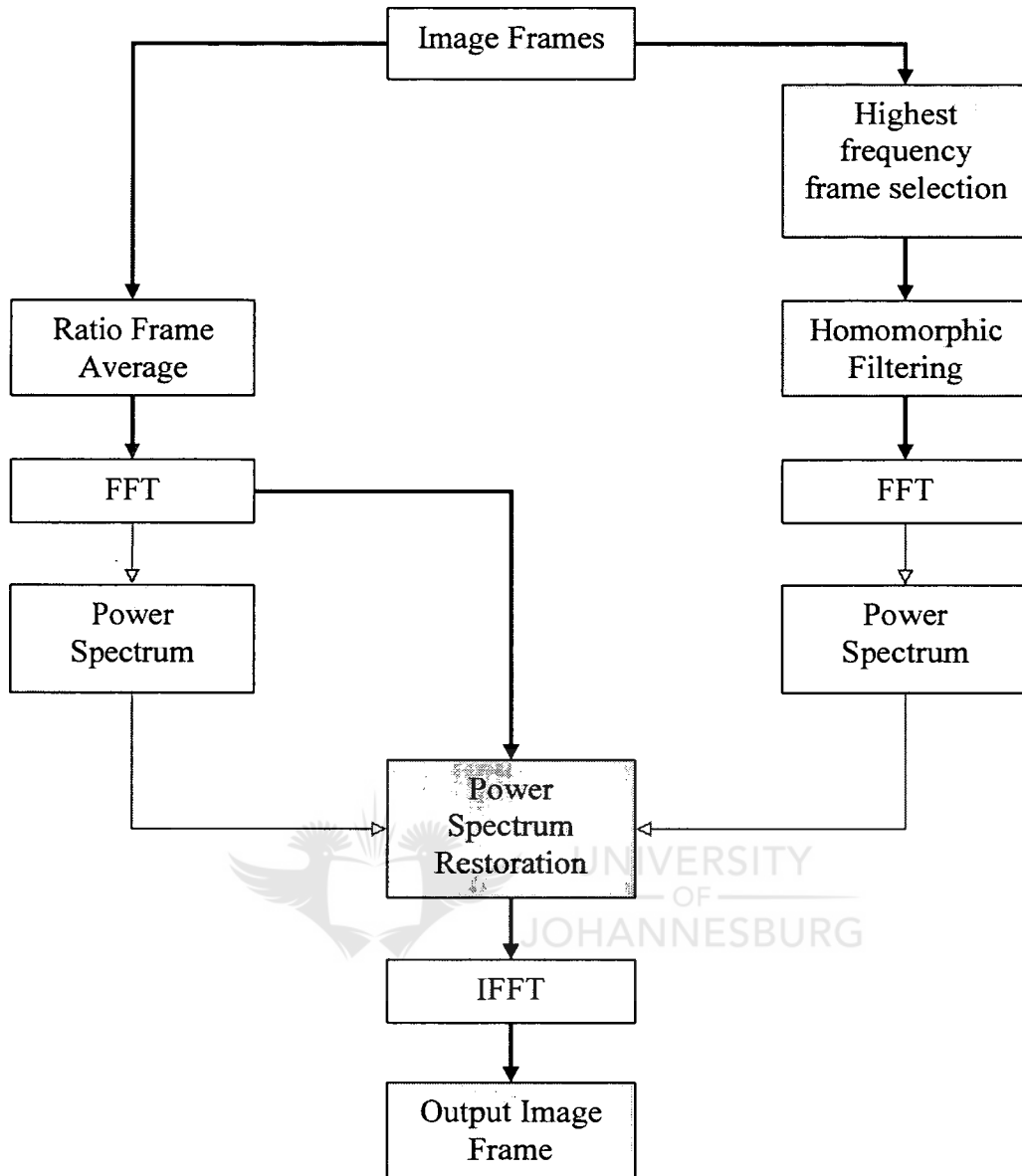


Figure 4-6: Block diagram of Homomorphic filtering and Power Spectrum Restoration based algorithm.

High frequency frame selection is performed by convolving the images with the convolutional mask shown in Figure 4-7. A Grey-level threshold is performed using the intensity value of 63 (8 bit). The number of values which exceed the threshold are counted and the image frame which has the highest count is selected as the image which is used in the Homomorphic Filtering step.

0	-1	0
-1	4	-1
0	-1	0

Figure 4-7: Convolutional mask.

For the Highest frequency image selection a sliding window of 8 images is used, with a new highest frequency image being selected from the set every 5 image frames. The reason for the selection is explained in [28] and is due to the fact that the high frequency Fourier modulus information does not change rapidly.

The process and functional block following the high frequency frame selection is the Homomorphic filtering. The filter used is a direct form fourth order FIR high pass filter with the discrete time domain transfer function:

$$H_{\text{ker}}(u) = \sum_{k=0}^4 b_k u^{-k}, \text{ with filter coefficients } b_0, \dots, b_4. \quad (4.5)$$

The operator is able to specify which filter coefficients the filter is to use, allowing for different cut-on frequencies to be selected. The two dimensional filtering is obtained by first applying the filter in the x direction and then filtering in the y direction.

Homomorphic filtering is structured by taking the natural logarithm of the image intensity values, transferring to the frequency domain, multiplication with the filter's frequency response, transference back to time domain and then finally outputting the exponential of the values. To reduce computational complexity this process can be performed in the time domain and convolution with the filters impulse response.

The power spectrum restoration filter is specified in [28] to have the following transfer function:

$$H_R(u, v) = \sqrt{\frac{W_F(u, v)}{|H_D(u, v)|^2 W_F(u, v) + W_N(u, v)}}, \quad (2.51)$$

where W_F is the ideal image power spectrum, W_N is the noise power spectrum, and H_D is the degradation transfer function which is unknown.

To approximate the above transfer function, the power spectrum from the Homomorphic filtered image is used as the numerator and the power spectrum of the image from the averaging is used as the denominator. Note that the square root does not fall away.

In Essence, the Fourier Modulus information from the Homomorphic filtered image is being incorporated into the Frame averaged image. The power spectrum filter that is created is then multiplied with the FFT (Fast Fourier Transform) output and the IFFT (Inverse Fast Fourier Transform) performed to recovery the turbulence suppressed image.

4.5 WIENER FILTERING APPROACH

The basic Wiener filtering approach is simply performing Wiener filtering on the ratio averaged image described by equation (4.4) for suppression of geometric distortions. In order to perform Wiener filtering a model of the PSF (Point Spread Function) or OTF (Optical Transfer Function) is required. The OTF model used in this method and used in the Control Grid Interpolation Method in [10] is:

$$H(u, v) = e^{-\lambda(u^2+v^2)^{5/6}}, \quad (4.7)$$

where u and v are the spatial frequencies and λ is a scalar which corresponds to the severity of the blur. As λ increases so does the severity of $H(u, v)$ (the OTF).

The Wiener filter is given by:

$$H_{wiener}(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + \left(\frac{W_N(u, v)}{W_i(u, v)} \right)}, \quad (4.8)$$

where $W_i(u, v)$ and $W_N(u, v)$ are the power spectrum of the signal and noise, respectively.

Using the model described by equation (4.7) the value for λ is found using kurtosis minimization as described in [98]. The kurtosis of a random variable (x) is described as the normalized fourth central moment:

$$kurtosis = \frac{E((x - \mu)^4)}{\sigma^4} \quad (4.9)$$

where μ is the mean of x , σ is its standard deviation and $E(.)$ is the expected value operation.

The λ parameter (also referred to as Lambda value within this study) in equation (4.7) is searched within a reasonable space Ω . For each λ parameter the image is Wiener filtered using equation (4.7) as the OTF and the kurtosis measured. The de-blurred image with the minimum kurtosis is chosen.

4.6 OWN ALGORITHM DEVELOPMENTS

Based on information gained through the literature study with regards to the atmospheric turbulence problem and the way various algorithms attempt to address the problem, algorithms have been constructed to compete with the existing mentioned algorithms.

In order to keep to keep the workload the algorithms need to perform to a minimum, this algorithm attempts to be efficient in outputting an enhanced image with little overhead. Efficiency is essential to producing a good output image at higher frames per second than corresponding algorithms.

4.6.1 Wiener Filtering using Laplacian Operator

This Wiener filtering method was produced as an alternative to the kurtosis minimization method.

Instead of using the kurtosis as a measure to the amount of blur in an image and then using the λ that produces the smallest kurtosis, the Laplacian Operator Sharpness Metric is used to determine how sharp an image is and then use the λ which produces the sharpest image.

The Laplacian Operator Sharpness Metric is described as:

$$S = \frac{1}{MN} \sum_m \sum_n |\nabla^2 f(m, n)|, \quad (4.10)$$

where $f(m, n)$ is the image being measured, M the number of rows and N the number of columns. This metric is equivalent to the image sharpening metric in [101] and [102].

4.6.2 Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression

This method is primarily based on the Illuminance-Reflectance video enhancement technique discussed by Tao et al. [93]. The method separates an image into two components; a lower frequency Illuminance estimate and a high frequency Reflectance estimate. A block diagram of the process is given in Figure 4-8.

This method is constructed such that it may operate using gray images or colour images. Tao et al. [93] presents some results indicating this methods excellent colour rendition and image enhancement in comparison to two other methods: Multi-Scale Retinex (MSR) and Luma Dependent Nonlinear Enhancement (LDNE) [93]. The construction and incorporation of some simplicity (in comparison to registration and memory hungry techniques) within the algorithm may yield some performance results achieving or reaching close to real time goals.

The first step in the algorithm uses a frame averaging technique described as:

$$g^{(t)}(m, n) = \alpha \cdot f(m, n) + (1 - \alpha)g^{(t-1)}(m, n), \quad (4.4)$$

where $g^{(t)}(m, n)$ is the resulting image frame ratio average, $f(m, n)$ the current image frame, $g^{(t-1)}(m, n)$ the frame average calculate on the previous iteration and alpha a scalar between 0 and 1 which is specified by the user and controls the weighting between the current frame and ratio average.

The frame averaging technique is used because it is able to produce some stabilization to the image using a low alpha value. The ratio averaging is inexpensive in processing time and is much more efficient than elastic registration or similar techniques [6].

The Reflectance and Illuminance adjustment component by Tao et al. [93] was chosen to be a part of the algorithm as it too runs fairly fast [93] while sharpening up the image and suppressing blur which is a major component atmospheric turbulence. The algorithm component also has an advantage of enhancing detail hidden in shadows.

Equation (4.12) describes an image frame comprising of Reflectance and Illuminance, whereby the Illuminance estimate is found by low pass filtering the image frame using a 4x4 sized Gaussian filter and the Illuminance is used together with equation (4.12) using element by element division to find the reflectance estimate. For colour images the greyscale image is used which is comprised using the maximum intensity across the RGB colour bands.

$$f(m, n) = L(m, n) \cdot R(m, n), \quad (4.12)$$

where $L(m, n)$ is the Illuminance estimate and $R(m, n)$ is the Reflectance estimate.

The Illuminance image is normalized and used in the Dynamic Range Compression step described by Tao et al. [93]. In this step the lower frequencies are suppressed.

There is a minor difference in the sigmoid function which is used and that documented by Tao et al.

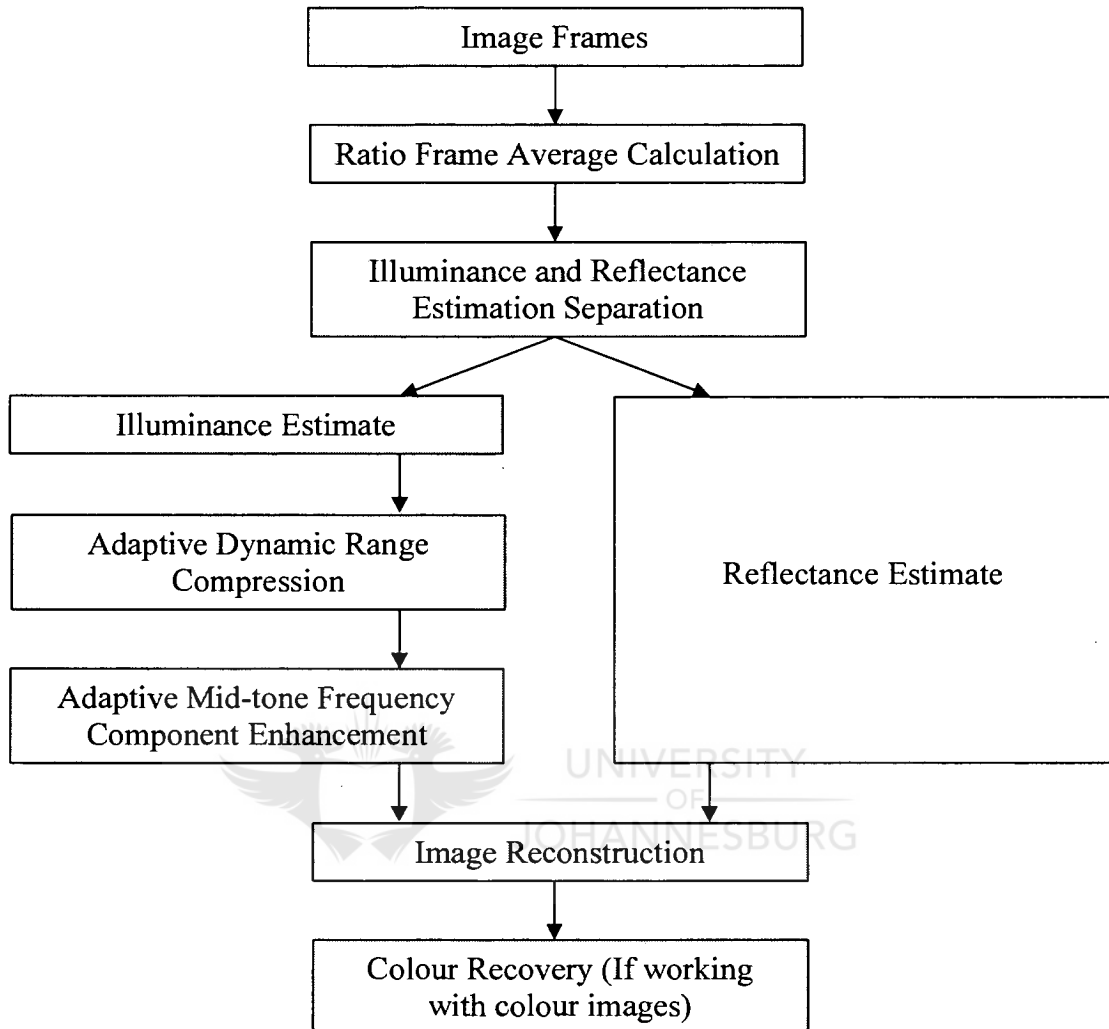


Figure 4-8: Block diagram of Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression Algorithm.

The Sigmoid function used is:

$$H_{sig}(k) = \frac{1}{1 + e^{ak}}, \quad (4.13)$$

where the a value used is 1.

The next step, Adaptive Mid-tone frequency Component Enhancement, takes place via the same technique as documented in [93]. Although the scalar P is found

dynamically through this step, it may be advantageous at times for this value to be overridden by a user specified scalar for contrast enhancement of particularly faint object detail.

Image restoration takes place through the dot product of the new Illuminance component and Reflectance component.

For colour images, the RGB colour bands are recovered using:

$$r' = \frac{I'}{I} r \quad g' = \frac{I'}{I} g \quad b' = \frac{I'}{I} b, \quad (4.14)$$

where r, g, b are the red, green and blue colour bands respectively from the frame ratio average. I' the greyscale version of the ratio frame average found through the maximum intensity approach discussed earlier. I is the new reconstructed grey image and r', g', b' the new red, green and blue colour bands respectively.

4.6.3 Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener Filter

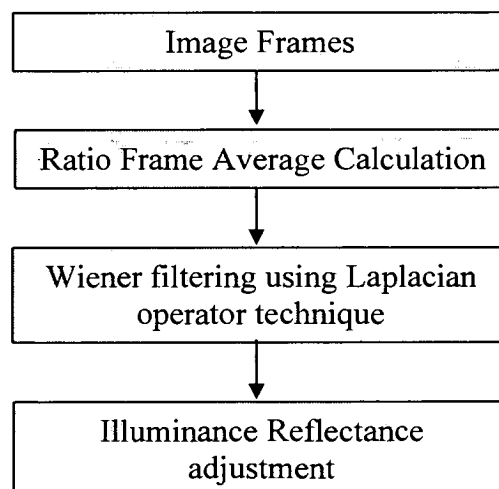


Figure 4-9: Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener filtering.



This method is the same as the previously described technique except that it additionally performs Wiener filtering after the ratio frame averaging using the Wiener filtering using the Laplacian Operator technique previously described in this section.

After Wiener filtering a full histogram stretch is performed to fully make use of the 8 bit grey level and provide maximum contrast. Once the histogram stretch had been performed the Illuminance Reflectance adjustment proceeded.

4.7 CONCLUSION

This chapter detailed the base design of the selected algorithms which are implemented and compared. Own algorithm developments are presented which make use of components or parts of components described by other authors.

The following chapter outlines the different experiments that will be performed, some of which are to compare the algorithms to one another.

5 CHAPTER 5: EXPERIMENTAL DESIGN

5.1 INTRODUCTION AND OVERVIEW

The planning of experimental work is essential in ensuring a successful outcome to any experiment. Also important in any experiment is repeatability and that of any tests so that results may be duplicated by others who follow the laid out experimental setup and method.

The experiments follow suggestions and guidelines from various authors. [105, 106, 107, 108, and 109]

This chapter lists the experiments undertaken in order to provide a comparison between the various algorithms on a FPS (Frames per Second) performance and image quality level. The imaging system used in the experiments and the setup of that imaging system is first described. Following the description of the imaging system setup, the experiments undertaken to analyse the algorithms are listed.

Each experiment has an aim, list of equipment used, method, discussion of experiment relevance and its relation to this project study, and expected experimental results. Some additional experiments are listed relating to the algorithm performance and quality investigation, as well as the basis atmospheric turbulence problem, such as the use of optical filters before video capture.

5.2 IMAGING SYSTEM

This section details the equipment used for capturing of test video footage used in some of the following experiments.

The digital video footage will be captured using a CCD video camera which has a detachable lens with a C/CS thread. The reason for the C or CS thread is because an off-the-shelf adapter component is available that allows connection of a C or CS threaded device to the 1.25 inch telescope ocular at prime focus of the telescope. This is important as one will then not have

distortions due to the eyepiece lens of the telescope (the eyepiece lens is bypassed and telescope becomes the lens for the camera). The CCD camera used will be an Arecont Vision AV3100 3 mega pixel IP camera due to cost considerations and its availability within the Electrical Engineering Department for this project.

While this camera has some suitable specifications, the AV2100 would be a more suitable camera for such purposes. This is because although it is a lower resolution camera (2 mega pixels) the size of the CCD is the same and has larger pixel bin sizes than the AV3100, thus it is more sensitive to light and performs better at very short exposure times which are desired for some of the algorithms.

The 'lens' of the imaging system is a Celestron Nextar 8 SE telescope which has a focal length of 2032mm and a diameter of 8 inches. The long focal length provides a much greater optical zoom while the large diameter allows for a greater collection of light allowing for the camera to more readily achieve short exposure times. The telescope is of a Schmidt-Cassegrain design which is a robust enclosed design and allows a wide range of focus adjustment. The focus adjustment is necessary for raw CCD imaging (CCD with no lens connected directly to the visual back of the telescope).

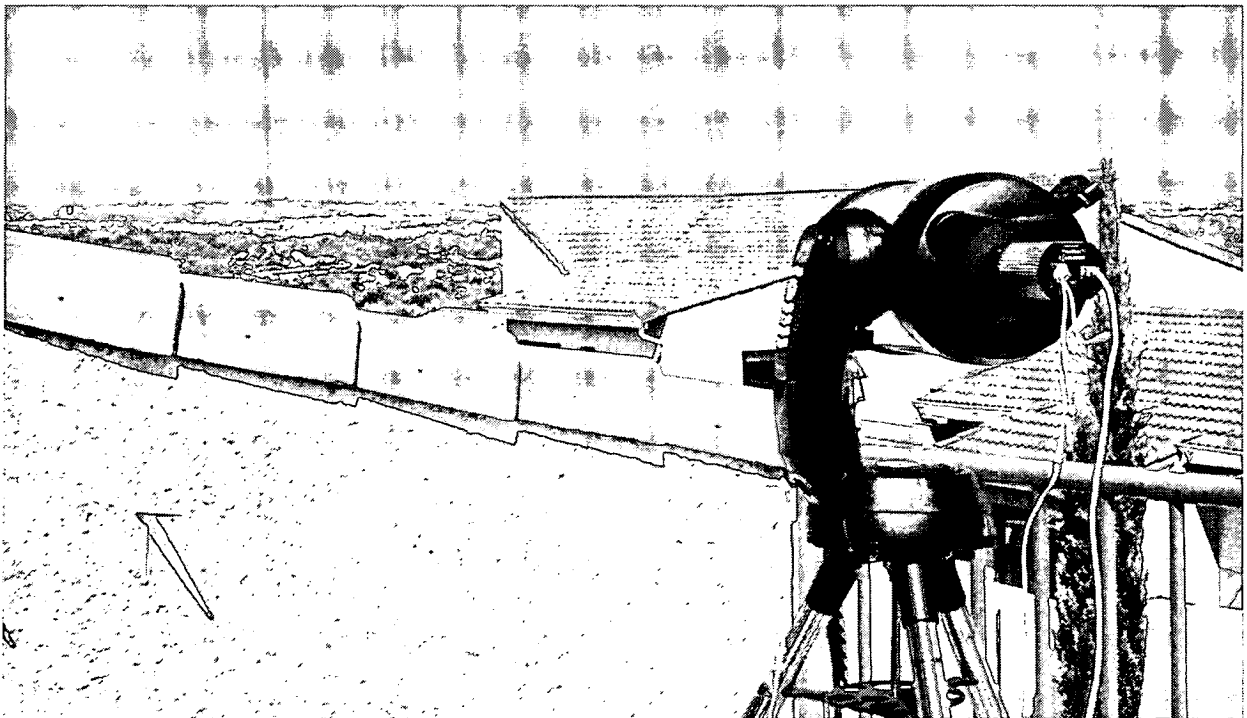


Figure 5-1: Imaging System

5.3 IMAGING SYSTEM EQUIPMENT SETUP

The imaging system is setup for recording video footage of a distant target as follows:

1. Assemble the telescope according to the manufacturer's instructions.
2. Point the telescope at the intended target to be recorded.
3. While looking through the eyepiece centre the target in the field of view.
4. Remove the eyepiece and 1.25" diagonal.
5. Unscrew c/cs mount lens from camera if one is attached exposing the ccd.
6. Attach the 1.25" to c/cs adapter to the camera.
7. Mount the camera and adapter to the back of the telescope (replacing the diagonal and eyepiece).
8. Connect the power supply and Ethernet cable to the camera.
9. Connect the Ethernet cable to the computer and setup a static Ip that has the same network address as the camera.
10. Use camera software to view streaming video.
11. Refocus image using focusing knob on telescope.
12. Video footage can now be captured of the target using the camera software.
13. The camera on its own gives roughly a 160x optical magnification of the target. Additionally a Barlow lens and 1.25" filters may be connected between the telescope and the 1.25" to c/cs adapter. The Barlow lens doubles the optical magnification.

5.4 OVERVIEW OF EXPERIMENTS

This section provides an experimental overview to link the experiments and their relationship to this study.

There are 7 experiments. Experiments 1 to 4 are used to compare algorithms on a performance and image quality level. Experiment 5 uses the classical mean square error metric to try measure and compare image quality. Experiment 6 investigates a performance improvement vs. image quality impact for the Wiener filtering based algorithms and Experiment 7 investigates the use of light filters and their effect on captured video.

Experiment 7 is an additional experiment relating to the atmospheric turbulence problem itself rather than the algorithms. Experiment 7 is considered relevant however since real time improvements may be made to an image before the image is captured by digital video footage and enhanced further by the atmospheric turbulence suppression algorithms. Figure 5-2 illustrates the relationship of these experiments.

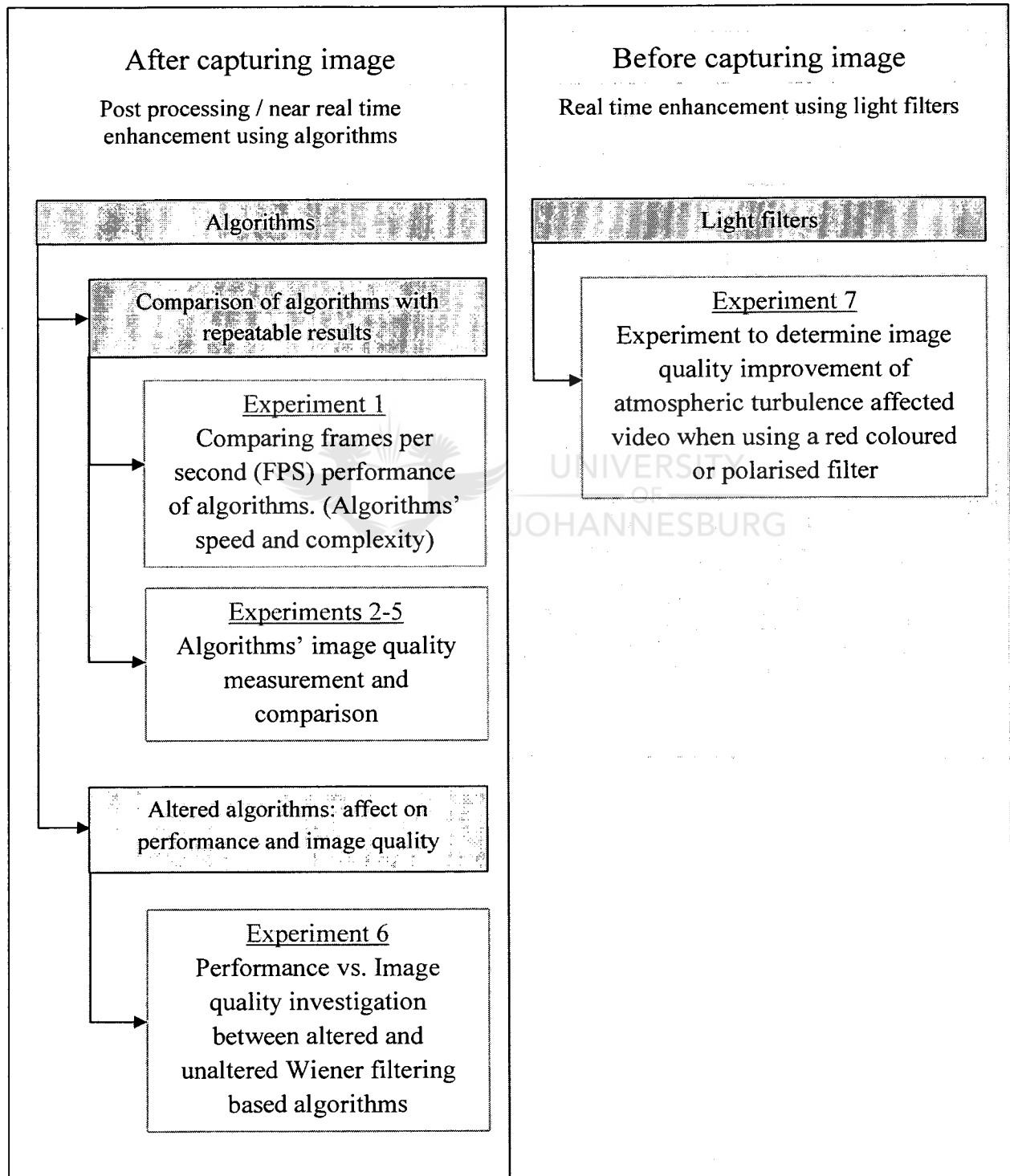


Figure 5-2: Overview of experiments

It is important when interpreting results that input parameters to the system that are not being investigated be set constant so as to insure that the changing system output is a result of the single parameter that is changing or under investigation.

All external programs are closed to ensure memory and processor resources are not being used or changed by outside programs (such as antivirus software or programs trying to automatically update).

5.5 EXPERIMENT 1: ALGORITHM PERFORMANCE EXPERIMENT

5.5.1 Aim

To find out how the algorithms compare with each other in terms of FPS (Frames per second) performance over various input image sizes.

5.5.2 Equipment

- Imaging system described in section 5.2,
- Algorithms detailed in chapter 4 implemented in Matlab, and
- Computer on which to run Matlab and algorithms.

5.5.3 Method

1. Using the imaging system, capture and record raw avi (Audio Video Interleave) video footage containing at least 100 image frames with a 10ms or less exposure time. Image size recorded must be 640x480 pixels or greater.
2. Crop the video into image sizes of 80x60, 160x120, 240x180, 320x240, 400x300, 480x360, 560x420, and 640x480 and save each resized video file.

3. Record the number of seconds it takes an algorithm to process 100 video frames. Repeat for each algorithm and the different video sizes.
4. Calculate average FPS for each image size by dividing 100 by the time it takes to process 100 frames.
5. Graph the results (FPS vs. Image size) and compare the performances.

5.5.4 *Experiment Relevance*

This experiment allows a determination of the complexity of an algorithm and relates to how easily an algorithm may be implemented to run in real time. The greater the complexity of an algorithm, the more processing is required and the lower the FPS will be. Thus to achieve real time processing, an algorithm with a lower FPS will need more expensive equipment that may also be bulkier and require far more energy than an algorithm with a higher FPS.

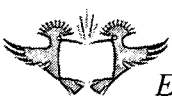
5.5.5 *Expected Outcome*



Graphed results are expected to show an exponential type decay as double the number of rows and columns of an image means that there are four times more pixels to process. The homomorphic filtering and power spectrum based method is expected to show the highest fps and the least drop off with increase in image size as it is simplistic in nature and originally developed and optimised to run on portable processing hardware. The image registration and fusion based method is expected to have the lowest fps due to the high level of complexity in the image registration.

5.5.6 *Uncertainty Control*

In this experiment, uncertainty in output measurements is removed by using a single video footage which has been scaled to different image sizes. This video set is processed by each of the algorithms.



Matlabs tic toc functions are used with each algorithm for measurement of time to process 100 image frames. All variables and parameters are cleared and memory freed before the next algorithm or video is used. Tic is implemented after loading the avi video and toc at the end of the algorithms.

5.6 EXPERIMENT 2: ALGORITHM SHARPNESS EXPERIMENT

5.6.1 Aim

To compute and compare the sharpness of video processed by each algorithm under review in this study.

5.6.2 Background - Laplacian Operator

The Laplacian operator is to be used with a still image and provides a measurement of the image sharpness. This metric is described below and associates an image with a greater image sharpness as one which contains a smaller amount atmospheric turbulence effects.

In order to have a fair comparison between the original unprocessed turbulent video footage and video footage processed through the algorithms, the same image frame, or average of image frames must be used.

The Laplacian operator approximates the linear second order derivative of an image $f(m,n)$ in the row and column directions (m and n),

$$S(m, n) = \nabla^2 f(m, n) \equiv \frac{\partial^2 f(m, n)}{\partial m^2} + \frac{\partial^2 f(m, n)}{\partial n^2}, \quad (5.1)$$

with the image sharpening metric defined as:

$$S_4 = \frac{1}{MN} \sum_m \sum_n |S(m, n)|, \quad (5.2)$$

where the image is of size MxN.

5.6.3 Equipment

- Imaging system described in section 5.2,
- Algorithms detailed in chapter 4 implemented in Matlab, and
- Computer on which to run Matlab and algorithms.

5.6.4 Method

1. Using the imaging system, capture and record raw avi (Audio Video Interleave) video footage containing at least 50 image frames with a 10ms or less exposure time. Image size recorded must be 640x480 pixels or greater.
2. Crop the video to an image frame size of 640x480.
3. Process the video using each algorithm and save the processed videos.
4. Calculate the Laplacian Operator Image Sharpness value for the first 50 frames in each processed video using equation (5.1) and (5.2). Graph the resulting values and calculate the average image sharpness indicative value.
5. Repeat point 4 for the unprocessed video.
6. Analyse and compare results.

5.6.5 Experiment Relevance

Since atmospheric turbulence blurs captured image due to the atmospheric PSF, this sharpness experiment may provide insight as to the amount of image blur removed by the algorithms and how much detail has been brought forward.

Sharpness is important in order to make out finer detail within the image. By calculating and graphing the sharpness value for multiple frames, information can be obtained as to the videos consistency in sharpness.

5.6.6 *Expected Outcome*

It is expected that there will be rippling of the sharpness value over the 50 frames for the processed and unprocessed videos due to the spatial and time varying nature of the atmospheric PSF causing differing blurs in each video frame.

The Illuminance-Reflectance adjustment based algorithm with Wiener filtering is expected to have the best sharpness value because during implementation and testing of the algorithm the result was noticeably sharper with detail hidden in shadows enhanced quite an amount.

5.6.7 *Uncertainty Control*

This experiment uses a single atmospheric turbulence affected video for all the algorithms. The video is processed by each algorithm and image sharpness measurements taken across the same span of image frames.

This ensures that each algorithm is exposed to the same image scene and identical atmospheric conditions. By measuring image sharpness across the same span of image frames, results between algorithms are directly comparable. None of the algorithms shift or delay image frames, i.e. the input frame number n forms part of the output frame at index n .

5.7 EXPERIMENT 3: ALGORITHM ABERRATION EXPERIMENT

5.7.1 Aim

To determine and compare the amount of aberration present in processed and unprocessed video due to atmospheric turbulence.

5.7.2 Equipment

- Imaging system described in section 5.2,
- Algorithms detailed in chapter 4 implemented in Matlab,
- Computer on which to run Matlab and algorithms, and
- Chequered target of uniformly sized black and white squares.

5.7.3 Method



1. Make an A0 sized chequered target as shown in Figure 5-3 with 20cm sized black and white squares.

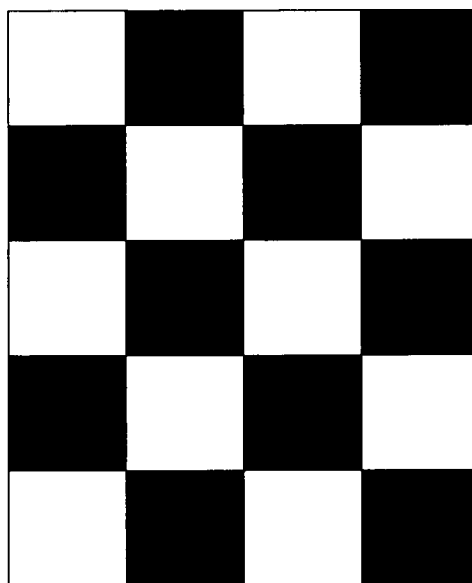


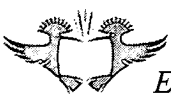
Figure 5-3: Checkerboard target example

2. Set the target at a distance of 1.2km from the imaging system.
3. Measure and record atmospheric conditions.
4. Setup the imaging system and capture video footage of the chequered target through atmospheric turbulence.
5. Process the captured video footage with each of the algorithms and save the processed videos.
6. Using the unprocessed video, display the frame average of the first 50 frames.
7. Measure and record the number of pixels a square spans. To ensure accuracy take measurements from multiple squares and calculate the average.
8. For unprocessed and all processed video footage follow the points below.
9. Using the first 50 frames of the video, construct an image consisting of the maximum pixel values.
10. Measure the number of pixels the white blocks span in the horizontal direction and calculate the average.
11. Use the reference calculated in 7 to determine the maximum horizontal displacement in mm.
12. Measure the number of pixels the white blocks span in the vertical direction and calculate the average.
13. Use the reference calculated in 7 to determine the maximum vertical displacement in mm.
14. Analyse and compare results.

5.7.4 *Experiment Relevance*

Atmospheric turbulence not only causes blurring of the object being viewed but also diverges light refracted from the object. This causes the object to appear to be moving (shaking) and warping. The movement is referred to as aberration.

This experiment allows one to measure and compare vertical displacements to horizontal displacements. It also allows a measurement as to the amount of aberration that is able to be suppressed by the algorithms.



By suppressing the aberration, the video footage is smoother over image frames and allows objects to be viewed much easier. Suppression of aberrations may help preserve object geometry by removing warping and distortion of the object.

5.7.5 *Expected Outcome*

There is expected to be an improvement in video stabilization when comparing the processed to unprocessed video.

Algorithms that use the ratio averaging are expected to have very similar residual displacements since the ratio averaging is common component to the algorithms and is the main factor in suppressing geometric distortions.

It is expected that horizontal displacement may be slightly greater if there is a cross wing across the optical path.

5.7.6 *Uncertainty Control*

A single video footage of the captured checkerboard target is processed by each of the algorithms. This ensures that the algorithms all process video of the target containing identical atmospheric turbulence conditions.

The same frame average reference image is used when measurements are taken between the different processed videos to ensure that only the algorithms themselves affect horizontal and vertical aberrations.

The imaging system is protected from wind ensuring any motion and aberration is due to the atmospheric turbulence acting over the optical path.



5.8 EXPERIMENT 4: ALGORITHM MTF EXPERIMENT

5.8.1 Aim

To calculate and compare the MTF for processed and unprocessed video of atmospheric turbulence.

5.8.2 Background

This experiment makes use of multiple spatial frequencies and the modulation transfer function of the different frequencies is determined using:

$$\text{MTF} = \left(\frac{I_{\max} - I_{\min}}{I_{\max} + I_{\min}} \right) / \left(\frac{W - B}{W + B} \right), \quad (5.3)$$

where I_{\max} is the maximum intensity value within that frequency and I_{\min} is the minimum intensity. W is the maximum luminance for white areas and B is the minimum luminance for the dark areas. A higher modulation index is more desirable as in a grey level image the darkest and lightest grey areas will be further apart and details will be more pronounced.

The W and B intensity values are obtained from very low frequencies.

5.8.3 Equipment

- Imaging system described in section 5.2,
- Algorithms detailed in chapter 4 implemented in Matlab,
- Computer on which to run Matlab and algorithms, and
- Three charts across which have spatial frequencies from 415mm/lp (first chart) to 10mm/lp (third chart).

The spatial frequency feature charts consist of sinusoidal printed patterns of different frequencies. Each frequency chart is printed on a A0 sized vertically positioned white page. The lowest spatial frequency chart is divided into four rows while for the higher frequency charts the page is divided into two columns and four rows. In each section a sinusoidal spatial pattern is printed. The lowest spatial frequency is at the top left corner and frequencies increase as you travel from each section from left to right and top to bottom. Thus the highest spatial frequency is found in the bottom right hand corner for that page.

There are three spatial frequency A0 sized charts. A low frequencies chart having periods of 415, 208, 138, and 104 mm/lp corresponding to 2, 4, 6 and 8 visible sinusoidal waves.

The rest of the frequencies have a period range from 92 mm/lp to 10mm/lp.

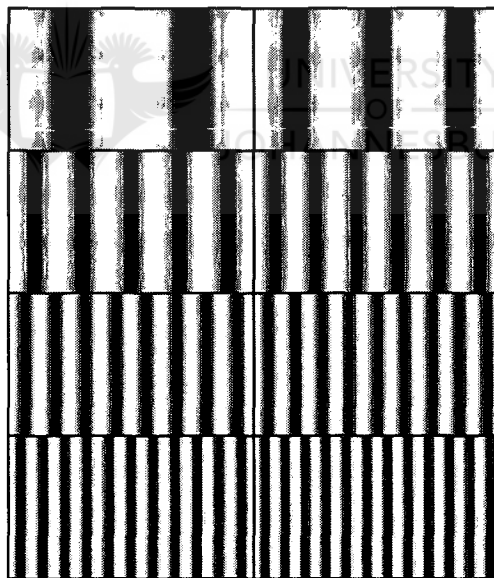


Figure 5-4: Spatial frequency chart example.

5.8.4 Method

1. Create the described spatial frequency charts which will be the videoed targets in this experiment.

2. Set the targets at a distance of 1.2km from the imaging system, as well as a low frequency reference object (An A4 sized printed black page and an A4 sized unprinted white page, both having intensities corresponding to the sinusoidal charts).
3. Measure and record atmospheric conditions.
4. Setup the imaging system and capture video footage of the chequered target through atmospheric turbulence.
5. Process the captured video footage with each of the algorithms and save the processed videos.
6. For each of the video footages repeat the following steps.
7. Record the minimum intensity values across 50 frames as well as the maximum intensity values.
8. Record more than 20 maximum and minimum values for each spatial frequency using the images obtained from step 7.
9. Calculate the minimum and maximum intensity average using the 20 or more values for each frequency.
10. Record the low frequency W and B intensity values from the white and black A4 pages respectively. Calculate the values using an average of values captured from the centre of the pages.
11. Using the W, B, maximum and minimum intensity values for the different frequencies construct and graph the MTF using equation (5.3).
12. Analyse and compare the results.

5.8.5 Experiment Relevance

Atmospheric turbulence causes blurring of an image and thus suppression of higher spatial frequencies.

The MTF allows a measure of the degradation in the clarity of spatial frequencies as the frequency increases. Information may then be obtained as to the effect the algorithms have on the MTF.

The MTF allows a way of measuring improvement in detail within the video by the algorithms.

5.8.6 Expected Outcome

The MTF is expected to drop off in a downward curve as spatial frequency increases until a cut off frequency is reached.

It is expected that there will be a slight improvement in the MTF curve for the algorithms particularly the Laplacian Operator based Wiener filtering algorithms.

An expected MTF curve the various video footages will follow is shown in figure 5-5.

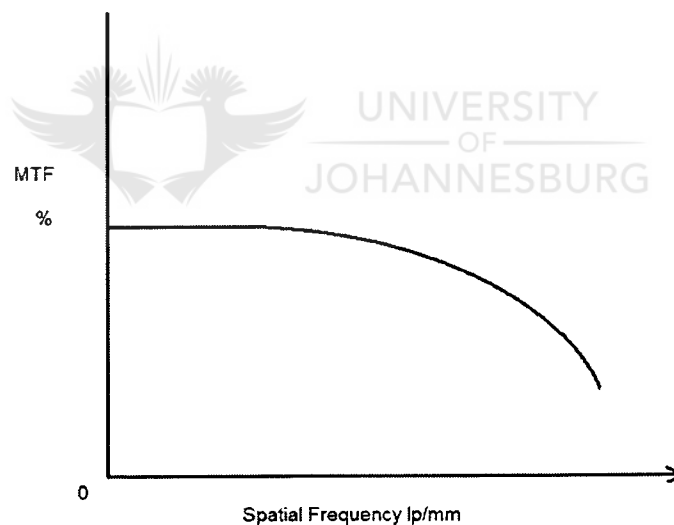


Figure 5-5: Expected MTF curve.

5.8.7 Uncertainty Control

Video footages for the high, mid, and low spatial frequencies is captured. Each of the algorithms processed these three videos. Ideally a single video footage should be containing all the spatial frequencies should be used, however due to the size of the spatial frequency charts and field of view this was not possible. An assumption is

therefore used that over the interval in which the required video is recorded, atmospheric conditions will not drastically change to have a significant difference between video footages. This assumption is considered reasonable if the time span required is less than half an hour and if atmospheric conditions in the last hour have not been noticeably different.

High spatial frequency results will be comparable between algorithms even if the assumption is false, the same applies to the mid and low frequency results. If the assumption is false the expected MTF curve may not follow the smooth curve as expected but doesn't stop result comparisons between algorithms for each of the spatial frequency bands.

5.9 EXPERIMENT 5: ALGORITHM MSE AND PSNR EXPERIMENT

5.9.1 Aim

To obtain MSE and PSNR measurements through the construction of a reference image of atmospheric turbulence affected video and processed video and see how they compare to MTF and sharpness measurements.

5.9.2 Background

The MSE metric is a classical metric used in determining image compression and decompression effectiveness. For this metric a reference image frame is necessary. The MSE of the turbulent unprocessed or processed video footage from the algorithms is defined as:

$$\text{MSE}(\mathbf{i}_d) = \frac{1}{MN} \sum_{m,n} (\mathbf{i}_d - \mathbf{i}_i)^2, \quad (5.4)$$

where, \mathbf{i}_d is the estimate or degraded image of the undistorted image, \mathbf{i}_i , of size $M \times N$.

The RMSE (Root mean square error) may also be calculated by simply taking the square root of the MSE.

The PSNR is also a classical error metric. The PSNR uses the RMSE for calculation.

The PSNR is defined as:

$$\text{PSNR}(\mathbf{i}_d) = 20 \log_{10} \left(\frac{255}{\text{RMSE}(\mathbf{i}_d)} \right), \quad (5.5)$$

for a grey level 8bit image.

5.9.3 Equipment

- Imaging system described in section 6.2,
- Algorithms detailed in chapter 4 implemented in Matlab,
- Computer on which to run Matlab and algorithms, and
- Chequered target of uniformly sized black and white squares.

5.9.4 Method

1. Make an A0 sized chequered target as shown in figure 6-2 with 20cm sized black and white squares.
2. Set the target at a distance of 1.2km from the imaging system.
3. Measure and record atmospheric conditions.
4. Setup the imaging system and capture video footage of the chequered target through atmospheric turbulence.
5. Crop the video footage so only the chequered blocks are visible. (crop out background around chequered target)
6. Process the cropped captured video footage with each of the algorithms and save the processed videos.

7. Construct the reference image by first performing a histogram stretch on the on the cropped unprocessed video. Perform ratio averaging with a $1/3$ lambda value for geometric stabilization. Using thresholding, set the white and black blocks to the maximum and minimum intensity values to construct the chequered pattern estimate. Save the reference video.
8. For the cropped unprocessed and processed video perform the following steps.
9. Perform a histogram stretch on the video to correct intensity values.
10. Use equation (6.4) and corresponding frame in reference video to calculate MSE for each frame.
11. Use equation (6.5) and calculated MSE to calculate PSNR.
12. Calculate average MSE and PSNR over image frames.
13. Analyse and compare results.

5.9.5 Experiment Relevance

MSE and PSNR metrics are classical metrics in signal processing. However, they require a reference, and in this image processing problem exact knowledge of the original object construction to be used as a reference is not known.

This experiment sets up to try and reconstruct a basic reference image and perform MSE and PSNR measurements. How the measurements correspond to MTF and sharpness measurements may provide insight as to whether MSE and PSNR metrics may be used for this particular problem with simple object reference construction.

5.9.6 Expected Outcome

It is expected that there will be very little difference in the MSE and PSNR measurements between the algorithms, however the processed algorithms will show a slightly better improvement.

5.9.7 Uncertainty Control

Same video footage is processed by each algorithm to ensure that the same level of atmospheric turbulence is processed. The same reference image is used when performing MSE and PSNR measurements for all the unprocessed and processed video. This ensures that changes in MSE and PSNR measurements are due to the algorithms themselves and their processing of the unprocessed video footage.

The imaging system is protected from wind ensuring any motion and aberration is due to the atmospheric turbulence.

5.10 EXPERIMENT 6: WIENER FILTERING LAMBDA SELECTION EXPERIMENT

5.10.1 Aim

To determine the necessity of calculating a lambda value for every frame in the Wiener filtering based algorithms.

5.10.2 Equipment

- Imaging system described in section 6.2,
- Algorithms detailed in chapter 4 implemented in Matlab,
- Computer on which to run Matlab and algorithms,
- Chequered target of uniformly sized black and white squares, and
- Spatial frequency charts.

5.10.3 Method

1. Setup the chequered and spatial frequency charts 1.2km from the imaging system.

2. Capture video footage of the charts.
3. Process the video footage using the Wiener filtering using the Laplacian Operator based algorithm. Save the processed video.
4. Record the average λ value used across 50 frames.
5. Modify the algorithm so the average λ value is used for the Wiener filtering of each of the 50 frames.
6. Process the captured video using the modified algorithm. Save the processed video.
7. Calculate frames per second processing performance, image sharpness, image aberration, MTF, MSE and PSNR as discussed in the previous experiments. Calculate for both of the processed and unprocessed video footages.
8. Analyse and compare results from the two processed videos against unprocessed video. Note any differences between the two processed videos.

5.10.4 Experiment Relevance

Algorithm speed is important if the algorithm is to be used in a real time system. A faster algorithm will be able to be run real time on more compact lower energy requirement hardware than a slower algorithm.

Calculating the λ used in the atmospheric OTF (Optical Transfer Function) construction for every image frame requires quite a bit of processing. If the λ value varies by a very small amount across image frames, the λ value could be updated after only a certain number of frames.

This experiment will provide insight into the image quality hit that would be taken for a performance gain when using a single lambda value for 50 image frames.

5.10.5 Expected Outcome

It is expected that the image quality hit with regard to sharpness, MTF and MSE PSNR measurements will be small compared to the increase in performance.

This is from observations of the λ value fluctuating slightly when video footage is processed. Small fluctuations however may actually be quite significant and this experiment will show if that is the case.

5.10.6 Uncertainty Control

Uncertainty control follows same principles used in experiment 1, 2, 3, 4, and 5. Same input parameter control principles are followed since the FPS performance, image sharpness, aberration, MTF, MSE and PSNR experiments are rerun.

5.11 EXPERIMENT 7: FILTERING RECEIVED LIGHT EXPERIMENT

5.11.1 Aim

To determine whether filtering atmospheric turbulence affected light received by the telescope before the camera captures it helps to improve contrast. This experiment separately looks at the use of a red filter and polarising filter.

5.11.2 Equipment

- Imaging system described in section 5.2,
- Red 1.25" optical filter Wratten number #24,
- Polarised 1.25" optical filter,
- Computer on which to run Matlab and metrics,
- Chequered target of uniformly sized black and white squares, and
- Spatial frequency charts.



5.11.3 Method

1. Setup chequered and spatial frequency targets at scene 1.2km from imaging system,
2. Setup imaging system with no filters attached and focus on targets,
3. Capture video footage of targets through the turbulent atmosphere and save unfiltered video,
4. Setup imaging system with the red filter attached and refocus,
5. Capture video footage of targets through the turbulent atmosphere and save colour filtered video,
6. Remove red colour filter and setup imaging system with polarised filter,
7. Capture video footage and save polarised video,
8. Calculate image sharpness, MTF, MSE and PSNR for unfiltered and filtered video,
9. Analyse and compare results.

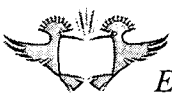
5.11.4 Experiment Relevance



Optical filters are small and require no processing or energy source. They may be useful as a preconditioning of received light to remove unwanted light which may cause noise and degradation in the received image.

Atmospheric turbulence causes blurring of the captured object and thus a decrease in contrast. As discussed in the literature study the atmospheric turbulence scatters and refracts light.

The above facts are used together with the knowledge that in photography, as reviewed in chapter 2, red filters are used for contrast enhancement of the captured image and the polarising filter can increase contrast by removing light scattered from a specific orientation (such as horizontal reflected light off water). The red filter allows just longer wave length light to be captured, and as discussed in chapter 2, longer wavelength light is refracted a smaller amount than shorter wavelengths.



The experiment thus analyses whether and how much an improvement the filters can have in the quality of the image.

5.11.5 Expected Outcome

It is expected that the filters will provide a better quality image in terms of the image sharpness, MTF, MSE and PSNR when compared to the unfiltered image. MTF is expected to increase further into the higher frequencies when using the red optical filter.

5.11.6 Uncertainty Control

Uncertainty control follows same principles used in experiment 2, 3, 4, and 5. Same input parameter control principles are followed since the image sharpness, aberration, MTF, MSE and PSNR experiments are rerun within this experiment for colour filtered video and polarised filtered video instead of algorithm processed video footage.

5.12 CONCLUSION

In this chapter various experiments have been detailed which allow for comparison to be made relating to the algorithms performance and image quality enhancements. Addition experiments relating to the algorithm performances and the atmospheric turbulence problem are also described. The Experiments have been designed in attempt to provide some repeatability of the results under similar atmospheric conditions.

The imaging System used in the experiments for capturing video footage and its setup has also been detailed along with the computer system platform used in the experiments.

The following chapter discusses some of the experimental issues that where encountered.

6 CHAPTER 6: RESULTS

6.1 IMPLEMENTATION ISSUES

When performing implementation of the algorithms it was expected that unforeseen implementation issues may arise. This section lists some of the issues encountered while implementing the various algorithms within the Matlab environment and the solution to those problems.

6.1.1 *Speckle Masking Issues*

A main problem encountered when implementing the Speckle Masking algorithm was discovered during its testing phase using test video footage, from which parts of the generated image contained artefacts and what can be described as smearing. This smearing would consist of a low frequency diagonal component within a sub-image region. This degradation is due to anisoplanatism as mentioned in [97] and [99]. To overcome this effect it is necessary to reduce the sub-image sizes. The degradation is notable particularly when a sub-image consists of a high frequency component amongst lower frequencies, an example being a thin telecommunications aerial surrounded by clear sky.

During implementation, a particular problem which related to the efficiency and realistic realization of the algorithm surrounded the recovery of the Fourier phases from the bispectrum. Since the Speckle Masking recovery is working with a two dimensional image rather than one dimensional projections, a four dimensional bispectrum object would be created. The bispectrum object contains a lot of redundant information and requires a large portion of memory, so a method was needed and devised which calculated and stored bispectrum values from the four dimensional bispectrum into a two dimensional array. Only the values that are needed for the phase recovery were calculated.

When performing apodization, the apodization function cannot be set to small (e.g. 20%) otherwise overlapping Hamming functions do not sum to one and result in a chequered grid of vertical and horizontal artefact lines.

6.1.2 Illuminance Reflectance Issues

The Illuminance reflectance mid tone enhancement describes the use of the following function in [93]:

$$H_{sig}(k) = \frac{1}{1 + e^{-ak}}, \quad (6.1)$$

however, this resulted in an inversion in the image and thus is possibly an error within the documentation. The following function used as an alternative which made greater theoretical sense and proved itself in practice and implementation of the algorithm is:


$$H_{sig}(k) = \frac{1}{1 + e^{ak}}. \quad (6.2)$$

6.1.3 Full Histogram Stretch

Since there is a PSF variation across the video frames there is a change in maximum and minimum pixel intensities. By performing a full histogram stretch to make use of the full 8 bit intensity range, a flickering in brightness sometimes becomes apparent in the output video image.

The flickering can be dampened by making use of the ratio averaging techniques as previously mentioned on the minimum and maximum intensity values.

6.1.4 Kurtosis Minimization

After implementation and thorough reanalysis of the kurtosis minimization code, it was found that with the test video footage (used as part of the testing phase of the implementation), the kurtosis minimization did not converge. The smallest λ value was always arrived at with higher λ values having larger calculated kurtosis.

This failure of convergence is mentioned in [100] and appears to be a known problem. The minimization is therefore dependant on the data content within the video, and the problem is noted as arising in cases when the content may consist of a large amount of low frequency data such as a large portion of sky in the image [100].

The solution to selecting a λ parameter is made by making use of the Laplacian Operator Image Sharpness metric from [101] and [102] for selection of the parameter based on image sharpness. This resulted in the construction of the Wiener Filtering using Laplacian Operator algorithm detailed in the preceding Algorithms Detail Design chapter.



6.1.5 General Programming Precautions

General programming precautions are those that should that should be kept in mind while programming. Some of the precautions in relation to the Matlab environment are described below.

Video footage which is to be processed and read using Matlab's 'aviread' function, needs to be in an uncompressed AVI (audio/video interleaved) format.

While working with video frame information, the data is in an 8 bit integer format and should be converted to double or other appropriate data type in order to maintain precision during calculations. After calculations have been completed the data is cast back to an 8 bit integer value (Assuming working with an 8 bit gray image or 24 bit colour image).



Most functions and associated code work with the normalized version of the image data rather than the 0-255 8 bit range, and care must be taken to normalize where necessary. This will obviously require the data type to be of type 'double'.

6.2 EXPERIMENTAL ISSUES

This section discusses some of the experimental issues encountered during design and implementation of the experiments and solutions used to overcome those problems.

6.2.1 *Providing for Comparability between Experiments*

Atmospheric conditions can change on a daily basis; this is evident since temperature, wind, barometric pressure and humidity readings differ daily as well as the amount of dust particles present in the air.

This is a problem if video footage used for each experiment is captured on separate days or even hours if results want to be compared between algorithms. Video footage captured on different days will capture different atmospheric turbulence and possibly bias one set of experimental results.

Since many of the experiments make use of the same imaging system and targets captured over the same distance, the simple solution is to capture all the required video footage for all the experiments within a 1 or 2 hour time span.

6.2.2 *Atmospheric Haze Degrading Seeing Conditions*

Dust Particles and other suspensions in the air (aerosols) absorb some of the light causing degradation in the seeing conditions. [12] These particles reduce visibility and the distance over which objects can be viewed. From observations during January 2008, Haze was particularly prominent during the afternoons when compared to

mornings; however some mornings aerosols such as fine water molecules were present. The effect the haze has on captured images is to reduce the images contrast.

This haze problem created some difficulty in timing and judging a clear late morning on which to capture video footage. Rain on a previous day with clear skies the next day helped to reduce aerosols in the air.

The solution was to judge a week when there will be clear skies and check each morning the atmospheric conditions. Suitable conditions were a visibility of greater than 10 km with clear skies in the morning, light wind less than 10 knots and having preferably rained in the last couple days.

6.2.3 *High Level of Atmospheric Turbulence*

Initially a length of 2 km was chosen between the imaging system and targets. This distance was selected since it allowed a 2x Barlow lens to be used while still allowing the target to be within the cameras field of view. The site at 2 km was also convenient allowing the Telescope to be setup in a secure location with power supply and clear visibility of the targets.

Unfortunately after testing the path length by capturing and reviewing video footage of the targets, the level of turbulent atmosphere was considerable. Some of the authors' algorithms seemed to be developed for light or medium turbulence. [1, 2, 5, 8] The high level of turbulence can hardly be considered surprising since it was the middle of summer, but it was still unexpected since some test video footage captured during the end of winter contained significantly less turbulence.

The solution was to find a site closer to the imaging system and postpone capturing video footage until the end of February, a task which involved judgement of conditions discussed in 7.3. A new site location was found at a distance of 1.2 km from the imaging system. Turbulence was still considerable but more manageable and viewable over the spatial frequency charts.

6.2.4 *Effects of Wind and Finding a Suitable Site*

Wind can cause instability in the captured image even with the imaging system appearing to be sturdy. This is because of the high optical magnification that is used, a slight movement of the imaging system results in a large displacement in the viewed image.

To ensure that wind which could move the imaging system did not corrupt video footage of the targets through atmospheric turbulence, a requirement was set to setup the imaging system in a protected location from the wind. Video footage was also captured only when the wind was little more than a breeze.

Requiring a protected and secure location in which to setup the imaging system provided some difficulty as it needed to also have a clear view of a target site at 1 to 2 km. Such a location was however found with the telescope setup and looking out through the open balcony doors of an upstairs bedroom. This also provided power for the telescope and IP camera although it was not necessary since a small generator or ups could be used at a location that did not have power.

6.2.5 *Creating Reference Image for MSE*

As discussed in the experimental setup, the MSE metric requires a reference image which is what would be viewed if atmospheric turbulence was not present. Slight wind causes large horizontal shifting and aberrations. The construction of the target was known (Chequered object), however the exact location of the reference target should be in the video frame is difficult to find.

A method was thus devised as discussed in the experiment MSE and PSNR measurement to reconstruct the reference image for each video frame. By cropping out background images and having a video containing only the chequered target, there is a priori knowledge of the construction of the reference video as required to construct a reference used by the MSE.



6.2.6 *Maintaining Short Exposure Times*

The Power Spectrum and Image Registration based algorithms require video frames to be captured with an exposure time of 10 ms or less to ‘freeze’ the atmospheric turbulence. To meet this requirement it was necessary to ensure that there was an adequate collection of light so that video frames would not be under exposed when using these low exposure times.

The solution to ensuring the video frames were captured at a low exposure setting, was to use a large aperture telescope, ensuring a large collection of light, and capture video footage on a clear day. The IP camera was additionally set to its high speed setting which selects exposure time of 10 ms or less based on the amount of light.

6.2.7 *Separate Video Footage Captured for Frequency Targets*

The spatial frequencies that were to be recorded for the MTF experiment were spread across three separate charts due to the low spatial frequencies and the video cameras field of view. Video footage captured for the four highest frequencies, another for the mid range frequencies, and one for the four lowest frequencies.

To try and avoid discrepancies in the level of atmospheric turbulence present in each of the video footages, video footage was captured within an hour assuring no changes in wind conditions.

6.3 SYSTEM PLATFORM

This section provides the specifications of the system platform on which the algorithms and related experiments are implemented, tested and analysed. This section is split into two parts: The hardware components/specifications of the system and the software components/specifications.



6.3.1 Hardware Specifications

Relevant computer hardware specifications:

- 2.8GHz Intel Core2Duo CPU socket LGA775,
- 2 GB DDR2 RAM running at 800mhz,
- Asus P5B deluxe Motherboard.

Imaging system hardware:

- Celestron Nexstar 8 SE 8 “ Schmidt-Cassegrain Telescope,
- AreCont Vision 3100 IP Camera,
- 1.25” to c/cs adapter,
- 1.25” red colour filter, #24 Wratten number,
- 1.25” polarised filter.

6.3.2 Software Specifications



Experiments make use of the following software:

- Windows XP professional Service Pack 2,
- Video is captured using AreCont Vision’s supplied software,
- Matlab 7 in which algorithms are implemented and experiments performed, and
- Microsoft Office 2007. Some results are exported to Microsoft Excel, data tabularised and graphs created from the experimental data.

6.4 ATMOSPHERIC CONDITIONS RECORD

Video footage for the experiments was taken under the following atmospheric conditions.

Date: 28 February 2008

Time: Between 9:30 and 10:30

Temperature: 25°C

Wind speed: 9km/h (5 knots)

Dew Point: 12°C

Humidity: 43%

Pressure: 1023 hPa

The optical path distance was 1.2km across suburban terrain.

6.5 EXPERIMENT 1: ALGORITHM PERFORMANCE EXPERIMENT RESULTS

The measurement of an algorithm's performance is taken in regard to the average amount of time it takes to process a single image frame. This section presents the results pertaining to the algorithms FPS (Frames per second) performance with an increase in video frame image size.

Additional individual graphs and tables for each of the algorithms relating to these experiment results may be found in Addendum H.

The Speckle Masking algorithm takes a series of image frames and produces a single image. The results presented for the Speckle Masking are the number of seconds it takes to produce an image. In the Speckle masking algorithm, sub image size can be selected. This is the size of the overlapping images that the image frames are broken up into. The smaller the sub image size, the smaller the number of artefacts present in the video.

Figure 6-1 presents the Speckle Masking FPS performance results for three different sub image sizes.

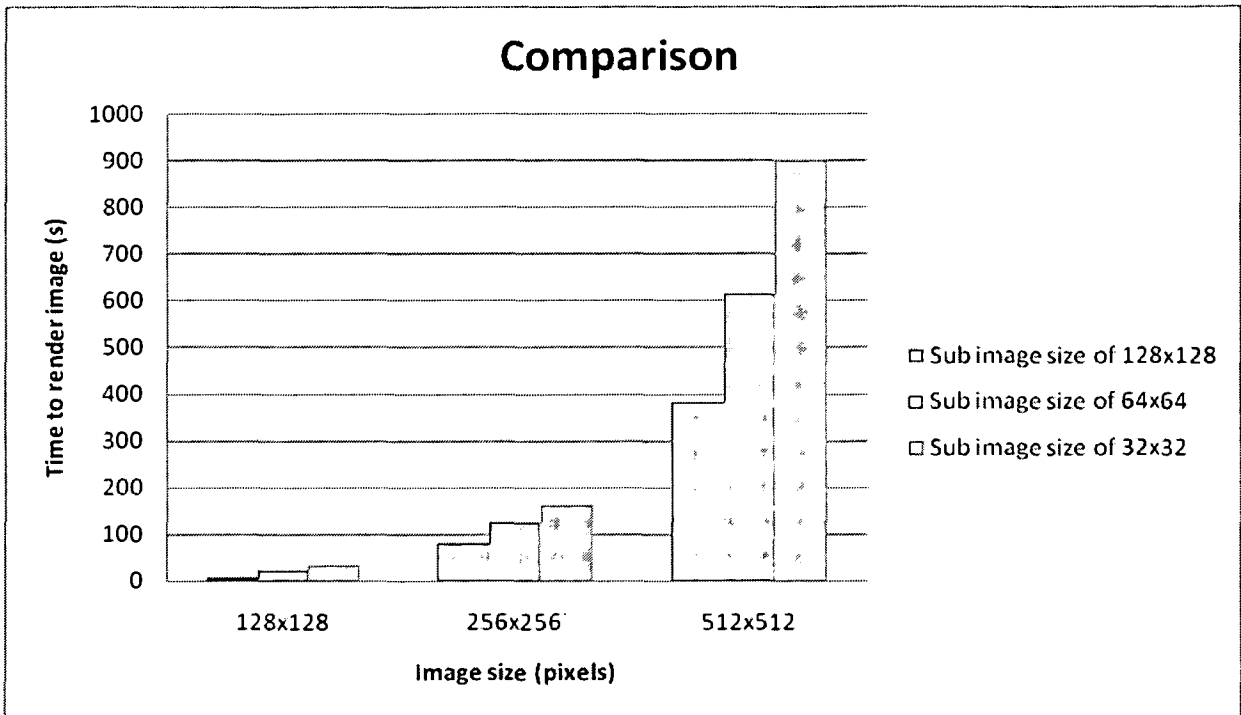
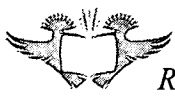


Figure 6-1: Speckle Masking sub image size comparison performance graph

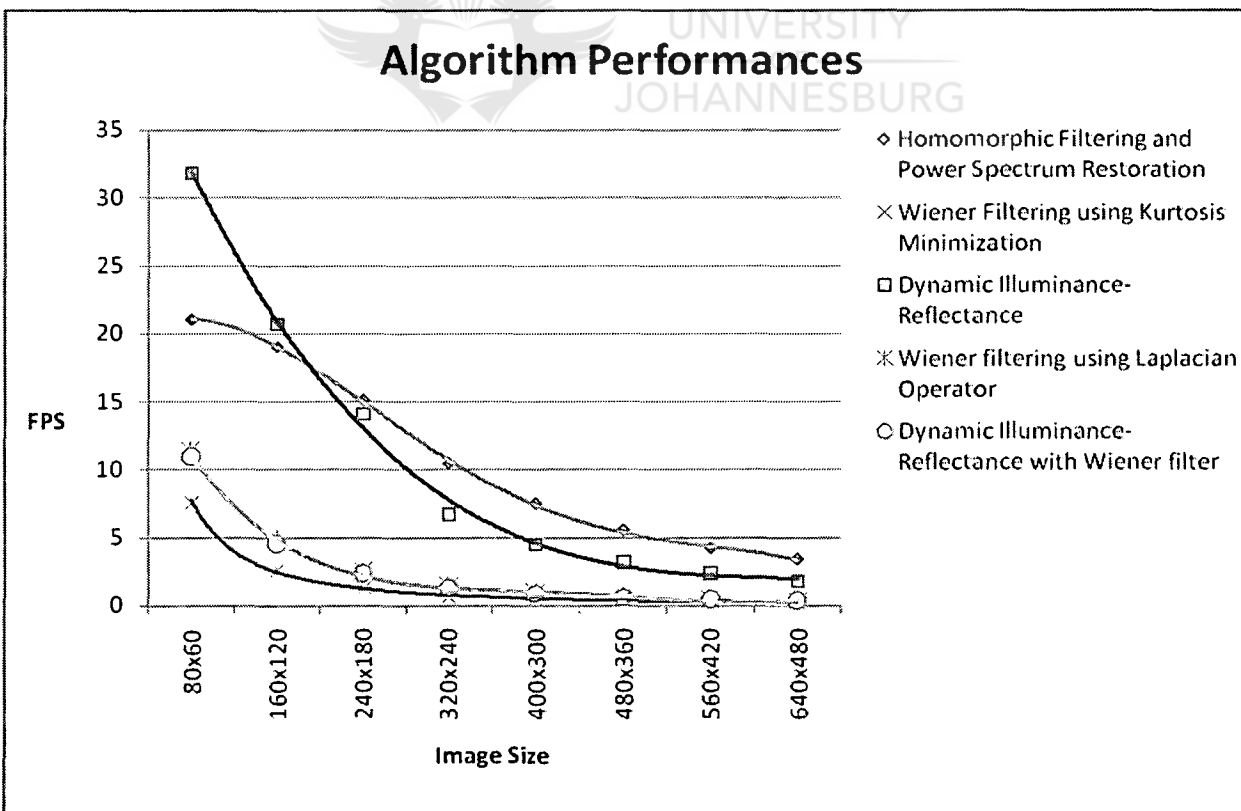


Figure 6-2: Algorithm comparison FPS performance graph

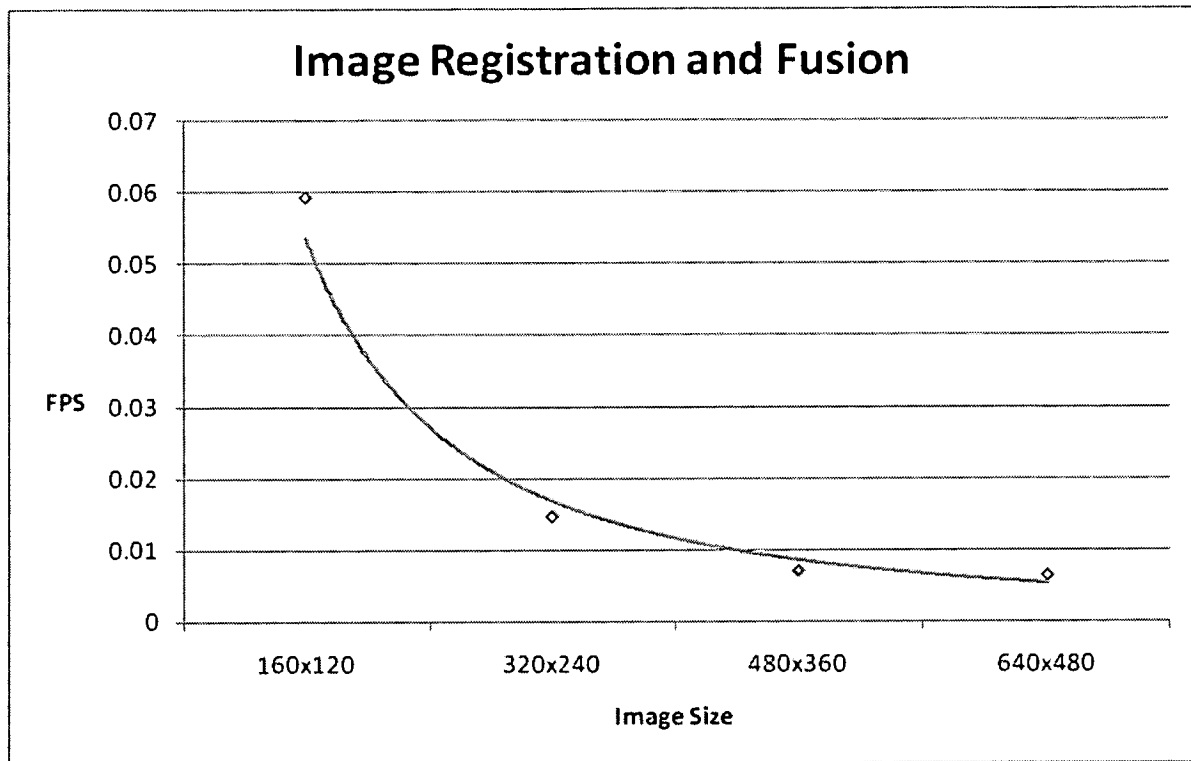


Figure 6-3: Image Registration and Laplacian Pyramid Image Fusion FPS performance graph

Figure 6-2 shows the FPS performance results for the algorithms. The Image Registration and Fusion algorithms FPS performance would appear to lie along the x-axis of Figure 6-2, so it is included in Figure 6-3.

Comparing the algorithms to one another we notice how they experience a general trend in which the frames per second drop off with an increase in image size. A sharper drop off indicates a more complex processing relationship with the image size. This sharp decrease in performance can be seen in the Image Registration and Fusion based algorithm as well as the Wiener filtering with Kurtosis Minimization algorithm.

An interesting performance curve which was verified through repeated results is that of the Homomorphic Filtering and Power Spectrum based method. The performance drops off gradually in much more of a linear fashion, whereas the performance curves of the other algorithms tend to better parallel one another. This shows that the algorithm is not highly dependent on image size. Figure 6-2 illustrates the trend by showing a plot of the algorithm performance curves on a single set of axis. The solid lines indicate the trend in the corresponding algorithms data points.



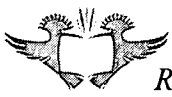
The Speckle Masking algorithm is the slowest, and requires a lot of memory and processing time. The time it takes to produce an image increases sharply as image size increases. (Figure 6-1). This is due to the algorithm being highly dependent on image size as the size of the bi-spectrum used for image reconstruction is determined from the image dimension. By doubling the image size results in an increase in the number of bi-spectrum elements by a factor of 4. These bi-spectrum elements are necessary for reconstruction and not redundant.

The Image Registration and Laplacian Pyramid image fusion algorithm is also very slow in comparison to the other algorithms. Referring to Figure 6-2, the Image Registration and Laplacian Pyramid algorithm would show as a line along the x axis. This poor performance in relation to the other algorithms is because of the way it attempts to remove or suppress geometric distortions in the video footage. The method uses elastic image registration which is complicated and requires a large amount of processing resources. The algorithms in Figure 6-2 however use ratio frame averaging for geometric distortion suppression which is a vastly simpler method.

On comparison of the Wiener filtering using the Laplacian operator method and the Dynamic Illuminance-Reflectance Atmospheric Turbulence suppression with Wiener filtering method, there is very little difference in performance, and both appear to lie on the orange trend line in Figure 6-2. This is interesting since the Dynamic Illuminance-Reflectance Atmospheric Turbulence suppression with Wiener filtering method uses the same Laplacian operator Lambda selected Wiener filtering. The reason is because the Illuminance-Reflectance processing component requires much less processing than the Wiener filtering component which searches for the best lambda value used in the OTF construction.

6.5.1 Concluding remarks

Results showed that for larger image sizes, the homomorphic filtering and power spectrum based algorithm had the highest frames per second. The algorithms using ratio averaging for geometric stabilisation ran much faster than the image registration and speckle masking methods, indicative of a much greater simplicity in processing and complexity than the latter mentioned methods.



The Dynamic Illuminance-Reflectance atmospheric turbulence suppression algorithm showed good performance when compared to the other algorithms and when coupled together with Wiener filtering it showed to add little overhead in terms of processing requirements when compared to the Wiener filtering method on its own.

With the conclusion and analysis of the different algorithms performance with regard to processing requirements and FPS, attention is turned to image quality and stability results of the algorithms.

6.6 EXPERIMENT 2: ALGORITHM SHARPNESS EXPERIMENT RESULTS

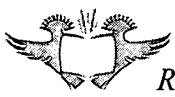
Atmospheric Turbulence causes a blurring of captured images. The blurring is suppressed by the algorithms resulting in an image which should be sharper with the recovery of suppressed high spatial frequencies.

The experiment results in this section consist of image frame sharpness measurements using a normalized cumulate of the Laplacian Operator performed on the image. In Addendum I, graphs of the sharpness value vs. frame number for 50 image frames are presented along with the average sharpness indicative value. A higher value would indicate an image is sharper than a lower value. Graphs are presented for each algorithm, as well as the original unprocessed video footage. The sharpness measurements for the unprocessed video provide a control reference.

Listing the average image sharpness measured from highest to lowest we have the following result as shown in Table 6-1 and illustrated on Figure 6-4.

All the methods were able to increase image sharpness with Illuminance-Reflectance based algorithms showing the greatest increase. This is due to the way the methods enhance detail hidden in flat zones such as shadows causing an increase in the image values when the Laplacian operator is performed.

The Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener Filtering algorithm has a far higher image sharpness. The reason for this high value is that



before the image is sharpened up using the Illuminance-Reflectance adjustment and enhancing detail hidden by shadow and overexposure, Wiener filtering is performed. The Wiener filter removes a lot of the image blur resident in the video due to the turbulent atmosphere. Image blur causes a flattening of the image intensities and decrease of image sharpness.

Algorithm	Average Image Sharpness Measurement
Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener Filtering	21.726
Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression	13.333
Image Registration and Laplacian Pyramid Image Fusion	11.256
Wiener Filtering Using Laplacian Operator	9.166
Speckle Masking	9.075
Homomorphic Filtering and power spectrum restoration based method	8.193
No algorithm - original unprocessed video	5.192

Table 6-1: Average image sharpness comparison, highest to lowest.

The rippling across image frames as seen in the graphs in Addendum I, is possibly due to the fluctuation of the atmospheric PSF across video frames and subsequent algorithm processing. The Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener Filtering algorithm shows a greater amount of rippling in the image sharpness value (Refer to Addendum I). Comparing frames 35-40 of the Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener Filtering algorithm there appears some correlation in the rise and falls to some of the other algorithms. It is possible that the Wiener

filtering in the method increases the discrepancy, and would be due to the Wiener filtering boosting some high frequency noise which is further increased through the Dynamic Illuminance-Reflectance adjustment. An increase in high frequency noise can cause an increase in image sharpness measurement.

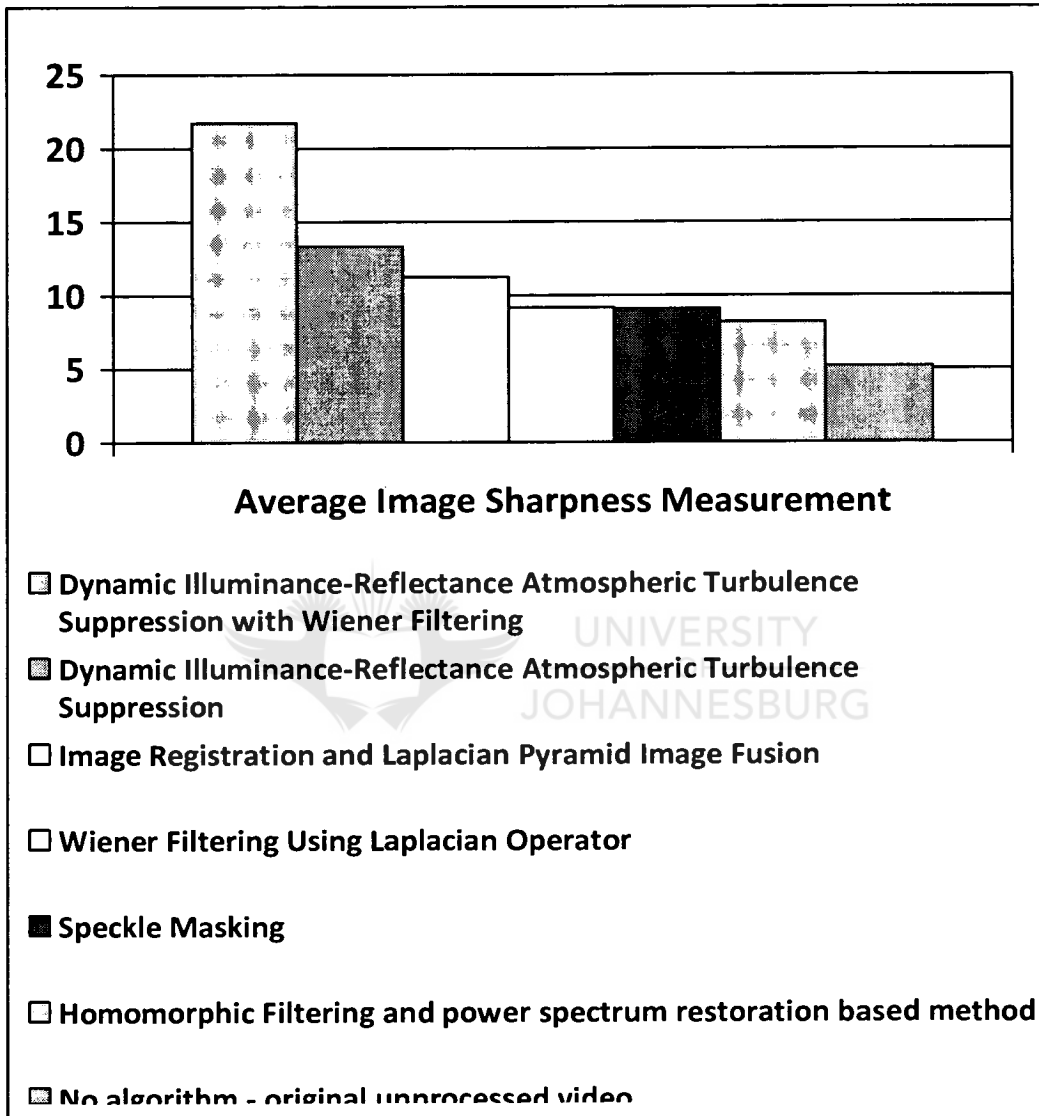


Figure 6-4: Average image sharpness comparison, highest to lowest.

While image Sharpness measurements are able to be made it is clear that they cannot be used as an image quality indicator on their own as high frequency noise can increase image sharpness measurements and distort results.

6.6.1 *Concluding Remarks*

This section has presented the results for image sharpness produced by the algorithms as well as the image sharpness for the unprocessed video. The unprocessed videos sharpness measurement acts as a control reference so comparisons may be made. All processed video showed an increase in image sharpness. The Illuminance-Reflectance based algorithms showed the greatest increase in sharpness primarily due to their ability to enhance detail hidden in flat tonal areas such as shadows and overly bright areas. Addendum I contains individual graphs for the algorithms' image sharpness across 50 image frames.

As discussed, high frequency noise may cause an increase in image sharpness measurements and for this reason image sharpness measurements should not be used as an image quality indicator on its own.

6.7 **EXPERIMENT 3: ALGORITHM ABERRATION EXPERIMENT RESULTS**

These results are used for the interpretation of a video's stability. The less warping and shaking there is in the video, the greater its stability will be. This is important when viewing the video footage as objects in the video will be easier to identify as they will not be moving and shaking across video frames which could appear slightly blurred at high frame rates.

Table 6-2 and Figure 6-5 display the results for horizontal aberration present in the video. Table 6-3 and Figure 6-6 present the vertical aberration results. Additional individual tables for each algorithm relating to the aberration measurements and results can be found in Addendum J.

The results show a greater horizontal than vertical displacement; this indicates a far greater horizontal movement possibly due to wind effects interacting with the heat waves. After processing, the video footages still contained a fair amount of horizontal displacement. This residual displacement shows that there is a low frequency horizontal displacement present.



Horizontal Displacement Results	
Algorithm	Average Displacement (mm)
Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression	41.176
Wiener Filtering using Laplacian Operator	41.667
Image Registration and Laplacian Pyramid Image Fusion	45.588
Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener Filtering	46.078
Homomorphic Filtering and Powers Spectrum based	50.000
Unprocessed video	54.412

Table 6-2: Video aberration results – Horizontal displacement.

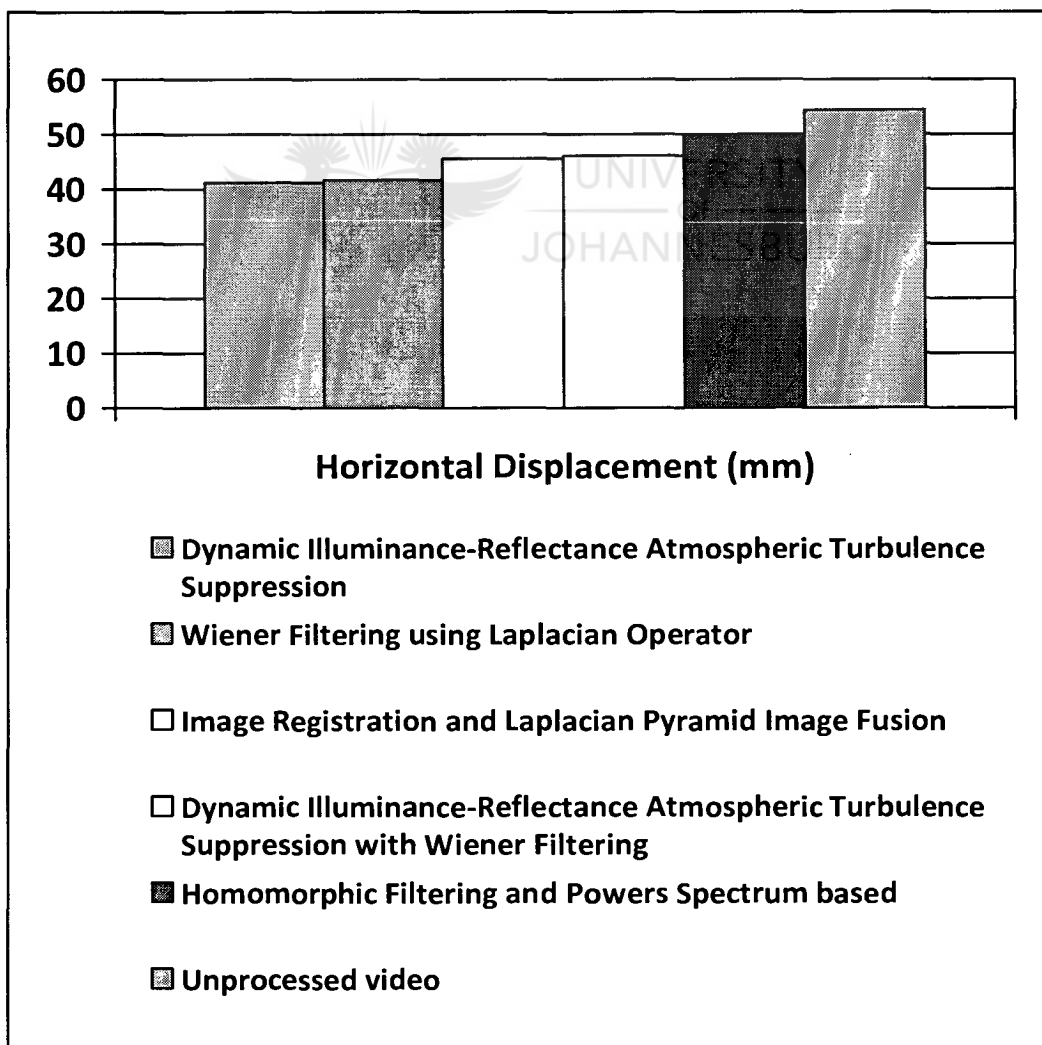


Figure 6-5: Video aberration results – Horizontal displacement.

Vertical Displacement Results	
Algorithm	Average Displacement (mm)
Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression	13.235
Wiener Filtering using Laplacian Operator	16.667
Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener Filtering	19.118
Image Registration and Laplacian Pyramid Image Fusion	19.608
Homomorphic Filtering and Powers Spectrum based	23.529
Unprocessed video	24.020

Table 6-3: Video aberration results – Vertical displacement.

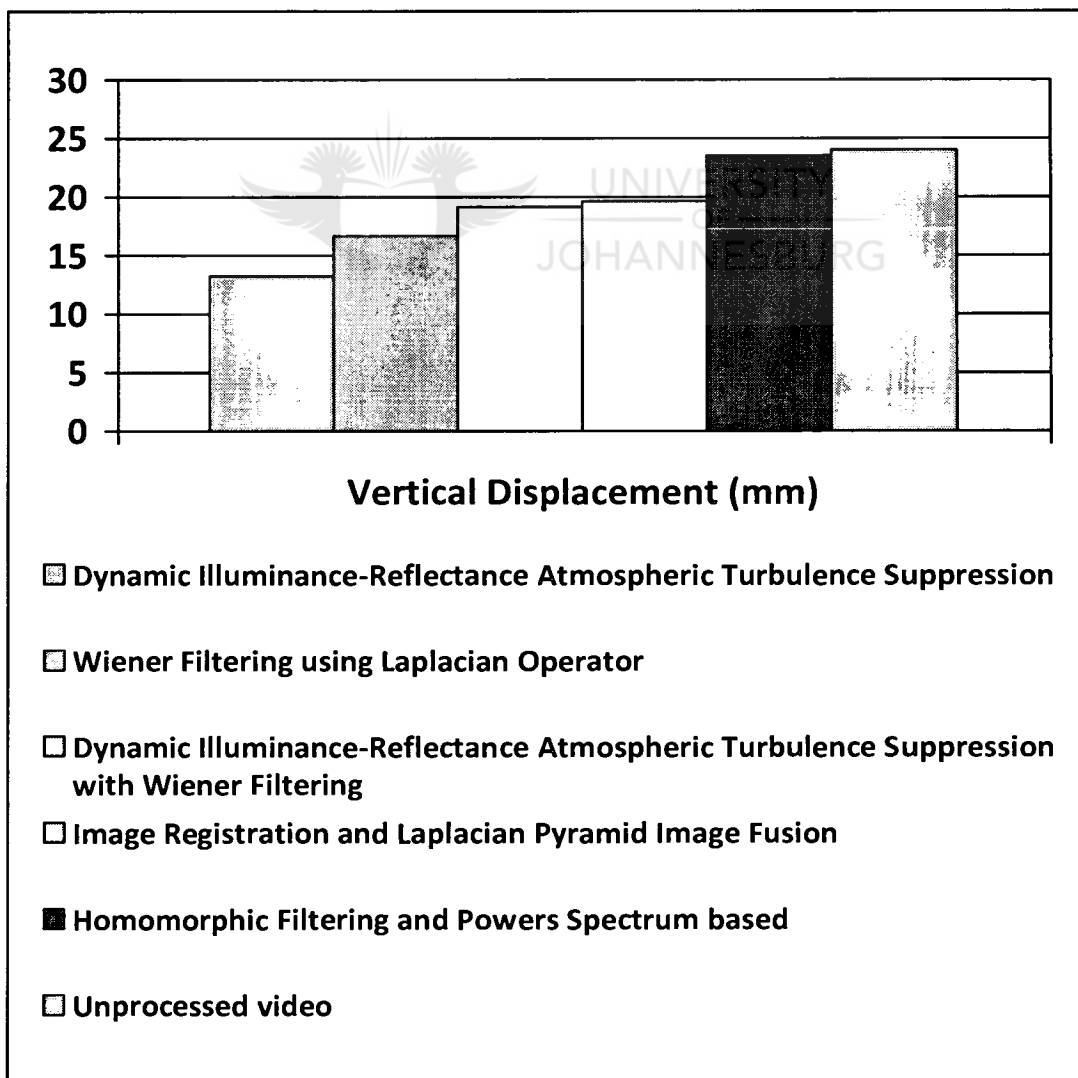
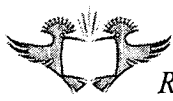


Figure 6-6: Video aberration results – Vertical displacement.



The results show a reduction in the horizontal and vertical displacements for all algorithms, however the results show very little improvement from the Homomorphic filtering and Power Restoration based method. Results are possible distorted due to the high frequency noise that is present in the processed video due to the power spectrum restoration and Homomorphic filtering outputted Fourier modulus information. When a histogram stretch is performed, the black squares contain some lower intensity high frequency noise and results in the black square average intensity being offset above the minimum intensity. This creates a problem when a threshold is performed and accurate determination of square sizes is required.

The Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression algorithm shows the greatest reduction in displacement. The Wiener filtering and Illuminance-Reflectance adjustment version however shows greater aberration, this is also be due to the propagation of high frequency from the Wiener filtering through the Illuminance-Reflectance adjustment, causing the black squares to be offset above the minimum. After Wiener filtering some high frequency noise is introduced. The Illuminance-Reflectance component then further enhances the noise amongst the flat field areas in an attempt to pull out detail.

The noise enhancement after the Wiener filtering by the Illuminance-Reflectance component is evident if we consider the Wiener filtering and Laplacian Operator algorithm. This algorithm has a lower measured displacement than the Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener filtering algorithm.

6.7.1 Concluding Remarks

It is evident that wind effects have caused greater horizontal motion then vertical motion across the image plane. All Algorithms showed a reduction in aberration when compared to the unprocessed video. The Homomorphic filtering method had the lowest reduction in aberration. These results are possibly corrupted by the amount of high frequency noise interfering in the histogram stretch.

Results for the rest of algorithms were in close proximity to one another with the Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression algorithm showing the greatest reduction in aberration.

6.8 EXPERIMENT 4: ALGORITHM MTF EXPERIMENT RESULTS

Atmospheric turbulence causes degradation in the higher spatial frequencies of an image. Atmospheric turbulence suppression algorithms try to increase object detail resulting in an enhancement of these suppressed higher frequencies.

MTF (Modulation Transfer Function) is the spatial frequency response of the video measured from which information can be obtained as to spatial frequency enhancements gained.

A low, medium, and high spatial frequency target was setup individually and video footage captured of each target. MTF was then calculated from the three video footages as laid out in the experimental method. Algorithms processed the three video footages and the outputted video used for their MTF calculation.

This section presents these results for the Algorithm MTF Experiment. The MTF curve obtained for unprocessed video is used as a control reference for the processed video. Results are graphed on a logarithmic scale.

Addendum K contains individual tables for each of the algorithms tabulating the MTF measurements.

The results for the unprocessed video show sharp decrease then a slight increase where the curve starts to flatten out. This slight increase is because the video footage of the four highest spatial frequencies were captured first just before the other frequencies and the atmospheric conditions may have been a bit better. Slightly better conditions may be the result of why the highest frequencies don't decrease as rapidly but noise and scintillation can also play a part.

Fluctuations in MTF among the four highest frequencies are due to the atmospheric PSF varying across an image frame as well as the aberration. The changing PSF and aberration causes a varied flickering of the intensities (scintillation) across video frames causing a flattening and slight variation in the MTF where visually the frequency bands are hard to discern.

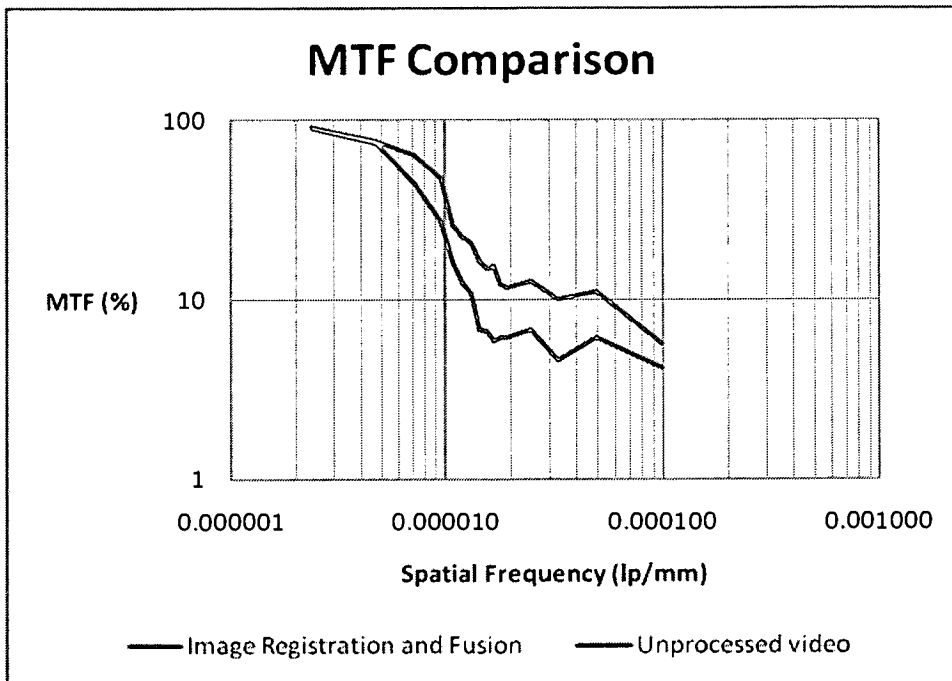


Figure 6-7: Image Registration and Laplacian Pyramid Image Fusion algorithm MTF curve.

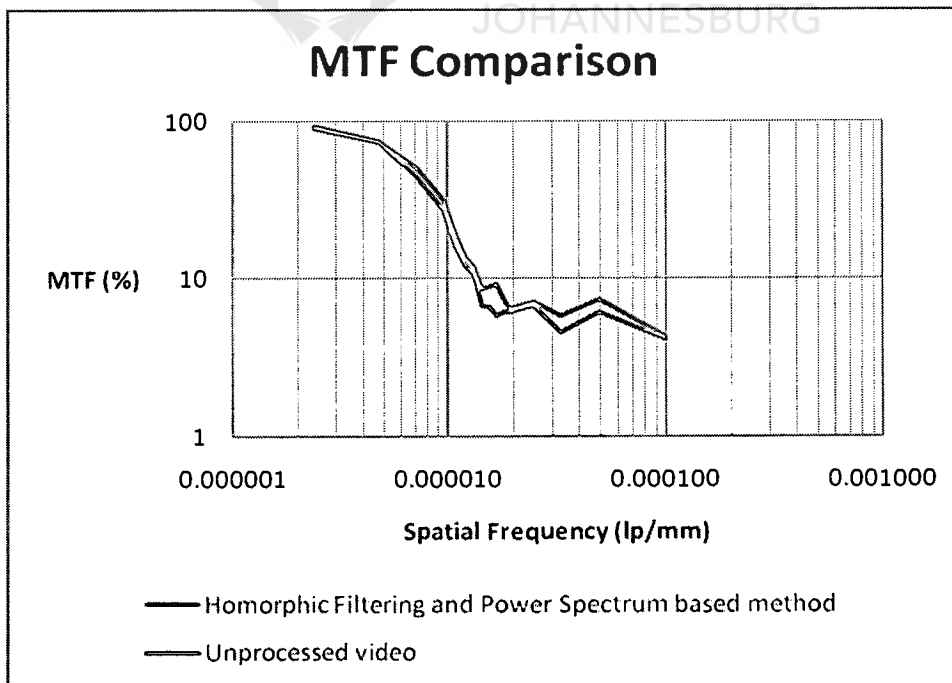


Figure 6-8: Homomorphic Filtering and Power Spectrum based method MTF curve.

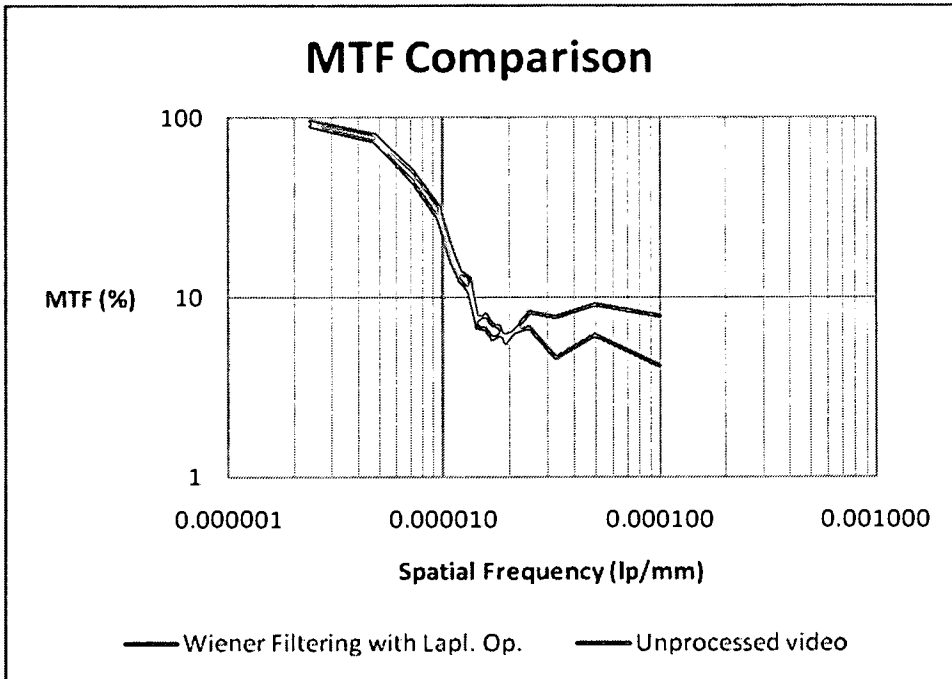


Figure 6-9: Wiener Filtering using Laplacian Operator algorithm MTF curve.

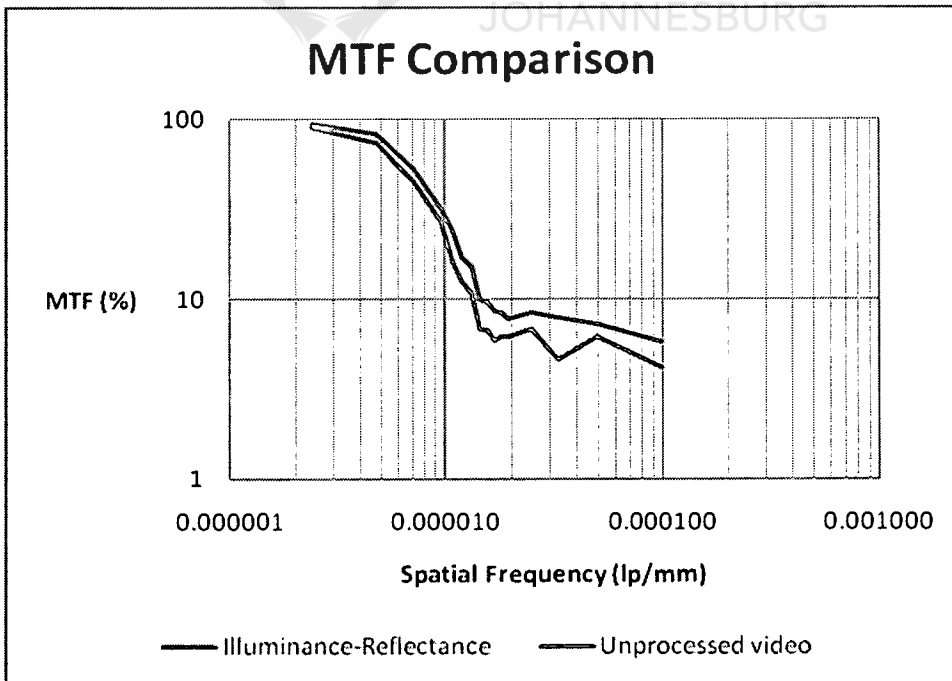


Figure 6-10: Dynamic Illuminance-Reflectance method MTF curve.

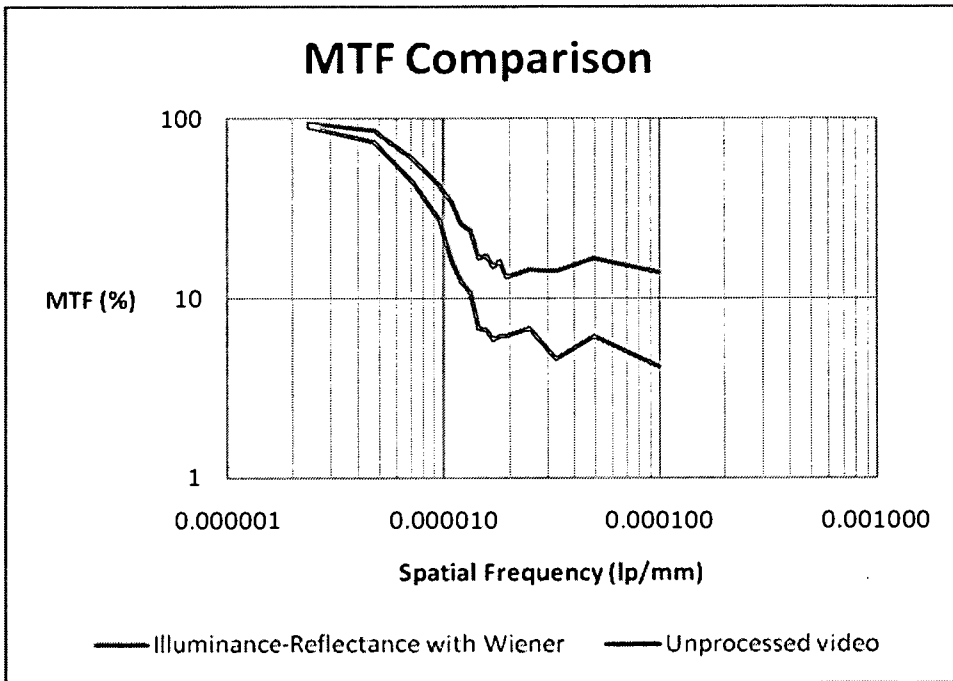


Figure 6-11: Dynamic Illuminance-Reflectance with Wiener filtering method MTF curve.

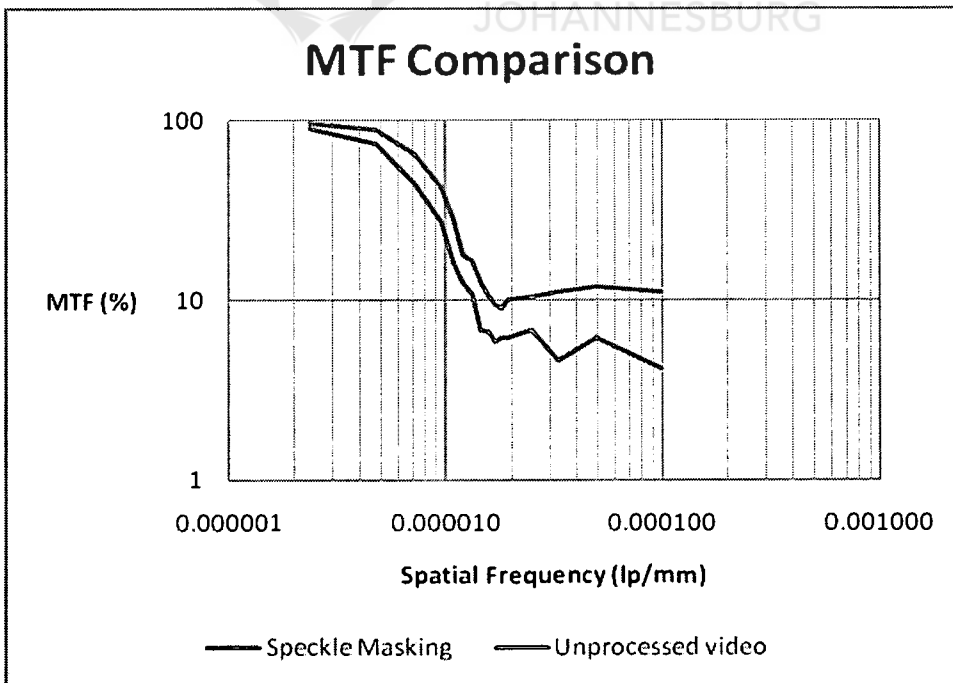


Figure 6-12: Speckle Masking algorithm MTF curve.

Considering some of the curves such as that from the Wiener filter, the MTF curve flattens out as we reach the high frequency noise level. This flattening is due to the process of recording maximum and minimum values across the video frame to overcome aberrations and varying PSF, from which MTF measurements are then made.

This flattening can be useful as it allows a measurement to be made of the high frequency noise present in the video. The MTF curve from video footage which shows a flattening of the highest spatial frequencies above that of another video's MTF, has a greater amount of high frequency noise present.

The Wiener filter based methods show an expected increase in high frequency noise when compared to the unprocessed video. The Speckle Masking algorithm and Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener filtering algorithm show the greatest amount of high frequency noise.

With regard to the rest of the MTF curve, all algorithms show an enhancement in the spatial frequencies. There is little improvement from the Homomorphic and Power Spectrum based method. The Speckle Masking algorithm, Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener filtering algorithm, and the Image Registration and Laplacian Pyramid Image Fusion algorithm showed the greatest improvement in spatial frequencies.

With regard to these spatial frequency response curves, the Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression algorithm shows similar performance to that of the Wiener filtering using Laplacian Operator algorithm. The Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression algorithm however has the advantage of not having the enhancement of the high frequency noise such as is evident in the Wiener filtering using Laplacian Operator algorithm's MTF curve.

6.8.1 Concluding Remarks

Results for unprocessed video have been presented and used as a comparison to the processed video results. From the MTF, a flattening of the curve at higher frequencies



can be indicative to high frequency noise present in the video. The high frequency noise and related MTF flattening is due to the method used for MTF measurement to overcome aberration and varying PSF corrupting measurements. This is useful for high frequency noise comparisons.

With regard to spatial frequency response, the Speckle Masking algorithm, Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener filtering algorithm, and the Image Registration and Laplacian Pyramid Image Fusion algorithm showed the greatest improvement in spatial frequencies. The Speckle Masking algorithm and Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener filtering algorithm however also showed the greatest amount of high frequency noise.

6.9 EXPERIMENT 5: ALGORITHM MSE AND PSNR EXPERIMENT RESULTS

MSE (Mean Square Error) and PSNR (Peak Signal to Noise Ratio) metrics are classically used in signal processing, however they require signal reference such as the signal at the source before travelling through a medium to the observed state. In this particular atmospheric turbulence problem, the reference image needs to be estimated since it is unknown. Reconstruction of an estimate is made easier if a priori knowledge is known about the image that is captured after travelling through the turbulence. This experiment made use of such a priori knowledge to construct an estimate.

From these results it appears that only the Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression algorithm is an improvement over the unprocessed video, and only a slight improvement at that. MSE and PSNR measurements are in close proximity to one another and are very low, indicative just how different the estimated ideal image is from the received and processed videos.

Algorithm	MSE
Unprocessed Video	0.0426
Image Registration and Laplacian Pyramid Image Fusion	0.0664
Homomorphic Filtering and Power Spectrum based method	0.0505
Wiener Filtering using Laplacian Operator	0.0450
Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression	0.0422
Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener Filtering	0.0476

Table 6-4: Algorithm MSE and PSNR Experiment - MSE Experiment results.

Algorithm	PSNR (dB)
Unprocessed Video	13.709
Image Registration and Laplacian Pyramid Image Fusion	11.775
Homomorphic Filtering and Power Spectrum based method	12.967
Wiener Filtering using Laplacian Operator	13.469
Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression	13.746
Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener Filtering	13.220

Table 6-5: Algorithm MSE and PSNR Experiment - PSNR Experiment results.

Unwanted noise and artefacts in the video play a big part in the MSE and PSNR measurements. The Wiener filter in the algorithms adds some high frequency noise and the Illuminance-Reflectance component in the Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener Filtering algorithm further enhances this noise. The Homomorphic Filtering and Power Spectrum based method has additive noise through the power spectrum adjustment. The Image Registration and Laplacian Pyramid Image Fusion algorithm processed video has some artefacts due to the alignment from the registration and



fusion of images. The registration process also uses a reference image which may lag the image sequence in the video, so image frames are registered to a previous geometric position.

These results are later further discussed in relation to all the other algorithm experiment results and conclusions.

6.9.1 Concluding Remarks

MSE and PSNR measurements showed an increase in noise for all algorithms except the Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression algorithm. These poor results are both due to unwanted high frequency noise and difficulty in creating an estimate reference image which is a true representation of what would be viewed without the high frequency aberration and PSF of the atmosphere.

6.10 EXPERIMENT 6: WIENER FILTERING LAMBDA SELECTION EXPERIMENT RESULTS

The Wiener Filtering Laplacian Operator algorithm spends a considerable amount of time searching for a lambda value. From observation of this algorithm, the lambda value, which determines the OTF shape, varies slightly across image frames.

The Wiener Filtering Lambda Selection Experiment investigates the performance gain and image quality loss when the lambda value is set instead of the lambda value being searched for in each frame. The set value is the average determined value across 50 image frames. The same video footage is processed by each algorithm to ensure the same atmospheric turbulence is used and any bias is removed during comparisons.

Tables of measurements and individual graphs may be found in Addendum L.

Comparing the two methods shows that there is a large performance gain by not calculating a λ value across the 50 image frames. Figure 6-13 shows a plot of the performance graphs on a single set of axes.

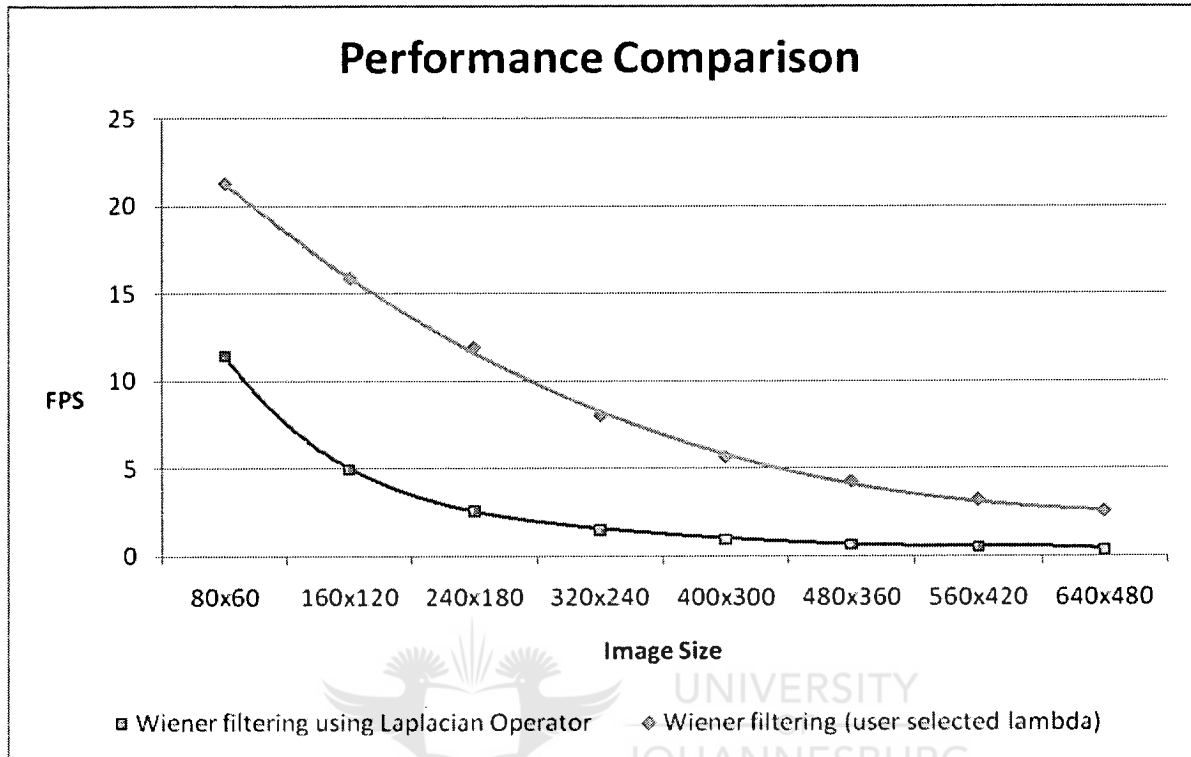


Figure 6-13: Wiener filtering FPS performance comparison graph

A sharper image will have a higher measured sharpness value. The Wiener Filtering using Laplacian algorithm has a slightly higher value of 9.166 compared to the user selected Lambda algorithm's 9.063. Results indicate a little difference in image sharpness when calculating a lambda value for each image frame over 50 frames or using the average lambda value for all the frames.

The small peaks, as seen in the image sharpness results graphs of Addendum L, are at the points where a new SNR estimate is made for the Wiener filtering. This SNR estimation is described in the Algorithm Detail Design chapter and is left unchanged between the algorithms.

The difference between average image sharpness results is less than the difference in SNR estimate affects between image frames.

The MTF Comparison (Figure 6-14) shows very little difference between the two methods. Towards higher frequencies the Wiener filtering without the sharpness measurement for each frame (blue plot), is unable to resolve the frequencies as well due to a slight difference in the lambda. The Wiener filtering without the sharpness measurement reaches the noise level slightly faster than the Wiener filtering using Laplacian Operator algorithm.

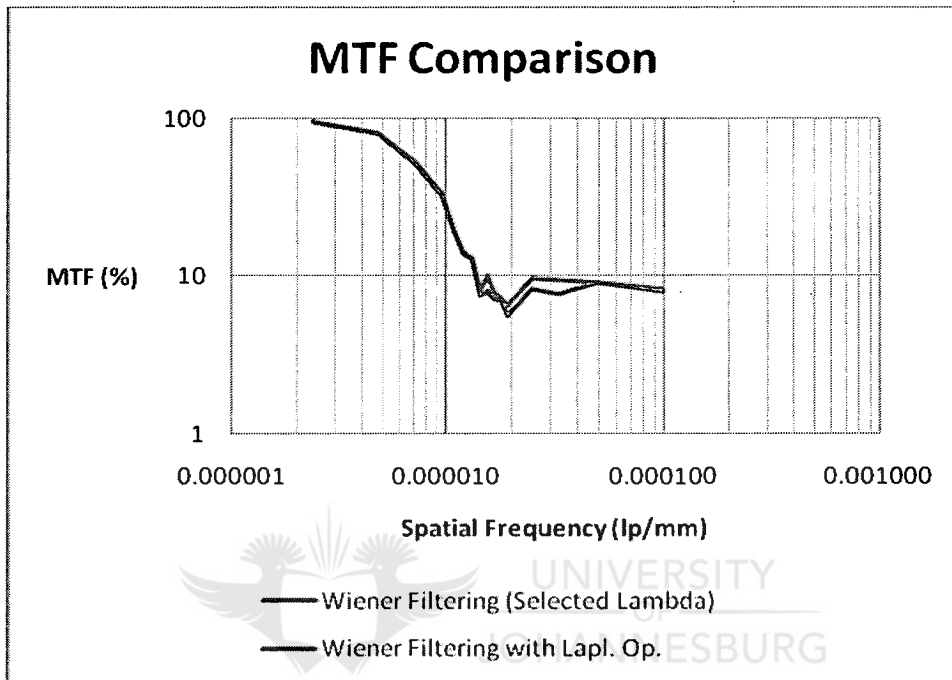


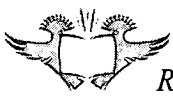
Figure 6-14: Wiener filtering MTF comparison

The MSE (Mean Square Error) and PSNR (Peak Signal to Noise Ratio), as discussed in the background of the Algorithm MSE and PSNR Experiment, were measured. Results for both algorithms were 0.045 for the MSE and 13.469dB for the PSNR.

The MSE and PSNR values indicate that there is no difference between the lambda selected via the Laplacian Operator method and the use of a single lambda value across the image frames.

6.10.1 Concluding Remarks

Over 50 image frames a single lambda value can be used, rather than estimating a lambda value for each image frame, with a result of a large performance gain with



little loss in image quality. An algorithm can make use of these results by only estimating a new lambda value periodically such as every 50th frame.

6.11 EXPERIMENT 7: FILTERING RECEIVED LIGHT EXPERIMENT RESULTS

After investigation into filtering electromagnetic waves, it was found that both red coloured and polarised filters can be used for contrast enhancement in images. Blurring caused by the Atmospheric turbulence causes a loss of image contrast. The Filtering Received Light Experiment investigates the image quality for atmospheric turbulence affected video footage, using no optical filter, red colour filter, and a polarised filter.

Unfiltered video footage is used as a control reference and video footage was captured within the hour to avoid discrepancies in atmospheric turbulence conditions. The red colour filter has a Wratten number of #24.

Additional tables and individual graphs relating to this experiment and measurements taken can be found in Addendum M.

The results (Table 6-6 and Figure 6-15) clearly show sharpness improvements by using either a red filter or polarised filter. The red filter provided a much greater gain in image sharpness. Sharpness improvement using the Polarised filter is small and is possibly due to a production of only a small amount of glare from the imaged scene which is removed by the polarised filter.

Filtered Video Footage Used	Average Image Sharpness Measurement
Unfiltered Video	2.677
Colour Filtered Video	5.907
Polarised Filtered Video	3.319

Table 6-6: Image sharpness results for unfiltered and filtered video.

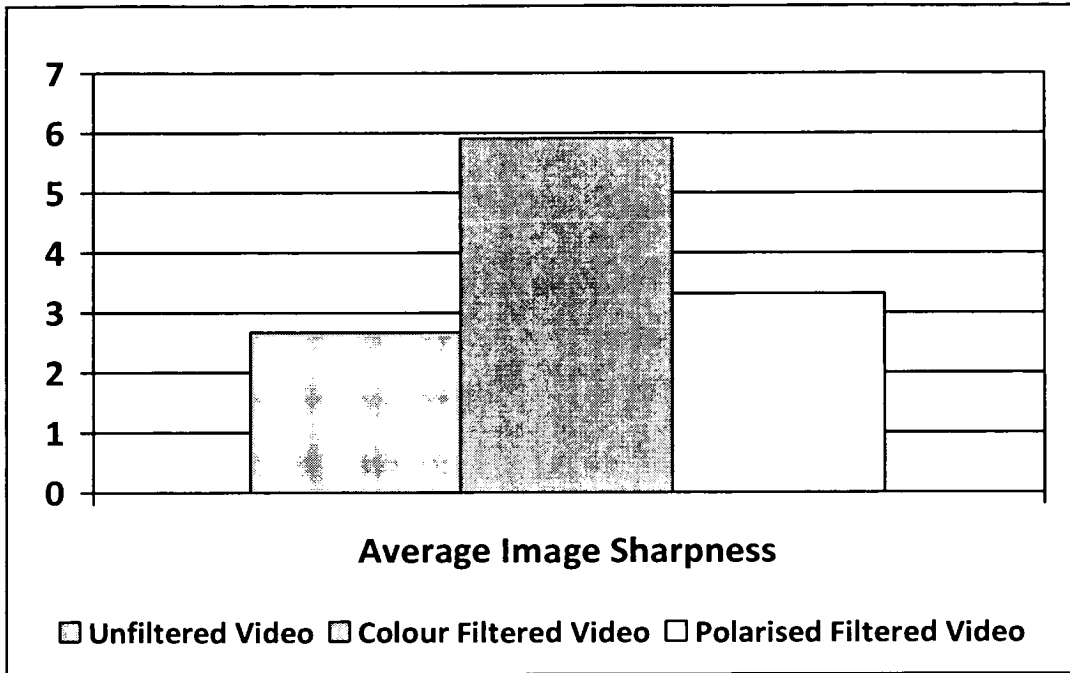


Figure 6-15: Image sharpness results of unfiltered and filtered video.

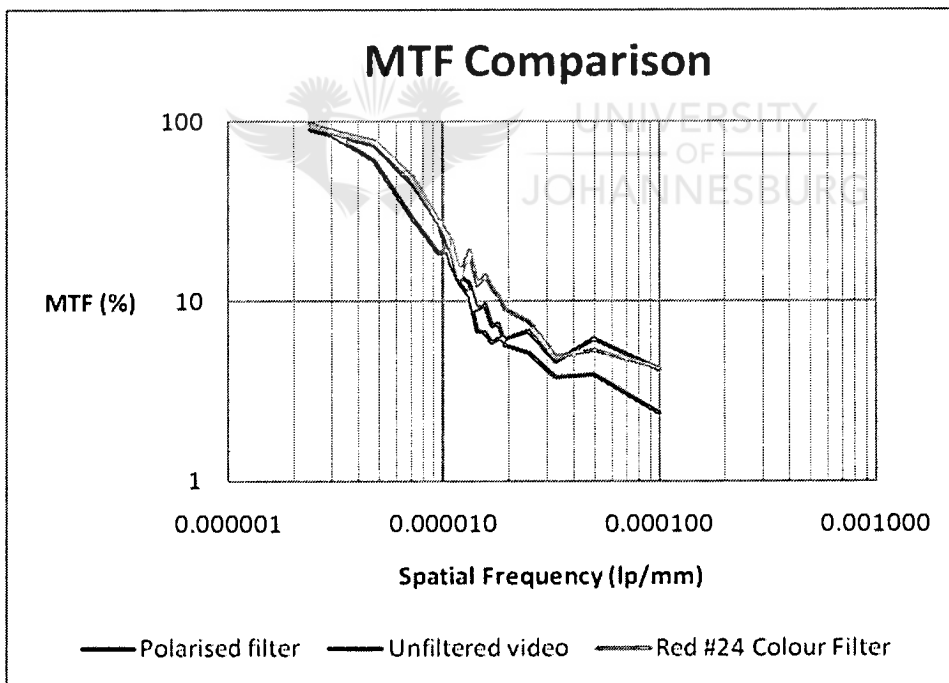


Figure 6-16: MTF results of unfiltered and filtered video.

Figure 6-16 shows the MTF plots for the unfiltered and filtered videos on a common axis. The red filtered video shows an enhancement in mid spatial frequencies before the noise level is reached in comparison to the unfiltered video response.

The polarising filter shows a reduction in high frequency noise as its curve does not flatten towards the highest frequencies. The polarising filter’s MTF is disappointing and is possibly due to, the filtering of the light from the broader white surfaces of the low spatial frequency target decreasing contrast, together with additive noise from the video camera.

Filtered Video Footage Used	MSE	PSNR (dB)
Unfiltered Video	0.0426	13.709
Colour Filtered Video	0.0187	17.286
Polarised Filtered Video	0.0359	14.450

Table 6-7: MSE and PSNR results for unfiltered and filtered video.

MSE and PSNR show improvement through the use of either filter. The Red #24 colour filtered footage has the greatest improvement in MSE and PSNR over unfiltered video footage. Figure 6-17 and 6-18 graphically illustrate MSE and PSNR differences.

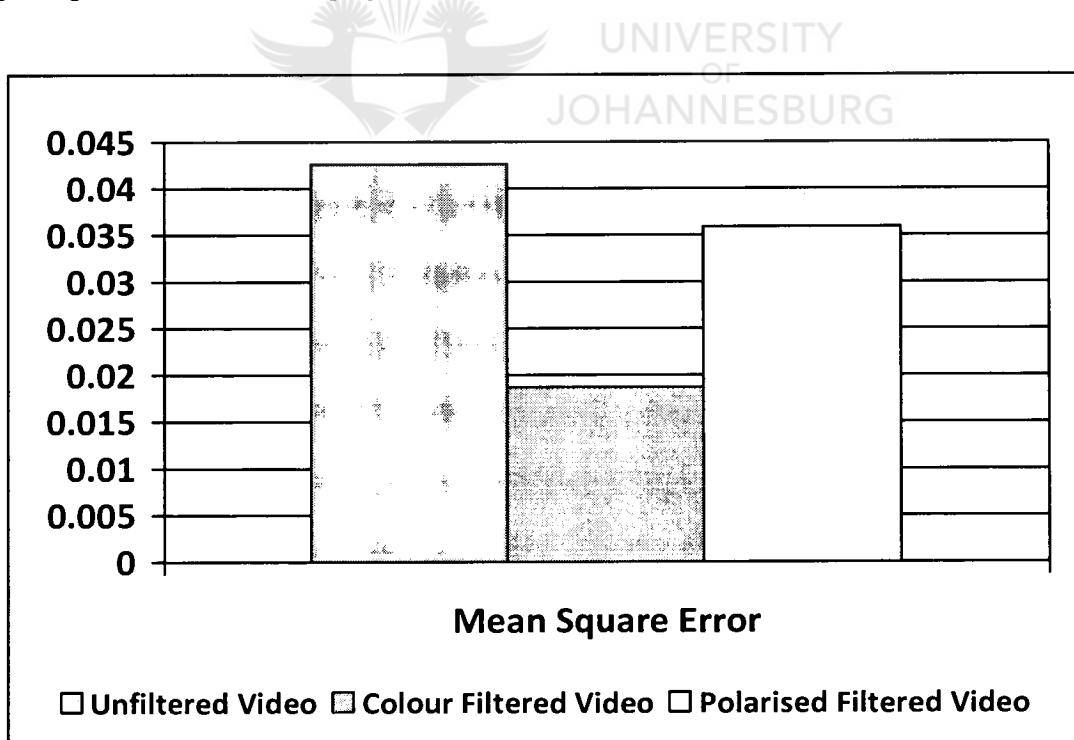


Figure 6-17: MSE results of unfiltered and filtered video (lower is better).

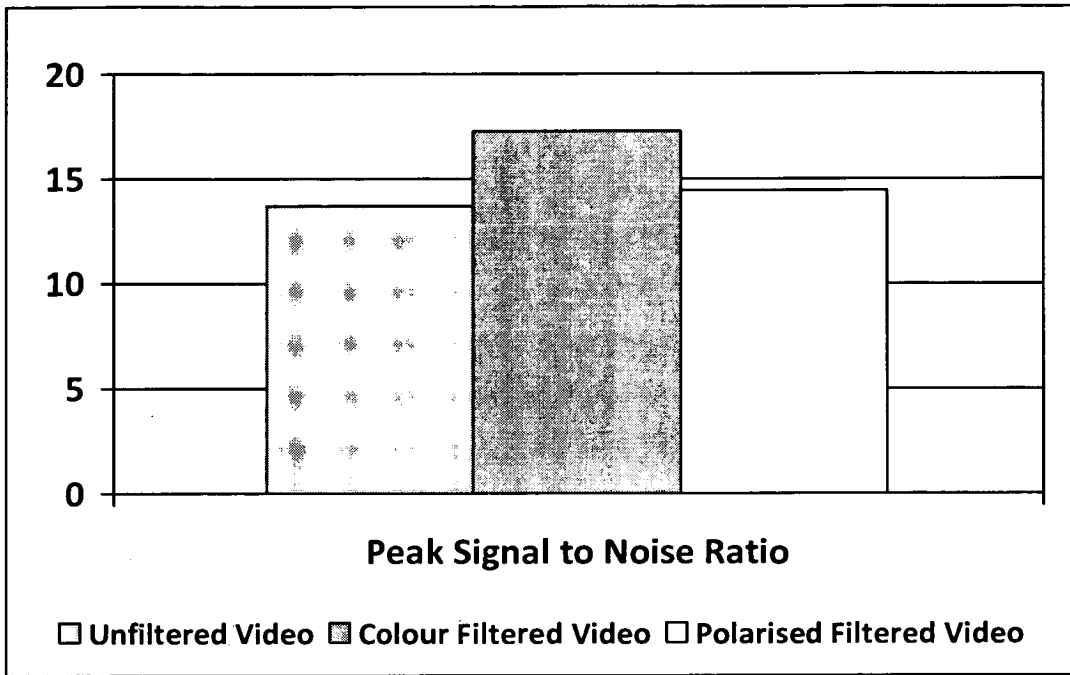


Figure 6-18: PSNR results of unfiltered and filtered video (higher is better).

6.11.1 Concluding Remarks

From the results it is clear that the use of a red #24 colour filter helps in improvement of image quality when viewing a scene through atmospheric turbulence. The polarising filter showed little improvement and its MTF curve suggests it helps to reduce high frequency noise but there is a loss of frequency detail. Considering results from the polarising filter it appears that it may have its own significant OTF that needs to be overcome, or else there may have been difficulty in obtaining focus with the imaging system for the video footages which used the polarising filter.

6.12 CONCLUSION

Various issues were encountered and corrected for during the implementation phase of the algorithms. A number of experimental design issues encountered are described as well as solutions that were used to counter those problems. Wind and aerosols in the air play a part as possible the biggest two problems since they jeopardise image contrast and clarity, and image stability respectively. This dissertation concentrates on analysis of atmospheric turbulence



suppression algorithms, however it is clear from some of the issues encountered that implementation and use of a portable system in the real world has many difficulties to overcome, and factors such as aerosols in the air can limit the effective range over which the system can be used, regardless of how turbulent the atmosphere is.

Various experimental results have been presented and individually discussed. The next chapter discusses the experimental results as a whole and also provides some example images for the algorithms.



7 CHAPTER 7: EXAMPLE IMAGES AND RESULTS DISCUSSION

7.1 INTRODUCTION

Experimental results have been presented and analysed in the previous chapters. Results have not yet been discussed and as a whole. This chapter summarises the results as a whole, with analyses and interpretation. Before the interpretation summary, example images of processed video frames are presented not just for the reader's interest, but also to help as a visual aid for the discussion of the results.

7.2 EXAMPLE IMAGES

A sample image frame is shown for unprocessed and processed video in the following figures (Figure 7-1 to Figure 7-7). The video footage used is courtesy of the CSIR. The video footage is of a building site 5.5 km from the imaging system. Video frame size is 640x480 pixels.

The video footage provides a various array of differing objects. There is complex scaffolding at the top of the video, foliage in the foreground, buildings and brickwork, various shadows, and higher spatial frequencies amongst flat fields such as the lampposts and fencing.

Differences between some of the example frames are small enough that it can be difficult to see once printed on paper. Wiener filtering using Laplacian Operator algorithm and the Dynamic Illuminance-Reflectance based algorithms' improvements are however easily visible on printed paper.



Figure 7-1: Unprocessed video frame



Figure 7-2: Image Registration and Laplacian Pyramid Image Fusion.



Figure 7-3: Homomorphic Filtering and Power Spectrum based method.



Figure 7-4: Wiener Filtering using Laplacian Operator.



Figure 7-5: Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression

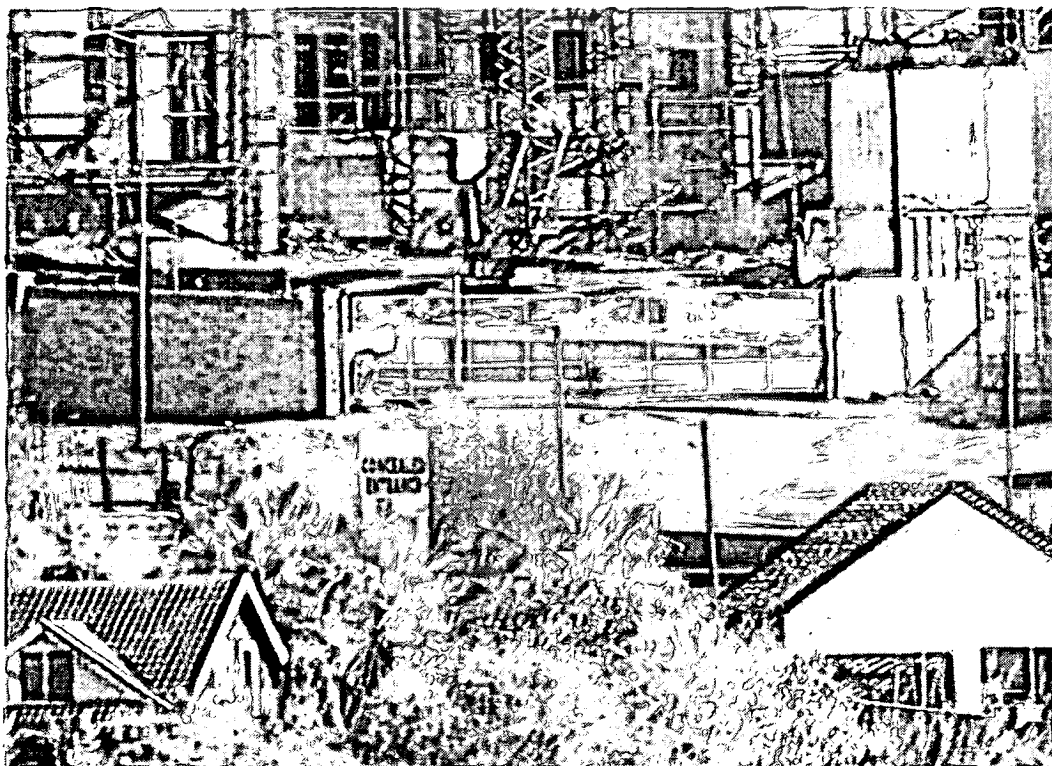


Figure 7-6: Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener Filtering

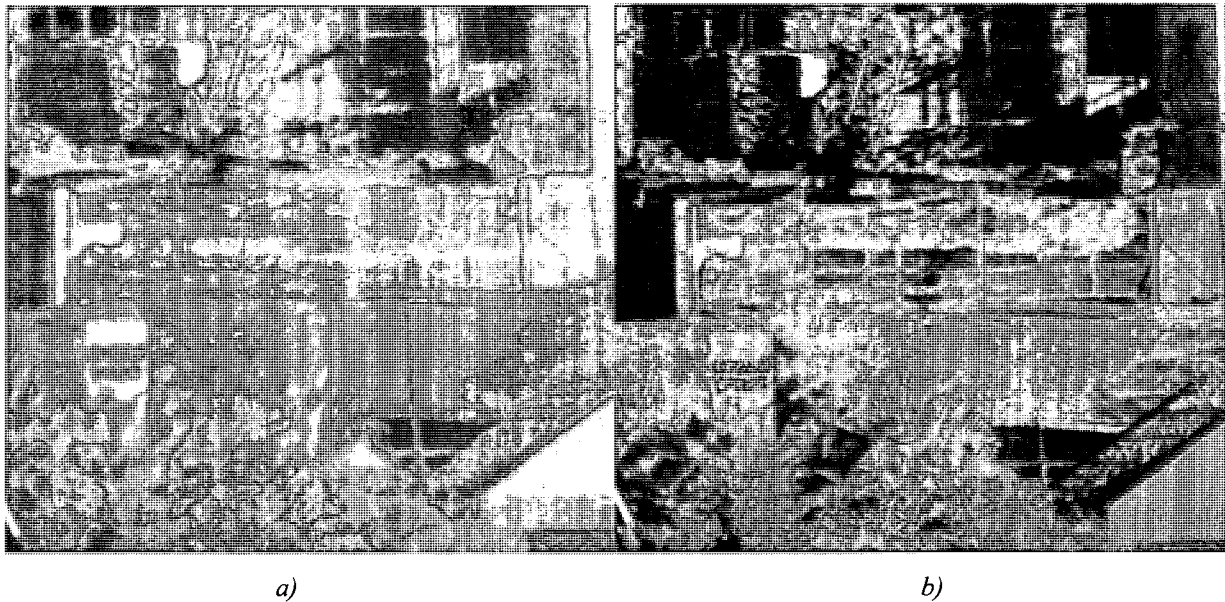


Figure 7-7: a) an unprocessed video frame. b) Speckle Masking after processing 100 frames.

7.3 SUMMARY AND LINKING OF RESULTS

7.3.1 Image Registration and Laplacian Pyramid Image Fusion

This algorithm has shown to improve an image's MTF, and provides a sharper image measurement due to the Laplacian Pyramid Image Fusion. The registration method however struggles to accurately align image frames resulting in only a small reduction in aberration across image frames. There is also evidence of small rippling and artefacts in the image as seen in Figure 7-2. These ripples and artefacts are due to the slight registration difference between frames and the fusion of these frames.

The fine registration difficulties can be attributed to the use of an image average as a reference. Blurring in the reference image can cause these small alignment errors.

The Image Registration and Laplacian Pyramid Image Fusion algorithm is very slow in comparison to the other algorithms presented here. The poor performance is due to the complicated registration procedure.

7.3.2 Homomorphic Filtering and Power Spectrum based method

This algorithm had the highest FPS for larger image sizes; however it had the poorest image quality enhancement. A fair amount of noise is introduced into the video through the Power Spectrum adjustment even if the edge of the image frame is tapered before processing to remove discontinuities. Figure 7-3 shows a tapered processed video frame.

7.3.3 Wiener Filtering using Laplacian Operator

The Wiener Filtering using Laplacian Operator method had results that placed it towards the centre of the reviewed algorithm results for both performance and image quality. The algorithm increased image sharpness and the MTF slightly but also increased the high frequency noise level.

The ratio averaging used together with the Wiener filtering helped reduce high frequency aberrations with only the Dynamic Illuminance-Reflectance algorithm showing a slightly better result.

7.3.4 Wiener Filtering using Selected Lambda for Multiple Image Frames

This method is a modification of the one above. The difference is that whereas the above method uses the Laplacian Operator metric to calculate a new lambda for each of the 50 image frames processed, this algorithm uses a single lambda value specified at the start of the algorithm. The lambda value is used when processing each of the 50 image frames without having to search for it. The lambda value specifies the shape of the OTF (Optical Transfer Function) used in the Wiener filtering.

The performance increase over the Wiener Filtering using Laplacian Operator algorithm is considerable. FPS performance is comparable to the Homomorphic and Power Spectrum based method. The decrease in image quality is slight in comparison to the Wiener Filtering using Laplacian Operator algorithm. MTF is marginally poorer for high frequencies with this algorithm converging to the same noise level at a slightly lower spatial frequency.

The use of a single lambda value across image frames shows that the small difference in the OTF's has minimal consequences for image quality and a viable trade off may be to update the lambda value and corresponding OTF after every few image frames.

7.3.5 Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression

This algorithm shows very good results. It has a good FPS performance and MTF curve similar to the above Wiener filtering algorithms but without having the high frequency noise levels the Wiener filtering based algorithms suffer with.

The Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression algorithm has the greatest reduction in the amount of high frequency aberration present in the video. It also has the second highest image sharpness levels. The high image sharpness levels are attributable to the enhancement of detail hidden in shadows and very bright areas by the dynamic Illuminance-Reflectance adjustment component. This can be clearly seen from Figure 7-5.

7.3.6 Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener Filtering

The Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener Filtering algorithm is a combination of the Wiener Filtering using Laplacian Operator Algorithm and Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression algorithm.

This algorithm has the highest image sharpness levels out of the algorithms reviewed as well as a good enhancement in the MTF curve. Due to the Wiener filtering and the boosting because of the dynamic Illuminance-Reflectance component, high frequency noise levels are higher.

The ratio frame average used in this algorithm reduces high frequency aberrations across video frames with only the Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression algorithm showing slightly better results.

Image sharpness and enhancement of detail hidden in shadow is clearly seen in Figure 7-6.

7.3.7 Speckle Masking

The Speckle Masking algorithm uses a 100 image frames to render a single image of the observed scene. This algorithm is very slow and shows enhancement of the MTF curve. High noise levels however were observed due to high frequency additive noise. Smudging effects are also evident in the rendered image as seen in Figure 7-7 amongst the lamppost and white sign which reads “Concealed Entrance”.

7.3.8 MSE and PSNR

It is clear from MSE and PSNR measurements for the algorithms in comparison to MTF and image sharpness measurements, that they are not suitable metrics on their own. It is very difficult to reconstruct a reference image and noise levels can greatly affect their values. Slight misalignment between the reference image and measured image can give poor MSE and PSNR readings. These small differences in MSE results for this atmospheric turbulence problem are not great enough for interpretation as to which algorithms provide better image enhancement. Where there are vast differences in readings, such as the in the Filtering Received Light Experiment, then

results have greater meaning and can be interpreted with some level of confidence as to which is better.

7.3.9 Imaging System and Optical Filters

From the results it is seen that due to the enhancement of high frequencies by the algorithm, the imaging system should be constructed so that minimal high frequency additive noise is present. Using a higher bit grey level may also help reduce noise levels.

The use of a red #24 colour optical filter can increase image sharpness and improve the MTF. The polarisation filter should be used where image glare may be a problem, its effect on image sharpness and MTF is less than the colour filter.

The use of an optical filter causes some of the light to be blocked and reflected. This causes a decrease in image contrast and longer exposure times will be required than when not using a filter.

7.3.10 Real time and post processing applications

None of the algorithms showed real time performance with their Matlab implementations; however real time performance is feasible for some of the algorithms if they are implemented in C or C++ or on hardware such as a GPU (Graphics Processing Unit). The Homomorphic Filtering and Power Spectrum based method, Wiener Filtering using Selected Lambda for Multiple Image Frames algorithm and Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression algorithm are the easiest to implement with real time performance since they are relatively low in complexity. Making use of GPU technology or FPGA (Field Programmable Gate-Array) it is expected that the Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener Filtering algorithm may also reach real time processing speeds.

If only post processing is necessary and video surveillance image quality enhancement was of utmost importance, then it might be expected that the most complex algorithm would give the best image. In this study the relatively less complicated Dynamic Illuminance-Reflectance based algorithms give a far better image quality. The Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener Filtering algorithm particularly shows large video surveillance enhancement.

7.4 CONCLUSION

The Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener Filtering algorithm shows the greatest increase in image quality and should be used in an imaging system that has a low amount of additive high frequency noise. The use of a red #24 optical filter also shows an improvement in image quality, however due to its light filtering, longer exposure times are necessary.

The simple ratio frame averaging using an α value of a 1/3, which is used in many of the reviewed algorithms, has shown to efficiently reduce high frequency aberrations across image frames.

The next chapter concludes this dissertation.

8 CHAPTER 8: CONCLUSION

8.1 OVERVIEW OF WORK

Video surveillance is used globally for gathering information, whether viewed in a security, recreational, research or other perspective. Some of these video surveillance systems utilize very high magnifications and could be used to survey targets up to 10 km from the imaging system. This is particularly useful for covert systems tasked with information gathering.

Digital video cameras are increasing with quality and decreasing with cost all the time making them prevalent in video surveillance systems. Coupling a digital video surveillance camera to a very long focal length lens to achieve high magnifications is possible with off-the-shelf hardware.

The problem with the high magnification systems is that between 1km and 10km, atmospheric turbulence scatters, refracts and absorbs light travelling from the surveyed scene. The atmospheric PSF (Point Spread Function), which defines how a point of light is scattered when travelling through the turbulence, varies both spatially and with time and is unknown. The result is that video captured of a stationary scene containing a unmoving object, appears to show a 'shimmering' movement of the scene and object as well as blurring of the scene and object, both varying spatially and temporally. This causes difficulty in identification of objects and detail.

Atmospheric turbulence is also known as heat shimmer, scintillation, and seeing, and a number of algorithms have been developed, as reviewed in the literature study, in an attempt at suppressing atmospheric turbulence effects. There is however a lack of comparison between the algorithms in relation to image quality improvements. Image quality analysis is made difficult since only the distorted image containing geometric distortion (seen as movement) and photometric distortion (seen as blur) is observed. The image scene that would be viewed without atmospheric turbulence is unknown.

This study is concerned with the comparison of the outputted video or image from various atmospheric turbulence suppression algorithms. This study is formed through a literature study which reviews the physics surrounding atmospheric turbulence and an understanding of the optics, image processing methods and related background information, atmospheric turbulence studies by other authors, atmospheric turbulence suppression algorithms by other authors, hardware components for construction of an imaging system, and metrics that may be useful for algorithm comparison. A study into the filtering of electromagnetic waves is also undertaken for a later experiment regarding the use of optical filters for image enhancement.

A number of selected algorithms were implemented in Matlab from the literature study review as well as some own developed algorithms constructed using the dynamic Illuminance-Reflectance adjustment component, described by Tao et al. [93], as a base.

An imaging system comprised of a Celestron Nexstar 8 SE eight inch diameter Schmidt Cassegrain telescope and an Arecont vision 3100 IP (Internet Protocol) surveillance camera was constructed. The imaging system was used to capture video footage through atmospheric turbulence of various targets for use with the experiments.

A number of experiments were designed to provide results to give insight into the algorithms' outputted image qualities and performance requirements. An experiment was also undertaken to investigate the use of a polarising filter and a red #24 colour filter on atmospheric turbulence degraded video.

The results indicated that all algorithms managed to increase image quality and stabilise the high frequency aberrations that were present across image. The results also showed that the newly constructed dynamic Illuminance-Reflectance adjustment based algorithms had the greatest image quality performances. The Illuminance-Reflectance adjustment has the added advantage of increasing detail hidden in shadows within the video.

The Speckle Masking and image registration algorithms were identified as requiring the greatest amount of processing resources. The Homomorphic Filtering and Power Spectrum based method had the best FPS performance and was able to suppress high frequency

aberrations but its image quality enhancement was the smallest. This method may give better image quality results with an adjustment of the homomorphic filtering component.

Out of the reviewed results the Illuminance-Reflectance adjustment based algorithms or Wiener Filtering (using a selected lambda) algorithm are best suited for a real time implementation. For post processing the dynamic Illuminance-Reflectance adjustment based algorithm showed the greatest image enhancement among the reviewed algorithms.

It is important that if atmospheric turbulence suppression using one of the algorithms is to be performed, that the imaging system be designed to minimize the amount of additive noise. Since atmospheric turbulence degrades higher spatial frequencies of an image, the atmospheric turbulence suppression algorithms boost these higher frequencies. High frequency noise will be boosted along with the frequencies and so it is important to minimise noise added from the imaging system and the signal digitization.

Using a red colour (Wratten #24) optical filter has shown to improve image quality of video captured through atmospheric turbulence. The filter however blocks a fair amount of light and this will cause an increase in exposure times. Care thus needs to be taken if considering using the filter together with an atmospheric turbulence suppression algorithm. Some atmospheric turbulence suppression algorithms, such as the Speckle Masking, image registration based, and optical flow field based algorithms, require exposure times of less than 10 ms to 'freeze' the atmospheric turbulence in place.

8.2 SUCCESSFUL PROBLEM SOLUTION

This study has been successfully undertaken by providing and demonstrating methods for comparing atmospheric turbulence suppression algorithms, both in terms of image quality relating to image sharpness and MTF (Modulation Transfer Function), and FPS (Frames per Second) performance. These methods allow for comparability of processed results which are repeatable.



Additionally new algorithms were developed based on Tao et al.'s Illuminance-Reflectance adjustment. This algorithm has shown excellent results in comparison to other algorithms reviewed.

The comparisons show the strengths and weaknesses of the different algorithms and provide clear results as to the image quality enhancement. Together with the performance results it is easy to determine the suitability of each algorithm in a real time or post processing environment.

Results have been interpreted for each of the algorithms. Interpretations were performed after each experiment, as well as after considering the results as a whole. From the results, algorithms providing the greatest FPS performance and those with the greatest image enhancement were identified. Post and real time processing requirements were discussed and suitable algorithms from those reviewed were identified.

The use of optical filters for image enhancement of images viewed through atmospheric turbulence has also been considered, investigated through experimentation, and concluded.



8.3 RELEVANCE OF EXPERIMENTS AND RESULTS

The experiments allow for measurement of FPS performance and importantly image quality in relation to long range video surveillance through atmospheric turbulence. The experiments allow algorithms to be compared with repeatable results, providing valuable information as to their strengths and weaknesses with regard to one another, their efficiency and processing requirements, allowing a algorithm to be selected for a particular post processing or real time processing situation, providing direction as to components which perform well, and providing measurements and better algorithm understanding for future compositions of new algorithms.

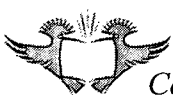
Experiments may be used with any video footage and various atmospheric turbulence affected video footage may be compared to one another. This can provide information such as how the image sharpness, video stability, and MTF changes over distance, temperature, terrain or other affecting factors such as wind.

The results show how some of the existing algorithm reviewed in this research compare. Comparison is on a FPS performance level, image quality level, and video stability. The implementation and comparison of two new algorithms constructed by this author and based on Tao et al.'s Illuminance-Reflectance adjustment model [93], demonstrates the strength of using Illuminance-Reflectance adjustment for this atmospheric turbulence affected video footage problem. Using a red #24 colour optical filter also shows that gains may be made in grey scale image quality for this problem.

This study and its results may be used for a greater understanding of this imaging problem and how best to treat it, helping to identify issues that may be encountered, the design of imaging systems, selection and implementation of one of the reviewed algorithms for a particular situation, tweaking the algorithms where they show weakness, compiling a new algorithm using a composition of components from the reviewed algorithms, or allowing for inspiration and lateral thinking for development of entirely new algorithms.

8.4 ISSUES ENCOUNTERED

Implementation issues stemmed primarily from the Speckle Masking algorithm. This algorithm is highly complicated and small details such as splitting up an image into smaller overlapping images have a huge effect on outputted image quality. Information on smaller details were scattered among the reviewed literature. A one dimensional signal creates a two dimensional bi-spectrum object which contains symmetries and a path is documented in the literature for navigation through this 2D object. An image however is two dimensional and creates a four dimensional bi-spectrum. Symmetry in this object had to be found as well as a path used to navigate through the object for Fourier phase reconstruction, overall a complicated job which needed continuous trouble shooting. Important processing details that were found in the literature are that the images must be broken up into small overlapping sub-images, image registration performed on the sub-images to remove small global low frequency aberrations, processing the sub-images separately using the Speckle Masking algorithm, and finally applying a hamming window to the processed sub images and piecing the sub-images together to create the final image.



For the Illuminance-Reflectance adjustment, equation (6.2) should be used instead of equation (6.1).

Video footage captured by the video camera first needed to be converted to raw AVI (Audio Video Interleave) before it could be used with Matlab. Matlab's `aviread` function's supported frame types are 8-bit, for indexed or greyscale images, 16-bit, for greyscale images, or 24-bit, for true colour.

When using a high magnification imaging system, the system must have a stable platform and support. If there is wind, the imaging system may need to be positioned in a sheltered area. Slight movements in the lens can cause large visible aberrations in the video footage, degrading the image and corrupting captured atmospheric turbulence.

Manual focusing is very difficult in atmospheric turbulence conditions. Focusing may be made easier by using a smaller aperture lens to increase the depth of field over which objects are in focus. However exposure times will increase as less light will be gathered with the smaller aperture. An alternative may be real time measurement of the image sharpness allowing identification of when the best focus is achieved.

8.5 FUTURE WORK

Further work may be undertaken to expand the number of algorithms compared using these experiments, and using the results for development of new algorithms and adjustment of existing algorithms for better image quality and performance through identification of strengths and weaknesses.

Experiments may be used for measurement of atmospheric turbulence and broadening its understanding. Valuable information and greater understanding may be found through the investigation of the effect on image sharpness and MTF as distance increases, temperature changes and how wind affects atmospheric turbulence.

Future work can be carried out on algorithms to develop them on FPGA (Field Programmable Graphics Array) or GPU (Graphics Processing Unit) hardware for real time implementations.

Work may also be carried into creation of an imaging system for use in the field either as a real time system or post processing system.

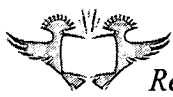
8.6 PUBLICATIONS AND PRESENTATIONS

- Presented a paper and presentation entitled 'Reduction of Heat Shimmer in Long Range Surveillance' to The Military Information and Communications Symposium of South Africa (MICSSA) held between the 24 and 26 of July 2007.
- Presented, received acceptance, and have had published a paper entitled 'Comparison of two terrestrial atmospheric turbulence suppression algorithms'. Paper was presented at the 8th Institute of Electric and Electronic Engineers (IEEE) 2007 Africon International Conference held in Windhoek, Namibia between the 26 and 28 September 2007.
- Presented a paper co-authored with Philip Robinson entitled 'Reduction of the effects of atmospheric turbulence in long range video surveillance' to the Pattern Recognition Association of South Africa during 2007 for consideration for inclusion in the proceedings of the Annual PRASA Symposium that was held between the 28th and 30th of November 2007..
- Part of the Council for Scientific and Industrial Research (CSIR) Programmed Reach into Science & Maths (PRISM) group. Presented presentations together with Philip Robinson regarding research on atmospheric turbulence undertaken at the University of Johannesburg at the CSIR Defence, Peace, Safety, and Security (DPSS) building on the 27th of July 2007 and the 31st January 2008. Another presentation is set for the 24th July 2008.

9 REFERENCES

- [1] W. Zhao, L. Bogoni, M. Hansen, "Video Enhancement by Scintillation Removal", *icme*, pp. 71, 2001 IEEE International Conference on Multimedia and Expo (ICME'01), 2001.
- [2] B. R. Frieden, "An exact, linear solution to the problem of imaging through turbulence", *Optical sciences centre*, University of Arizona, Tucson, AZ 85721, USA, 1997.
- [3] S. D. Ford¹, B. M. Welsh¹, M.C. Roggemann², "Constrained least-squares estimation in deconvolution from wave-front sensing", *Optics communications*, ¹Department of Electrical and Computing Engineering, Air Force Institute of Technology, 2950 P Street, Bldg 640, Wright-Patterson Air Force Base, OH 45433-7765, USA, ²Department of Electrical Engineering, Michigan Technological University, 1400 Townsend Drive, 121 EERC bldg, Houghton, MI 49931-1295, USA. 1997.
- [4] G. Barnard. Restoration of Turbulence Degraded Images, Literature study report, Doc. No. QT-RA02-D-P047-001, Thales Advanced Engineering (PTY) Ltd.
- [5] G. Thorpe¹, A. Lambert², D. Fraser², "Atmospheric Turbulence Visualization through Image Time-Sequence Registration", ¹*Boeing Australia Limited*, 363 Adelaide Street, Brisbane, QLD 4000, Australia, ²*School of Electrical Engineering*, University College, University of New South Wales, ADFA Canberra ACT 2600, Australia.
- [6] B. D. Walters, "Heat Shimmer Removal from Images Study and Algorithm Implementation", Dissertation, University of Johannesburg, 2006.
- [7] R. A. Serway, R. J. Beichner, J. W. Jewett, *PHYSICS For Scientists and Engineers with Modern Physics*, fifth edition, part III Thermodynamics and part V Light and Optics, Harcourt College Publishers, 2000.

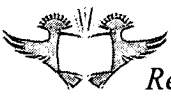
- [8] B. Cohen, V. Avrin, M. Belitsky, I. Dinstein, "Restoration of an image representing a video sequence recorded under turbulent effects", Electrical and Computer Engineering Department, Ben Gurion University of the Negev, Israel.
- [9] R. L. Fante, "Electromagnetic Beam Propagation in Turbulent Media", *Proceedings of the IEEE*, Vol. 63, No. 12, pp 1669-1692, 1975.
- [10] D. Li¹, R. M. Mersereau¹, D. H. Frakes², M. J. T. Smith³, "A New Method for Suppressing Optical Turbulence in Video", ¹School of ECE, Georgia Institute of Technology, Atlanta, ²4-D Imaging Inc, Atlanta, ³School of ECE, Purdue University, West Lafayette, Indiana.
- [11] S. Gepshtein, A. Shtainman, B. Fishbain, L. P. Yaroslavsky, "Restoration of Atmospheric Turbulent Video Containing Real Motion Using Rank Filtering and Elastic Image Registration", Department of Interdisciplinary studies, Faculty of Engineering, Tel-Aviv University, Israel.
- [12] D. Saddot, N. S. Kopeika, S.R. Rotman, "Incorporation of Atmospheric Blurring Effects in Target Acquisition Modelling of Thermal Images", *Infrared Physics Technology*, vol. 36, No. 2, pp. 551 – 564, Elsevier Science Ltd., Great Britain, 1995.
- [13] H. Okayama, L. Wang, "Spatial Coherence Degradation of Light Influenced by Temperature and Aerosol by Use of Atmospheric Turbulence Chamber", *Remote Sensor Environment*, vol. 69, pp. 189-193, Elsevier Science Inc, 655 Avenue of the Americas, New York, NY 10010, 1999.
- [14] J. Sloup, "Visual Simulation of Refraction Phenomena in the Earth's Atmosphere," *Seventh International Conference on Information Visualization (IV'03)*, pp. 452-457, 2003.
- [15] J. L. Lumley, H. A. Panofsky, "The structure of atmospheric turbulence", *Interscience Monographs and Texts in Physics and Astronomy*, New York, Wiley, 1964.



- [16] F. Roddier, "The Effects of atmospheric turbulence in optical astronomy", *Progress in Optics*, Vol. 19, Amsterdam, North-Holland Publishing Co., pp. 281-376, 1981.
- [17] L. G. Brown, "A Survey of Image Registration Techniques", *ACM Computing Surveys*, Vol. 24, No. 4, pp. 325-376, 1992.
- [18] B. Cong, "Encoding Neural Networks to Compute the Atmospheric Point Spread Function", Dept. of Computer Science, California State University, Fullerton, CA 92834 - USA.
- [19] R. C. Gonzales, R. E. Woods, *Digital Image Processing*, Second Edition, Prentice Hall, Upper Saddle River, New Jersey, ISBN 0-201-18075-8, 2001.
- [20] A. Bovik, *Handbook of Image and Video Processing*, Academic Press, Harcourt Place, 32 Jamestown Road, London, 2000.
- [21] J. Duan, G. Qiu, "Novel histogram processing for colour image enhancement", *Proceedings, Third International Conference on Image and Graphics*, pp. 55-58, Dec. 2004.
- [22] H. Shen, P. Li, L. Zhang, Y. Zhao, "A MAP algorithm to super-resolution image reconstruction", *Proceedings Third International Conference on Image and Graphics*, pp. 544-547, Dec 2004.
- [23] A. Zomet, A. Rav-Acha, S. Peleg, "Robust Super-Resolution", *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, vol. 1, pp. 645-650, Dec. 2001.
- [24] M. Elad, Y. Hel-Or, "A Fast Super-Resolution Reconstruction Algorithm for Pure Translation Motion and Common Space-Invariant Blur", *IEEE Transactions on Image Processing*, Vol. 10, No. 8, pp. 1187-1193, August 2001.
- [25] D. R. Cok, Signal Processing method and apparatus for producing interpolated chrominance values in a sampled color image signal. U.S. Patent No. 4,642,678, 1987.

- [26] M. Irani, S. Peleg, "Improving Resolution by Image Registration", *ICVGIP: Graphical Models and Image Processing*, Vol. 53, No. 3, pp. 231-239, May 1991.
- [27] E. C. Ifeachor, B. W. Jervis; *Digital Signal Processing: A Practical Approach*, Second Edition, Prentice Hall, Pearson Education Ltd., 2002.
- [28] I. Goss-Ross, G. Barnard, B. Coetzer. *Restoration of Turbulence Degraded Images*, Algorithm Development, Doc. No. RP-RA02-S-C047-003, Thales Advanced Engineering (PTY) LTD., March 1991.
- [29] P. A. Jansson, *Deconvolution of Images and Spectra*, Second Edition, Academic Press, ISBN 0-12-380222-9, 1997.
- [30] M. Nitzburg, T. Shiota, "Nonlinear Image Filtering with Edge and Corner Enhancement", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 14, No. 8, pp. 826-833, August 1992.
- [31] J. R. Hare, J. P. Reilly, "The iterative deconvolution of linearly blurred images using non-parametric stabilising functions", *Proceedings, 2000 International Conference on Image Processing*, Vol. 3, pp. 770-773, 2000.
- [32] L. I. Rudin, S. Osher, E. Fatemi, "Nonlinear total variation based noise removal algorithms", *Physica D: Nonlinear Phenomena*, Vol. 60, Issue 1-4, pp. 259-268, November 1992.
- [33] D. Bevc, "Compensating for irregular sampling and rugged topology", Stanford Exploration Project, report 84, May 9, 2001.
- [34] D. Kundur, D. Hatzinakos, "Blind image deconvolution", *IEEE Signal Processing Magazine*, Vol. 13, Issue 3, pp. 43-64, May 1996.
- [35] T. F. Chan, Chiu-Kwong Wong, "Total variation blind deconvolution", *IEEE Transactions on Image Processing*, Vol. 7, Issue 3, pp. 370-375, March 1998.

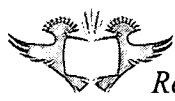
- [36] M. Cannon, "Blind deconvolution of spatially invariant image blurs with phase", *IEEE Acoustics, Speech, and Signal Processing* [see also *IEEE Transactions on Signal Processing*], Vol. 24, Issue 1, pp. 58-63, Feb 1976.
- [37] T. J. Schulz, "Multiframe blind deconvolution of astronomical images", *Journal of Optics Society of America*, Vol. 10 pp. 1064, 1993.
- [38] D. Li, S. Simske, R. M. Mersereau, "Blind Image Deconvolution using constrained variance maximization", *In the Proceedings of the 38th Asilomar Conference on Signals, Systems and Computers*, Pacific grove, CA, USA, 2004.
- [39] A. D. Hillery, R. T. Chin, "Iterative Wiener filters for image restoration", *IEEE Transactions on Signal Processing*, Vol. 39, Issue 8, pp. 1892-1899, Aug 1991.
- [40] Z. Mu, R. J. Plemmons, "Regularization Methods for Image Restoration Based on Autocorrelation Functions", *Proc. SPIE*, Vol. 4116, pp 375-384, November 2000.
- [41] P. J. Burt, E. H. Andelson, *The Laplacian Pyramid as a Compact Image Code*, IEEE, *Transactions on Communications*, vol. com-31, No. 4, April 1983.
- [42] R. A. Serway, R. J. Beichner, J. W. Jewett, *PHYSICS For Scientists and Engineers with Modern Physics*, fifth edition, Chapter 34 – electromagnetic waves, pp 1075-1097, Harcourt College Publishers, 2000.
- [43] Electromagnetic spectrum page, <http://www.cbu.edu/~jvarrian/252/emspec.html>, referenced on 30 April 2007.
- [44] Astronomy and light, <http://physics.fortlewis.edu/Astronomy/astronomy%20today/CHAISSON/AT303/HTML/AT30303.HTM>, referenced on 2 March 2007.



- [45] Converting a colour photo into black and white tutorial, tutorials page, <http://www.cambridgeincolour.com/tutorials/color-black-white.htm>, referenced 4 May 2007.
- [46] H. Okayama, L. Wang, "Spatial Coherence Degradation of Light Influenced by Temperature and Aerosol by Use of Atmospheric Turbulence Chamber", *Remote Sensor Environment*, vol. 69, pp. 189-193, Elsevier Science Inc, 655 Avenue of the Americas, New York, NY 10010, 1999.
- [47] J. McCreary, *Filter options for digital cameras: Pre-processing light for capture*, Digital Photography For What Its Worth, 2004, article can be viewed on <http://dpfwiw.com/filters.htm>, referenced 7 May 2007.
- [48] A. Klaus, K. Karner, P. Elbischger, R. Perko, H. Bischof, "Camera calibration from a single night sky image", *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, pp. 151-157, July 2004.
- [49] Y. Matsushita, E. Ofek, G. Weina, T Xiaoou, S. Heung-Yeung, "Full-frame video stabilization with motion inpainting", *IEEE Transactions on pattern analysis and Machine Intelligence*, Vol. 28, Issue 7, pp. 1150-1163, July 2006.
- [50] F. Gasparini, R. Schettini, "Color Correction for Digital Photographs", *12th International Conference on Image Analysis and Processing*, pp. 646-651, 2003.
- [51] R. A. Serway, R. J. Beichner, J. W. Jewett, *PHYSICS For Scientists and Engineers with Modern Physics*, fifth edition, Chapter 38 – Diffraction and Polarization, pp 1211-1237, Harcourt College Publishers, 2000.
- [52] M. Born, E. Wolf, *Principles of optics, Electromagnetic theory of propagation, interference and diffraction of light*, seventh edition, Cambridge University Press, 2003.



- [53] C. Y. Liu, A. W. Lohmann, "High resolution image formation through the turbulent atmosphere", *Optics Communications*, Vol. 13, no. 1, p. 22-27 1996ol. 8, pp. 372-377, Aug 1973.
- [54] J. Bally, D. Theil, Y. Billawalla, D. Potter, R. F. Loewenstein, F. Mrozek, "A Hartmann differential image motion monitor (H-DIMM) for atmospheric turbulence characterisation", *Publications Astronomical Society of Australia*, Vol. 13, No. 1, pp. 22-27, 1996.
- [55] J. F. Cardosa, A. Souloumiac, "Blind beamforming for non-gaussian signals", *IEE Proceedings F Radar and Signal Processing*, Vol. 140, Issue 6, pp. 362-370, Dec 1993.
- [56] M. A. Alharbi, "Fast Video Stabilization Algorithms", Thesis, Department of Electrical and Computer Engineering, Graduate School of Engineering and Management, Air Force Institute of Technology, Air University, Wright-Patterson Air Force Base, Ohio, June 2006.
- [57] K. L. Baker, E. A. Stappaerts, D. Gavel, S. C. Wilks, J. Tucker, D. A. Silva, J. Olsen, S. S. Olivier, P. E. Young, M. W. Kartz, L. M. Flath, P. Kruelevitch, J. Crawford, and O. Azucena, " High-speed horizontal-path atmospheric turbulence correction with a large-actuator-number microelectromechanical system spatial light modulator in an interferometric phase-conjugation engine," *Optics Letters*, Vol. 29, pp. 1781-1783, 2004.
- [58] E. K. Hege, S. M. Jefferies, M. Loyd-Hart, "Computing and telescopes at the frontiers of optical astronomy", *Computing in Science & Engineering*, Vol. 5, Issue 6, pp. 42-51, Nov.-Dec. 2003.
- [59] M. Cohen, G. Cauwenberghs, M. Vorontsov, G. Carhart, "Focal-Plane Image and Beam Quality Sensors for Adaptive Optics", Army Research Laboratory, Intelligent Optics Laboratory, Adelphi, MD20783, 2001.



- [60] A. Ardeberg, T. Anderson, "Matching and improving the best atmospheric turbulence conditions with very large telescopes", *New Astronomy reviews*, pp 557–562, Elsevier Science, 1998.
- [61] J. Mallozzi, "Adaptive Optics Goes Earth-Bound", Recent Editorial, R&D Magazine, <http://www.rdmag.com/ShowPR.aspx?PUBCODE=014&ACCT=1400000100&ISSUE=0410&RELTYPE=PR&ORIGRELTYPE=FE&PRODCODE=00000000&PRODLETT=S&CommonCount=0>, Referenced on 6 March 2007.
- [62] D. Slater, "MIST, an instrument for long range horizontal path imaging research", SPIE Newsroom, <http://newsroom.spie.org/x822.xml?highlight=x544>, referenced on 6 March 2007.
- [63] G. Angers, "Choosing an Eyepiece", *The Journal of the Royal Astronomical Society of Canada*, Vol. 92, No. 6, pp 329-330, 1998.
- [64] Resource Centre, Buying Guides – Types of Telescope Designs page, http://www.escience.ca/resource/buying_guides/TeleDesigns.html, referenced on the 7th March 2007.
- [65] QUESTAR MFL 90 Manual, Multi Focal Length Lens on Consumer Lines LD Surveillance Systems page, <http://www.company7.com/questar/surveillance/quemfl90.html>, referenced on 6 March 2007.
- [66] Nexstar 8 SE, telescope products page, <http://www.celestron.com/c2/product.php?CatID=13&ProdID=416>, referenced 7 March 2007.
- [67] Telescopes, ETX Premier Edition Page, http://www.meade.com/etx_premier/index.html, referenced 7 March 2007.
- [68] Telescopes, LX90GPS page, <http://www.meade.com/lx90gps/index.html>, referenced 7 March 2007.

- [69] Video Surveillance Cameras and Video Astrophotography page, <http://www.lafterhall.com/surveillancecameras.html>, referenced 9 March 2007.
- [70] AV3100, Products page, http://www.arecontvision.com/products_3100.html, referenced 9 March 2007.
- [71] PixeLINK PL-A741 MV Camera datasheet, Industrial and machine vision, products page, http://www.pixelink.com/products_info.asp?id=11, referenced on the 27 March 2007.
- [72] S. Winkler, *Digital Video Quality: vision models and metrics*, John Wiley & Sons Ltd., The Atrium, southern Gate, Chichester, West Sussex PO19 8SQ, England, 2005.
- [73] P. Teo, D. Heeger, "Perceptual Image Distortion", *First IEEE International Conference on Image Processing*, vol. 2, pp 982-986, November 1994.
- [74] S. J. Ko, S. H. Lee, and K. H. Lee, "Digital image stabilizing algorithms based on bit-plane matching," *IEEE Transaction on Consumer Electronics*, vol. 44, no. 3, pp. 617–622, 1998.
- [75] S. Erturk, "Digital image stabilization with sub-image phase correlation based global motion estimation," *IEEE Transaction on Consumer Electronics*, vol. 49, no. 4, pp. 1320–1325, 2003.
- [76] D.H. Frakes, J.W. Monaco, M.J.T. Smith, "Suppression of Atmospheric Turbulence in Video Using an Adaptive Control Grid Interpolation Approach", School of Electrical and Computing Engineering, Georgia Institute of Technology, Atlanta, GA, 2001.
- [77] A. Robles-Kelly, E. R. Hancock, "Vector Field smoothing via heat flow", *Proceedings of the 17th International conference on pattern recognition*, Vol. 2, pp. 94-97, August 2004.



- [78] H. W. Haussecker, D. J. Fleet, "Computing Optical Flow with Physical Model of Brightness", *IEEE transactions on pattern analysis and machine intelligence*, Vol. 23, No. 6, pp. 661-673, June 2001.
- [79] C. Hung-Chang, L. Shang-Hong, L. Kuang-Rong, "A robust and efficient video stabilization algorithm", *IEEE International conference on multimedia and expo*, Vol. 1, pp. 29-32, June 2004.
- [80] A. Labeyrie, "Attainment of diffraction-limited resolution in large telescopes by Fourier analyzing speckle patterns in star images", *Astron. Astrophys.*, Vol. 6, pp. 85-87, 1970
- [81] M. F. Hoffmann, "A Search for Binary Stars Using Speckle Interferometry", Senior Research Thesis, Centre for Imaging Science, Rochester Institute of Technology, May 2000.
- [82] C. L. Matson, M. Fox, E. K. Hege, L. Hluck, J. Drummond, and D. Harvey, "Deep-space satellite-image reconstructions from field data by use of speckle imaging techniques: images and functional assessment", *Applied Optics*, Vol. 36, pp. 3120-3126, 1997.
- [83] S. N. Karbelkar, R. Nityananda, "Atmospheric Noise on the Bispectrum in Optical Speckle Interferometry", *Journal of Astrophysics and Astronomy*, Vol. 8, pp. 271-274, 1987.
- [84] A. W. Lohmann, G. Weigelt, and B. Wirtitzer, "Speckle masking in astronomy: triple correlation theory and applications", *Applied Optics*, Vol. 22, pp. 4028-4037, 1983.
- [85] H. Bartelt, A. W. Lohmann, and B. Wirtitzer, "Phase and amplitude recovery from bispectra", *Applied Optics*, Vol. 23, pp. 3121-3129, 1984
- [86] C. J. Carrano, "Speckle imaging over horizontal paths", *Proceedings of the SPIE — High Resolution Wavefront Control: Methods, Devices, and Applications IV*, 4825, 2002.



- [87] S.A. Dianat and M.R. Raghuveer, "Fast Algorithms for Phase and Magnitude Reconstruction From Bispectra", *Optical Engineering*, Vol. 29, No. 5, pp. 504-512, 1990.
- [88] S. K. Saha, "High resolution imaging by employing passive and active approaches", Indian Institute of Astrophysics, Bangalore, India, April 2003.
- [89] A. Hyvärinen, E. Oja, "Independent Component Analysis: Algorithms and Applications", Neural Networks Research Centre, Helsinki University of Technology, Finland, 2000.
- [90] J.-F. Cardoso, P. Comon, "Independent Component Analysis, a Survey of Some Algebraic Methods", *IEEE International Symposium on Circuits and Systems*, vol. 2, pp. 93-96, Atlanta, USA, May 1996.
- [91] J.-F. Cardoso, "Multidimensional Independent Component Analysis", *proceedings of the 1998 IEEE International Conference on Acoustics, Speech, and Signal Processing*, vol. 4, pp. 1941-1944, Seattle, WA, USA, May 1998.
- [92] I. Kopriva, Q. Du, H. Szu, "Independent Component Analysis Approach to Image Sharpening in the Presence of Atmospheric Turbulence", *Optics Communications*, vol. 233, Issue 1-3, pp. 7-14, March 2004.
- [93] L. Tao, R. Tompkins, V. K. Asari, "An Illuminance-Reflectance Nonlinear Video Enhancement Model for Homeland Security Applications", *Proceedings of the 34th Applied Imagery and Pattern Recognition Workshop*, ISBN: 0-7695-2479-6, October 2005.
- [94] H. T. Ngo, L. Tao, V. K. Asari, "A nonlinear technique for enhancement of color images: an architectural perspective for real time applications", *Proceedings of the 33rd Applied Imagery Pattern Recognition workshop*, pp. 124-129, Oct 2004.



- [95] O. Nuri, Capt. Ender, "A non-linear technique for the enhancement of extremely non-uniform lighting image", *Journal of Aeronautics and Space Technologies*, Vol. 3, No. 2, pp. 37-47, June 2007.
- [96] D. Korff, "Analysis of a method for obtaining near-diffraction-limited information in the presence of atmospheric turbulence", *Journal of the Optical Society of America*, vol. 63, no. 8, pp 971-980, August 1973.
- [97] E. Pehlemann, O. Von der Luehe, "Technical aspects of the speckle masking phase reconstruction algorithm", *Astronomy and Astrophysics*, vol. 216, pp 337-346, 1989.
- [98] D. Li, R. M. Mersereau, S. Simske, "Blur Identification Based on Kurtosis Minimization", *IEEE International Conference on Image Processing*, Vol. 1, pp. 905-908, Sept 2005.
- [99] O. Von der Luehe. "Speckle imaging of solar small scale structure", *Astronomy and Astrophysics*, vol. 268, pp 374-390, 1993.
- [100] D. Li, "Restoration of Atmospheric Turbulence Degraded Video using Kurtosis Minimization and Motion Compensation", Doctorate of Philosophy, School of Electrical and Computer Engineering, Georgia Institute of Technology, May 2007.
- [101] J.C. Russ, *The Image Processing Handbook*, CRC Press, Boca Raton, pp. 101, 1992.
- [102] M.C. Roggemann, B. Welsh, *Turbulence Effects on Imaging Systems*, CRC Press, Boca Raton, pp. 57, 1996.
- [103] H. Ingensand, "Concepts and solutions to overcome the refraction problem in terrestrial precision measurement", *Integration of Techniques and Corrections to Achieve*



Accurate Engineering Survey, International Congress, Washington, D.C., USA, April 2002.

- [104] P. Shirley, B. Smits, H. Hu, E. Lafortune, "A practitioners' assessment of light reflection models", *Proceedings, The Fifth Pacific Conference on Computer Graphics and Applications*, pp. 40-49, Oct 1997.
- [105] J.L.C. Sanz, T.T. Huang, *Theorems and Experiments on Image Reconstruction from Zero Crossings*, IBM Almaden Research Center, Published 1987.
- [106] A. M. Waxman, K. Wohn, "Image theory: a framework for 3-D inference from time-varying imagery", In *C. Brown (Ed.), Advances in Computer Vision*, Vol. 1, pp. 165-224, 1988.
- [107] C. F. Wu, M. Hamada, *Experiments: Planning, Analysis, and Parameter Design Optimization*, Wiley, Published 2000.
- [108] K. Fang, A. Sudjianto, *Design and Modeling for Computer Experiments*, CRC Press, Published 2006.
- [109] R.T. Apteker, J.A. Kisimov, H. Neishlos, "Video acceptability and frame rate", *IEEE Multimedia*, Vol. 2, Issue 3, pp. 32-40, 1995.
- [110] B.D. Walters, "Comparison of two terrestrial atmospheric turbulence suppression algorithms", *IEEE Africon 2007*, ISBN: 978-1-4244-0987-7, pp. 1-7, Oct. 2007.

10 ADDENDUM A

SPECKLE MASKING ALGORITHM CODE

10.1 'MAKEHANNING' FUNCTION

```
function hWindow = makeHanning(length,percent)
%this function constructs a two dimensional square hanning window
  based on a given
%length and apodization percentage. The two dimensional window is
  returned

han2D = ones (length,length);

%% construct hanning function
windowLength = round(length*(percent)/100); %determine length of
  hanning function
han1D = hann(windowLength,'periodic');

%% construction of full length of function
han1Dfull = ones(length);
halfWindowLength = ceil(windowLength/2); %calculate integer half
restWindowLength = windowLength - halfWindowLength;
han1Dfull(1:halfWindowLength) = han1D(1:halfWindowLength);
han1Dfull(length-restWindowLength+1:length) =
  han1D(halfWindowLength+1:windowLength);

%% construction 2D window from function
for k = 1:length, %iterate row
  for l = 1:length, %iterate column
    %han2D(k,l) = min(han1Dfull(k),han1Dfull(l)); %create 2D
  window
    han2D(k,l) = (han1Dfull(k)*han1Dfull(l)); %create 2D window
  end
end

hWindow = han2D;
```

10.2 'IMAGEREGISTRATION' FUNCTION

```
function regIm = imageRegistration(imageToReg,refImage,border)
%performs image registration on imageToReg removing translational
  motion
%given reference image refImage. The border edge width on imageToReg
  across which
```



```
%registration is performed is specified by the scalar border.
%parameter sizes:   refImage           NxM
%                   imageToReg        (N+2*border) x (M+2*border)
%                   regIm             NxM
%
%function developed to coincide and support speckle masking
%procedure
%By BD Walters

[rw,cl] = size(refImage);
bestFitImage = zeros(rw,cl);
currentError = rw*cl;
twiceBorder = 2*border;

% cc = normxcorr2(refImage,imageToReg);
% [max_cc, imax] = max(abs(cc(:)));
% [ypeak, xpeak] = ind2sub(size(cc),imax(1));
% corr_offset = [ (ypeak-size(imageToReg,1)) (xpeak-
%               size(imageToReg,2)) ];
%
% bestFitImage = imageToReg((1-corr_offset(1)):(rw-
%               corr_offset(1)),(1-corr_offset(2)):(cl-corr_offset(2)));

for rborder = 0:twiceBorder,
    for cborder = 0:twiceBorder,

        %temp =
        corr2(imageToReg((1+rborder):(rw+rborder),(1+cborder):(cl+cborder)),refImage);
        temp =
        sumsqr(imageToReg((1+rborder):(rw+rborder),(1+cborder):(cl+cborder)) - refImage);

        if temp<currentError
            bestFitImage =
            imageToReg((1+rborder):(rw+rborder),(1+cborder):(cl+cborder));
            currentError = temp;
        end

    end

end

regIm = bestFitImage;
```

10.3 'BSPECALC' FUNCTION

```
function finalBSpec = BSPECALC(IG,imsize,numtoavg)
%Bispectrum creation following path q through 4D space
%optimized to calculate some necessary parts only corresponding with
%phase
%reconstruction

%created by BD Walters
```



```
BSpec = zeros(imsizes,imsizes,numtoavg);

%% calculate 0X except 01
num = 1;
qr = 0;
qc = 1;
kiter = 0;
for liter = 2:imsizes-1
    pr = kiter - qr;
    pc = liter - qc;

    BSpec(kiter+1,liter+1,num)=
        (IG(pr+1,pc+1)*IG(qr+1,qc+1)*conj(IG(kiter+1,liter+1)));
end

%% calculate X0 except 10
qr = 1;
qc = 0;
liter = 0;
for kiter = 2:imsizes-1
    pr = kiter - qr;
    pc = liter - qc;

    BSpec(kiter+1,liter+1,num)=
        (IG(pr+1,pc+1)*IG(qr+1,qc+1)*conj(IG(kiter+1,liter+1)));
end

%% calculate 11
pr = 0;
pc = 1;
qr = 1;
qc = 0;

kiter = 1;
liter = 1;
BSpec(kiter+1,liter+1,num)=
    (IG(pr+1,pc+1)*IG(qr+1,qc+1)*conj(IG(kiter+1,liter+1)));

%% calculate 1X except 11
qr = 1;
qc = 1;
kiter = 1;
for liter = 2:imsizes-1,
    pr = kiter - qr;
    pc = liter - qc;

    BSpec(kiter+1,liter+1,num)=
        (IG(pr+1,pc+1)*IG(qr+1,qc+1)*conj(IG(kiter+1,liter+1)));
end

%% calculate X1 except 11
qr = 1;
qc = 1;
```



```
liter = 1;
for kiter = 2:imsize-1,
    pr = kiter - qr;
    pc = liter - qc;

    BSpec(kiter+1,liter+1,num)=
        (IG(pr+1,pc+1)*IG(qr+1,qc+1)*conj(IG(kiter+1,liter+1)));
end

%% calculate all other k and l indices

occurs = 0;%initialize for use

for kiter = 2:imsize-1,
    for liter = 2:imsize-1,

        qr = floor(kiter/2);
        qc = floor(liter/2);
        occurs = qr+qc-1; %total possible average number

        if (occurs>numtoavg) %ensure doesn't exceed specified number
to avg
            occurs = numtoavg;
        end

        for num = 1:occurs, %iterate through averages

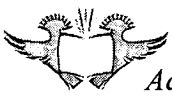
            pr = kiter - qr;
            pc = liter - qc;

            BSpec(kiter+1,liter+1,num)=
                (IG(pr+1,pc+1)*IG(qr+1,qc+1)*conj(IG(kiter+1,liter+1)));

            if (qr>qc) %decrease greater vector component of q for
next iteration
                qr = qr - 1;
            else
                qc = qc - 1;
            end
        end
    end
end

%% end

finalBSpec = BSpec;
%end of bispectrum calculation
```



10.4 'PHASESCALC' FUNCTION

```
function calcPhases = phasesCalc(BSpec,IG,imsize,numtoavg)
%calculates the fourier phases from the 4D bispectrum created by
    BSpecCalc
%optimized to make use of higher spectral data in bispectrum
%created by BD Walters

expphases = zeros(imsize,imsize);

expphases(2,1) = IG(2,1);%exp(i*angle(IG(2,1)));%initial unknown
    1,0
expphases(1,2) = IG(1,2);%exp(i*angle(IG(1,2)));%initial unknown
    0,1

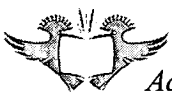
%% calculate 0X except 01
num = 1;
qr = 0;
qc = 1;
kiter = 0;
for liter = 2:imsize-1
    pr = kiter - qr;
    pc = liter - qc;
    expphases(kiter+1,liter+1) =
        exp(i.*(angle(expphases(qr+1,qc+1))+angle(expphases(pr+1,pc+1))-
            angle(BSpec(kiter+1,liter+1,num)))));
end

%% calculate X0 except 10

qr = 1;
qc = 0;
liter = 0;
for kiter = 2:imsize-1
    pr = kiter - qr;
    pc = liter - qc;
    expphases(kiter+1,liter+1) =
        exp(i.*(angle(expphases(qr+1,qc+1))+angle(expphases(pr+1,pc+1))-
            angle(BSpec(kiter+1,liter+1,num)))));
end

%% calculate 11
qr = 1;
qc = 0;
pr = 0;
pc = 1;
kiter = 1;
liter = 1;

expphases(kiter+1,liter+1) =
    exp(i.*(angle(expphases(qr+1,qc+1))+angle(expphases(pr+1,pc+1))-
        angle(BSpec(kiter+1,liter+1,num)))));
```



```
%% calculate lX except 11
```

```
qr = 1;  
qc = 1;  
kiter = 1;  
for liter = 2:imsize-1,  
    pr = kiter - qr;  
    pc = liter - qc;  
    expphases(kiter+1,liter+1) =  
        exp(i.*(angle(expphases(qr+1,qc+1))+angle(expphases(pr+1,pc+1))-  
            angle(BSpec(kiter+1,liter+1,num))));  
end
```

```
%% calculate Xl except 11
```

```
qr = 1;  
qc = 1;  
liter = 1;  
for kiter = 2:imsize-1,  
    pr = kiter - qr;  
    pc = liter - qc;  
    expphases(kiter+1,liter+1) =  
        exp(i.*(angle(expphases(qr+1,qc+1))+angle(expphases(pr+1,pc+1))-  
            angle(BSpec(kiter+1,liter+1,num))));  
end
```

```
%% calculate all other k and l indices
```

```
occurs = 0;%initialize for use
```

```
for kiter = 2:imsize-1,  
    for liter = 2:imsize-kiter,  
  
        qr = floor(kiter/2);  
        qc = floor(liter/2);  
        occurs = qr+qc-1; %total possible average number  
  
        if (occurs>numtoavg) %ensure doesn't exceed specified number  
            to avg  
                occurs = numtoavg;  
            end  
  
        for num = 1:occurs, %iterate through averages  
  
            pr = kiter - qr;  
            pc = liter - qc;  
  
            temp =  
            exp(i.*(angle(expphases(qr+1,qc+1))+angle(expphases(pr+1,pc+1))-  
                angle(BSpec(kiter+1,liter+1,num))));  
  
            expphases(kiter+1,liter+1) = expphases(kiter+1,liter+1)  
            + temp/occurs;  
  
            if (qr>qc) %decrease greater vector component of q for  
                next iteration
```



```
        qr = qr - 1;
    else
        qc = qc - 1;
    end
end

end

end

%% end
phases = angle(expphases);
checkflag = 0;
    if mod(imesize,2)==0
        checkflag=1;
    end
    for kt = 2:imesize,
        for lt = 2:imesize-kt+2,
            if (checkflag==1)
                if (((imesize-kt+2)==kt)&&(imesize-lt+2)==lt)

                    else
                        phases(imesize-kt+2,imesize-lt+2) = 0-
phases(kt,lt);
                    end
                else
                    phases(imesize-kt+2,imesize-lt+2) = 0-
phases(kt,lt);
                end
            end
        end
    end

end

calcPhases = phases;
```

10.5 MAIN FILE (RUN)

```
%4D method image2d

clc;
clear all;
close all;

%%load video
directoryToOpen = ('D:\workvids2\noiseRem\');
fileToOpen = ('Mid freq no filt0001-0');

mov = aviread(strcat(directoryToOpen,fileToOpen, '.avi'), [1:110]);
    %open avi

[M,N] = size(mov); % determine number of frames

framenum = 100; %number of frames to use
    ***set***
```



```
framenumstart = 1;
framenumend = framenumstart + framenum-1;

border = 5; %border around which image registration takes place
    ***set***
twiceBord = 2*border;
imsize = 470;%size of image
    ***set***
xshift = 80; %horizontal offset within video
    ***set***
yshift = 0; %vertical offset down within video
    ***set***

%%tiling and apodization setup
subISize = 32; %sub image size
    ***set***
hPercent = 100; %percent apodization
    ***set***

overlapLength = round((subISize*(hPercent)/100)/2);

numtoavg = 16; %number of phases to average over if available
    ***set***

%calculate number of sub images across row and column
offset = subISize-overlapLength;
rowSubINum = floor((imsize-overlapLength)/(offset));
colSubINum = floor((imsize-overlapLength)/(offset));

w = (makeHanning(subISize,hPercent)); %create hanning apodization
    window

newImage = zeros(imsize,imsize); %pre-create a blank image for
    memory
tile = zeros(subISize,subISize); %pre-create a blank tile

%%create reference bispectrum Ispec
Ispec = zeros(subISize,subISize);
for x = -subISize/2:0,
    for y = -subISize/2:0,
        f = sqrt(x^2 + y^2);
        m = y + subISize/2 + 1;
        n = x + subISize/2 + 1;

        Ispec(m,n) = psReferenceSE(0.15,0.055,f/(subISize));
        Ispec((m),(subISize+1-n)) = Ispec(m,n);
        Ispec((subISize+1-m),(n)) = Ispec(m,n);
        Ispec((subISize+1-m),(subISize+1-n)) = Ispec(m,n);
    end
end

Ispec = fftshift(Ispec);

%%perform tiling image creation
tic %start stopwatch
```




```
d = mov(framenumstart); %\  
truecimg = d.cdata;      %- get first image frame  
img = truecimg(:,:,:);  %/  
  
Igfull = rgb2gray(img);  
%first frame used for reference in image registration  
refIg =  
    double(Igfull(1+yshift+border:imsize+yshift+border,1+xshift+border:  
        imsize+xshift+border));  
  
rFrom = 1+yshift  
rTo = imsize+yshift+2*border  
cFrom = 1+xshift  
cTo = imsize+xshift+2*border  
  
for ktile = 0:rowSubINum-1, %iterate through row subimage numbers  
    for ltile = 0:colSubINum-1, %iterate through column subimage  
        numbers  
  
            PS = zeros(subISize,subISize);  
            BSpect = zeros(subISize,subISize,numtoavg);  
            phases = zeros(subISize,subISize);  
  
            averageIG = zeros(2,2,'double');  
  
            for fnum = framenumstart:1:framenumend,  
  
                %get image frame and sub portion  
  
                d = mov(fnum);  
                truecimg = d.cdata;  
                img = truecimg(:,:,:);  
                Igfull = rgb2gray(img);  
                Ig = double(Igfull(rFrom:rTo,cFrom:cTo)); %crop to size  
  
                mtile = (1+(ktile*offset));  
                ntile = (1+(ltile*offset));  
  
                %perform basic global image registration  
                tile(1:subISize,1:subISize) =  
                imageRegistration((Ig(mtile:(mtile+subISize+twiceBord-  
                    1),ntile:(ntile+subISize+twiceBord-  
                    1))./255),refIg(mtile:(mtile+subISize-1),ntile:(ntile+subISize-  
                    1))./255,border);  
  
                %%Fourier transform image  
                IG = fft2(tile);  
  
                temp = averageIG(1,2);  
                averageIG(1,2) = temp + (IG(1,2))/framenum;  
                temp = averageIG(2,1);  
                averageIG(2,1) = temp + (IG(2,1))/framenum;  
  
                apoIG = IG;  
                PS = PS + (apoIG.*conj(apoIG)/(subISize^2))/framenum;
```



```
        IG(1,1) = 0;
        BSpect = BSpect +
BSpecCalc(IG,subISize,numtoavg)./framenum;

    end

    %%adjust power spectrum
    PS = PS./((Ispec));

    %%Fourier phase calculation
    phases = phasesCalc(BSpect,averageIG,subISize,numtoavg);

    newIG = sqrt(PS*(subISize^2)).*exp(i.*phases);
    imageOut = abs(ifft2(newIG)).*255;

    tile = imageOut;
    tile = tile.*w; %apply apodization

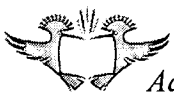
    newImage(mtile:(mtile+subISize-1),ntile:(ntile+subISize-1))
= newImage(mtile:(mtile+subISize-1),ntile:(ntile+subISize-1)) +
tile;

end
end

newIm = imadjust(uint8(newImage(16:imshow-33,16:imshow-
33)),stretchlim(uint8(newImage(16:imshow-33,16:imshow-33)),[0
1]),[]);

toc %stop stopwatch and display elapsed time

figure
imshow(uint8(Ig)),title('original') %display original image
figure
imshow(uint8((newIm))),title('computed') %display reconstructed
image
```



11 ADDENDUM B

IMAGE REGISTRATION AND LAPLACIAN PYRAMID IMAGE FUSION ALGORITHM CODE

11.1 'IMAGEFUSION' FUNCTION

```
function [imageOut] = imagefusion(im1,im2,t)
%fuse two images using laplacian pyramid decomposition and energy
  given threshold t

%% gaussian

a = 0.4;
b = 0.25;
c = b - a/2;
wt = [c b a b c];

%im1

%fprintf(1,'gau im1: level0\n');
Glevel10 = im1;
Glevel11 = greduce(Glevel10,wt);
Glevel12 = greduce(Glevel11,wt);
Glevel13 = greduce(Glevel12,wt);

%im2

Glevel20 = im2;
Glevel21 = greduce(Glevel20,wt);
Glevel22 = greduce(Glevel21,wt);
Glevel23 = greduce(Glevel22,wt);

%% Laplacian

%im1

%fprintf(1,'lap im1: level0\n');
level10 = Glevel10 - gexpand(Glevel11,wt);
level11 = Glevel11 - gexpand(Glevel12,wt);
level12 = Glevel12 - gexpand(Glevel13,wt);
level13 = Glevel13 - gexpand(Glevel14,wt);

%im2

level20 = Glevel20 - gexpand(Glevel21,wt);
```



```
level21 = Glevel21 - gexpand(Glevel22,wt);
level22 = Glevel22 - gexpand(Glevel23,wt);
level23 = Glevel23 ;%- gexpand(Glevel24,wt);

%energy fusion

%fprintf('START');
%t = 0.25; %threshold override

[k,l] = size(level13);

energy13 = imgenergy(level13,k,l);
energy23 = imgenergy(level23,k,l);
match33 = imgmatch(level13,level23,k,l);
[wt13,wt23] = imgweights(match33,energy13,energy23,t);

level3 = ceil(wt13.*level13 + wt23.*level23); %level 3 of
recombination

[k,l] = size(level12);
energy12 = imgenergy(level12,k,l);
energy22 = imgenergy(level22,k,l);
match32 = imgmatch(level12,level22,k,l);
[wt12,wt22] = imgweights(match32,energy12,energy22,t);

level2 = ceil(wt12.*level12 + wt22.*level22) + gexpand(level3,wt);
%level 2 of comb

[k,l] = size(level11);
energy11 = imgenergy(level11,k,l);
energy21 = imgenergy(level21,k,l);
match31 = imgmatch(level11,level21,k,l);
[wt11,wt21] = imgweights(match31,energy11,energy21,t);

level1 = ceil(wt11.*level11 + wt21.*level21) + gexpand(level2,wt);
%level 1 of comb

[k,l] = size(level10);
energy10 = imgenergy(level10,k,l);
energy20 = imgenergy(level20,k,l);
match30 = imgmatch(level10,level20,k,l);
[wt10,wt20] = imgweights(match30,energy10,energy20,t);

level0 = ceil(wt10.*level10 + wt20.*level20) + gexpand(level1,wt);
%level 0 of comb

imageOut = level0;
```



11.2 'GREDUCE' FUNCTION

```
function IResult = greduce(I, Wt)

[h w]= size(I);

hnew = ceil(h * 0.5);
wnew = ceil(w * 0.5);

%% Pad the boundaries.

I = [ I(1,:) ; I(2,:) ; I ; I(h-1,:); I(h,:) ]; % Add two rows
      towards the beginning and the end.
I = [ I(:,1) I(:,2) I I(:, w -1) I(:,w) ]; % Add two
      columns towards the beginning and the end.

I = double(I);

IResult = zeros(hnew, wnew); % Initialize the array in the beginning
for i = 0 : hnew -1
    for j = 0 : wnew -1
        A = [];
        for m = -2 : 2
            for n = -2 : 2
                tmpval = I ( 2 * i + m + 3, 2 * j + n + 3 ) * Wt(m +
3) * Wt(n + 3);
                A = [A, tmpval] ;
            end
        end
        IResult(i + 1, j + 1)= sum(A);
        % Bad array arithmetic - Matlab array indices start at 1.
        % and the algorithm assumes 0. Hence got to do it.
    end
end
end
```

11.3 'GEXPAND' FUNCTION

```
function IResult = gexpand(I, Wt)

[h w]= size(I);
hnew = ceil(h * 2);
wnew = ceil(w * 2);

%% Pad the boundaries.

weight = Wt;
I = [ I(1,:) ; I ; I(h,:) ]; % Pad the top and bottom rows.
I = [ I(:,1) I I(:,w) ]; % Pad the left and right columns.

I = double(I);

IResult = zeros(hnew, wnew); % Initialize the array in the beginning
```



```

for i = 0 : hnew - 1
% fprintf('\n %d', i);
  for k = 0 : wnew - 1
    A = [];
    for m = -2 : 2
      for n = -2 : 2
        pixeli = (i - m)/2 ;
        pixelj = (k - n)/2 ;
        if ( (floor(pixeli) == pixeli) & (floor(pixelj) ==
pixelj ) & pixeli >= 0 & pixelj >= 0 )
          pixeli = pixeli + 1;
          pixelj = pixelj + 1;
          tmpval = I (pixeli, pixelj) * weight(m + 3) *
weight(n + 3);
          A = [A, tmpval] ;
        end
      end
    end
    IResult(i + 1, k + 1)= 4 * sum(A);
    % Bad array arithmetic - Matlab array indices start at 1.
    % and the algorithm assumes 0. Hence got to do it.
  end
end
end

```

11.4 'IMGENERGY' FUNCTION

```

function calcEnergy = imgenergy(img, x, y)
%calculates the local energy of the image 'img' with intensity
  values
%between 0 and 1 using a 3x3 window, given the rows 'x' of the image
%and columns 'y'
% BY BD WALTERS

img1 = [img(1,:) ; img ; img(x,:)]; % Pad top and bottom row
img1 = [img1(:,1)  img1  img1(:,y)]; % Pad left and right column

calcEnergy = zeros(x,y);
for m = 2:x,
  for n = 2:y,
    for k = -1:1,
      for l = -1:1,
        calcEnergy(m,n) = calcEnergy(m,n) + (img1(m + k, n +
1))^2;
      end
    end
  end
end
end
end

```



11.5 'IMGMATCH' FUNCTION

```
function match = imgmatch(img1, img2, x, y)
%This function calculates the match between two images 'img1' and
'img2'
%using their local energy based on a 3x3 window, function also
receives the
%rows and columns of the two same sized images as 'x' and 'y'

%img1 and img2 have intensities between 0 and 1

% BY BD WALTERS

newimg1 = [img1(1,:) ; img1 ; img1(x,:) ]; %Pad top and bottom
row
newimg1 = [newimg1(:,1) newimg1 newimg1(:,y)]; %Pad left and right
column

newimg2 = [img2(1,:) ; img2 ; img2(x,:) ]; %Pad top and bottom
row
newimg2 = [newimg2(:,1) newimg2 newimg2(:,y)]; %Pad left and right
column

match = zeros(x,y); %initialization

for m = 2:x, %iterate rows
    for n = 2:y, %iterate columns
        for k = -1:1,
            for l = -1:1,
                match(m,n) = match(m,n) + (newimg1(m + k, n +
1))* (newimg2(m + k, n + l));
            end
        end
    end
end

match = match./ (imgenergy(img1,x,y) + imgenergy(img2,x,y) + 0.001);
```

11.6 'IMGWEIGHTS' FUNCTION

```
function [wt1,wt2] = imgweights(match12, energy1, energy2,
threshold)
%This function calculates the weight matrices for two images based
on the
%images match and their energies. In selection of the appropriate
weights,
%a threshold is also used. The two weight matrices can then be used
to fuse
%two images. note it is assumed two images where of the same size.
ie
```



```
%dimensions of first three parameters are the same. parameter  
  threshold is  
%between 0 and 1
```

```
% BY BD WALTERS
```

```
[x,y] = size(match12); %get image sizes
```

```
wt1 = zeros(x,y); %initialisation of weight 1
```

```
for m = 1:x,  
  for n = 1:y,  
    if (match12(m,n) <= threshold)  
      if (energy1(m,n) > energy2(m,n))  
        wt1(m,n) = 1;  
      else  
        wt1(m,n) = 0;  
      end  
    else  
      if (energy1(m,n) > energy2(m,n))  
        wt1(m,n) = 0.5 + 0.5*(1 - match12(m,n))/(1 -  
threshold);  
      else  
        wt1(m,n) = 0.5 - 0.5*(1 - match12(m,n))/(1 -  
threshold);  
      end  
    end  
  end  
end  
end
```



```
wt2 = ones(x,y) - wt1;
```

11.7 MAIN FILE (RUN)

```
%main file  
%  
%matlab code by (where not specified):  
%  
%           Bryn Walters  
%           University Of Johannesburg  
%  
%2d Registration and associated code by:  
%  
%Dartmouth College, by Senthil Periaswamy under the direction of  
  Hany Farid.  
%  
% References:  
% -----  
% 1) S. Periaswamy and H. Farid. "Elastic registration in the  
  presence of  
  intensity variations", IEEE Transactions on Medical Imaging, in  
  press, 2003.
```




- % 2) S. Periaswamy. "General-purpose medical image registration",
Ph.D.
% Dissertation, Department of Computer Science, Dartmouth
College, 2003
- % 3) S. Periaswamy, J.B. Weaver, D.M. Healy Jr., D. Rockmore, P.J.
Kostelec,
% and H. Farid. "Differential Affine Motion Estimation for
Medical Image
% Registration" Proceedings of the SPIE - The International
Society for
% Optical Engineering, 4119, pp. 1066-1075, 2000.

close all;
clear all

%% operator parameters

alpha = 1/8; %averaging amount, 1/2 1/4 1/8 1/16 or 1/32

%% get images

mov = aviread('d:\workvids\fusion\chousel_bw_bright.avi'); %30 fps

[M,N] = size(mov); % determine number of frames

a = 1; %image pixels to focus on; from a to b (rows)

b = 240;

c = 1; %c to d (columns)

d = 320;

dig = mov(1);

truedimg = dig.cdata;

img = truedimg(a:b,c:d,:);

img = rgb2gray(img);

imggray = img;

%% run algorithm

[imgx,imgy] = size(imggray); %get size of video frames

imgavg = double(imggray); %initial image average set as first image

z = zeros(imgx,imgy);

%% parameters for registration

%%

params.main.boxW = 5;

params.main.boxH = 5;

params.main.model = 4;

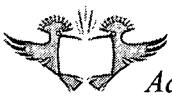
params.main.minSize = 32;

params.main.dispFlag = 0;

params.main.padSize = 2;

params.smooth.loops_inner = 30;

params.smooth.loops_outer = 3;



```

params.smooth.lamda      = [1e11 1e11 1e11 1e11 1e11 1e11 1e11 1e11
1e11];
params.smooth.deviant    = [0 0 0 0 0 0 1 0];
params.smooth.dweight    = [0 0 0 0 0 0 0 0];

params.glob.flag         = 1;
params.glob.model        = 4;
params.glob.iters        = 7;
params.glob.numLevels    = 100;
params.glob.dispFlag     = 0;
params.glob.minSize      = 32;

params.em.sigma_global   = 0.01;
params.em.sigma_local    = 0.1;
params.em.useEdgeMask    = 0;
params.em.applybcFlag    = 0;
params.em.flagSmooth     = 1;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
N=100; %override number of frames to process
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

previmg = double(imggray); %previous fused image
figure
tic
for count = 1:N, %work with sets of eight images

    %%
    fprintf('count: %5.0d \n',count);
    %%

    dig = mov(count);
    truecimg = dig.cdata;
    img = truecimg(a:b,c:d,:);
    imggray = double(rgb2gray(img));

    imgavg = ratioAvg(imggray,imgavg,alpha); %calculate avg at step
    count

    i1 = double(imggray) / 255;
    i2 = double(imgavg) / 255;

    [M,ib,ic] = register2dem_global(i1,i2,params); %register
    image to obtain flow field

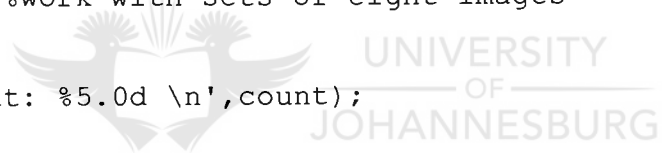
    ilw = double(aff_warp(i1,M)) * 255; %warp image using flow
    field

    fusedimage = imagefusion(imgavg,ilw,0.5); %fuse image

    imshow(uint8(fusedimage))
    F(count) = getframe;

end
toc
movie2avi(F,'d:\workvids\fusion\chousel_eighth_t_half_fuseavg.avi','f
ps',30)

```



12 ADDENDUM C

HOMOMORPHIC FILTERING AND POWERS SPECTRUM RESTORATION BASED ALGORITHM CODE

12.1 'RATIOAVG' FUNCTION

```
function outImage = ratioAvg(image1, image2, u)
%function averages two inputted images using the ratio coefficient
  such
outImage = (u).*image1 + (1-u).*image2
%
%matlab code by:
%
%           Bryn Walters
%           University Of Johannesburg
%           as part of Honours year requirements for
%           Baccalaureus Ingeneria Degree
%           in
%           Electrical and Electronic Engineering

outImage = (u).*image1 + (1-u).*image2;
```

12.2 'CHOOSEHIGHFREQIMG' FUNCTION

```
function [outImage,num] = chooseHighFreqImg(images,imgx,imgy,
  prevHF, prevnum)
%this function chooses an image with the highest amount of high
  frequency
%from the image set 'images'.
%INPUTS: images:  image set containing 8 images
%         prevHF:  previous high frequency image found
%         prevnum: prevHF's number in previous set of images
%
%matlab code by:
%
%           Bryn Walters
%           University Of Johannesburg
%           as part of Honours year requirements for
%           Baccalaureus Ingeneria Degree
%           in
%           Electrical and Electronic Engineering

%initialization
```



```
look = 8;
HFnum = 0;
HFimg = images(1:imgx,1:imgy);

if (prevnum < 4) %first three images of old img set will not have
    been forgotton
    look = 5; %compare first five to old
    HFimg = prevHF;
    HFnum = hFreqNum(HFimg);
    %eg. first image previously was HFimg therefore on second step
    with five
    %    new images we only need to check first five
end

for n = 1:look, %assuming images was correctly sent as length of 8

    %deconcatenate current image
    current = images(1:imgx,imgy*(n-1)+1:imgy*n);

    tempnum = hFreqNum(current);
    if (tempnum > HFnum) %check if higher frequency
        HFimg = current;
        HFnum = tempnum;
    end

end

end

outImage = HFimg;
num = HFnum;
```



12.3 'HFREQNUM' FUNCTION

```
function num = hFreqNum(image)
%function hFreqNum returns the number of high frequency components
    in the
%inputted image using the following convolution mask and
    thresholding at 63
% mask = [0 -1 0;-1 4 -1;0 -1 0]
%
%note the inputted must be a grayscale image
%
%matlab code by:
%
%           Bryn Walters

mask = [0 -1 0;-1 4 -1;0 -1 0];

%padding = padImage(image,2,2); %pad image with a 1 depth border of
    zeros
%[ic,ir] = size(padding); % get size of padded image

%temp = 0;
%totnum = 0;
```



```
%for m = 2:(ic-1), %iterate through original image columns
%   for n = 2:(ir-1), %iterate through original image rows
%       temp = 0;
%       for k = -1:1,
%           for l = -1:1,
%               temp = temp + padding(m+k,n+1)*mask(2+k,2+1);
%           end
%       end
%       if (temp > 63) %threshold
%           totnum = totnum + 1;
%       end
%   end
%end

%% matlab conv2 method which is handled more efficiently
%%enhanced code starts here

convImage = conv2(double(mask),double(image)); %convolute image and
    mask
[imx,imy] = size(convImage);
totnum = 0;
for m = 2:imx-1,%iterate through original image
    for n = 2:imy-1, %iterate through original image
        if (convImage(m,n) > 63)
            totnum = totnum +1;
        end
    end
end
num = totnum;
```



12.4 'PADIMAGE' FUNCTION

```
function outimage = padImage(image, height, width)
%function pads an image with zeros of given padding height and width
%The padding is split on either side of image so the image is
    centred
%image is image to pad
%height is TOTAL height of zeros to add to image
%width is TOTAL width of zeros to add
%eg if height = width = 2
%output image will contain a single border of zeros
%
%matlab code by:
%
%           Bryn Walters
%           University Of Johannesburg
%           as part of Honours year requirements for
%           Baccalaureus Ingeneria Degree
%           in
%           Electrical and Electronic Engineering

[ix,iy] = size(image);
```



```

lp = floor(width/2);    %padding thickness on the left of image
rp = width - lp;       %right
tp = floor(height/2);  %top
bp = height - tp;     %bottom

outimageprev = [zeros(ix,lp) image zeros(ix,rp)];
outimage = [zeros(tp, iy+width); outimageprev; zeros(bp, iy+width)];

```

12.5 'HOMOMORPHICFILT' FUNCTION

```

function outImage = homomorphicFilt(image, num)
%this function performs homomorphic filtering on the inputted image
  where
%num specifies the coefficient set to use in filtering
%
%matlab code by:
%
%           Bryn Walters
%           University Of Johannesburg
%           as part of Honours year requirements for
%           Baccalaureus Ingeneria Degree
%           in
%           Electrical and Electronic Engineering

b1 = [-0.000016 -0.000108 0.9998 -0.000108 -0.000016];%1
b2 = [-0.000032 -0.000216 0.9996 -0.000216 -0.000032];%2
b3 = [-0.000064 -0.000432 0.9992 -0.000432 -0.000064];%4
b4 = [-0.000128 -0.000864 0.9984 -0.000864 -0.000128];%8
b5 = [-0.000256 -0.001728 0.9968 -0.001728 -0.000256];%16
b6 = [-0.000384 -0.002592 0.9952 -0.002592 -0.000384];%24
b7 = [-0.000512 -0.003456 0.9936 -0.003456 -0.000512];%32
b8 = [-0.000768 -0.005183 0.9904 -0.005183 -0.000768];%48
b9 = [-0.001024 -0.006910 0.9872 -0.006910 -0.001024];%64
b10 = [-0.001532 -0.010362 0.9808 -0.010362 -0.001532];%96
b11 = [-0.002039 -0.013809 0.9744 -0.013809 -0.002039];%128
b12 = [-0.003104 -0.021115 0.9608 -0.021115 -0.003104];%196
b13 = [-0.004026 -0.027529 0.9488 -0.027529 -0.004026];%256
b14 = [-0.005908 -0.041071 0.9232 -0.041071 -0.005908];%384
b15 = [-0.007638 -0.054347 0.8976 -0.054347 -0.007638];%512
b16 = [-0.012222 -0.103118 0.7952 -0.103118 -0.012222];%1024

%-----
%-----

switch num
  case 1
    b = b1;
  case 2
    b = b2;
  case 3

```



```
        b = b3;
case 4
        b = b4;
case 5
        b = b5;
case 6
        b = b6;
case 7
        b = b7;
case 8
        b = b8;
case 9
        b = b9;
case 10
        b = b10;
case 11
        b = b11;
case 12
        b = b12;
case 13
        b = b13;
case 14
        b = b14;
case 15
        b = b15;
case 16
        b = b16;
end

im = (double(image))/255; % Rescale values 0-1 (and cast to
'double')
logIm = log(im+0.01);      % first part of homomorphic is natural
log values

H = b*b'; %create 2d filter
tempimage = filter2(H,logIm); %perform 2d filtering through rows and
columns

outImage = exp(tempimage)*255; %undo log with exponential
```

12.6 MAIN FILE (RUN)

```
%main file
%
%matlab code by:
%
%           Bryn Walters
%           University Of Johannesburg
%           as part of Honours year requirements for
%           Baccalaureus Ingeneria Degree
%           in
%           Electrical and Electronic Engineering
```

```
close all;
clear all;
clc;

%% operator parameters

alpha = 1/8; %averaging amount, 1/2 1/4 1/8 1/16 or 1/32
num = 5; %coefficient number set to use for homomorphic filtering(1-
    16)

%% get images

mov = aviread('d:\workvids\chouserof.avi');

[M,N] = size(mov); % determine number of frames

a = 1; %image pixels to focus on; from a to b (rows)
b = 240;
c = 1; %c to d (columns)
d = 320;

dig = mov(1);
truecimg = dig.cdata;
img = truecimg(a:b,c:d,:);
img = rgb2gray(img);
imggray = img;

%% run algorithm

[imgx,imgy] = size(imggray); %get size of video frames

imgavg = imggray; %initial image average set as first image
z = zeros(imgx,imgy);

prevHFreq = z; %previous high frequency frame found in count-1
prevHFNum = 1; %the frame number of the previous high frequency
    frame
update = 5; %max frequency frame selection performed every 5 frames

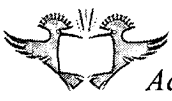
imageSet = [z z z z z z z z]; %image set of 8 images we work with
specHF = z; %spectrum of high frequency selected image

%%%%%%
%N = 150; %override number of frames to process
%%%%%%

tic %start stopwatch to record processing time

for count = 1:N, %iterate frames

    dig = mov(count);
    truecimg = dig.cdata;
    img = truecimg(a:b,c:d,:);
    imggray = rgb2gray(img);
```

```
imgavg = ratioAvg(imggray,imgavg,alpha); %calculate avg at step
count
fftavg = fft2(double(imgavg)); %fft of avg
specavg = fftavg.*conj(fftavg); %power spectrum of avg

%shift old image set and add current image
imageSet = [imggray imageSet(1:imgx,1+imgy:imgy*8)];

if (update == 5) %must look for new HF image
    [prevHFFreq,prevHFNum] =
    chooseHighFreqImg(imageSet,imgx,imgy,prevHFFreq,prevHFNum);

    fftHF = fft2(double(homomorphicFilt(prevHFFreq, num)));%fft
of HF image
    specHF = fftHF.*conj(fftHF); %spectrum of HF image
    update = 0; %reset update counter
end .

update = update +1; %increment counter till update
count
specFilt = sqrt(specHF./specavg); %power spectrum filter
fftOut = fftavg.*specFilt; %filter output of FFT average with PS
filter

ifftOut = real(ifft2(fftOut));
imageOut = ifftOut;

imshow(uint8(imageOut));% show image
F(count) = getframe;

end

toc %stop stopwatch
movie2avi(F,'d:\workvids\thales\chouserroof_eighth_5.avi','fps',30)
%working with file
```

13 ADDENDUM D

WIENER FILTERING USING KURTOSIS MINIMIZATION ALGORITHM CODE

13.1 'RGB2INTENSITY' FUNCTION

```
function intensityImage = rgb2intensity(rgbImage)
% takes a rgb colour image and returns an intensity image comprised
% of the maximum intensities of the r g and b channels

[rw,cl,clSpace] = size(rgbImage);
tempI = zeros(rw,cl); %precreate blank image

tempI = max(rgbImage,[],3);

intensityImage = tempI;
```

13.2 'RATIOAVG' FUNCTION

```
function outImage = ratioAvg(image1, image2, u)
%function averages two inputted images using the ratio coefficient
such
%outImage = (u).*image1 + (1-u).*image2
%
%matlab code by:
%
%           Bryn Walters
%           University Of Johannesburg

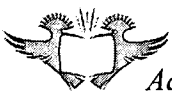
outImage = (u).*image1 + (1-u).*image2;
```

13.3 'ATMOSOTF' FUNCTION

```
function otf = atmosOtf(sizeR,sizeC,lamda)

tf = zeros(sizeR,sizeC);

for k = -sizeR/2:sizeR/2-1
```



```
for l = -sizeC/2:sizeC/2-1
    tf(k+sizeR/2+1,l+sizeC/2+1) = exp(-
        lamda.*((k)^2+(l)^2)^(5/6));
end
end

otf = tf;
```

13.4 'WIENEROPT' FUNCTION

```
function imgOut = WienerOpt(img, nmax, nmin, num, snr)
%returns best Wiener filtered image image using constraints provided
and kurtosis measurements

lamda = linspace(nmin,nmax,num);

[imgr,imgc] = size(img);

IMG = fft2(img);

OTF = ifftshift(atmosOtf(imgr,imgc,lamda(1)));

newimg = real(ifft2(IMG.*conj(OTF)./((abs(OTF)).^2 + 1/snr)));
k = kurtosis(newimg(:));

biter = 1;
bestimg = newimg;
minK = k;

for iter = 2:num,
    OTF = ifftshift(atmosOtf(imgr,imgc,lamda(iter)));

    newimg = real(ifft2(IMG.*conj(OTF)./((abs(OTF)).^2 + 1/snr)));
    k = kurtosis(newimg(:));
    if k < minK
        bestimg = newimg;
        minK = k;
        biter = iter;
    end
end

imgOut = bestimg;
```

13.5 MAIN FILE (RUN)

```
%close all;
clear all;
clc;

%% operator parameters
```



```
alpha = 1/3; %averaging amount, 1/2 1/4 1/8 1/16 or 1/32

%initial search parameters
lamdamin = 0.0001;
lamdamax = 0.001;
num = 50;

%% get images

directoryToOpen = ('D:\workvids2\noiseRem\');
fileToOpen = ('Mid freq no filt0001-0');
detail = '';
mov = aviread(strcat(directoryToOpen,fileToOpen,'.avi'),1:100);
fps = 20;

[M,N] = size(mov); % determine number of frames

a = 1; %image pixels to focus on; from a to b (rows)
b = 480;
c = 1; %c to d (columns)
d = 640;

dig = mov(1);
truecimg = dig.cdata;
imgcolor = truecimg(a:b,c:d,:);
img = double(rgb2gray(imgcolor));

[imgr,imgc] = size(img);
update = 10;

N = 100;
tic
for count = 1:N, %iterate frames

    dig = mov(count);
    truecimg = dig.cdata;
    imgcolor = truecimg(a:b,c:d,:);
    imggray = double(rgb2gray(imgcolor));

    img = ratioAvg(imggray,img,alpha);

    if (update == 10)
        %%snr est
        % estimate noise variance
        wsize = 5;
        numdiv = floor((imgr)/wsize)*floor((imgc)/wsize);
        nvar = 0;

        for m = wsize:wsize:imgr
            for n = wsize:wsize:imgc
                nvar = nvar + ((std2(img(m-wsize+1:m,n-
wsize+1:n)))^2)/numdiv;
            end
        end
    end
end
```

```
snr = ((std2(img))^2-nvar)/nvar;

update = 0;
end
update = update + 1;

%%filter using Wiener

[filteredImg,lamdaO] = WienerOpt(img,lamdamin,lamdamax,num,
snr);

%search optimization
lamdamin = lamdaO - 0.00008;
lamdamax = lamdaO + 0.00008;
num = 10;

lamdaO

imgEnhanced =
imadjust(uint8(filteredImg),stretchlim(uint8(filteredImg),[0
1]),[]);
% figure
imshow((imgEnhanced))
F(count) = getframe;
end
toc

nameOfFile = strcat('d:\workvids2\Wiener filtering average
laplacian\_',fileToOpen,'_',int2str(1/alpha),detail);
movie2avi(F,nameOfFile,'fps',fps)
```

14 ADDENDUM E

WIENER FILTERING USING LAPLACIAN OPERATOR ALGORITHM CODE

14.1 'RGB2INTENSITY' FUNCTION

```
function intensityImage = rgb2intensity(rgbImage)
% takes a rgb colour image and returns an intensity image comprised
% of the maximum intensities of the r g and b channels

[rw,cl,clSpace] = size(rgbImage);
tempI = zeros(rw,cl); %precreate blank image

tempI = max(rgbImage, [], 3);

intensityImage = tempI;
```

14.2 'RATIOAVG' FUNCTION

```
function outImage = ratioAvg(image1, image2, u)
%function averages two inputted images using the ratio coefficient
such
%outImage = (u).*image1 + (1-u).*image2
%
%matlab code by:
%
%           Bryn Walters
%           University Of Johannesburg

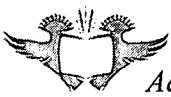
outImage = (u).*image1 + (1-u).*image2;
```

14.3 'ATMOSOTF' FUNCTION

```
function otf = atmosOtf(sizeR,sizeC,lamda)

tf = zeros(sizeR,sizeC);

for k = -sizeR/2:sizeR/2-1
```



```

for l = -sizeC/2:sizeC/2-1
    tf(k+sizeR/2+1,l+sizeC/2+1) = exp(-
        lamda.*((k)^2+(l)^2)^(5/6));
end
end

otf = tf;

```

14.4 'CUM LAP' FUNCTION

```

function cumLapNum = cumLap(I)
%perform Laplacian based sharpness measurement

[r,c] = size(I);

%option 1
H = fspecial('laplacian',0.2);
LI = imfilter(I,H,'replicate');

%option 2 (discrete laplacian)
%LI = 4*del2(I);

cumLapNum = sum(abs(LI(:)))/(r*c);

```

14.5 'WIENER OPT' FUNCTION

```

function [imgOut,lamdaout] = WienerOpt(img, nmax, nmin, num, snr)

lamda = linspace(nmin,nmax,num);

[imgr,imgc] = size(img);

IMG = fft2(img);

OTF = ifftshift(atmosOtf(imgr,imgc,lamda(1)));

newimg = real(ifft2(IMG.*conj(OTF)./((abs(OTF)).^2 + 1/snr)));

k = cumLap(newimg);

biter = 1;
bestimg = newimg;
maxK = k;

for iter = 2:num,
    OTF = ifftshift(atmosOtf(imgr,imgc,lamda(iter)));

    newimg = real(ifft2(IMG.*conj(OTF)./((abs(OTF)).^2 + 1/snr)));

    k = cumLap(newimg);

```



```
    if k > maxK
        bestimg = newimg;
        maxK = k;
        biter = iter;
    end
end

imgOut = bestimg;

lamdaout = lamda(biter);
```

14.6 MAIN FILE (RUN)

```
%close all;
clear all;
clc;

%% operator parameters

alpha = 1/3; %averaging amount, 1/2 1/4 1/8 1/16 or 1/32

%initial search parameters
lamdamin = 0.0001;
lamdamax = 0.001;
num = 50;

%% get images

directoryToOpen = ('D:\workvids2\noiseRem\');
fileToOpen = ('Mid freq no filt0001-0');
detail = '';
mov = aviread(strcat(directoryToOpen,fileToOpen, '.avi'),1:100);
fps = 20;

[M,N] = size(mov); % determine number of frames

a = 1; %image pixels to focus on; from a to b (rows)
b = 480;
c = 1; %c to d (columns)
d = 640;

dig = mov(1);
truecimg = dig.cdata;
imgcolor = truecimg(a:b,c:d,:);
img = double(rgb2gray(imgcolor));

[imgr,imgc] = size(img);
update = 10;

N = 100;
tic
```




```
for count = 1:N, %iterate frames

    dig = mov(count);
    truecimg = dig.cdata;
    imgcolor = truecimg(a:b,c:d,:);
    imggray = double(rgb2gray(imgcolor));

    img = ratioAvg(imggray,img,alpha);

    if (update == 10)
        %%snr est
        % estimate noise variance
        wsize = 5;
        numdiv = floor((imgr)/wsize)*floor((imgc)/wsize);
        nvar = 0;

        for m = wsize:wsize:imgr
            for n = wsize:wsize:imgc
                nvar = nvar + ((std2(img(m-wsize+1:m,n-
wsize+1:n)))^2)/numdiv;
            end
        end

        snr = ((std2(img))^2-nvar)/nvar;

        update = 0;
    end
    update = update + 1;

    %%filter using Wiener

    [filteredImg,lamda0] = WienerOpt(img,lamdamin,lamdamax,num,
snr);

    %search optimization
    lamdamin = lamda0 - 0.00008;
    lamdamax = lamda0 + 0.00008;
    num = 10;

    lamda0

    imgEnhanced =
    imadjust(uint8(filteredImg),stretchlim(uint8(filteredImg),[0
1]),[]);
    % figure
    imshow((imgEnhanced))
    F(count) = getframe;
end
toc

nameOfFile = strcat('d:\workvids2\Wiener filtering average
laplacian\',fileToOpen,'_',int2str(1/alpha),detail);
movie2avi(F,nameOfFile,'fps',fps)
```



15 ADDENDUM F

DYNAMIC ILLUMINANCE-REFLECTANCE ATMOSPHERIC TURBULENCE SUPPRESSION ALGORITHM CODE

15.1 'RGB2INTENSITY' FUNCTION

```
function intensityImage = rgb2intensity(rgbImage)
% takes a rgb colour image and returns an intensity image comprised
% of the maximum intensities of the r g and b channels
```

```
[rw,cl,clSpace] = size(rgbImage);
tempI = zeros(rw,cl); %precreate blank image
tempI = max(rgbImage,[],3);
```

```
intensityImage = tempI;
```



15.2 'RATIOAVG' FUNCTION

```
function outImage = ratioAvg(image1, image2, u)
%function averages two inputted images using the ratio coefficient
such
```

```
%outImage = (u).*image1 + (1-u).*image2
```

```
%
```

```
%matlab code by:
```

```
%
```

```
%
```

```
%
```

Bryn Walters
University Of Johannesburg

```
outImage = (u).*image1 + (1-u).*image2;
```

15.3 'WIS' FUNCTION

```
function f = WIS(v,a)
%Windowed Inverse Sigmoid
```

```
f = 1/(1+exp(a*v));
```



15.4 MAIN FILE (RUN)

```
%illuminance reflectance based higher frequency enhancement method
clc;
close all;
clear all;

directoryToOpen = ('D:\workvids2\noiseRem\');
fileToOpen = ('Mid freq no filt0001-0');
fps=20;
mov = aviread(strcat(directoryToOpen,fileToOpen, '.avi'),1:100);

[M,N] = size(mov); % determine number of frames

a = 1; %image pixels to focus on; from a to b (rows)
b = 480;
c = 1; %c to d (columns)
d = 640;

framenum = 50;
offsetframe = 1;

dig = mov(offsetframe);
truecing = dig.cdata;
I = truecing(a:b,c:d,:);

avgRatio = 1/3;

updatetime = 4; %when to calculate new vmin and p values (1 is every
    frame)
update = updatetime; %initialize to get new values on first frame

gsize =4;
sigma = gsize/sqrt(2);
hsize = [gsize gsize];
h = fspecial('gaussian',hsize,sigma); %create gaussian filter
h2 = fspecial('gaussian',[10 10],10/sqrt(2)); %create gaussian
    filter

%init
IgMean = 0;
a      = 1;
vMax   = 3;
vMin   = 0;
sDev   = 0;
p      = 0;

tic
for count = 1:framenum,

    dig = mov(count+offsetframe);
    truecing = dig.cdata;
    I = ratioAvg(truecing(a:b,c:d,:), I, avgRatio);
```

```
[rw,cl,rgbspace] = size(I);

%Ig = rgb2gray(I); %create grayscale image (Hue and saturation
set to 0 in
%NTSC colour space

% alternate method to NTSC colour space transition
Ig = double(rgb2intensity(I)); %transfer to maximum intensity
values

Lum = filter2(h,Ig); %find luminance estimate
LumN = Lum./255;

Ref = Ig./Lum;%exp(log(Ig)-log(Lum)); %find Reflectance estimate

if (update == updatetime) %time to refresh values and
recalculate vMin and p

    IgMean = mean2(Ig); %calculate mean of intensity image

    if (IgMean <= 70) %dynamically find vMin based on Ig
        vMin = -6;
    elseif (IgMean >= 150)
        vMin = -3;
    else
        vMin = (IgMean-70)/80*3-6;
    end
    sDev = std2(Ig);

    if (sDev<=30) %dynamically find p based on standard
deviation
        p = 2;
    elseif (sDev > 80)
        p = 0.5;
    else
        p = -0.03*sDev + 2.9;
    end

    update = 0;
end
update = update + 1;

LumN1 = (LumN).*(WIS(vMax,a)-WIS(vMin,a))+WIS(vMin,a);
LumN2 = (1/a).*log(1./(LumN1) - 1); %inverse sigmoid function
LumNenh =((LumN2 - vMin)./(vMax - vMin)); %calculate adjusted
illumiance with normalisation

%midtone enhancement
midExp = (filter2(h2,Ig)./(Ig+1)).^p;
LumNenh2 = LumNenh.^midExp;
```



```
Ig2 = uint8(LumNenh2.*255.*Ref);

Ig2 = imadjust(uint8(Ig2),stretchlim(uint8(Ig2),[0.005
0.995]),[]);

imshow(uint8(Ig2))
F(count) = getframe;
end

toc

nameOfFile = strcat('d:\workvids2\Image average, Dynamic
illuminance_reflectance adjustment, and sharpening (video,
colour)\',fileToOpen,'_',int2str(1/avgRatio),'th');
movie2avi(F,nameOfFile,'fps',fps)
```



16 ADDENDUM G

DYNAMIC ILLUMINANCE-REFLECTANCE ATMOSPHERIC TURBULENCE SUPPRESSION WITH WIENER FILTERING ALGORITHM CODE

16.1 'RGB2INTENSITY' FUNCTION

```
function intensityImage = rgb2intensity(rgbImage)
% takes a rgb colour image and returns an intensity image comprised
% of the maximum intensities of the r g and b channels

[rw,cl,clSpace] = size(rgbImage);
tempI = zeros(rw,cl); %precreate blank image

tempI = max(rgbImage,[],3);
intensityImage = tempI;
```

16.2 'RATIOAVG' FUNCTION

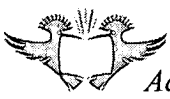
```
function outImage = ratioAvg(image1, image2, u)
%function averages two inputted images using the ratio coefficient
such
%outImage = (u).*image1 + (1-u).*image2
%
%matlab code by:
%
%           Bryn Walters
%           University Of Johannesburg

outImage = (u).*image1 + (1-u).*image2;
```

16.3 'ATMOSOTF' FUNCTION

```
function otf = atmosOtf(sizeR,sizeC,lamda)

tf = zeros(sizeR,sizeC);
```



```
for k = -sizeR/2:sizeR/2-1
    for l = -sizeC/2:sizeC/2-1
        tf(k+sizeR/2+1,l+sizeC/2+1) = exp(-
            lamda.*((k)^2+(l)^2)^(5/6));
    end
end
end

otf = tf;
```

16.4 'CUMLAP' FUNCTION

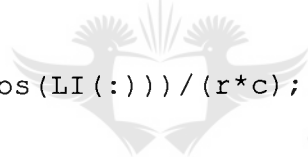
```
function cumLapNum = cumLap(I)
%perform Laplacian based sharpness measurement

[r,c] = size(I);

%option 1
H = fspecial('laplacian',0.2);
LI = imfilter(I,H,'replicate');

%option 2 (discrete laplacian)
%LI = 4*del2(I);

cumLapNum = sum(abs(LI(:)))/(r*c);
```



16.5 'WIENEROPT' FUNCTION

```
function [imgOut,lamdaout] = WienerOpt(img, nmax, nmin, num, snr)

lamda = linspace(nmin,nmax,num);

[imgr,imgc] = size(img);

IMG = fft2(img);

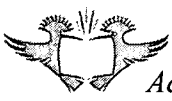
OTF = ifftshift(atmosOtf(imgr,imgc,lamda(1)));

newimg = real(ifft2(IMG.*conj(OTF)./((abs(OTF)).^2 + 1/snr)));

k = cumLap(newimg);

biter = 1;
bestimg = newimg;
maxK = k;

for iter = 2:num,
    OTF = ifftshift(atmosOtf(imgr,imgc,lamda(iter)));
```



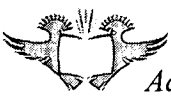
```
newimg = real(ifft2(IMG.*conj(OTF)./((abs(OTF)).^2 + 1/snr)));  
  
k = cumLap(newimg);  
  
if k > maxK  
    bestimg = newimg;  
    maxK = k;  
    biter = iter;  
end  
end  
  
imgOut = bestimg;  
  
lamdaout = lamda(biter);
```

16.6 'WIS' FUNCTION

```
function f = WIS(v,a)  
%Windowed Inverse Sigmoid  
  
f = 1/(1+exp(a*v));
```

16.7 MAIN FILE (RUN)

```
%close all;  
clear all;  
clc;  
  
%% operator parameters  
  
alpha = 1/3; %averaging amount, 1/2 1/4 1/8 1/16 or 1/32  
  
%initial search parameters  
lamdamin = 0.0001;  
lamdamax = 0.001;  
num = 10;  
  
%% get images  
  
directoryToOpen = ('D:\workvids2\noiseRem\');  
fileToOpen = ('Mid freq no filt0001-0');  
  
detail = 'snr update 2 frames';  
mov = aviread(strcat(directoryToOpen,fileToOpen, '.avi'),1:100);  
fps=20;  
  
[M,N] = size(mov); % determine number of frames  
  
a = 1; %image pixels to focus on; from a to b (rows)  
b = 480;
```

```
c = 1; %c to d (columns)
d = 640;

dig = mov(1);
truecimg = dig.cdata;
imgcolor = truecimg(a:b,c:d,:);
img1 = double(rgb2gray(imgcolor));

[imgr,imgc] = size(img1);
updatesnr = 2; %when to update snr est
update = updatesnr;
snr = 0;

N = 100;

%%illumiance reflectance setup
updatetime = 4; %when to calculate new vmin and p values (1 is every
    frame)
update2 = updatetime; %initialize to get new values on first frame

gsize =4;
sigma = gsize/sqrt(2);
hsize = [gsize gsize];
h = fspecial('gaussian',hsize,sigma); %create gaussian filter
h2 = fspecial('gaussian',[10 10],10/sqrt(2)); %create gaussian
    filter

%init
Ig = zeros(imgr,imgc);
IgMean = 0;
a      = 1;
vMax   = 3;
vMin   = 0;
sDev   = 0;
p      = 0;
%%end of setup

tic
for count = 1:N, %iterate frames

    dig = mov(count);
    truecimg = dig.cdata;
    imgcolor = truecimg(a:b,c:d,:);
    imggray = double(rgb2intensity(imgcolor)); %transfer to maximum
        intensity values

    img1 = ratioAvg(imggray,img1,alpha);

    if (update == updatesnr)
        %%snr est
        % estimate noise variance
        wsize = 5;
        numdiv = floor((imgr)/wsize)*floor((imgc)/wsize);
        nvar = 0;
```



```
for m = wsize:wsize:imgr
    for n = wsize:wsize:imgc
        nvar = nvar + ((std2(img1(m-wsize+1:m,n-
wsize+1:n)))^2)/numdiv;
    end
end

snr = ((std2(img1))^2-nvar)/nvar;

update = 0;
end
update = update + 1;

%%filter using Wiener

[filteredImg,lamda0] = WienerOpt(img1,lamdamin,lamdamax,num,
snr);

%search optimization
lamdamin = lamda0 - 0.00008;
lamdamax = lamda0 + 0.00008;
num = 10;

lamda0

Ig =
double(imadjust(uint8(filteredImg),stretchlim(uint8(filteredImg)
,[0 1]),[]));

%%illumination adjustment section
%%~~~~~

Lum = filter2(h,Ig); %find luminance estimate
LumN = Lum./255;

Ref = Ig./Lum;%exp(log(Ig)-log(Lum)); %find Reflectance estimate

if (update2 == updatetime) %time to refresh values and
recalculate vMin and p

    IgMean = mean2(Ig); %calculate mean of intensity image

    if (IgMean <= 70) %dynamically find vMin based on Ig

        vMin = -6;
    elseif (IgMean >= 150)

        vMin = -3;
    else

        vMin = (IgMean-70)/80*3-6;
    end

    sDev = std2(Ig);
```

```
if (sDev<=30) %dynamically find p based on standard
deviation
    p = 2;
elseif (sDev > 80)
    p = 0.5;
else
    p = -0.03*sDev + 2.9;
end

update2 = 0;
end
update2 = update2 + 1;

LumN1 = (LumN).*(WIS(vMax,a)-WIS(vMin,a))+WIS(vMin,a);
LumN2 = (1/a).*log(1./(LumN1) - 1); %inverse sigmoid function
LumNenh = ((LumN2 - vMin)./(vMax - vMin)); %calculate adjusted
illumiance with normalisation

%midtone enhancement
midExp = (filter2(h2,Ig)./(Ig+1)).^p;
LumNenh2 = LumNenh.^midExp;

Ig2 = uint8(LumNenh2.*255.*Ref);

Ig2 = imadjust(uint8(Ig2),stretchlim(uint8(Ig2),[0.005
0.995]),[]);

imshow(uint8(Ig2))
F(count) = getframe;
end
toc

nameOfFile = strcat('d:\workvids2\Wiener, Laplacian, Illuminance
reflectance\',fileToOpen,'_',int2str(1/alpha),detail);
movie2avi(F,nameOfFile,'fps',fps)
```

17 ADDENDUM H

EXPERIMENT 1: ALGORITHM FPS PERFORMANCE ADDITIONAL GRAPHS AND TABLES

Additional individual graphs and tables for the algorithms' results for Experiment 1: Algorithm FPS Performance.

17.1 IMAGE REGISTRATION AND LAPLACIAN PYRAMID IMAGE FUSION

The performance results of the Image Registration and Laplacian Pyramid Image Fusion algorithm are shown below in the following table. This table was used to construct the graph displayed in the Results section for the Image Registration and Laplacian Pyramid Image Fusion algorithm.




Image size	FPS
160x120	0.05933
320x240	0.01476
480x360	0.00719
640x480	0.00650

Table 17-1: Image Registration and Laplacian Pyramid Image Fusion FPS performance

17.2 HOMOMORPHIC FILTERING AND POWER SPECTRUM BASED METHOD

The performance results of the Homomorphic Filtering and Power Spectrum Restoration based algorithm are shown below in the following table and graph. In the graph, Blue dots represent data points and the green line is a fitted trend line.

Image size	FPS
80x60	21.053
160x120	19.048
240x180	15.129
320x240	10.46
400x300	7.502
480x360	5.540
560x420	4.191
640x480	3.406

Table 17-2: Homomorphic Filtering and Power Spectrum method FPS performance

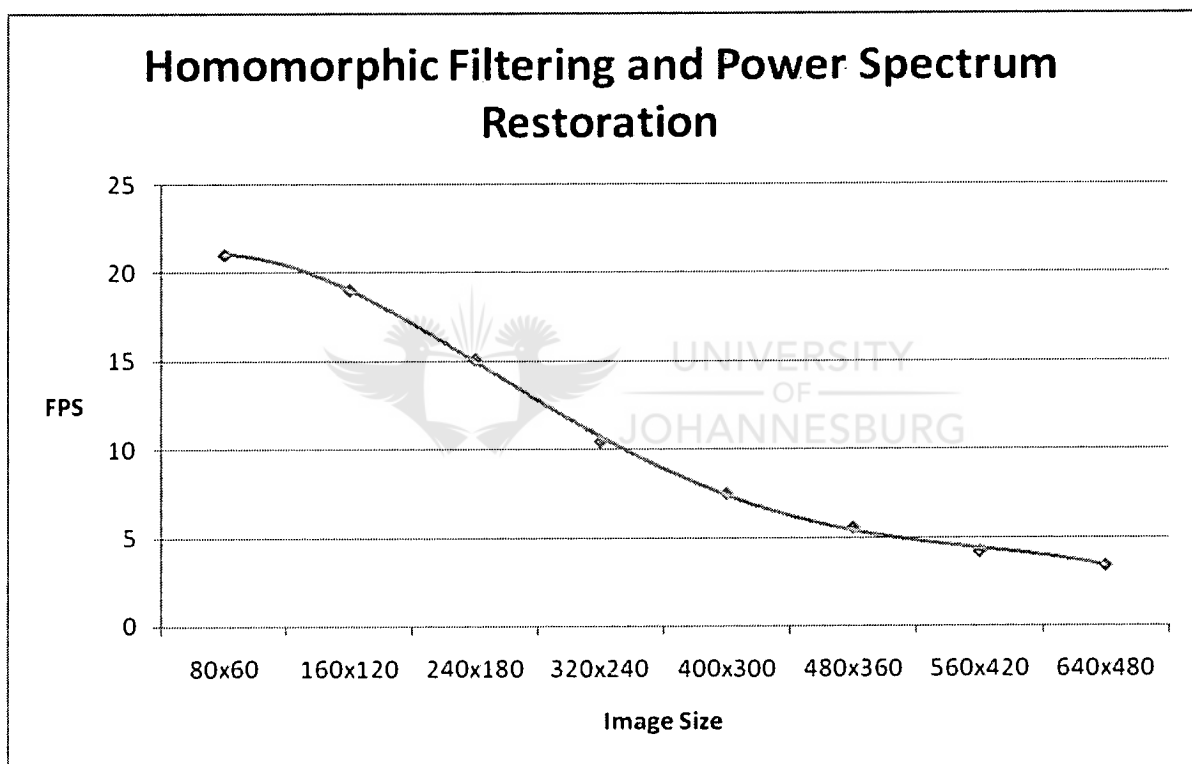


Figure 17-1: Dynamic Homomorphic Filtering and Power Spectrum FPS performance graph

17.3 WIENER FILTERING USING KURTOSIS MINIMIZATION

The following table and graph presents the FPS performance for the Wiener filtering approach method with kurtosis minimization as described in the Algorithm Detailed Design chapter. In the graph, Blue dots represent data points and the green line is a fitted trend line.

Image size	FPS
80x60	7.576
160x120	2.604
240x180	1.323
320x240	0.713
400x300	0.528
480x360	0.434
560x420	0.319
640x480	0.245

Table 17-3: Wiener Filtering using Kurtosis Minimization FPS performance

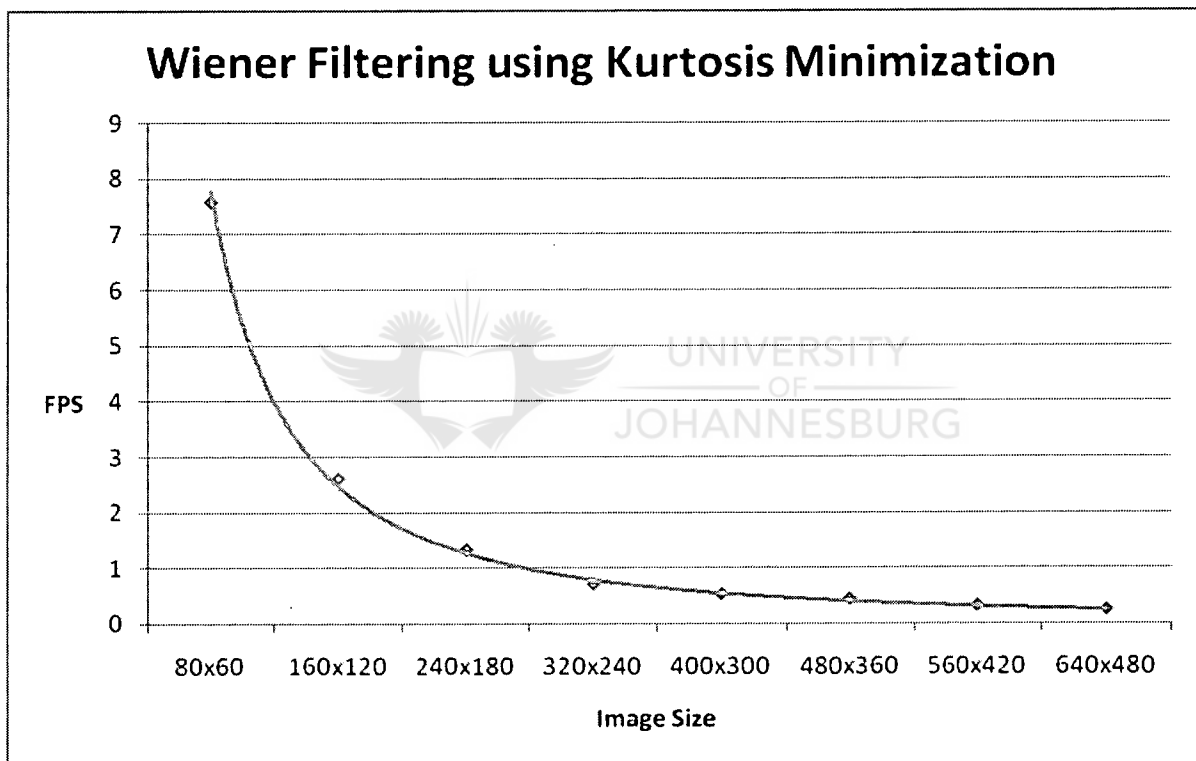


Figure 17-2: Wiener Filtering using Kurtosis Minimization FPS performance graph

17.4 WIENER FILTERING USING LAPLACIAN OPERATOR

The Wiener Filtering using the Laplacian Operator is a method similar to the kurtosis minimization except the Laplacian Operator is used in a metric for selection of the filter shape. The table and graph below show the performance of this method that was created. In the graph, Blue dots represent data points and the green line is a fitted trend line.

Image size	FPS
80x60	11.439
160x120	4.973
240x180	2.588
320x240	1.497
400x300	0.972
480x360	0.682
560x420	0.499
640x480	0.387

Table 17-4: Wiener filtering using Laplacian Operator FPS performance

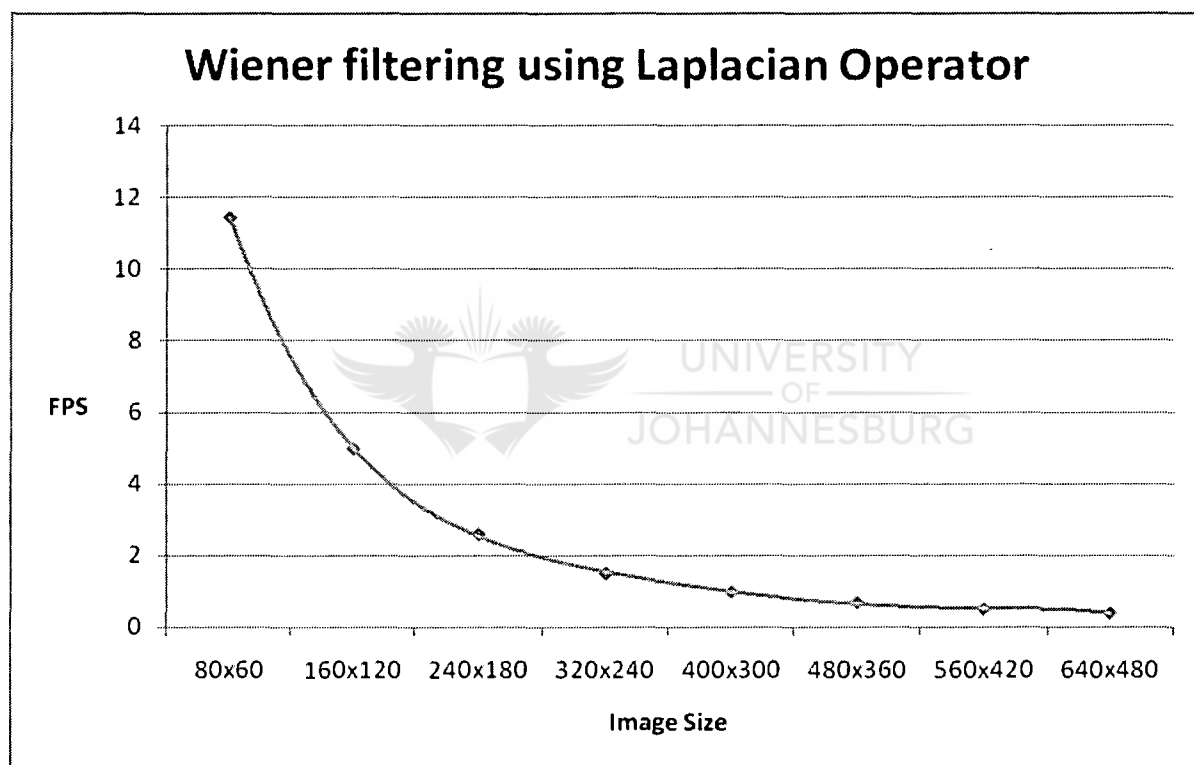


Figure 17-3: Wiener Filtering using Laplacian Operator FPS performance graph

17.5 DYNAMIC ILLUMINANCE-REFLECTANCE ATMOSPHERIC TURBULENCE SUPPRESSION

The performance results of the Illuminance-Reflectance adjustment based algorithm are shown below in the following table and graph. In the graph, Blue dots represent data points and the green line is a fitted trend line.

Image size	FPS
80x60	30.769
160x120	19.573
240x180	13.530
320x240	6.670
400x300	4.469
480x360	3.176
560x420	2.353
640x480	1.792

Table 17-5: Dynamic Illuminance-Reflectance Adjustment FPS performance

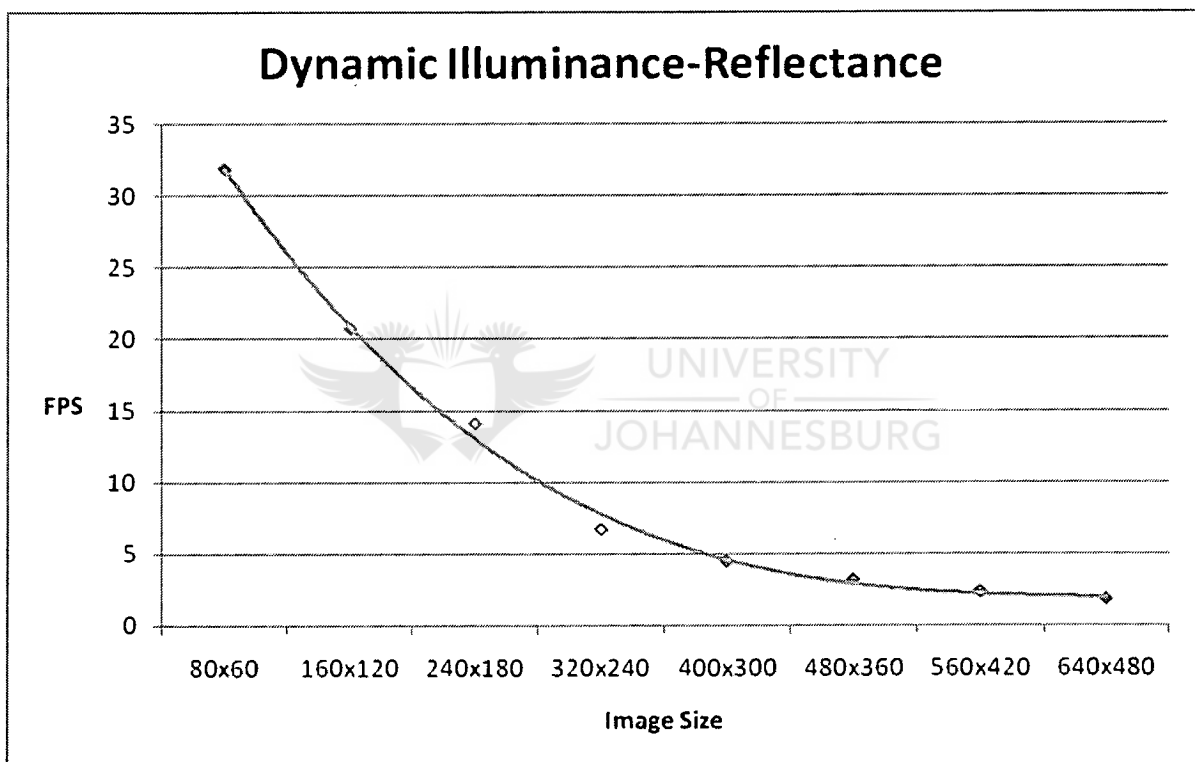


Figure 17-4: Dynamic Illuminance-Reflectance Adjustment FPS performance graph

17.6 DYNAMIC ILLUMINANCE-REFLECTANCE ATMOSPHERIC TURBULENCE SUPPRESSION WITH WIENER FILTERING

This algorithm is similar to the Dynamic Illuminance-Reflectance Atmospheric Turbulence suppression algorithm except it includes the Wiener Filtering Using the Laplacian Operator before Illuminance-Reflectance adjustment is performed. It is an interesting method in relation to just performing the Wiener filtering using Laplacian Operator, as the additional



overhead required from the Illuminance-Reflectance may be visualized on comparison of the two graphs. In the graph, Blue dots represent data points and the green line is a fitted trend line.

Image size	FPS
80x60	10.978
160x120	4.507
240x180	2.342
320x240	1.306
400x300	0.838
480x360	0.592
560x420	0.432
640x480	0.332

Table 17-6: Dynamic Illuminance-Reflectance with Wiener filtering FPS performance

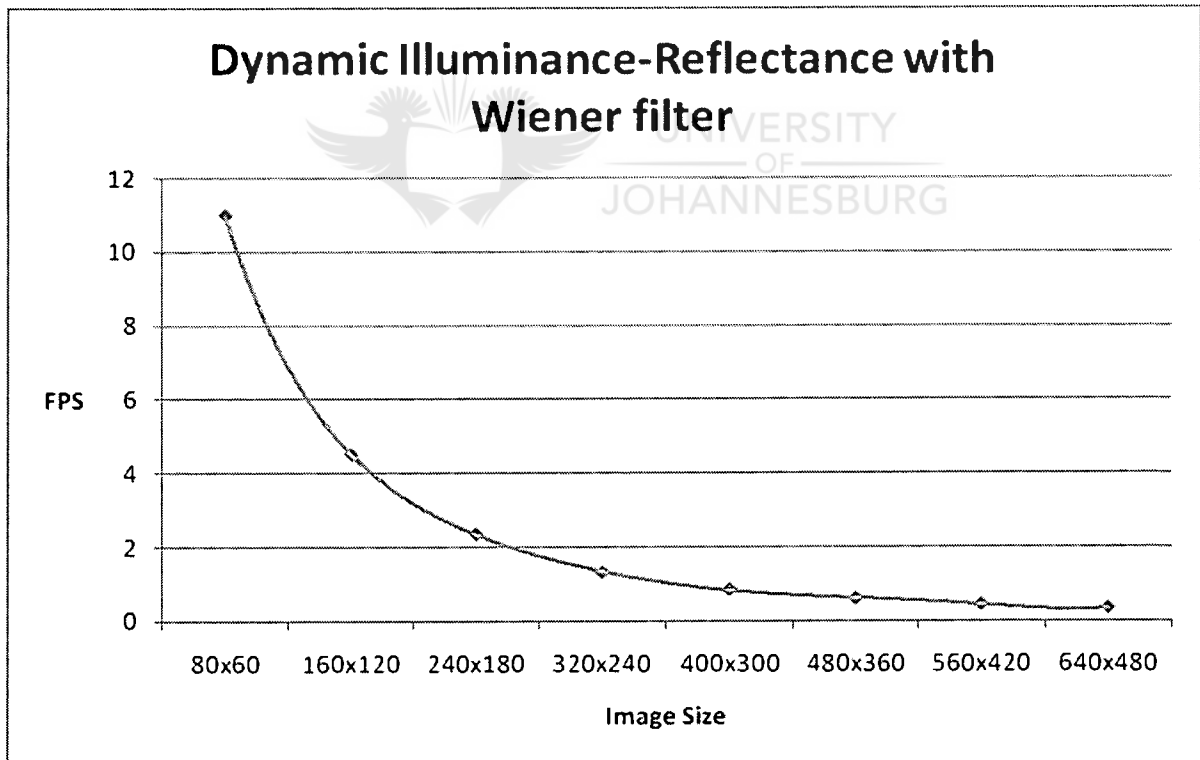


Figure 17-5: Dynamic Illuminance-Reflectance with Wiener filtering FPS performance graph



17.7 SPECKLE MASKING

The Speckle Masking algorithm takes a series of image frames and produces a single image. The results presented for the Speckle Masking are thus as the number of seconds it takes to produce an image.

In the Speckle masking algorithm, sub image size can be selected. This is the size of the overlapping images that the image frames are broken up into. The smaller the sub image size, the smaller the number of artefacts present in the video.

Image Size	Seconds to process sub image size of:		
	128x128 (s)	64x64 (s)	32x32 (s)
128x128	8.406	22.703	34.515
256x256	80.953	125.047	163.875
512x512	384.469	616.093	900.703

Table 17-7: Speckle Masking with sub image size of 128x128, 64x64, and 32x32 performance

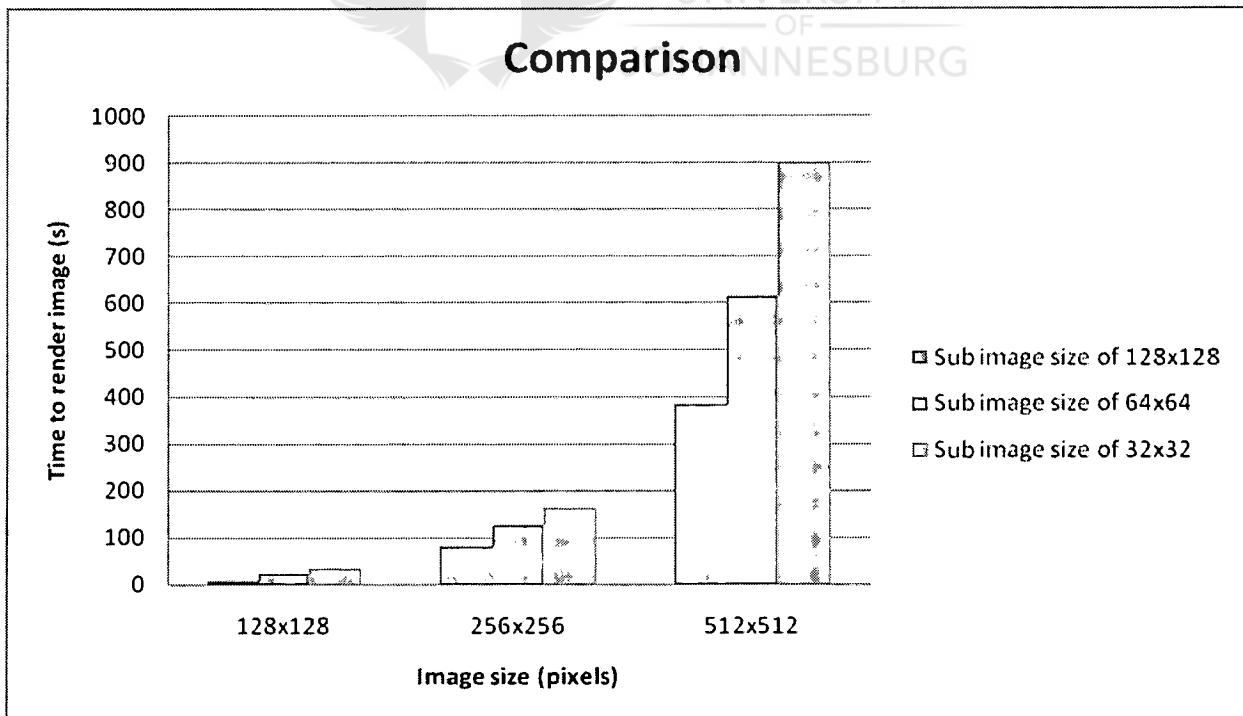


Figure 17-6: Speckle Masking sub image size comparison performance graph

18 ADDENDUM I

EXPERIMENT 2: ALGORITHM FPS PERFORMANCE ADDITIONAL GRAPHS

18.1 UNPROCESSED VIDEO

The following graph shows the sharpness measurement results for the unprocessed video. Each of the algorithms used this video footage and attempt to improve the high spatial frequencies which were suppressed by the atmospheric turbulence, i.e. the increase of image sharpness through high frequency enhancement. It is expected that this graph will have the lowest sharpness values and is used as a comparison to the other graphs to determine improvement in image sharpness. The average image sharpness measurement obtained across the 50 image frames is 5.192.

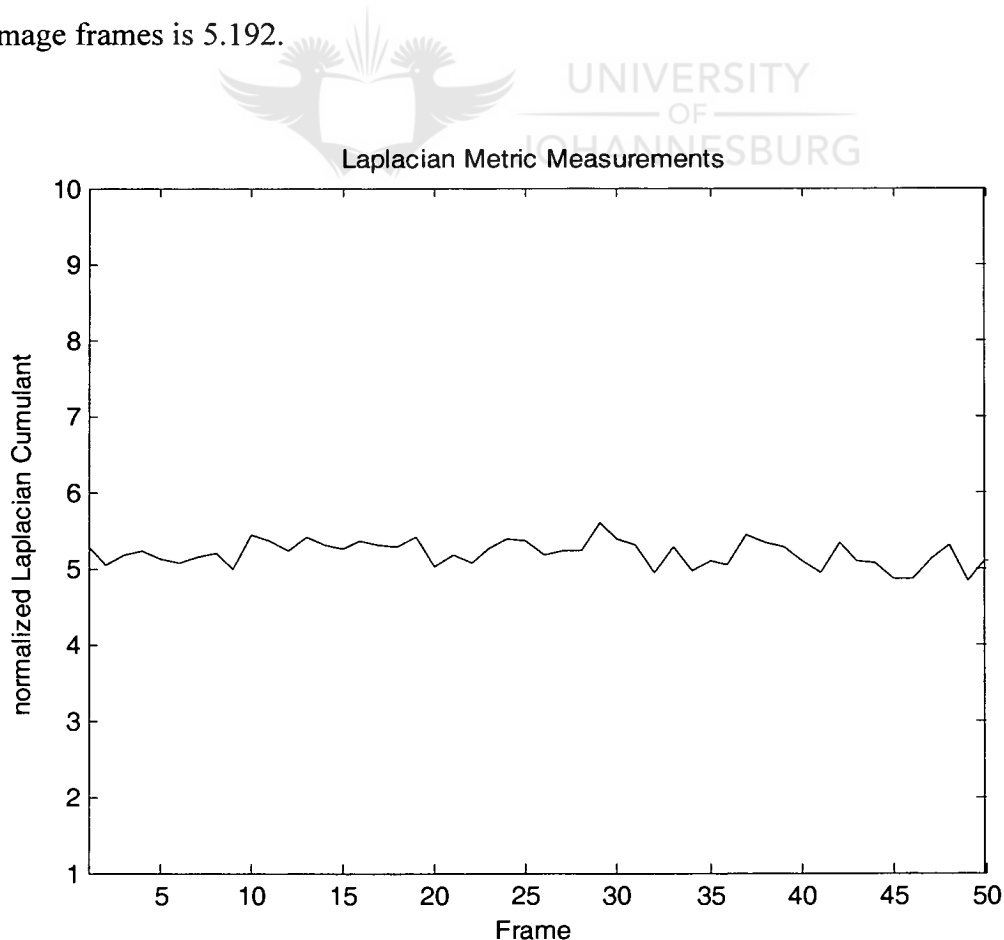


Figure 18-1: Unprocessed video - image sharpness vs. frame number graph.



18.2 IMAGE REGISTRATION AND LAPLACIAN PYRAMID IMAGE FUSION

Image Registration and Laplacian Pyramid image Fusion algorithm's outputted video sharpness results are presented. The frames analysed correspond to those in the unprocessed video. Figure 18-2 presents the results in graphical form.

The average image sharpness measurement obtained across the 50 frames is 11.256.

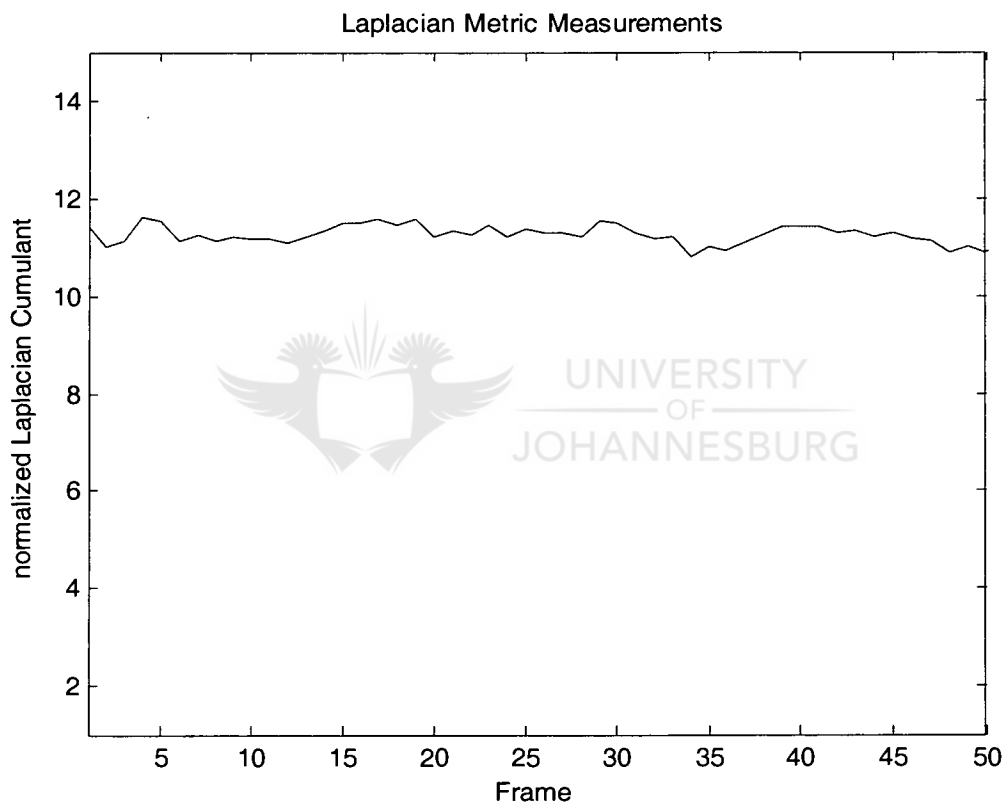


Figure 18-2: Image registration and fusion - image sharpness vs. frame number graph.



18.3 HOMOMORPHIC FILTERING AND POWER SPECTRUM BASED METHOD

The Homomorphic Filtering and Power Spectrum based method's outputted video sharpness results are presented. The frames analysed correspond to those in the unprocessed video. Figure 18-3 presents the results in graphical form.

The average image sharpness measurement obtained across the 50 frames is 8.193.

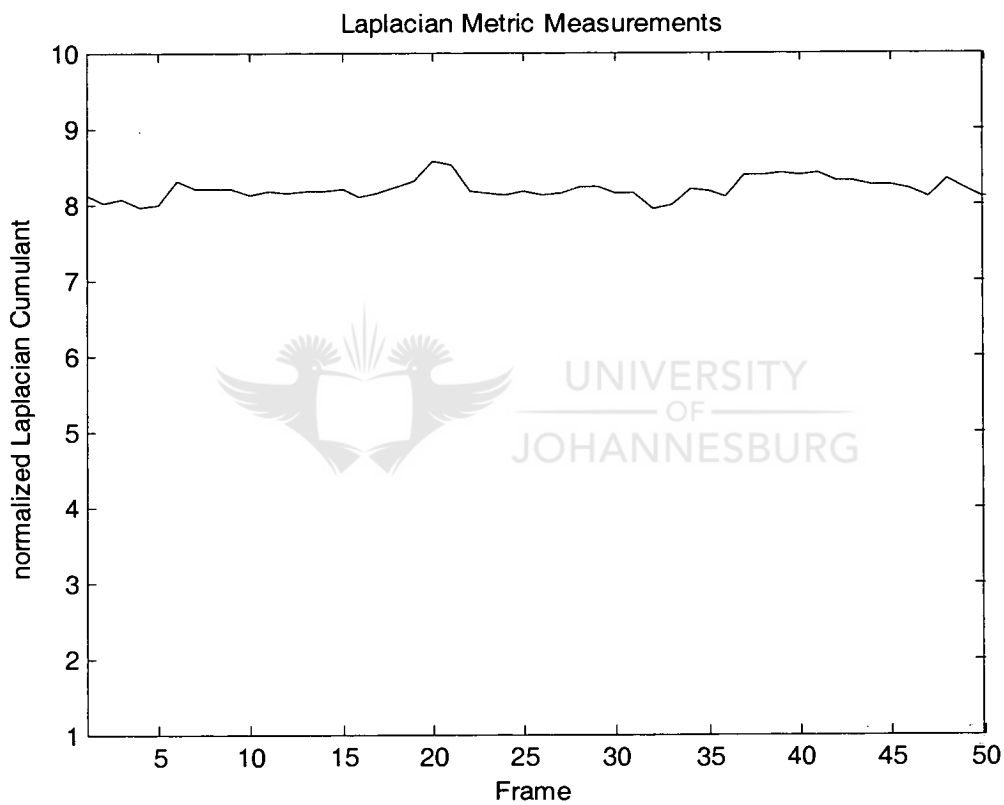


Figure 18-3: Homomorphic filtering and power spectrum method – image sharpness vs. frame number graph.

18.4 WIENER FILTERING USING LAPLACIAN OPERATOR

Wiener filtering using Laplacian Operator algorithm's outputted video sharpness results are presented. The frames analysed correspond to those in the unprocessed video. Figure 18-4 presents the results in graphical form.

The average image sharpness measurement obtained across the 50 frames is 9.166.

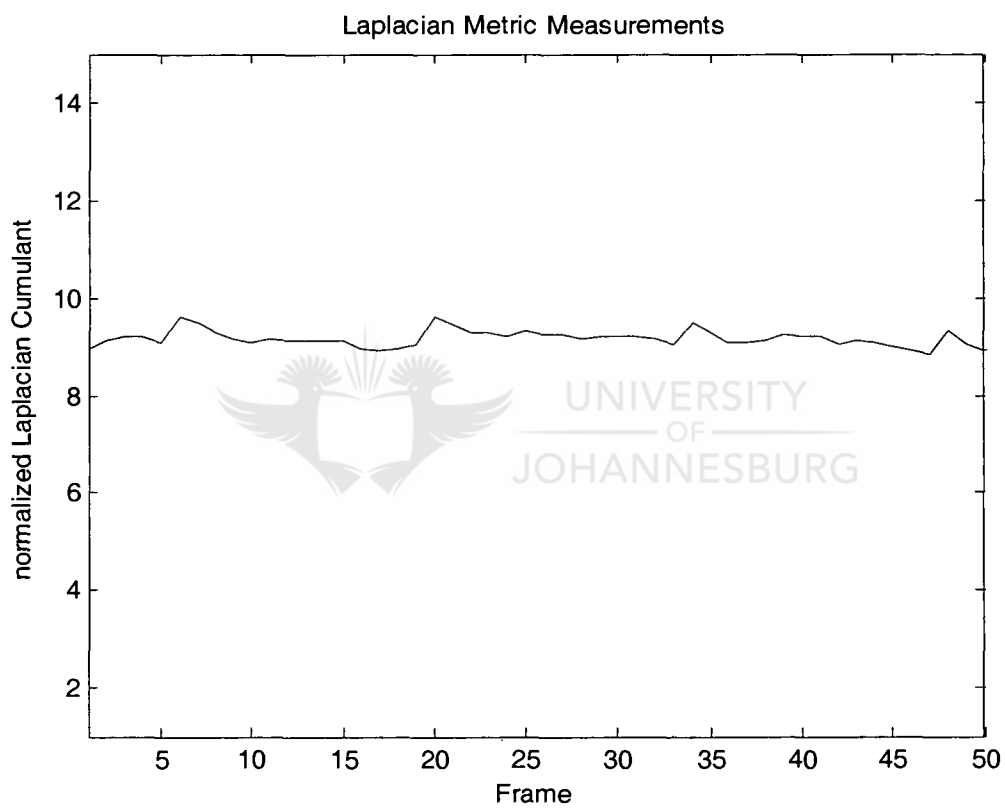


Figure 18-4: Wiener filtering using Laplacian Operator - image sharpness vs. frame number graph.

18.5 DYNAMIC ILLUMINANCE-REFLECTANCE ATMOSPHERIC TURBULENCE SUPPRESSION

Dynamic Illuminance-Reflectance atmospheric turbulence suppression algorithm's outputted video sharpness results are presented. The frames analysed correspond to those in the unprocessed video. Figure 18-5 presents the results in graphical form.

The average image sharpness measurement obtained across the 50 frames is 13.333.

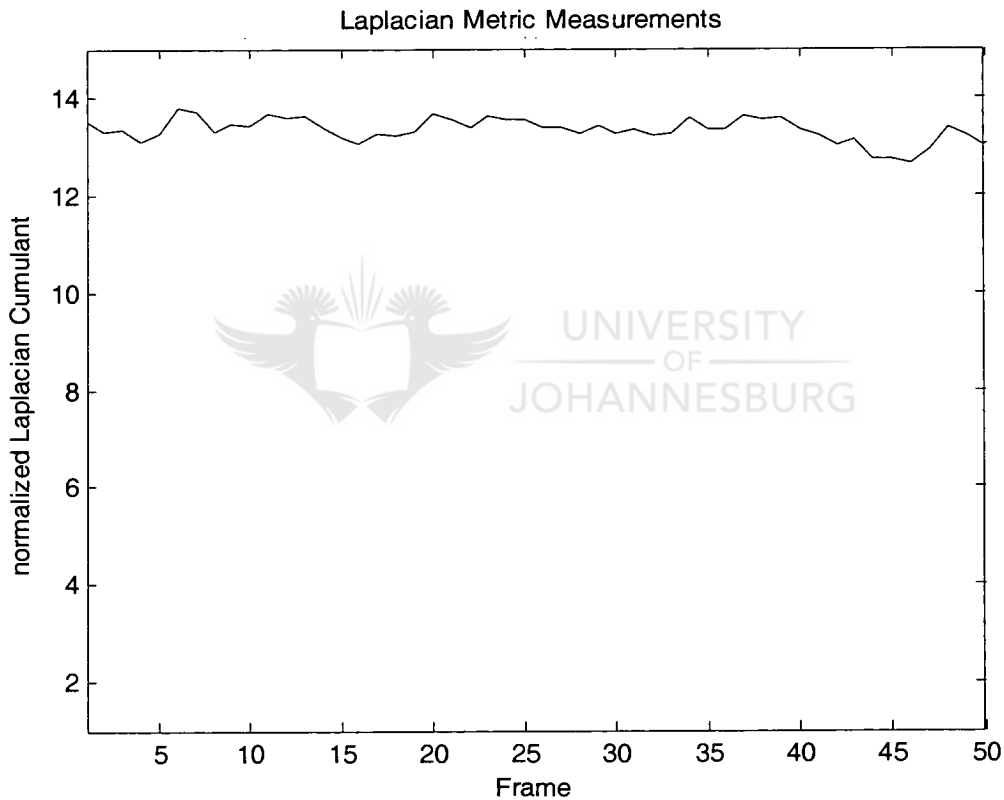


Figure 18-5: Dynamic Illuminance-Reflectance atmospheric turbulence suppression - image sharpness vs. frame number graph.



18.6 DYNAMIC ILLUMINANCE-REFLECTANCE ATMOSPHERIC TURBULENCE SUPPRESSION WITH WIENER FILTERING

Dynamic Illuminance-Reflectance atmospheric turbulence suppression with Wiener filtering algorithm's outputted video sharpness results are presented. The frames analysed correspond to those in the unprocessed video. Figure 18-6 presents the results in graphical form.

The average image sharpness measurement obtained across the 50 frames is 21.726.

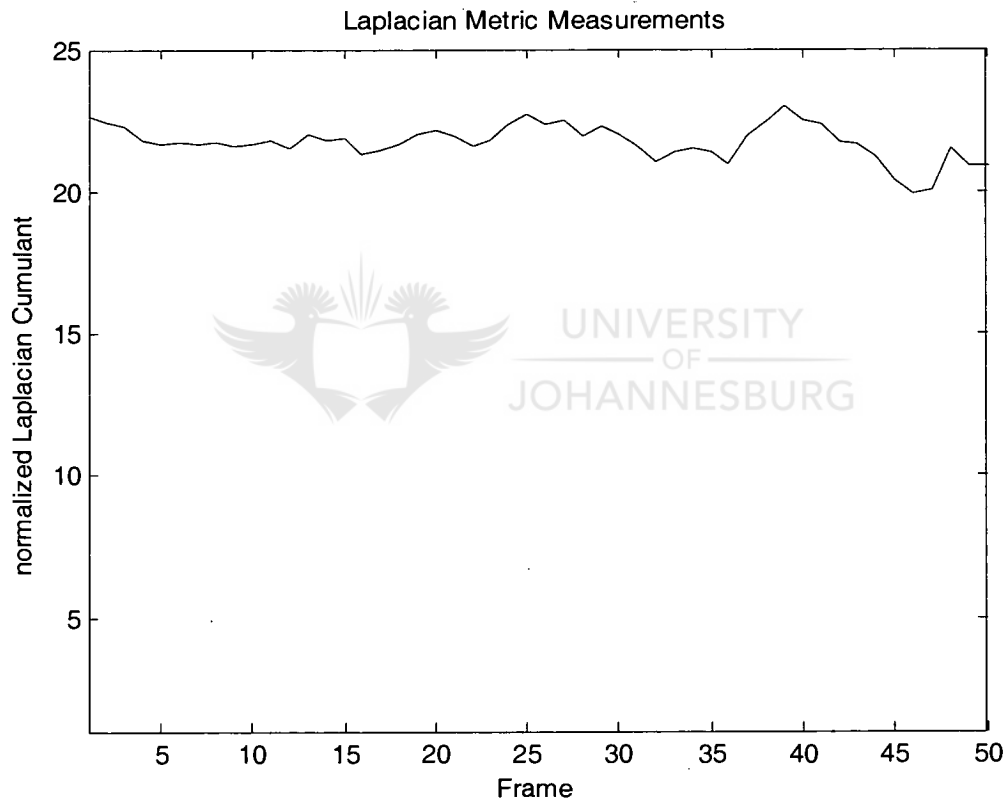


Figure 18-6: Dynamic Illuminance-Reflectance atmospheric turbulence suppression with Wiener filtering - image sharpness vs. frame number graph.

19 ADDENDUM J

EXPERIMENT 3: ALGORITHM ABERRATION

ADDITIONAL TABLES

19.1 MEASUREMENT RESULTS

Results are presented showing measured square size in pixels, calculated actual size in mm, and the calculated displacement in both a horizontal and vertical direction.

Size of the squares is known to be 200mm.

Averaging frames and measuring multiple square sizes and averaging measurement square size is measured to span 51 pixels.

$$\begin{aligned} \text{Therefore mm/pixel} &= 200/51 \text{ mm/pixel} \\ &= 3.921569 \text{ mm/pixel} \end{aligned}$$

This is used for converting the measured lengths in the video from pixels to mm.

Original Unprocessed Video		
Direction Measured	Horizontal	Vertical
Measurement 1 (pixels)	82	64
Measurement 2 (pixels)	78	63
Measurement 3 (pixels)	79	63
Measurement 4 (pixels)	76	63
Average (pixels)	78.750	63.250
Size (mm)	308.824	248.039
Displacement (mm)	54.412	24.020

Table 19-1: Unprocessed video aberration result.



Image Registration and Laplacian Pyramid Image Fusion		
Direction Measured	Horizontal	Vertical
Measurement 1 (pixels)	74	61
Measurement 2 (pixels)	75	61
Measurement 3 (pixels)	74	61
Measurement 4 (pixels)	74	61
Average (pixels)	74.250	61.000
Size (mm)	291.177	239.216
Displacement (mm)	45.588	19.608

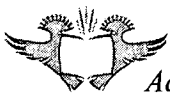
Table 19-2: Image Registration and Laplacian Pyramid Image Fusion aberration result.

Homomorphic Filtering and Power Spectrum Based Method		
Direction Measured	Horizontal	Vertical
Measurement 1 (pixels)	76	63
Measurement 2 (pixels)	76	64
Measurement 3 (pixels)	76	63
Measurement 4 (pixels)	78	62
Average (pixels)	76.500	63.000
Size (mm)	300.000	247.059
Displacement (mm)	50.000	23.529

Table 19-3: Homomorphic Filtering and Power Spectrum based method aberration result.

Wiener Filtering using Laplacian Operator		
Direction Measured	Horizontal	Vertical
Measurement 1 (pixels)	70	60
Measurement 2 (pixels)	73	60
Measurement 3 (pixels)	71	60
Measurement 4 (pixels)	75	58
Average (pixels)	72.250	59.500
Size (mm)	283.333	233.333
Displacement (mm)	41.667	16.667

Table 19-4: Wiener Filtering using Laplacian Operator aberration result.



Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression		
Direction Measured	Horizontal	Vertical
Measurement 1 (pixels)	70	58
Measurement 2 (pixels)	71	59
Measurement 3 (pixels)	73	57
Measurement 4 (pixels)	74	57
Average (pixels)	72.000	57.750
Size (mm)	282.353	226.471
Displacement (mm)	41.176	13.235

Table 19-5: Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression aberration result.

Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener Filtering		
Direction Measured	Horizontal	Vertical
Measurement 1 (pixels)	74	63
Measurement 2 (pixels)	73	61
Measurement 3 (pixels)	75	59
Measurement 4 (pixels)	76	60
Average (pixels)	74.500	60.750
Size (mm)	292.157	238.235
Displacement (mm)	46.078	19.118

Table 19-6: Dynamic Illuminance-Reflectance Atmospheric Turbulence Suppression with Wiener Filtering aberration result.

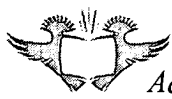
20 ADDENDUM K

EXPERIMENT 4: ALGORITHM MTF RESULT

ADDITIONAL TABLES

Spatial Frequency	Line pairs/mm	Minimum Average	Maximum Average	MTF (%)
Low	0.002410	67.024	254.070	90.452721
	0.004819	89.213	250.310	73.675596
	0.007231	130.135	235.811	44.839913
	0.009634	154.390	220.878	27.511048
Medium	0.010846	186.954	225.047	16.114237
	0.012048	197.767	228.023	12.384421
	0.013245	191.683	216.707	10.679481
	0.014451	199.022	215.087	6.761394
	0.015674	194.311	209.800	6.680079
	0.016863	198.791	212.628	5.861734
	0.018083	193.222	207.333	6.139881
High	0.019268	195.000	209.419	6.214082
	0.025000	183.861	201.000	6.790133
	0.033333	193.263	205.290	4.600930
	0.050000	183.250	198.694	6.165528
	0.100000	189.576	200.303	4.195233

Table 20-1: Unprocessed video MTF measurements.



Spatial Frequency	Line pairs/mm	Minimum Average	Maximum Average	MTF (%)
Low	0.002410	67.707	255.000	92.212580
	0.004819	88.282	255.000	77.163014
	0.007231	109.270	255.000	63.562712
	0.009634	136.781	255.000	47.942912
Medium	0.010846	187.233	255.000	25.808719
	0.012048	195.047	255.000	22.436418
	0.013245	198.098	253.878	20.785621
	0.014451	205.283	249.935	16.520393
	0.015674	204.978	244.711	14.881231
	0.016863	210.163	252.302	15.346410
	0.018083	205.472	237.750	12.265318
High	0.019268	204.710	235.000	11.602028
	0.025000	200.167	234.833	12.661936
	0.033333	208.553	236.763	10.065198
	0.050000	197.472	226.917	11.023469
	0.100000	203.394	218.485	5.683366

Table 20-2: Image Registration and Laplacian Pyramid Image Fusion algorithm MTF measurements.

Spatial Frequency	Line pairs/mm	Minimum Average	Maximum Average	MTF (%)
Low	0.002410	75.293	255.000	93.978316
	0.004819	100.718	255.000	74.915337
	0.007231	133.784	245.838	50.984406
	0.009634	159.732	232.585	32.075637
Medium	0.010846	185.070	229.326	19.246764
	0.012048	198.279	231.535	13.944062
	0.013245	193.829	221.024	11.814032
	0.014451	202.783	223.044	8.574897
	0.015674	193.844	214.089	8.943764
	0.016863	201.442	223.488	9.350276
	0.018083	198.167	215.444	7.528288
High	0.019268	202.774	217.968	6.507955
	0.025000	183.556	203.472	7.175462
	0.033333	198.290	215.658	5.850480
	0.050000	184.972	205.944	7.480605
	0.100000	193.818	206.182	4.309846

Table 20-3: Homomorphic Filtering and Power Spectrum based method MTF measurements.

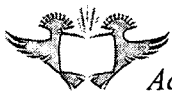


Spatial Frequency	Line pairs/mm	Minimum Average	Maximum Average	MTF (%)
Low	0.002410	64.220	255.000	95.745450
	0.004819	83.359	252.154	80.597867
	0.007231	119.405	231.243	51.096374
	0.009634	142.683	214.049	32.049600
Medium	0.010846	180.907	223.884	19.464532
	0.012048	190.907	222.349	13.948576
	0.013245	185.146	213.195	12.909228
	0.014451	194.652	211.196	7.473181
	0.015674	189.222	206.378	7.950438
	0.016863	194.651	210.279	7.075576
	0.018083	188.667	203.583	6.971855
High	0.019268	192.581	204.645	5.568227
	0.025000	181.556	202.778	8.215391
	0.033333	190.790	211.658	7.714824
	0.050000	181.778	205.528	9.123381
	0.100000	189.152	210.394	7.910126

Table 20-4: Wiener Filtering using Laplacian Operator algorithm MTF measurements.

Spatial Frequency	Line pairs/mm	Minimum Average	Maximum Average	MTF (%)
Low	0.002410	63.951	255.000	94.386412
	0.004819	74.718	240.923	82.973727
	0.007231	102.351	203.757	52.200603
	0.009634	123.366	186.634	32.159789
Medium	0.010846	157.861	205.093	23.857854
	0.012048	168.209	202.837	17.109576
	0.013245	162.122	191.122	15.050975
	0.014451	171.391	190.935	9.888813
	0.015674	166.044	184.356	9.580632
	0.016863	172.721	189.605	8.543043
	0.018083	164.944	180.806	8.410374
High	0.019268	169.581	184.645	7.796844
	0.025000	143.833	161.806	8.400342
	0.033333	152.211	169.974	7.876240
	0.050000	144.056	159.611	7.317939
	0.100000	149.758	162.485	5.822949

Table 20-5: Dynamic Illuminance-Reflectance method MTF measurements.



Spatial Frequency	Line pairs/mm	Minimum Average	Maximum Average	MTF (%)
Low	0.002410	88.073	255.000	94.781390
	0.004819	95.308	246.154	86.055006
	0.007231	117.838	223.676	60.369438
	0.009634	135.732	210.732	42.168456
Medium	0.010846	145.163	204.977	34.469000
	0.012048	155.698	201.744	25.993146
	0.013245	151.268	192.195	24.043351
	0.014451	159.000	187.630	16.665887
	0.015674	155.267	184.267	17.233884
	0.016863	161.256	187.070	14.953319
	0.018083	155.556	182.889	16.295672
	0.019268	159.484	181.774	13.179551
High	0.025000	157.972	202.194	14.368179
	0.033333	164.790	210.105	14.145069
	0.050000	156.611	209.167	16.813824
	0.100000	164.576	209.394	14.024321

Table 20-6: Dynamic Illuminance-Reflectance with Wiener filtering method MTF measurements.

Spatial Frequency	Line pairs/mm	Minimum Average	Maximum Average	MTF (%)
Low	0.002410	54.148	253.630	96.102776
	0.004819	62.179	248.464	88.917766
	0.007231	85.263	216.553	64.499768
	0.009634	110.594	199.969	42.671446
Medium	0.010846	151.000	213.944	28.027461
	0.012048	169.233	210.900	17.811749
	0.013245	161.919	198.405	16.454778
	0.014451	173.184	202.553	12.701351
	0.015674	170.683	194.488	10.593118
	0.016863	176.800	198.600	9.436601
	0.018083	169.889	189.756	8.976472
	0.019268	172.615	195.487	10.096825
High	0.025000	155.400	181.886	10.470135
	0.033333	165.467	195.567	11.116238
	0.050000	156.061	186.697	11.917616
	0.100000	164.657	194.400	11.044817

Table 20-7: Speckle Masking algorithm MTF measurements.

21 ADDENDUM L

EXPERIMENT 6: WIENER FILTERING LAMBDA SELECTION ADDITIONAL GRAPHS AND TABLES

21.1 ALGORITHM PERFORMANCE RESULTS

21.1.1 Wiener Filtering User Selected Lambda Algorithm

Image size	FPS
80x60	21.28
160x120	15.87
240x180	11.9
320x240	8
400x300	5.62
480x360	4.22
560x420	3.16
640x480	2.51

Table 21-1: Wiener filtering without sharpness measurement λ selection FPS performance

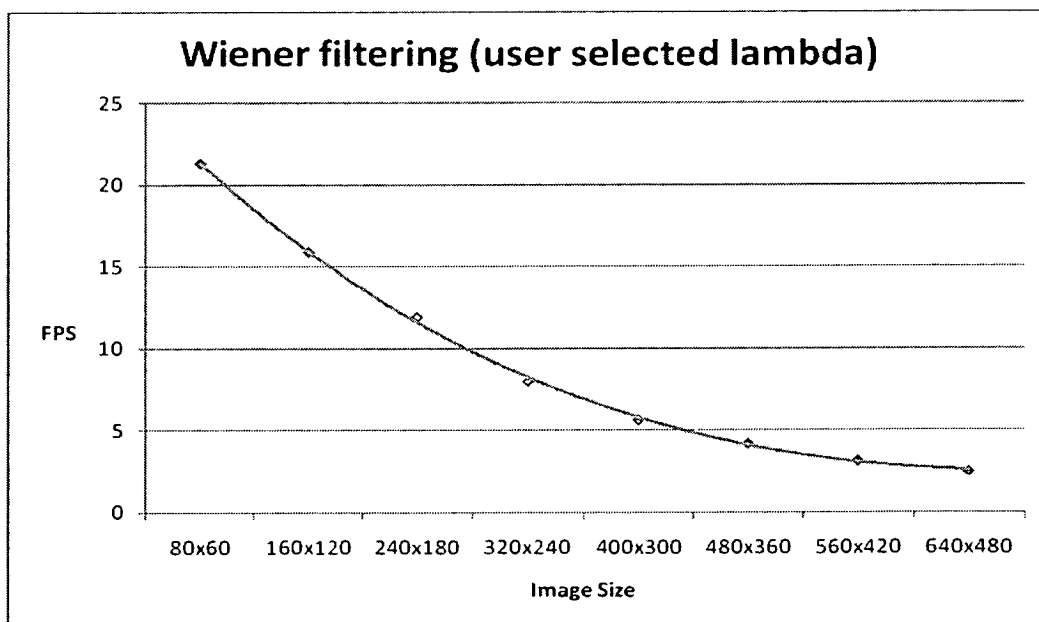


Figure 21-1: Wiener filtering without sharpness measurement λ selection FPS performance graph

21.1.2 Wiener Filtering using Laplacian Operator

Image size	FPS
80x60	11.439
160x120	4.973
240x180	2.588
320x240	1.497
400x300	0.972
480x360	0.682
560x420	0.499
640x480	0.387

Table 21-2: Wiener filtering using Laplacian Operator FPS performance

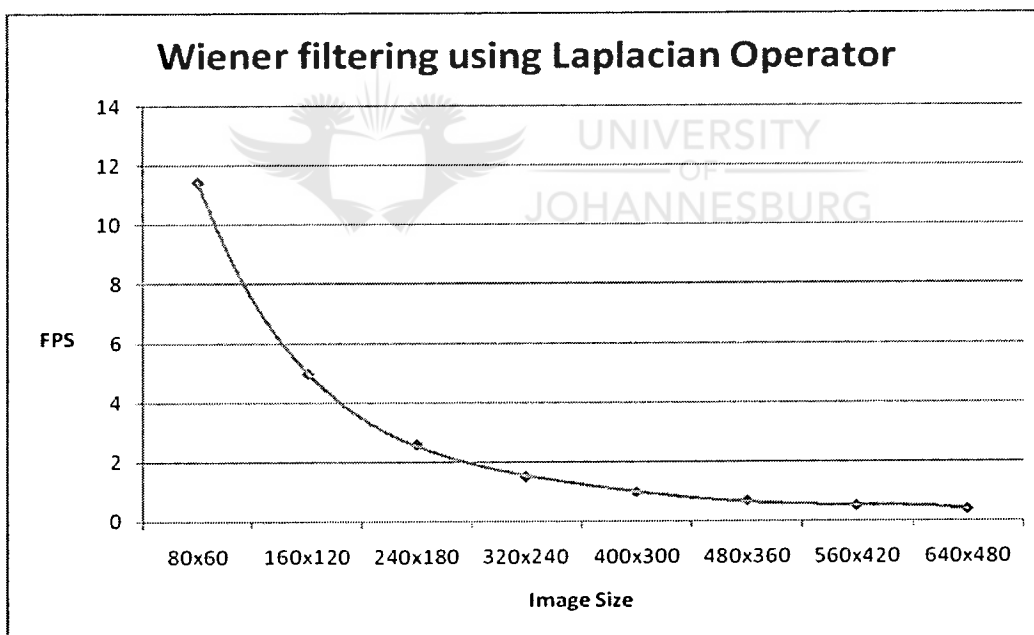


Figure 21-2: Wiener filtering using Laplacian Operator FPS performance graph

21.2 IMAGE SHARPNESS RESULTS

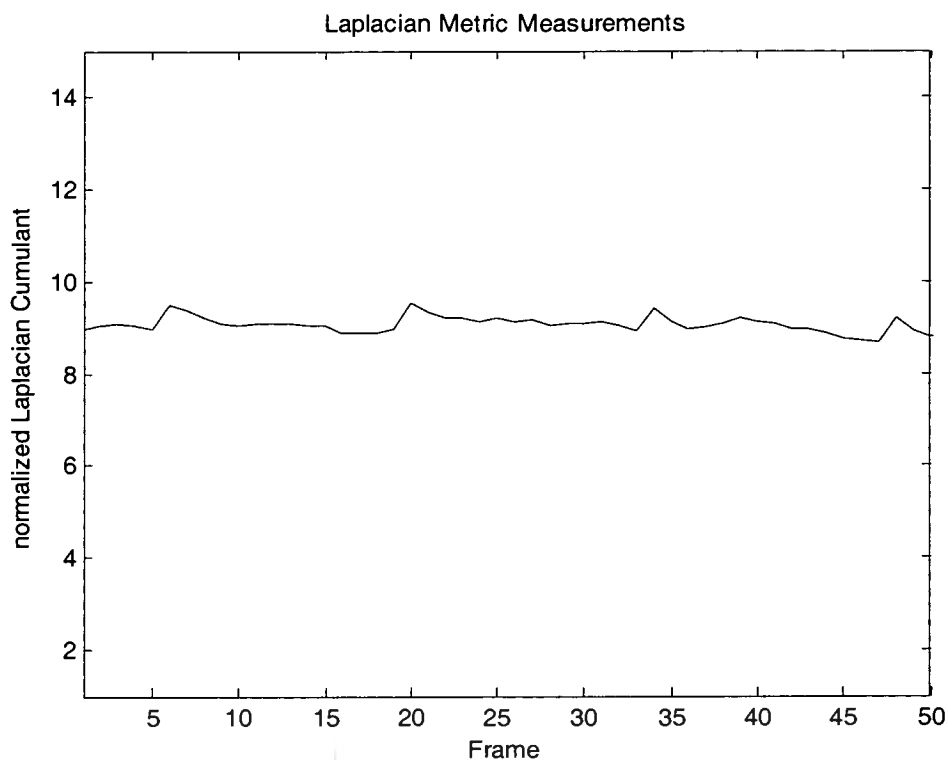


Figure 21-3: Wiener filtering without sharpness measurement λ selection - sharpness measurement graph

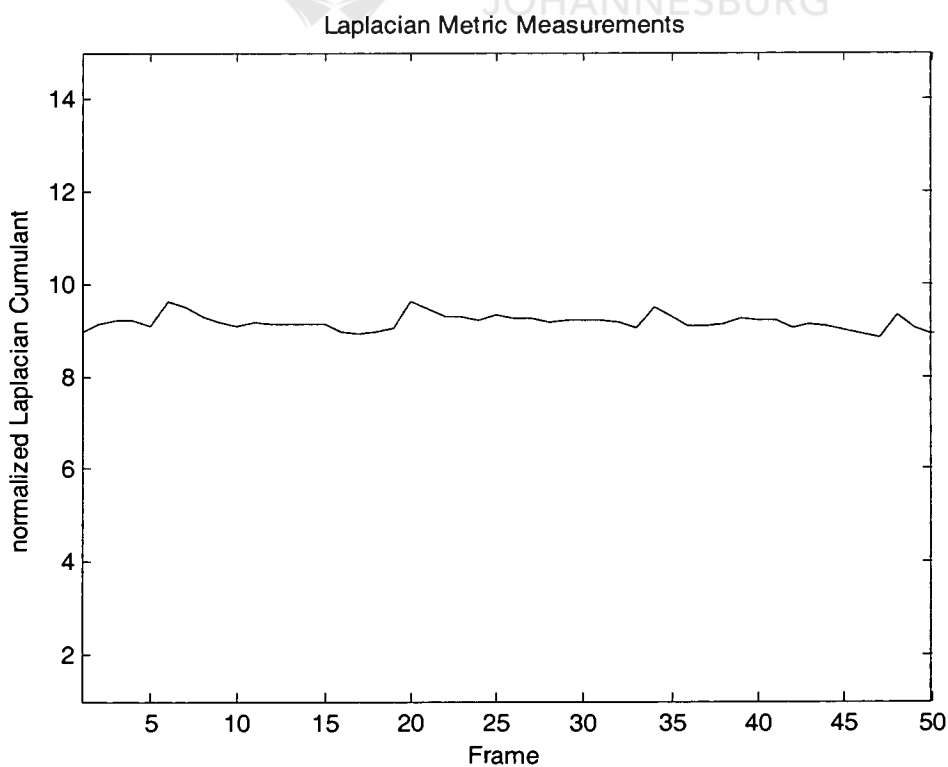


Figure 21-4: Wiener filtering using Laplacian Operator – sharpness measurement graph

21.3 MTF RESULTS

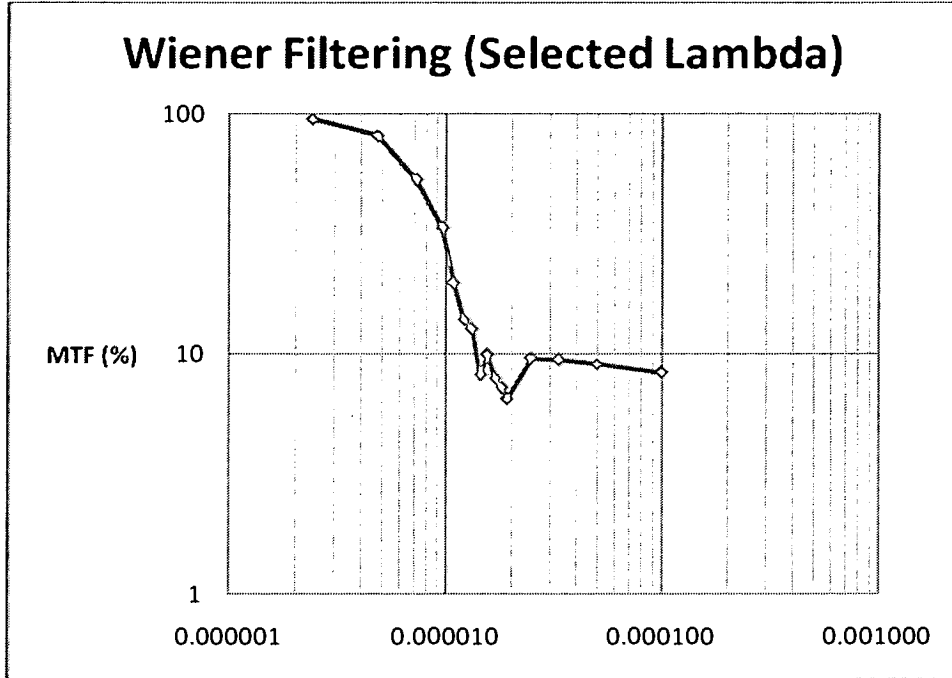


Figure 21-5: Wiener filtering without sharpness measurement λ selection – MTF graph

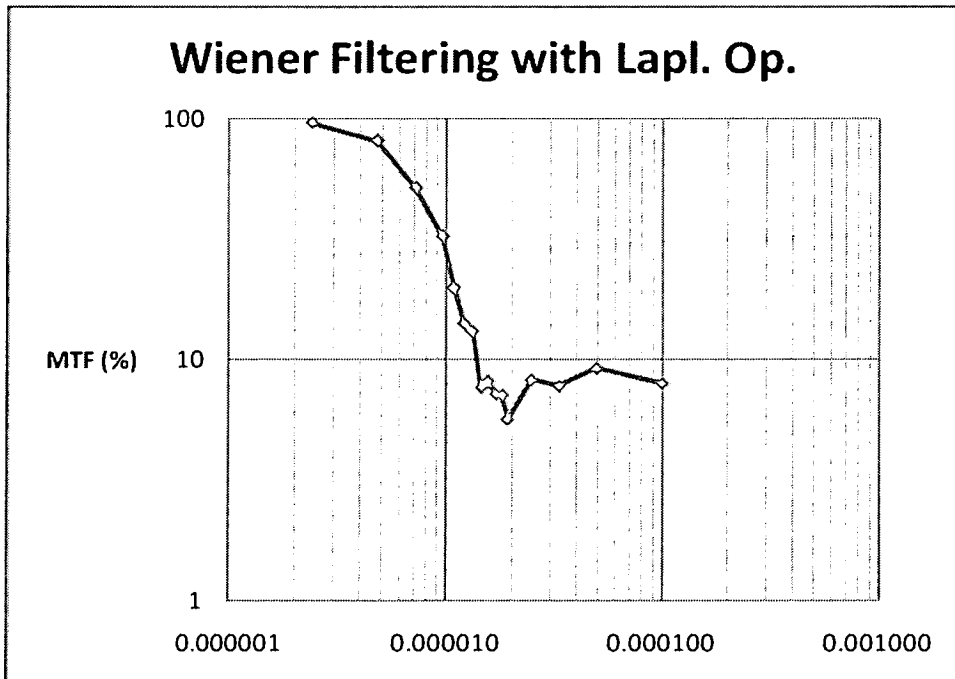


Figure 21-6: Wiener filtering using Laplacian Operator – MTF graph

22 ADDENDUM M

EXPERIMENT 7: FILTERING RECEIVED LIGHT ADDITIONAL GRAPHS AND TABLES

22.1 IMAGE SHARPNESS RESULTS ACROSS 50 FRAMES

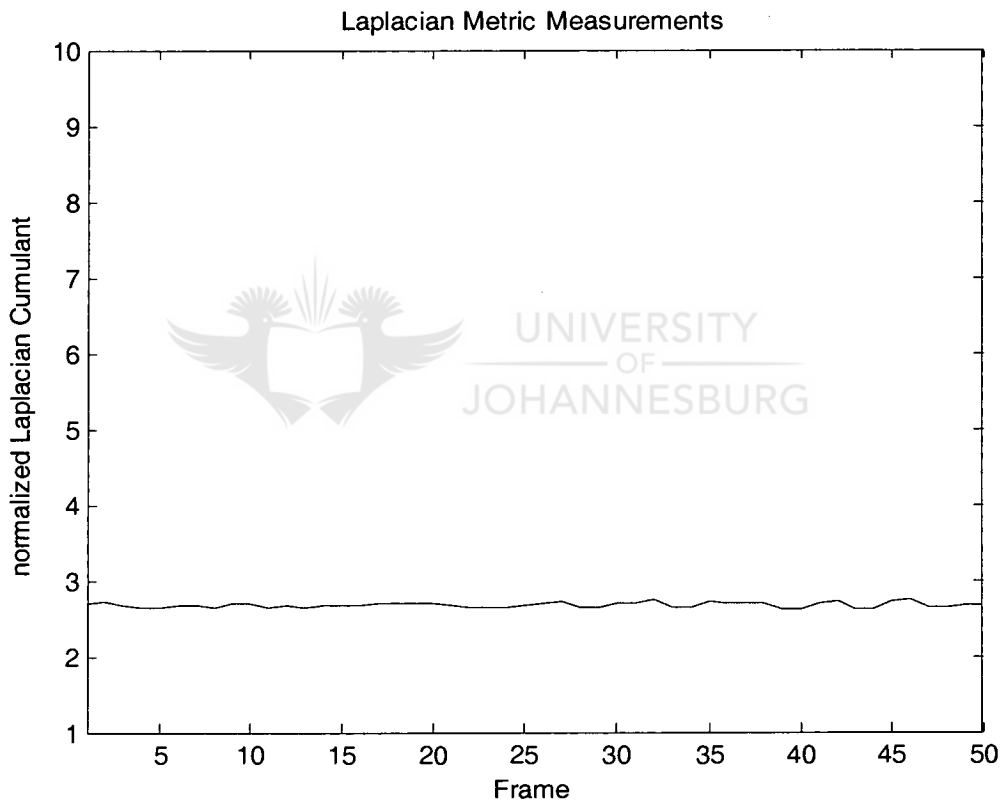


Figure 22-1: Unfiltered video - sharpness result.

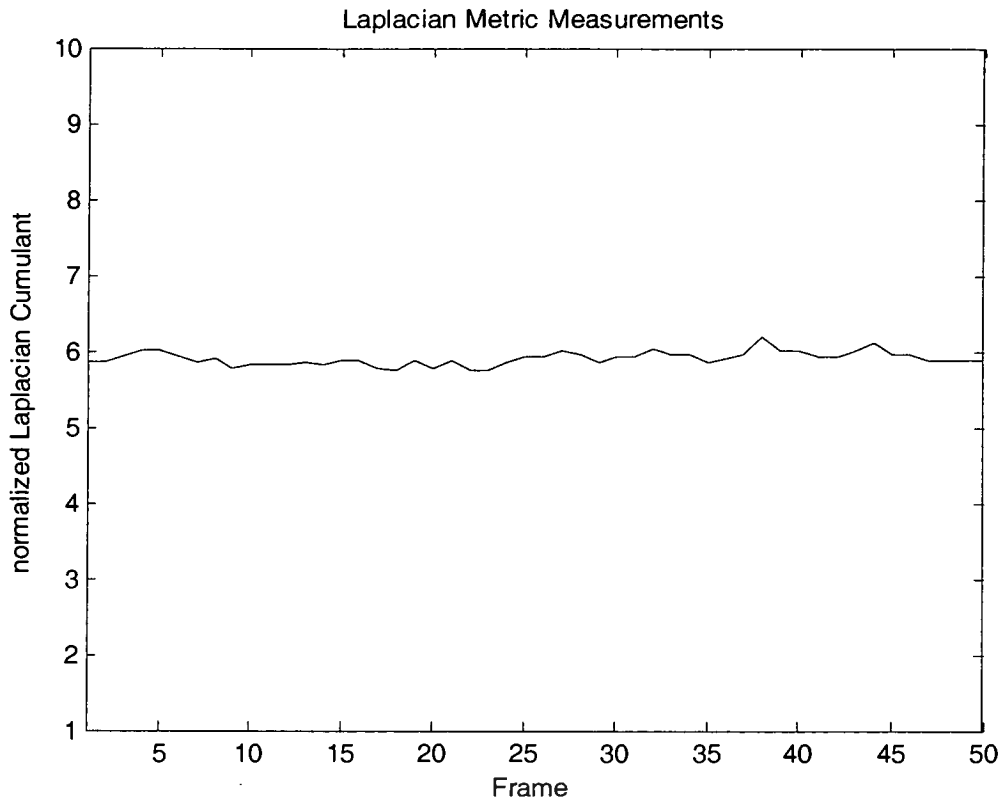


Figure 22-2: Colour filtered video - sharpness result.

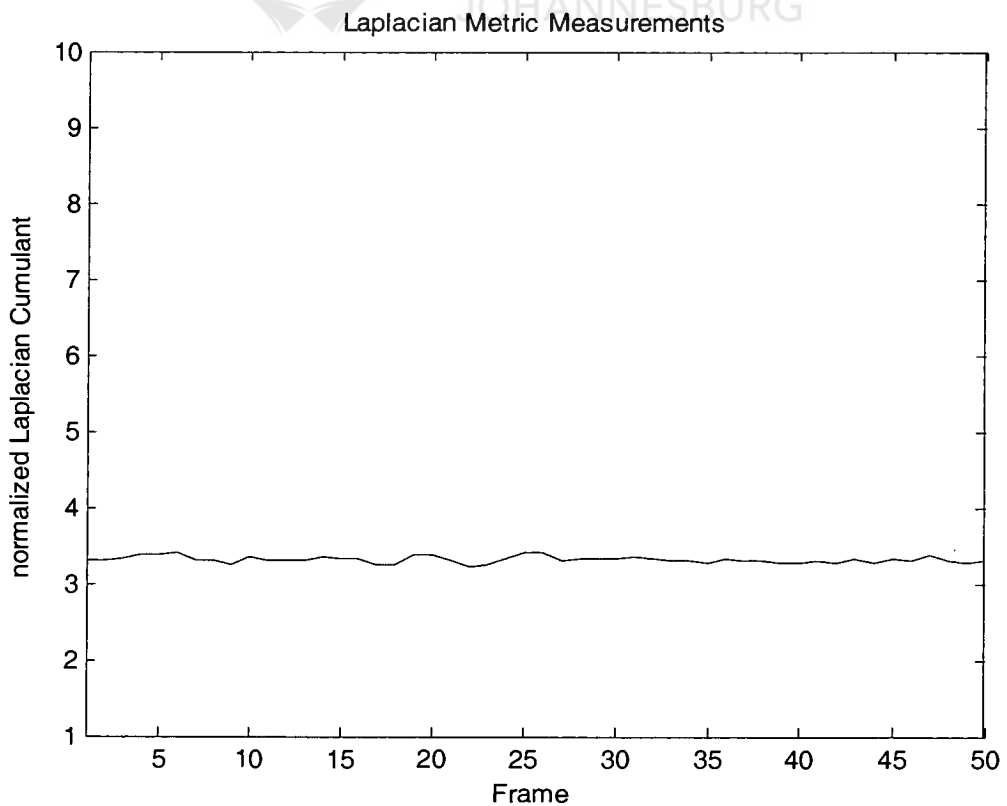


Figure 22-3: Polarised filtered video - sharpness result.

22.2 MTF RESULTS

Spatial Frequency	Line pairs/mm	Minimum Average	Maximum Average	MTF (%)
Low	0.002410	67.024	254.070	90.452721
	0.004819	89.213	250.310	73.675596
	0.007231	130.135	235.811	44.839913
	0.009634	154.390	220.878	27.511048
Medium	0.010846	186.954	225.047	16.114237
	0.012048	197.767	228.023	12.384421
	0.013245	191.683	216.707	10.679481
	0.014451	199.022	215.087	6.761394
	0.015674	194.311	209.800	6.680079
	0.016863	198.791	212.628	5.861734
	0.018083	193.222	207.333	6.139881
	0.019268	195.000	209.419	6.214082
High	0.025000	183.861	201.000	6.790133
	0.033333	193.263	205.290	4.600930
	0.050000	183.250	198.694	6.165528
	0.100000	189.576	200.303	4.195233

Table 22-1: Unfiltered video - MTF measurements.

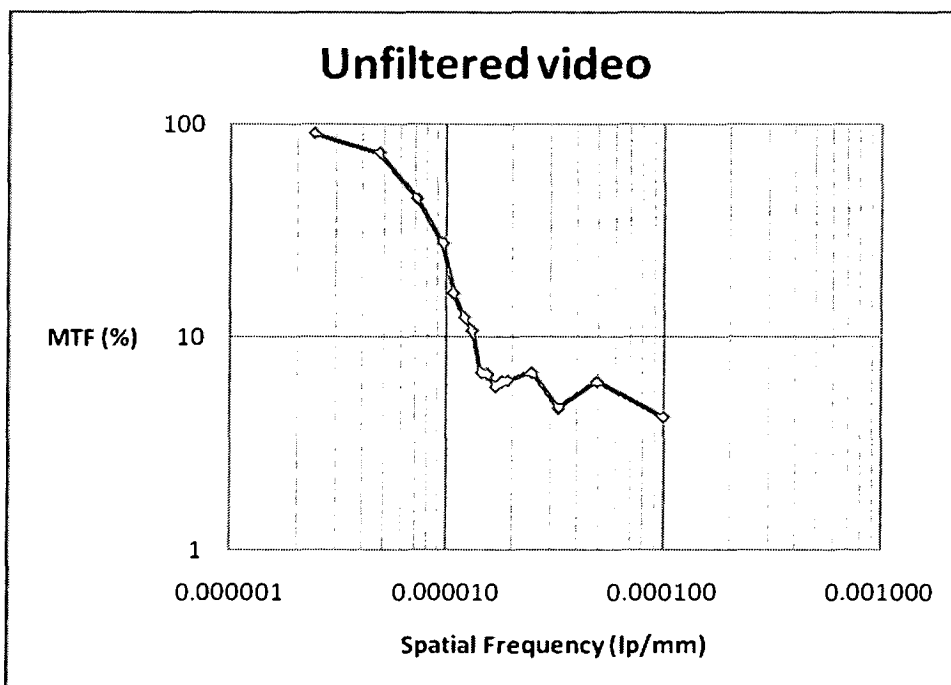


Figure 22-4: Unfiltered video - MTF curve.



Spatial Frequency	Line pairs/mm	Minimum Average	Maximum Average	MTF (%)
Low	0.002410	34.903	147.871	95.907881
	0.004819	48.191	146.429	78.326552
	0.007231	71.769	136.308	48.129297
	0.009634	86.784	124.622	27.773097
Medium	0.010846	116.744	147.974	21.910130
	0.012048	127.186	147.628	13.814268
	0.013245	119.441	146.765	19.061802
	0.014451	127.531	145.594	12.281787
	0.015674	124.077	144.308	13.999121
	0.016863	125.853	143.118	11.920643
	0.018083	121.065	135.645	10.548274
	0.019268	126.667	139.548	8.985870
High	0.025000	152.636	167.000	7.651488
	0.033333	158.962	168.462	4.940299
	0.050000	152.194	162.065	5.348274
	0.100000	155.778	163.704	4.224173

Table 22-2: Red #24 Colour filtered video - MTF measurements.

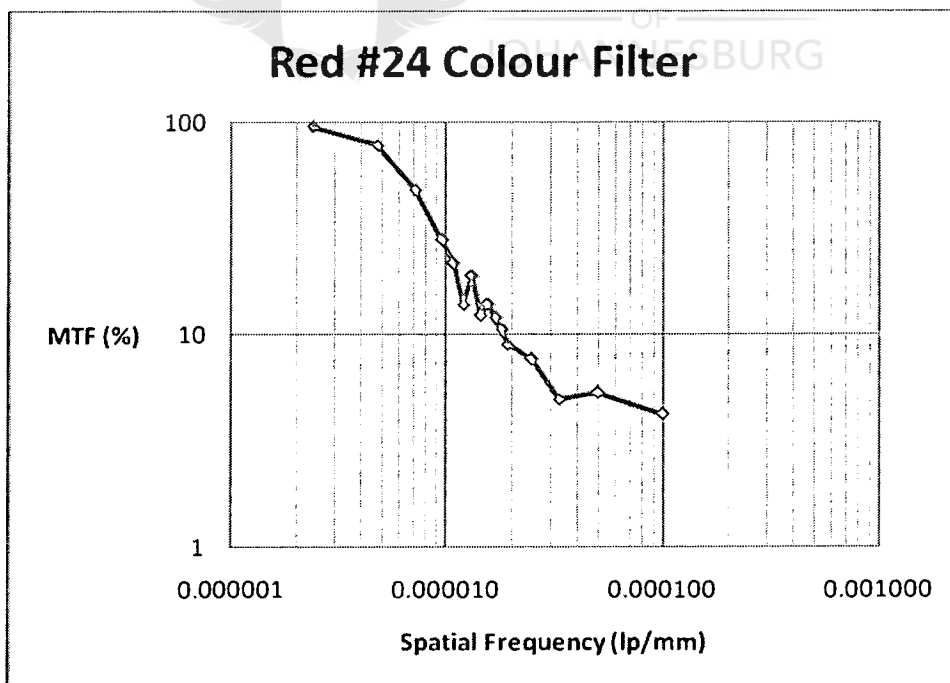
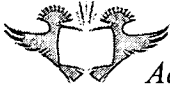


Figure 22-5: Red #24 colour filtered video - MTF curve.



Spatial Frequency	Line pairs/mm	Minimum Average	Maximum Average	MTF (%)
Low	0.002410	70.806	242.556	97.403343
	0.004819	108.818	218.273	59.468571
	0.007231	142.487	195.568	27.904616
	0.009634	148.333	181.767	17.999372
Medium	0.010846	179.794	219.647	19.374307
	0.012048	181.222	208.694	13.681690
	0.013245	179.029	204.088	12.701242
	0.014451	185.000	203.031	9.023568
	0.015674	180.962	199.885	9.648522
	0.016863	183.412	197.647	7.254250
	0.018083	179.097	193.581	7.546881
	0.019268	182.429	193.500	5.718935
High	0.025000	173.556	183.667	5.170762
	0.033333	180.929	188.571	3.778616
	0.050000	173.222	180.722	3.871086
	0.100000	178.650	183.350	2.371832

Table 22-3: Polarised video - MTF measurements.

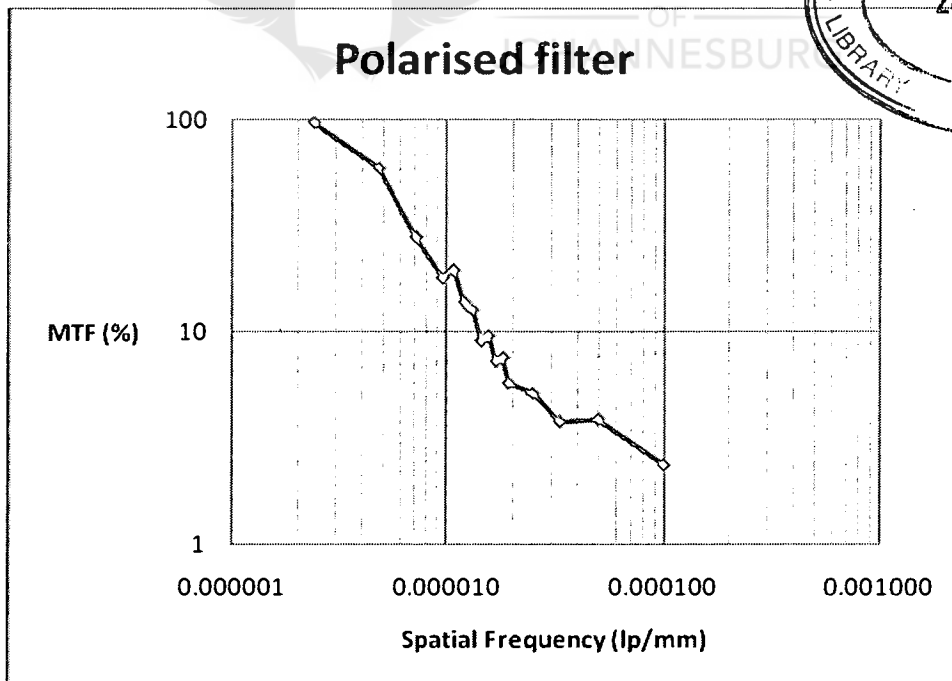
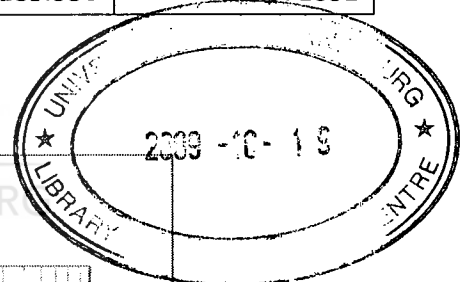


Figure 22-6: Polarised video - MTF curve.