DATA ANALYTICS AND METHODS FOR IMPROVED FEATURE SELECTION AND MATCHING

A thesis submitted to the University of Manchester for the degree of Doctor of Philosophy in the Faculty of Engineering and Physical Sciences

2012

By
Michael May
School of Computer Science
3.2.2 Debevec and Malik .............................................. 75
3.2.3 Other Methods of Generating HDR Images .................. 77
3.2.4 HDR Cameras .................................................. 79
3.2.5 Tone Mapping .................................................. 80
3.2.6 Exposure Fusion .............................................. 82
3.2.7 HDR Video .................................................... 85
3.2.8 HDR Feature Matching ........................................ 86
3.3 3D Reconstruction ................................................ 87
  3.3.1 Stereo Vision ............................................... 87
  3.3.2 Structure from Motion ..................................... 88
  3.3.3 Bundler and Bundle Adjustment ......................... 89
  3.3.4 Using 3D Information for Feature Matching and Object
        Recognition .................................................. 91
3.4 Summary .......................................................... 93
3.5 Focus of Thesis .................................................. 93

4 Preliminary Analysis of the SIFT .................................. 95
  4.1 The SIFT Descriptor ........................................... 95
     4.1.1 Distance Distributions .................................. 97
     4.1.2 Why the Feature Space is Not Filled .................. 100
  4.2 Image Ground Truth: Homography and Annotation ............ 104
     4.2.1 Homography Based Match Checking ..................... 104
     4.2.2 Annotation Based Match Checking ...................... 105
  4.3 Metrics for Measuring Feature Matches ....................... 106
  4.4 A Contrario Methodology ..................................... 109
     4.4.1 Background Model ........................................ 109
     4.4.2 One to One and One to Many Feature Matching ....... 111
     4.4.3 Examples of A Contrario Feature Matching for Object
            Detection .................................................. 112
  4.5 Conclusion .................................................... 113

5 SIFTing Through Parameter Space ................................. 115
  5.1 GPGPU Techniques ............................................. 117
     5.1.1 GPGPUs and CUDA ........................................ 117
     5.1.2 Methodology .............................................. 119
     5.1.3 Choice of Parameters for Investigation ............... 119
  5.2 Genetic Algorithms for Parameter Investigation .............. 123
  5.3 Data Visualization and Analysis Techniques .................. 127
  5.4 Results, Analysis and Recommendations ....................... 128
     5.4.1 Data Sets Used for Analysis ............................. 130
     5.4.2 Results .................................................. 132
     5.4.3 Analysis - General Observations ...................... 132
     5.4.4 Validation of the Choice of Analysis Data .......... 138
     5.4.5 Further Analysis of the Sweep Data ................. 138
5.5 Conclusion ................................................. 144

6 High Dynamic Range Fusion Features ................. 147
  6.1 High Dynamic Range Fusion Features ............... 148
  6.2 Fusion Feature Selection .............................. 149
    6.2.1 Feature Blending ................................. 153
    6.2.2 Evaluation Methodology .......................... 153
    6.2.3 Results and Analysis ............................ 156
  6.3 Stereo Fusion Features ............................... 165
    6.3.1 Results ........................................ 168
    6.3.2 Analysis ........................................ 169
    6.3.3 Conclusion ..................................... 169

7 A contrario SIFT Feature Weighting .................. 170
  7.1 FEWER ................................................ 171
  7.2 Calculating Feature Weights ........................ 175
  7.3 Evaluation Methodology .............................. 176
    7.3.1 Examples of Matching and Weighting ............ 178
    7.3.2 Selecting a Threshold ............................ 178
  7.4 RANSAC Comparison ................................. 181
    7.4.1 Results ........................................ 181
    7.4.2 Thresholding Using 0.7 ......................... 184
    7.4.3 Error Cases ..................................... 185
  7.5 Evaluation ........................................... 187
  7.6 Conclusion ........................................... 188

8 Conclusion and Future Work ............................ 190
  8.1 Summary ............................................. 190
  8.2 Contributions ........................................ 191
    8.2.1 Preliminary Analysis of the SIFT. ............... 191
    8.2.2 SIFTing through Parameter Space ................. 191
    8.2.3 Multi-exposure, High Dynamic Range, Fusion Features. 191
    8.2.4 A contrario SIFT Feature Weighting ............. 192
  8.3 Future Work .......................................... 192
    8.3.1 SIFTing through Parameter Space ................ 192
    8.3.2 Multi-exposure, High Dynamic Range, Fusion Features. 193
    8.3.3 A contrario SIFT Feature Weighting ............. 193

A .............................................................. 195
  A.1 Oxford Test Images ..................................... 195
  A.2 Sweep/GA Graphs ...................................... 197
    A.2.1 Oxford Sweep .................................... 197
  A.3 Lowe vs Sweep Histogram Values ..................... 200
  A.4 MBDA Test Images ..................................... 201
## List of Tables

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.1</td>
<td>Table showing examples of matches and the resulting <em>correspondence ratio.</em></td>
<td>108</td>
</tr>
<tr>
<td>5.1</td>
<td>The main parameters of the SIFT algorithm and Lowe’s default values.</td>
<td>118</td>
</tr>
<tr>
<td>5.2</td>
<td>The sweep input parameters.</td>
<td>122</td>
</tr>
<tr>
<td>5.3</td>
<td>The sweep output parameters.</td>
<td>123</td>
</tr>
<tr>
<td>5.4</td>
<td>The GA input range.</td>
<td>124</td>
</tr>
<tr>
<td>5.5</td>
<td>Observed parameters hierarchy.</td>
<td>137</td>
</tr>
<tr>
<td>6.1</td>
<td>The mean and median results of FAW for twenty eight test exposure image pairs.</td>
<td>157</td>
</tr>
<tr>
<td>7.1</td>
<td>The stages used for extracting and weighting features with FEWER.</td>
<td>175</td>
</tr>
<tr>
<td>7.2</td>
<td>Weighting values calculated from experimental data for different aspects of the weighting process.</td>
<td>176</td>
</tr>
<tr>
<td>7.3</td>
<td>The results for test case A for RANSAC and the left pair of images using the FEWER technique.</td>
<td>184</td>
</tr>
<tr>
<td>7.4</td>
<td>The results for test case B for RANSAC and the left pair of images using the FEWER technique.</td>
<td>184</td>
</tr>
<tr>
<td>7.5</td>
<td>The results for test case C for RANSAC and the left pair of images using the FEWER technique.</td>
<td>185</td>
</tr>
<tr>
<td>7.6</td>
<td>The results for the selected test frames with a threshold value of 0.7185</td>
<td>185</td>
</tr>
<tr>
<td>A.1</td>
<td>The <em>precision</em> and number of correct matches for feature matching when using the Lowe’s default SIFT parameters and the parameters obtained in the sweep from for the Oxford set using the ‘best 500’ values.</td>
<td>200</td>
</tr>
<tr>
<td>A.2</td>
<td>The <em>precision</em> and number of correct matches for feature matching when using the Lowe’s default SIFT parameters and the parameters obtained in the sweep from for the MBDA set using the ‘best 500’ values.</td>
<td>208</td>
</tr>
</tbody>
</table>
List of Figures

1.1 Framework for the proposed 3D HDR Video SIFT object recognition system. ........................................... 24
1.2 An example of SIFT matching. ........................................... 26
1.3 An example of the results of the parameter sweep and genetic algorithm. ........................................... 27
1.4 An example of the fusion feature technique. ....................... 28
1.5 A typical example of weighted feature matching with FEWER. .. 29

2.1 Example of Harris features. ........................................... 34
2.2 The SUSAN mask on a simple sample image. ....................... 37
2.3 The filters used for CenSurE and STAR ............................ 38
2.4 The Difference of Gaussian (DoG) technique used to extract an initial set of features at different scales. Image from [70]. ........ 40
2.5 The scale space extrema are selected by comparing each pixel with its 26 neighbours in the DoG pyramid. Image from [70]. ....... 41
2.6 An example of an interpolated sub-pixel location. .................. 42
2.7 An example of the SIFT orientation histogram stage. .............. 44
2.8 A 2 × 2 example of the feature descriptor grid used in the SIFT. .. 45
2.9 How the image gradients relate to a single set of eight histogram bins. ........................................... 45
2.10 The 4 × 4 SIFT descriptor. ........................................... 46
2.11 The log-polar location grid used by GLOH. Image from [86]. ... 47
2.12 An example of the issue that can arise without the use of colour information. ................................. 48
2.13 The sampling strategy used by SIFT with Irregular Orientation Histogram Binning. .................. 50
2.14 The sampling strategy used by DAISY. ............................ 51

3.1 Speed graph for different SIFT implementation. .................... 73
3.2 An example of tone mapping. ........................................... 74

4.1 An average SIFT feature descriptor (2 Million features) ......... 96
4.2 The frequency distribution for each dimension of two million SIFT feature vectors .................................... 97
<table>
<thead>
<tr>
<th>Section</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3</td>
<td>Graphs of feature distributions for SIFT bins 40 to 47 from 2 million features</td>
</tr>
<tr>
<td>4.4</td>
<td>A flow chart demonstrating the steps of the SIFT algorithm.</td>
</tr>
<tr>
<td>4.5</td>
<td>The Euclidean distance distribution for 100,000,000 random feature pair combinations.</td>
</tr>
<tr>
<td>4.6</td>
<td>The matching distance for 100,000,000 combinations of random features.</td>
</tr>
<tr>
<td>4.7</td>
<td>The matching distance for 858,802,832 features from corresponding pairs of images.</td>
</tr>
<tr>
<td>4.8</td>
<td>Normalised match distributions with a log scale for the y-axis.</td>
</tr>
<tr>
<td>4.9</td>
<td>Features and their distribution extracted from a test image.</td>
</tr>
<tr>
<td>4.10</td>
<td>Two blurred images from the scale space used to extract the SIFT features.</td>
</tr>
<tr>
<td>4.11</td>
<td>A DoG image with the features generated from that approximate scale.</td>
</tr>
<tr>
<td>4.12</td>
<td>Examples of pairs of feature regions with various amounts of overlap and their approximate overlap error.</td>
</tr>
<tr>
<td>4.13</td>
<td>An example of an image pair with annotation boxes showing corresponding regions.</td>
</tr>
<tr>
<td>4.14</td>
<td>Example of object detection using the correspondence ratio.</td>
</tr>
<tr>
<td>4.15</td>
<td>Correlation graphs between correspondence ratio and other metrics.</td>
</tr>
<tr>
<td>4.16</td>
<td>The average results of matching random sets of features extracted from Flickr to 1062 images.</td>
</tr>
<tr>
<td>4.17</td>
<td>The distribution of the correspondence ratio for the million features matched to a video sequence.</td>
</tr>
<tr>
<td>4.18</td>
<td>Example test images for the a contrario technique.</td>
</tr>
<tr>
<td>4.19</td>
<td>First example of thresholding using the a contrario methodology.</td>
</tr>
<tr>
<td>4.20</td>
<td>Second example of thresholding using the a contrario methodology.</td>
</tr>
<tr>
<td>5.1</td>
<td>The data used to select the parameter sweep limits.</td>
</tr>
<tr>
<td>5.2</td>
<td>An example of the values generated from a dual objective genetic algorithm optimisation.</td>
</tr>
<tr>
<td>5.3</td>
<td>Stochastic uniform selection selects the next generation for the genetic algorithm.</td>
</tr>
<tr>
<td>5.4</td>
<td>Parallel coordinates displaying a 10 dimensional dataset with 90525 elements.</td>
</tr>
<tr>
<td>5.5</td>
<td>A pareto front generated from a parameter sweep.</td>
</tr>
<tr>
<td>5.6</td>
<td>Histograms of three parameters after the data has been brushed.</td>
</tr>
<tr>
<td>5.7</td>
<td>Box plot used to summarise the histogram data.</td>
</tr>
<tr>
<td>5.8</td>
<td>Images from the Oxford test data set.</td>
</tr>
<tr>
<td>5.9</td>
<td>Images from the MBDA test data set.</td>
</tr>
<tr>
<td>5.10</td>
<td>An example of the result data from the sweep and the histogram analysis.</td>
</tr>
<tr>
<td>5.11</td>
<td>Oxford GA Pareto Front - Sigma.</td>
</tr>
</tbody>
</table>
5.12 Oxford GA Pareto Front - Contrast. .................................. 133
5.13 Oxford GA Pareto Front - Curvature. ................................. 134
5.14 Oxford GA Pareto Front - Intervals. ................................. 134
5.15 Oxford GA Pareto Front - Octaves. ................................. 134
5.16 Oxford GA Pareto Front - Peak. ...................................... 134
5.17 Oxford GA Pareto Front - Match Ratio. ............................. 135
5.18 This shows the correlation between mean sigma and intervals for pareto front of the Oxford sweep. .......................... 136
5.19 This shows typical examples of the ‘best 500’ sweep histograms. 138
5.20 Graphs of Lowe’s values plotted against the histogram peak values for the MBDA set using the ‘best 500’ values ........................ 139
5.21 Graphs of Lowe’s values plotted against the histogram peak values for the Oxford set using the ‘best 500’ values ........................ 139
5.22 Pareto fronts and sweep distribution for the Oxford boats dataset which shows increasing change in zoom and rotation. .......... 140
5.23 Pareto fronts and sweep distribution for the Oxford bikes dataset which shows increasing changes in the level of blur. ............. 141
5.24 Pareto fronts and sweep distribution for the Oxford bricks dataset which shows increasing changes in viewpoint in angle. ......... 142
5.25 Pareto fronts and sweep distribution for the Oxford cars dataset which shows increasing change in lighting. ...................... 142
5.26 Pareto fronts and sweep distribution for the Oxford ubc dataset which shows decreasing levels of JPEG compression. ............ 143
5.27 GA and sweep results from Set 1 of the MBDA data set. ........ 145

6.1 A example where a single LDR exposure cannot capture all the information in the scene due to its high dynamic range. ........... 148
6.2 An example of two aligned input images taken at different exposures. 149
6.3 The normalised weightings generated from the exposure measures for the images in Fig 6.2 ............................................. 152
6.4 The set of fusion features displayed on a rough exposure fusion image. ................................................................. 154
6.5 Example set of high contrast images used for testing. ............. 154
6.6 FAW Feature matching examples represented by the parallel lines. 155
6.7 Graph of the recall results for 28 sets of features extracted from different image types or from exposure fusion. ................. 156
6.8 Graph of the 1 – precision results for 28 sets of features extracted from different image types or from exposure fusion. ........... 158
6.9 The distribution of the exposure fusion measure when compared to results from a single exposure. ................................. 158
6.10 The distribution of the exposure fusion measure when compared to tone mapped results. .............................................. 159
6.11 The distribution of the exposure fusion measure when compared to exposure fusion results. ........................................ 159
6.12 The result of the well-exposedness (E) measure returned for a pixel value. ........................................... 162
6.13 The mean values of $1 - \text{precision}$ for FAW compared to an LDR images............................................. 164
6.14 Test case 3, 19 and 27: Examples of the test cases which performs worse than the single exposure for $1 - \text{precision}$. The show the first exposure image, the exposure fusion image and the target image. The exposure fusion shows how the high dynamic range scene compares to the target image. ........................................... 165
6.15 A pair of stereo images at $10^\circ$ and a 2 EV difference. ................. 166
6.16 A graph showing the number of feature that are used to create a set of 3D points from stereo pairs at various angles and exposures. 167
6.17 The set of features selected from two stereo input images ............. 167
6.18 A graph showing the correspondence ratio for fusion features and features extracted from exposure fusion images generated from 28 pairs of stereo images. ................................. 168

7.1 A stereo pair of target images displaying the SIFT features extracted from them. There are 2176 in the left image and 2087 in the right image........................................... 171
7.2 The set of 3D feature positions generated from a stereo pair. ..... 172
7.3 The final set of clustered type2 and type3 features. ............... 172
7.4 The set of clustered features in a scene input image. .............. 173
7.5 This shows the two cases of type3 mismatches. .................... 173
7.6 The left hand view of each of the target image pairs ................ 177
7.7 Stereo background model image for the $a$ contrario technique. . 177
7.8 A typical example of weighted feature matching using FEWER. 178
7.9 A typical example of FEWER matching displaying matches from the left hand target image to the left scene image. ....... 179
7.10 This shows the correspondence ratio before and after applying a threshold on the FEWER feature weightings. .......... 179
7.11 These are the mean correspondence ratio graphs for the three feature types for matches from both target to both scene images. . 180
7.12 This shows the mean number of matches as the weighting threshold for 4400 frames which do not contain the target object. .... 181
7.13 The number of features remaining in a test set after various thresholds have been applied. ................................. 182
7.14 Histograms with logarithmic y axes of the weightings across each of the sets. ........................................... 183
7.15 Scatter plots of the number of selected matches in test cases A and C. ................................................................. 185
7.16 An example of a FEWER edge case. ................................. 186
7.17 An example motion blur causing an error in FEWER. ............ 186
A.1 The Oxford bark images; variation in zoom. 195
A.2 The Oxford bikes images; variation in blur. 195
A.3 The Oxford boat images; variation in zoom and rotation. 195
A.4 The Oxford bricks images; variation in viewpoint 196
A.5 The Oxford cars images; variation in lighting 196
A.6 The Oxford graffiti images; variation in viewpoint. 196
A.7 The Oxford trees images; variation in blur. 196
A.8 The Oxford ubc images; variation in jpeg compression. 196
A.9 Oxford Sweep Best 500 - Sigma. 197
A.10 Oxford Sweep Best 500 - Contrast. 197
A.11 Oxford Sweep Best 500 - Curvature. 197
A.12 Oxford Sweep Best 500 - Intervals. 198
A.13 Oxford Sweep Best 500 - Octaves. 198
A.14 Oxford Sweep Best 500 - Match Ratio. 198
A.15 Oxford Sweep Pareto Front - Sigma. 198
A.16 Oxford Sweep Pareto Front - Contrast. 199
A.17 Oxford Sweep Pareto Front - Curvature. 199
A.18 Oxford Sweep Pareto Front - Intervals. 199
A.19 Oxford Sweep Pareto Front - Octaves. 199
A.20 Oxford Sweep Pareto Front - Match Ratio. 200
A.21 The MBDA set 1 images. 201
A.22 The MBDA set 2 images. 201
A.23 The MBDA set 3 images. 201
A.24 The MBDA set 4 images. 201
A.25 The MBDA set 5 images. 202
A.26 The MBDA set 6 images. 202
A.27 The MBDA set 7 images. 202
A.28 The MBDA set 8 images. 202
A.29 MBDA GA Pareto Front - Sigma. 203
A.30 MBDA GA Pareto Front - Contrast. 203
A.31 MBDA GA Pareto Front - Curvature. 203
A.32 MBDA GA Pareto Front - Intervals. 203
A.33 MBDA GA Pareto Front - Octaves. 204
A.34 MBDA GA Pareto Front - Match Ratio. 204
A.35 MBDA Sweep Best 500 - Sigma. 204
A.36 MBDA Sweep Best 500 - Contrast. 204
A.37 MBDA Sweep Best 500 - Curvature. 205
A.38 MBDA Sweep Best 500 - Intervals. 205
A.39 MBDA Sweep Best 500 - Octaves. 205
A.40 MBDA Sweep Best 500 - Match Ratio. 205
A.41 MBDA Sweep Pareto Front - Sigma. 206
A.42 MBDA Sweep Pareto Front - Contrast. 206
A.43 MBDA Sweep Pareto Front - Curvature. 206
A.44 MBDA Sweep Pareto Front - Intervals. 206
Abstract

This work, entitled *Data Analytics and Methods for Improved Feature Selection and Matching*, was submitted in June 2012 by Michael May to The University of Manchester for the degree of Doctor of Philosophy.

This work focuses on analysing and improving feature detection and matching. After creating an initial framework of study, four main areas of work are researched. These areas make up the main chapters within this thesis and focus on using the Scale Invariant Feature Transform (SIFT).

The preliminary analysis of the SIFT investigates how this algorithm functions. Included is an analysis of the SIFT feature descriptor space and an investigation into the noise properties of the SIFT. It introduces a novel use of the *a contrario* methodology and shows the success of this method as a way of discriminating between images which are likely to contain corresponding regions from images which do not.

Parameter analysis of the SIFT uses both parameter sweeps and genetic algorithms as an intelligent means of setting the SIFT parameters for different image types utilising a GPGPU implementation of SIFT. The results have demonstrated which parameters are more important when optimising the algorithm and the areas within the parameter space to focus on when tuning the values.

A multi-exposure, High Dynamic Range (HDR), fusion features process has been developed where the SIFT image features are matched within high contrast scenes. Bracketed exposure images are analysed and features are extracted and combined from different images to create a set of features which describe a larger dynamic range. They are shown to reduce the effects of noise and artefacts that are introduced when extracting features from HDR images directly and have a superior image matching performance.

The final area is the development of a novel, 3D-based, SIFT weighting technique which utilises the 3D data from a pair of stereo images to cluster and class matched SIFT features. Weightings are applied to the matches based on the 3D properties of the features and how they cluster in order to attempt to discriminate between correct and incorrect matches using the *a contrario* methodology. The results show that the technique provides a method for discriminating between correct and incorrect matches and that the *a contrario* methodology has potential for future investigation as a method for correct feature match prediction.
Declaration

No portion of the work referred to in this thesis has been submitted in support of an application for another degree or qualification of this or any other university or other institute of learning.
Copyright

i. The author of this thesis (including any appendices and/or schedules to this thesis) owns certain copyright or related rights in it (the “Copyright”) and s/he has given The University of Manchester certain rights to use such Copyright, including for administrative purposes.

ii. Copies of this thesis, either in full or in extracts and whether in hard or electronic copy, may be made only in accordance with the Copyright, Designs and Patents Act 1988 (as amended) and regulations issued under it or, where appropriate, in accordance with licensing agreements which the University has from time to time. This page must form part of any such copies made.

iii. The ownership of certain Copyright, patents, designs, trade marks and other intellectual property (the “Intellectual Property”) and any reproductions of copyright works in the thesis, for example graphs and tables (“Reproductions”), which may be described in this thesis, may not be owned by the author and may be owned by third parties. Such Intellectual Property and Reproductions cannot and must not be made available for use without the prior written permission of the owner(s) of the relevant Intellectual Property and/or Reproductions.

iv. Further information on the conditions under which disclosure, publication and commercialisation of this thesis, the Copyright and any Intellectual Property and/or Reproductions described in it may take place is available in the University IP Policy (see http://documents.manchester.ac.uk/DocuInfo.aspx?DocID=487), in any relevant Thesis restriction declarations deposited in the University Library, The University Library’s regulations (see http://www.manchester.ac.uk/library/aboutus/regulations) and in The University’s policy on presentation of Theses.
Acknowledgements

I would like to thank my supervisors, Martin Turner and Tim Morris, for their ideas, help and support throughout this process. Thanks to my fellow PhD students for making my time spent doing this research more interesting and providing me with other areas of research to learn about besides my own; Ben, David, Merna, Sam and Steve.

I appreciate greatly the opportunity provided to me by MBDA, EPSRC and The University of Manchester by providing the funding and academic support for this project. I would particularly like to thank Keith Markham and Graham Wallis for providing time, expertise and criticism of my work throughout the research period.

I would like to thank all my friends for providing much needed distractions and entertainment in the times when I needed a break from work. In particular Adam, Craig, Jimmy, Kate, Liam, Matthew, Sam, Steve and Tom.

I would like to thank my family for always supporting me. Without your time, advice and support I would not be where I am today.

I would like to especially thank Jane for putting up with me for the last three years when I was stressed, grumpy and fed up and I greatly appreciate your help, support and encouragement (and for proof reading my thesis and checking for spelling mistakes - good work).

Manchester, March 2012

Michael May
Publications

The following main publications are based on the work in this thesis:


III and IV were presented at The Fifth Pacific-Rim Symposium on Image and Video Technology 2011 (PSIVT 2011) in Gwangju, South Korea.

The following publications occurred through the CASE studentship in association with MBDA:


Abbreviations and Notation

List of Abbreviations

The most frequent abbreviations are described below:

2D Two Dimensional
3D Three Dimensional
AWF Alignment, Weighting and then Features
BBF Best Bin First
BRIEF Binary Robust Independent Elementary Features
CSIFT Colour SIFT
CUDA Compute Unified Device Architecture
EV Exposure Value
FAST Features from Accelerated Segment Test
FAW Features, Alignment and then Weighting
FEWER Feature Extraction and Weighting for Enhanced Recognition
FLD Fisher’s Linear Discriminant analysis
FLD-SIFT Fisher’s Linear Discriminant analysis SIFT
GA Genetic Algorithm
GDOH Gradient Distance and Orientation Histogram
GLOH Gradient Location and Orientation Histogram
GPGPU General Purpose Graphical Processing Unit
HDR High Dynamic Range
HOG Histogram of Orientated Gradients
HVS Human Visual System
LDR Low Dynamic Range
LESH Local Energy based Shape Histogram
<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>LMA</td>
<td>Levenberg-Marquardt Algorithm</td>
</tr>
<tr>
<td>MSER</td>
<td>Maximally Stable Extremal Regions</td>
</tr>
<tr>
<td>MTB</td>
<td>Median Threshold Bitmap</td>
</tr>
<tr>
<td>ND</td>
<td>Neutral Density (filter)</td>
</tr>
<tr>
<td>ORB</td>
<td>Oriented FAST and Rotated BRIEF</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal Component Analysis</td>
</tr>
<tr>
<td>PSNR</td>
<td>Peak Signal to Noise Ratio</td>
</tr>
<tr>
<td>QMF</td>
<td>Quadrature Mirror Filters</td>
</tr>
<tr>
<td>RANSAC</td>
<td>RANdom SAmple Consensus</td>
</tr>
<tr>
<td>RGB</td>
<td>Red, Green, Blue</td>
</tr>
<tr>
<td>RIFT</td>
<td>Rotation Invariant Feature Transform</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver Operating Characteristic</td>
</tr>
<tr>
<td>ROI</td>
<td>Region Of Interest</td>
</tr>
<tr>
<td>SIFT</td>
<td>Scale Invariant Feature Transform</td>
</tr>
<tr>
<td>SURF</td>
<td>Speeded Up Robust Features</td>
</tr>
<tr>
<td>SUSAN</td>
<td>Smallest Unvalue Segment Assimilating Nucleus</td>
</tr>
<tr>
<td>UAV</td>
<td>Unmanned Air Vehicle</td>
</tr>
</tbody>
</table>
## List of Notation

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$A$</td>
<td>The second moment, or autocorrelation, matrix</td>
</tr>
<tr>
<td>$\lambda_1, \lambda_2$</td>
<td>The eigenvalues of $A$</td>
</tr>
<tr>
<td>$M \otimes N$</td>
<td>The convolution of matrices $M$ and $N$</td>
</tr>
<tr>
<td>$\text{trace}(M)$</td>
<td>The sum of the elements on the main diagonal of $M$</td>
</tr>
<tr>
<td>$\det(M)$</td>
<td>For a $2 \times 2$ matrix $M = \begin{vmatrix} a &amp; b \ c &amp; d \end{vmatrix}$, $\det(M) = ad - bc$</td>
</tr>
<tr>
<td>$M_c$</td>
<td>The Harris measure, $\det(A) - \alpha \text{trace}(A)^2$</td>
</tr>
<tr>
<td>$I(x, y)$</td>
<td>The pixel intensity in image $I$ at $(x, y)$</td>
</tr>
<tr>
<td>$\sigma$</td>
<td>Standard deviation</td>
</tr>
<tr>
<td>$G(x, y, \sigma)$</td>
<td>A Gaussian kernel with a standard deviation of $\sigma$</td>
</tr>
<tr>
<td>$L(x, y)$</td>
<td>A Laplacian operator</td>
</tr>
<tr>
<td>$L(x, y, \sigma)$</td>
<td>An image blurred with Gaussian of standard deviation $\sigma$.</td>
</tr>
<tr>
<td>$\text{LoG}(x, y, \sigma)$</td>
<td>A Laplacian of Gaussian operator</td>
</tr>
<tr>
<td>$D(x, y, \sigma)$</td>
<td>A Difference of Gaussian operator</td>
</tr>
<tr>
<td>$H$</td>
<td>A Hessian matrix of second derivatives</td>
</tr>
<tr>
<td>$M(x, y)$</td>
<td>The magnitude of an image gradient</td>
</tr>
<tr>
<td>$\theta(x, y)$</td>
<td>The orientation of an image gradient</td>
</tr>
<tr>
<td>$D_e(u, v)$</td>
<td>The Euclidean distance between two vectors</td>
</tr>
<tr>
<td>$D_{\text{city}}(u, v)$</td>
<td>The city-block distance between two vectors</td>
</tr>
<tr>
<td>$D_{\text{chess}}(u, v)$</td>
<td>The chessboard distance between two vectors</td>
</tr>
<tr>
<td>$\mathbf{H}_{ab}$</td>
<td>A homography which relates the images $I_a$ and $I_b$</td>
</tr>
<tr>
<td>$\epsilon$</td>
<td>The overlap error for two matched feature patches</td>
</tr>
<tr>
<td>$Z_{ij}$</td>
<td>The value of the pixel from a set of exposure images at image position $i$ and image exposure $j$</td>
</tr>
<tr>
<td>$G(Z_{ij})$</td>
<td>A logarithmic response function derived from a set of exposure images</td>
</tr>
<tr>
<td>$C(x, y)$</td>
<td>The contrast measure</td>
</tr>
<tr>
<td>$S(x, y)$</td>
<td>The saturation measure</td>
</tr>
<tr>
<td>$E(x, y)$</td>
<td>The well-exposedness measure</td>
</tr>
<tr>
<td>$H(p)$</td>
<td>The entropy measure (Shannon’s entropy)</td>
</tr>
<tr>
<td>$W(x, y)$</td>
<td>A weighting for a pixel</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

This thesis concerns the use of features for matching images. It reviews the current state of the art and introduces novel analysis and techniques to improve the detection and matching of image features.

1.1 Overview

Some of the earliest work on image matching is based on global descriptions of an object. This means that the whole image of the object is taken into account and histograms of the global colour or texture are created and matched to other images. The shape of the object has been considered [94, 102, 115] but the main problem with a global descriptor is when the view point is varied even by small amounts or when the object is partially occluded then image matching may not occur.

Further research in this area has led to local descriptors being accepted as a more robust method for object recognition. These describe smaller regions within the image of an object and are matched individually therefore, to classify an object, only a subset of these features need to be seen. There are two parts; the feature detector and the feature descriptor. Feature detectors find the regions within the image upon which a feature descriptor is applied. A detector and descriptor combination must be as invariant to changes as possible to reliably find the same points in different images of the same scene.

Initial work in feature detectors used the Harris and Stephens [44] corner detector but this did not provide sufficient invariance other than for translation,
rotational and illumination [2,147]. After this developments led to feature detectors being used with larger degrees of invariance and combined with novel descriptors [1,10,70,86]. Detectors and their corresponding descriptors are described in detail within the next chapter.

The research interests for this thesis span the areas of object recognition, feature detection, high dynamic range imaging, general purpose graphics processing units (GPGPUs), parameter analysis, 3D reconstruction and structure from motion. The common thread of the research is in understanding the scale invariant feature transform (SIFT) [13,70] and analysing its performance.

1.2 Framework

This thesis addresses the problem of object recognition by improving on current local feature based recognition techniques. A framework has been created (Figure 1.1) which describes a methodology for recognising objects from a moving platform using a combination of scale invariant features, high dynamic range video and stereoscopic depth information generated from motion parallax and stereopsis. The framework indicates how the constituent parts would be combined in an attempt to make an effective system for object recognition and can be broken down into the following research objectives:

1. Create a high dynamic range video stream.

2. Locate scale, rotation, illumination and viewpoint invariant features within each image of the video stream.

3. Generate 3D information from the video stream which will allow clustering of features and differentiation between individual objects within a scene.

4. Generate 3D keypoint information for target objects in a database.

5. Match target objects from the database to the video stream using the features and depth information.

6. Track located objects.

7. Recognise changes in a scene from previous images of the same location.
CHAPTER 1. INTRODUCTION

8. Enable communication and collaboration between multiple mobile platforms.

9. Create a system capable of running in real-time.

Figure 1.1: Framework for the proposed 3D HDR Video SIFT object recognition system. The green boxes highlight the areas that have be focused on in this thesis.

The initial input is a series of LDR images that are used to create one of three options of HDR input; (a) the original LDR images, creating a set of pseudo HDR features, (b) a full HDR image and (c) a tone mapped image created from the HDR image. It should be noted that the input proposed is in the form of a continuous real-time video stream.

The feature detector will generate features which uniquely describe areas within the image and are invariant to scale, viewpoint, rotation and lighting. Currently SIFT is one of the best solutions [86] so for this project a large emphasis has been placed on investigating improving accuracy of this algorithm. Following this the system uses the feature data to generate 3D information from multiple stereo cameras, motion parallax or both. The depth information is important to help cluster the features extracted by highlighting the boundaries between objects. Depth maps of objects will generate more information about their surface and the relative position of features in 3D space. This allows more discriminative keypoint matching and give an indication of the actual size of objects recognised.

There are two possible matching scenarios that we will focus on in this project. The first involves matching the input to a database of features generated from target images. These features must also have depth information associated with them giving a pseudo 3D model of the object. The second scenario involves
matching the input images to previously recorded features from a scene. This will allow differences in the scenes to be highlighted from the last time that the mobile platform was in a specific location. Once an object has been recognised it can be tracked by the system in consecutive frames of the video stream until it leaves the camera’s field of view.

The modularity of the framework allows all the processing to be carried out on board the mobile platform or different parts may be computed in remote locations from the camera. The system also incorporates the ability to share data between multiple mobile platforms. The database therefore can be held centrally or distributed. Processing at the central location would allow for data to be integrated. For example, if two mobile platforms are near each other superior 3D and keypoint information about the scene could be generated by using images from both.

The framework also allows different areas to be researched and developed individually and new modules can be added if required. As new research is completed and new solutions discovered a module can be easily swapped. Modules can also be used for other projects. The system as a whole will utilise multi-threaded processing techniques to help achieve real-time performance. Speed and accuracy in all areas of this framework are important factors to allow real-time responses to recognised objects. The applications for such a system include use in robotics, security and vehicles as all become increasingly independent of the necessity of human supervision and guidance.

1.3 Background and Motivation

1.3.1 The Scale Invariant Feature Transform

Work has focused on testing and adapting the SIFT [70] which is a local image descriptor. A detailed literature review was completed on the SIFT and related work. This highlighted its success when compared to other similar algorithms [86, 87] and as such has been the main focus as the algorithm for image matching throughout this research.

1.3.2 Key Problems Addressed in This Thesis

The main problems that this thesis addresses are as follows:
CHAPTER 1. INTRODUCTION

Figure 1.2: An example of SIFT matching. The features matched between the images are shown by the blue lines.

- The scale invariant feature transform has many parameters that effect the results of detection and matching. It is not well understood how varying these parameters effect the matching of different image types and how best to select the parameters. Thus, parameter analysis and estimation are necessary for the implementation of the SIFT.

- High dynamic range environments pose a problem when matching features as an 8-bit low dynamic range camera often cannot capture all the information in a scene. This means that areas in the image will be under or over exposed and features will not match to such areas.

- False positive matches often occur when matching images due to feature vectors being generated from different areas of images which correspond well enough to be matched mistakenly. Identifying which features matches are correct when matching images is important.

1.4 Methods for Addressing the Problems

The main areas of research in this thesis are as follows:

1.4.1 Preliminary Analysis of the SIFT

This work investigates how the SIFT algorithm functions. This includes an analysis of the SIFT feature descriptor space using millions of features downloaded from Flickr and an investigation into the noise properties of SIFT. This section includes an overview of the noise properties of SIFT matching and introduces a novel use
1.4. METHODS FOR ADDRESSING THE PROBLEMS

of the *a contrario* methodology for detecting images which are likely to match based on the rate of false positive matches which occur for non-corresponding images.

1.4.2 SIFTing Through Parameter Space

This work focuses on optimising the SIFT by adjusting the parameters for specific image types. The parameter space for the SIFT is substantial. There are over twenty variables which effect how the SIFT will respond to an image. This makes selecting the best parameters for matching between a pair of images difficult. Image pairs have been collected from multiple sources including infrared cameras. Parameter sweeps and genetic algorithm based optimisation have been used to explore the parameter space to better understand how different image types effect the algorithm and that the default parameters require adjustment for better matching [80].

![Figure 1.3: An example of the results of the parameter sweep and genetic algorithm. The blue points are the results of the GPU parameter sweep and the pink points are the pareto front generated by a genetic algorithm. The default SIFT parameter values are shown as the green point on the graph.](image)

1.4.3 Multi-exposure, High Dynamic Range, Fusion Features

This work introduces an improved process where fusion features assist matching SIFT image features from high contrast images. FAW defines the preferred order
for extracting features: First Feature extraction, then Alignment of images and finally pixel Weighting. The process uses up to four quality measures to select features from a series of differently exposed images and weights the features in favour of those areas that are defined as well exposed. The results show an advantage in using these features over features extracted from the common alternative techniques of exposure fusion and tone mapping which extract the features as AWF; Alignment of images, Weighting of pixels then extraction Features [79].

Figure 1.4: The top two images are an example of two aligned input images taken at different exposures. The arrows represent the scale, orientation and position of the SIFT features. The bounding box in each shows the areas within which SIFT features have been matched between the images using RANSAC during the alignment process [134]. The bottom two images are the set of fusion features displayed on a binary fusion image on the left and on a rough exposure fusion image on the left. The binary image shows which areas are best exposed in each image.

1.4.4 A Contrario SIFT Feature Weighting

This work focuses on object detection and feature match weighting utilising stereoscopic image pairs, the scale invariant feature transform and 3D reconstruction. The object detection technique is based on a contrario noise subtraction utilising false positive matches from random features. The feature weighting process utilises 3D spatial information generated from the stereoscopic pairs. The features are divided into three different types, matched from the target to the scene and weighted based on their 3D data and spatial clustering properties. The weightings are computed by analysing a large number of false positive matches.
1.5. CONTRIBUTIONS OF THIS THESIS

The techniques described reduces the occurrence of false positives and can create a reduced set of highly relevant features [78].

Figure 1.5: A typical example of weighted feature matching displaying matches from the left hand target image to the left scene image. Some of the correct matches are green and red indicating higher weightings. The mismatched features in this scene have received low weightings and are coloured blue. The feature matches with low weightings can be removed by adjusting the weighting threshold which is set at 0 in these cases. The graph below the image shows the weightings for each of the 33 matched features and whether they match correctly.

1.5 Contributions of this Thesis

The main contributions of this thesis are as follows:

1. A study of the SIFT feature space and the introduction of the correspondence ratio in conjunction with the a contrario methodology.

2. A framework for investigating the SIFT parameter space and an analysis of sample data.

3. A method for extracting a HDR set of features from the multiple exposure images.

4. An a contrario method for weighting feature matches from stereo image pairs.
1.6 Structure of this Thesis

The thesis is structured as follows:

1. **Introduction.** An overview of the problems that this thesis aims to solve and introduction to the chapters contained within the thesis.

2. **Overview of Feature Detectors and Descriptors.** Outline the context within which the work is focused and a literature review of the relevant areas related to image features.

3. **Overview of Other Related Work.** A literature review focusing on the other relevant areas of work related to this thesis including GPGPUs, HDR and 3D reconstruction.

4. **Preliminary Analysis of the SIFT.** Areas of research conducted relating to the following chapters and the background of the SIFT’s functionality.

5. **SIFTing through Parameter Space.** Outline the work conducted on the analysis of the SIFT parameter space.

6. **Multi-exposure, High Dynamic Range, Fusion Features.** Outline the work conducted utilising HDR imagery for improved SIFT matching.

7. **A contrario SIFT Feature Weighting** Outline the work conducted utilising stereoscopic imagery to weight SIFT matches.

8. **Conclusion** Summary of the work conducted, the contributions made, the conclusions reached and the areas for further work.
Chapter 2

Feature Detectors and Descriptors

This chapter provides background material to the work that has been conducted relating to feature detectors and descriptors.

2.1 Feature Detectors

There are a large selection of methods for detecting repeatable features based on the pixel properties. In order to describe a patch around a point using a descriptor one must first have a way of reliably generating the points such that they are still localised after changes in lighting, viewpoint, scale, blur and rotation. The following section outlines a number of notable techniques which have been used to implement a higher level feature detector. It demonstrates other possible techniques that could be utilised for feature detection before introducing the focus of this thesis, the scale invariant feature transform, in the next section.

The detectors can be grouped into two main types, both of which are widely used:

1. **Edge Based.** A corner is a point where two edges of different orientations meet. These techniques generally utilise a detector that responds to edges and from the identified edges selects corners as feature candidates.

2. **Region Based.** These methods rely on the observation that a corner is a region where one quadrant of a region is of one type, and the remaining
three quadrants are of a different type. This indicates that the image patch is a corner.

2.1.1 Harris and Stephens / Plessey

The edge based corner detector generally referred to as the Harris corner detector [44] is one of the most commonly used. It is invariant to translation, rotation and illumination and it is based on the principle that at a corner the gradient will be high in multiple directions. A window on a corner point should have a change in intensity when moved in any direction over a corner point which is not the case with an edge or a planar region. The algorithm passes a window over the image and looks for areas where the gradient changes substantially in more than one direction. The Harris detector uses two scales, the integration scale that is used in the weighting, and the differentiation scale used when computing the derivatives, but in what follows a single concept of scale will be used as the two scales are usually related by a simple factor.

The Harris detector uses the second moment matrix as the basis of its corner decisions. The matrix $A$ is the second moment matrix, also called the autocorrelation matrix, has values closely related to the derivatives of image intensity. For each pixel, and the surrounding patch $(u, v)$ in the image $I$, the autocorrelation matrix is calculated:

$$A = \sum_{u,v} W(u,v) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix}$$

(2.1)

where $I_x$ and $I_y$ are the derivatives of pixel intensity in the $x$ and $y$ directions. The off-diagonal entries are the product of $I_x$ and $I_y$, while the diagonal entries are squares of the respective derivatives. A Gaussian weighting, $W(u,v)$, can be used to emphasise the values at the centre of the window. The eigenvalues of $A$; $\lambda_1$ and $\lambda_2$, are not explicitly calculated but if both are large than the location is a corner. The Harris measure based on the trace and determinant is used where $\alpha$ is an empirically determined constant:

$$M_c = \det(A) - \alpha \text{trace}(A)^2 = \lambda_1 \lambda_2 - \alpha(\lambda_1 + \lambda_2)^2$$

(2.2)

$M_c$, the Harris measure, indicates that a location is a corner, as a matrix with large positive eigenvalues will also return a large value from this equation.
2.1. FEATURE DETECTORS

2.1.2 Shi and Tomasi

Shi and Tomasi [122] is a variant of the Harris and Stephens corner detector. Instead of using \( M_c = \lambda_1 \lambda_2 - \alpha (\lambda_1 + \lambda_2)^2 \), \( \min(\lambda_1, \lambda_2) \) is used which is shown to be a better measure for finding the same corner under affine transformations. This method is more computationally expensive than the Harris corner detection algorithm because it directly calculates the eigenvalues of the autocorrelation matrix, \( A \).

2.1.3 Laplacian of Gaussian

The Laplacian of Gaussian (LoG) operator calculates the second spatial derivative of an image. It can be computed using a convolution filter across an image to create a new image which highlights edges. In areas where the image has a constant intensity, the LoG response will be zero. Marr and Hildreth [76] used this method to detect edges. This is advantageous over a first derivative function as the zero-crossing means a global edge threshold is not required.

An image is convolved with a Gaussian kernel at a specified scale \( \sigma \).

\[
G(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2 + y^2}{2\sigma^2}}
\]  

(2.3)

The result is convolved with a Laplacian operator, \( L(x, y) \), which is a differential operator defined as follows where \( I(x, y) \) is the pixel intensity at a point in an image:

\[
L(x, y) = \frac{\partial^2 I}{\partial x^2} + \frac{\partial^2 I}{\partial y^2}
\]  

(2.4)

The final LoG equation is then given as:

\[
\text{LoG}(x, y, \sigma) = G(x, y, \sigma) \otimes L(x, y)
\]  

(2.5)

Choosing \( \sigma \) effects the scale of the features that are detected as a larger value will result in more high frequency data being smoothed. The process can be done at multiple scales to extract different sized features in an image.
2.1.4 Harris-Laplace

The Harris-Laplace detector, developed by Mikolajczyk and Schmid [84], localises points in scale-space using a scale-adapted Harris function making the Harris detector scale invariant. It selects the points for which the Laplacian of Gaussian (LoG) attains a maximum over scale as the Harris detector alone cannot find optima in three dimensions. The LoG response detects the extremum over scale and the Harris measure detects the corner in two dimensions at that scale.

It starts by blurring the image with varying scales of Gaussian kernel and calculating the LoG response. The local maxima are found in the neighbourhood of the point, its 26 surrounding pixels, based on LoG response. If the LoG response attains no extremum or the responses are below a threshold the point is rejected.

For each feature at a scale, the maximum of the Harris measure for the pixel in an $8 \times 8$ window surrounding a feature is selected as the new location feature. The two steps, LoG and Harris measure, are repeated until the feature is stable and does not change position or until another feature has converged at the same location and scale.

The result of the Harris-Laplace is a single or reduced set of features which correspond to a single local structure in the image. If the same point was represented by the scale adapted Harris it would consist of multiple features at different differentiation scales (Figure 2.1). The scale invariance of this means that features generated using Hessian-Laplace from scaled images will always generate a very similar single feature to represent an image characteristic and as such it is scale invariant.

Figure 2.1: The left is a set of scale-adapted Harris features found at different scales. The right shows the final Harris-Laplace feature generated from them. Image from [84]
2.1.5 Harris-Affine

This was also developed by Mikolajczyk and Schmid [84] to increase the affine invariance of Harris features. The initial features are detected using the Harris-Laplace method and then transformed via affine transformation to a normalised form. Two features extracted from the same characteristic in the same scene at different angles will be transformed so that they are as similar as possible in terms of position and scale. An iterative algorithm is applied to warp the shape of the patch such that the centre point of the patch that it represents is always in the same place given changes of viewpoint of the scene. The steps of the algorithm are as follows:

1. Locate features using the Harris-Laplace detector.

2. For each initial point, normalise the region to be affine invariant using affine shape adaptation. Affine invariance can be accomplished from measurements of the multi-scale windowed second moment matrix.

3. Iteratively estimate the affine region: selection of proper integration scale, differentiation scale and spatially localise interest points.

4. Update the affine region using the newly calculated scales and spatial localisations.

5. Repeat steps 3 and 4 until the difference between the two successive second-moment matrices is sufficiently small.

2.1.6 Difference of Gaussian

This is the method used in Lowe’s Scale Invariant Feature Transform (SIFT) [68, 70] algorithm for locating points to be encoded using the descriptor. It is an approximation of the LoG method and designed to be faster. The image is blurred with a Gaussian kernel at different scales and subtracted to create octaves before locating extrema in three dimensional scale space are selected as possible features. For the SIFT, their sub-pixel locations are estimated and they can be rejected based on their contrast and localisation along edges. This is explained in further detail in Section 2.2.2.
2.1.7 Hessian-Laplace and Hessian-Affine

Hessian-Laplace and Hessian-Affine [83] are the same as the Harris-Laplace and Harris-Affine detectors respectively except the initial points are detected using a scale adapted Hessian detector. The scale adaptation is added by blurring the image at different scales before applying the Hessian detector.

2.1.8 Maximally Stable Extremal Regions

Maximally stable extremal regions (MSERs) [77] is a feature detection technique which selects features that are invariant to affine and photometric changes. The regions are defined solely by an extremal property of the intensity function in the region and on its outer boundary.

For an image \( I \) thresholded over \( t \), each pixel will be either black, below the threshold, or white, above or equal to the threshold. As \( t \) is varied from the minimum to the maximum the image will gradually change from white to black. Black spots corresponding to local minima will appear to grow until finally the image appears completely black. The term extremal region means that all the surrounding pixels have a higher or lower intensity than the region i.e. it is an extrema.

For a sequence images created over the full set of thresholds, nested extremal regions are found. Nested extremal regions are such that the region in the previous image in the sequence is a subset of a region in the next. The set of all connected components in the sequence is the set of all extremal regions. An extremal region is deemed to be maximally stable where the change in the region size is smallest over a subset of thresholded images.

The regions generated can be used to position descriptors and testing has shown that the use of MSERs are successful for the generation of repeatable and reliable regions across changes in viewpoint, lighting and scale but fails to perform well with image blur [85].

2.1.9 Smallest Univalence Segment Assimilating Nucleus

Smallest Univalence Segment Assimilating Nucleus (SUSAN) [124] is a corner detection algorithm. Each image pixel is used as the central point, or nucleus, of a small circular mask. Figure 2.2 shows an example of the masks in different locations of an image.
A check is carried out to compare the pixel values of the nucleus with the surrounding areas within the circle. Pixels with values close to that of the nucleus are assumed to be of the same object/surface. Pixels with values close to the centre value are labelled white and the other black. This white region is the univalve segment assimilating nucleus or USAN. The smallest USAN for an area is found using non-maximal suppression. A corner will have the centroid, the centre of USAN region, far from the nucleus and all points on the line from the nucleus through the centroid out to the edge of the mask are in the USAN area. If these requirements are met the exact corner location can be found by calculating the local minimum in the USAN area.

2.1.10 Features from Accelerated Segment Test

Features from Accelerated Segment Test (FAST) [110, 111] uses a Bresenham midpoint circle algorithm of radius three around each pixel in an image. The algorithm determines which pixels at radius three relate to the border of a circle. It is a more efficient version of the SUSAN corner detector.

The segment test criterion operates by considering a circle of sixteen pixels around the corner candidate. It classifies the point as a corner if there exists a set of nine contiguous pixels in the circle which are all brighter or darker than the intensity of the centre candidate pixel by a set threshold.
CHAPTER 2. FEATURE DETECTORS AND DESCRIPTORS

2.1.11 CenSurE and STAR

The CenSurE or STAR feature detector is designed with the purpose of achieving a full spatial resolution in a multiscale detector. STAR is based on a Centre Surround Extrema (CenSurE) feature detector and has been created for the Open Computer Vision Library (OpenCV). The CenSurE paper [3] compares the detector to SURF and the SIFT which both perform subsampling causing the accuracy of the feature localisation to be affected. CenSurE uses a bi-level approximation of the ‘Mexican-hat’ shape of the Laplacian used in the Laplacian of Gaussian method. The difference between the STAR and CenSurE filters are shown in Figure 2.3.

Figure 2.3: The filters used for CenSurE (left) and STAR (right). The STAR filter is made from two squares, one rotated by 45 degrees, to approximate a circle. Image from [3].

The process has three stages. The primary stage is to calculate the response to a simplified bilevel Laplacian of Gaussian and filter week responses leading to detected edges. The algorithm calculates the sum of the inner area of the kernel and the outer area of the kernel and computes the difference. The size of the filter controls the scale of the features that are detected and this is computed at multiple scales.

In the second step, the local extrema are detected in the 3x3x3 area surrounding the pixel and points which are not local maxima are suppressed. The final step, uses the Harris measure to detect if the local extrema has a strong corner response.

2.2 Feature Descriptors

This section describes the algorithms used to encode information about a point found using a feature detector so that it can be matched to other features. The use of independent point or region based descriptors means that matching can still occur if the object is semi-occluded, as the entire object does not need to be
considered for a match, only the features that are in view. The SIFT is outlined in greater detail (including the feature detection stage) as this is the key area of focus of this thesis.

2.2.1 Correlation Based Similarity Measures

This defines one of the most basic descriptors. A square image patch of a set size is centred on a feature location and the values are directly compared to other image patches of the same size. Comparisons between patches use various methods to generate a similarity score which can be thresholded to decide if the match is valid. These include sum of the absolute differences (SAD), sum of squared differences (SSD) and normalized cross-correlation (NCC). NCC is a technique for situations in which the brightness of the image can vary due to lighting and exposure conditions. The images can also be first normalised which is generally done by subtracting the mean and dividing by the standard deviation.

2.2.2 Scale Invariant Feature Transform

The original SIFT algorithm by David Lowe [68, 70] is a four stage process that creates unique and highly descriptive features from an image. These features are invariant to scale and rotation and robust to changes in illumination, noise as well as small changes in viewpoints. This transform is the central focus of this thesis.

The features can be used to indicate if there is any correspondence between areas within images. Clusters of features from an image that are similar to a cluster of features from another indicate, with a high likelihood, matches. This allows object recognition to be implemented by comparing features generated from input images to features generated from images of known target objects. The four stages of the SIFT algorithm are as follows:

Scale-space Extrema Detection

The first step is to create the Gaussian scale-space pyramid. Blurred images, \( L(x, y, k\sigma) \), are produced from the convolution of Gaussian functions, \( G(x, y, k\sigma) \) with an input image, \( I(x, y) \). \( \sigma \) is the scale and \( k \) is a constant factor separating each image. The effect of this is to create an octave; a column of identically sized images blurred to different degrees. The second image from the top of the
The first octave is then subsampled, by taking every second value from each row and column. This becomes the first image in the next octave. The blurring process is then repeated using this image to produce the second octave. Further octaves are created in the same way until the images in an octave become smaller than the blurring filter.

The Gaussian filters, \( G(x, y, \sigma) \), are used such that the images created are separated by a constant factor, \( k \), in each octave. Each octave has a \( \sigma \) value double that of the previous one. An integer, \( s \), is chosen that indicates the value of \( k \) using the formula \( k = 2^{1/s} \). A value of 3 for \( s \) was recommended by [70] for producing the most stable features. A total of \( s + 3 \) scale-space images are produced in each octave.

![Figure 2.4: The Difference of Gaussian (DoG) technique used to extract an initial set of features at different scales. Image from [70].](image)

The Difference of Gaussian (DoG), \( D(x, y, \sigma) \), is calculated as the difference between two consecutive images within an octave. The output from each octave is \( s + 2 \) DoG images.

The initial set of candidate features are selected by comparing each point in the DoG images to its 26 neighbours. The eight values directly surrounding each point at its scale and the nine in the scales directly above and below it are compared. If it is the maximum or minimum value in its neighbourhood its location is stored as a candidate feature.
2.2. FEATURE DESCRIPTORS

Figure 2.5: The scale space extrema are selected by comparing each pixel with its 26 neighbours in the DoG pyramid. Image from [70].

**Feature Localisation and Rejection**

Interpolation occurs to locate the exact, sub-pixel, location of the candidate features before eliminating the points that are in areas of low contrast and those that have a high edge response.

The sub-pixel localisation uses a second order Taylor series expansion of the Difference-of-Gaussian scale-space function, $D(x, y, \sigma)$, with its origin at the candidate feature location to estimate a more accurate location. This gives an improvement for the stability and matching of the final features. Using the Taylor expansion with a sample point as the origin, where $\mathbf{x} = (x, y, \sigma)^T$ is the offset from this point.

$$D(\mathbf{x}) = D + \frac{\partial D^T}{\partial \mathbf{x}} \mathbf{x} + \frac{1}{2} \mathbf{x}^T \frac{\partial^2 D}{\partial \mathbf{x}^2} \mathbf{x}$$  \hspace{1cm} (2.6)

Taking the derivative with respect to $\mathbf{x}$, and setting it to 0, gives the equation below.

$$\dot{\mathbf{x}} = -\frac{\partial^2 D^{-1}}{\partial \mathbf{x}^2} \frac{\partial D}{\partial \mathbf{x}}$$  \hspace{1cm} (2.7)

The location of the extremum, $\dot{\mathbf{x}}$, within the DoG scale-space, $D(x, y, \sigma)$, is determined using this. The Hessian ($\frac{\partial^2 D}{\partial \mathbf{x}^2}$) and the derivative ($\frac{\partial D}{\partial \mathbf{x}}$) of $D$ are approximated using the difference in neighbouring sample points and the resulting $3 \times 3$ linear system which can easily be solved. It returns an offset from the original position and if it is greater than 0.5 in any direction a new candidate feature is selected and the process is repeated. The offset (if $\leq 0.5$) is then added to the
candidates location for a better estimate of the extremum’s location.

Figure 2.6: An example of an interpolated sub-pixel location. The red crosses are pixel locations but by interpolation it can be seen that the actual maximum of the curve, the green cross, is located between these pixel locations. By using the sub-pixel location rather than the pixel location it has been shown that the features are more stable.

Low contrast points are rejected next. They are unstable because the low contrast means there is a higher chance that the extremum was detected as a result of image noise and may not be visible in other images of the same scene. $D(\hat{x})$ is used and any extremum with a value of $|D(\hat{x})|$ less than 0.03 is rejected as a possible feature.

$$D(\hat{x}) = D + \frac{1}{2} \frac{\partial D^T}{\partial \hat{x}} \hat{x}$$

The next step is to remove the features which are localised on edges. Points on edges often do not have enough information to uniquely encode their position as two points from different locations along the same edge can result in very similar features. This means that points with changes in gradient in more than one direction, corners, are required. To calculate whether the feature is located on an edge or a corner the principal curvatures are computed at the location and the scale of each feature is calculated using a $2 \times 2$ Hessian matrix, $H$.

$$H = \begin{bmatrix} \frac{\partial^2 D}{\partial x^2} & \frac{\partial^2 D}{\partial x \partial y} \\ \frac{\partial^2 D}{\partial x \partial y} & \frac{\partial^2 D}{\partial y^2} \end{bmatrix}$$

The derivatives are estimated by taking differences of neighbouring pixel values. The ratio between the eigenvalues of $H$ are used to eliminate edges. The eigenvalues of $H$ are proportional to the principal curvature of the derivatives.
The sum of the eigenvalues can be computed from the trace of $H$ and their product from their determinant. The technique is based on the Harris corner detector [44]. Given the two eigenvalues of $H$, $\alpha$ the larger one and $\beta$ the smaller one, their sum and product can be calculated without determining the actual eigenvalues:

$$\text{trace}(H) = \alpha + \beta$$  \hspace{1cm} (2.10)  \\
$$\text{det}(H) = \alpha \beta$$  \hspace{1cm} (2.11)

If the determinant is negative then the curvatures have different signs and the point is therefore not an extremum and is discarded. The ratio between the largest and smallest eigenvalues is called $r$ and its value is such that $\alpha = r \beta$. Therefore, by substitution, the following is true;

$$\frac{\text{trace}(H)^2}{\text{det}(H)} = \frac{(\alpha + \beta)^2}{\alpha \beta} = \frac{(r \beta + \beta)^2}{r \beta^2} = \frac{(r + 1)^2}{r}$$  \hspace{1cm} (2.12)

The value $(r + 1)^2/r$ is smallest when the two eigenvalues are equal. The ratio of the principal curvatures can be computed without calculating the exact eigenvalues and points dismissed if they are below a threshold, $r_{th}$, using the following equation, the result of which is very efficient to compute:

$$\frac{\text{trace}(H)^2}{\text{det}(H)} < \frac{(r_{th} + 1)^2}{r_{th}}$$  \hspace{1cm} (2.13)

Lowe uses 10 as the default value of $r_{th}$ within his experiments [70].

**Orientation Assignment**

This stage calculates one or more orientation for each feature. This adds the rotational invariance to the descriptor and is calculated using the gradient directions in a feature’s neighbourhood using a Gaussian weighting. The following equations are used to calculate the gradient magnitude $M(x, y)$ and orientation $\theta(x, y)$ in the scale-space Gaussian blurred image, $L$, for each feature.

$$M(x, y) = \sqrt{(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2}$$  \hspace{1cm} (2.14)
\[ \theta (x, y) = \tan^{-1} \left( \frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)} \right) \] (2.15)

The orientation assignment uses a 36 bin histogram, where each bin represents 10 degrees. Each orientation within the point’s neighbourhood is calculated and added to the histogram. The values are weighted by the magnitude and by a Gaussian-weighted circular kernel with a \( \sigma \) that is 1.5 times that of the scale of the feature.

The highest peaks in the orientation histogram are selected as the orientation for the descriptor. Any peaks which are eighty percent of the highest peak are selected as a second orientation for which a feature is also generated in the next stage. A curve is fit to the 3 histogram values closest to each peak to interpolate the peak position for a more accurate measure of its orientation.

![Figure 2.7: An example of the SIFT orientation histogram stage. The dominant orientation can be estimated by binning the gradient direction weighted by the magnitude across 36 orientation bins. A second feature is generated if there is another peak that is 80 percent of the largest peak.](image)

Creating the Feature Descriptor

The final feature descriptor is a 128 dimensional vector which describes the gradient directions in the feature’s neighbourhood. It is created using a \( 4 \times 4 \) array
of 16 histograms centred on the original feature and rotated to match the orientation calculated in the previous step. The gradient magnitude and direction are calculated for each point in the region and the magnitude is weighted again by a Gaussian-weighted circular kernel with a \( \sigma \) that is 1.5 times that of the scale of the feature.

![Feature descriptor grid](image)

Figure 2.8: A \( 2 \times 2 \) example of the feature descriptor grid used in the SIFT. The left image shows the image gradients and the right side shows the related histogram bins. Image from [70].

For each square in the \( 4 \times 4 \) array an 8 bin histogram is generated, each bin representing an angle of 45 degrees (Figure 2.10). The magnitude values are added to the histograms based on which orientation they are closest to using a linear weighting. A value half way between two orientations will have half its value added to both bins.

![Histograms](image)

Figure 2.9: This shows how the image gradients relate to a single set of eight histogram bins. Each histogram is calculated from gradient orientations weighted by their magnitudes. The values are added anti-clockwise starting in the \( x \) direction. Image from [70].

The histograms are then normalised by dividing all bin heights by the maximum height occurring in the whole descriptor reducing the affect of brightness.
or contrast on the descriptor. The influence of large gradients in the descriptor is reduced by limiting the maximum height to 0.2, and then renormalising again to bring back to unit length. The values are stored in a vector starting at the top left histogram row by row to the bottom right with the indices for the orientations as shown in Figure 2.10.

![Figure 2.10: The $4 \times 4$ SIFT descriptor. The orientations in each circle represent a vector value, with a total of 128. The order in which the values are stored in the vector is given by the indices on the arrows. Lowe’s SIFT implementation convention is different to standard convention: The $y$ axis points upwards and the angles are measured counter-clockwise.](image)

2.3 Variants of the SIFT Algorithm

There are many variations of the SIFT algorithm and the following section lists some of the key versions which have been developed.

2.3.1 GLOH

The GLOH (Gradient Location and Orientation Histogram) [86] feature descriptor uses the same first three steps as the SIFT but differs in the final stage of the algorithm. The final stage uses an altered location grid and uses PCA to reduce its dimensionality in an attempt to make it more robust and distinctive.
A log-polar location grid is used with 17 bins, 3 bins in the radial direction and 8 in the angular direction (Figure 2.11). The gradient orientations are calculated using 16 bins in a $4 \times 4$ array. The result is a 272 bin histogram which is reduced, using PCA, to a 128 dimensional vector.

![Figure 2.11: The log-polar location grid used by GLOH. Image from [86].](image)

### 2.3.2 SURF

SURF (Speeded Up Robust Features) implements a similar algorithm to the SIFT but reduces the time taken by simplifying and approximating the steps. Box filters (a Gaussian second order derivative approximation) and integral images are used to create the scale space pyramid. All layers of the pyramid can be generated from the original image by up-scaling the filter size rather than taking the output from a previous filtered layer which is significantly faster and independent of the previous filter size. To localise interest points non-maximal suppression is used at each scale with a $3 \times 3 \times 3$ neighbourhood. The maxima are then interpolated to determine their exact position.

Orientation assignment occurs by calculating Haar-wavelet responses in $x$ and $y$ directions in a circular neighbourhood around the feature with a radius six times the scale of the feature. The results are weighted with a Gaussian centred at the feature. The dominant direction is estimated calculating the sum of all responses within a sliding orientation window covering an angle of $\pi/3$ and selecting the direction with the largest value.

The feature descriptor is created by centring and orientating a square region, with a size twenty times the scale, on a feature. The region is split into $4 \times 4$, equal sized, sub-regions and within each Haar-wavelet responses are calculated in
both \( x \) and \( y \) directions at \( 5 \times 5 \) regularly spaced points. The responses are then summed over each sub-region and the results are used as the first values in the vector. The absolute values for the responses are summed for each region and added to the vector. The final vector has 64 dimensions.

### 2.3.3 Colour SIFT

Colour SIFT (CSIFT) [1] is an attempt to incorporate colour information into the SIFT algorithm to further increase the distinctiveness of features; Figure 2.12.

![Figure 2.12: An example of the issue that can arise without the use of colour information. Note how the highlighted regions appear to be the same without colour information and when shown in colour they can be distinguished from each other. Image from [1].](image)

The process of adding colour to the SIFT descriptor requires a method for storing colour in a way invariant to scene changes. The colour invariance model was developed by Geusebroek et.al [41] and the paper describes ways of calculating different invariants from RGB images where \( E \) is the reflected spectrum in the viewing direction \( x \) is a 2D vector which defines the position in the image and \( \lambda \) is the wavelength of the reflected light.

- \( H = \frac{E_x}{E_{\lambda\lambda}} \) is the reflectance property which is independent of viewpoint, surface orientation, illumination direction, intensity, and Fresnel reflectance coefficient.

- \( C_\lambda = \frac{E_\lambda}{E} \) is given as an invariant to the viewpoint, surface orientation, illumination direction and illumination intensity.

- \( W_x = \frac{E_x}{E} \) is given as an invariant to the changes in the illumination intensity.
2.3. VARIANTS OF THE SIFT ALGORITHM

- \( N_{\lambda_x} = \frac{E_x E - E_y E_z}{E_x^2} \) is a point’s reflectance property that is independent of the viewpoint, surface orientation, illumination direction, illumination intensity, and illumination colour.

The invariants can be calculated from an RGB image using the equation below which converts the values to a Gaussian colour model. The resulting values can be substituted into the invariant equations described above at a given scale of the Gaussian colour model.

\[
\begin{bmatrix}
\hat{E} \\
\hat{E}_x \\
\hat{E}_{\lambda} \\
\hat{E}_{\lambda\lambda}
\end{bmatrix} =
\begin{bmatrix}
0.06 & 0.63 & 0.27 \\
0.3 & 0.04 & -0.35 \\
0.34 & -0.6 & 0.17
\end{bmatrix}
\begin{bmatrix}
R \\
G \\
B
\end{bmatrix}
\] (2.16)

The CSIFT descriptor is built in the same way as the standard SIFT descriptor but instead the gradients of the invariant values are used. The experimentation within the paper uses the \( H \) invariant to generate the features on images where the lighting has been varied. The results show that the CSIFT features are more consistent between the images and the paper states that there is a 1.5 times improvement in matching performance between the images where lighting has been varied. The data presented and test results shown are limited but promising.

2.3.4 PCA-SIFT

PCA-SIFT (Principal Component Analysis SIFT) uses PCA to implement a different final stage of the SIFT algorithm. PCA is used to reduce the dimensionality of the classifier leaving only the most descriptive values making it more compact. A \( 41 \times 41 \) patch is centred at the feature and aligned with the orientation calculated in step three of the SIFT algorithm. The gradient is calculated in both the \( x \) and \( y \) direction for each point in the patch and the values concatenated to form a vector. This results in a 3042 dimensional vector \((2 \times 39 \times 39)\) and used as the input for the PCA which reduces the vector to 36 dimensions. The paper shows results of experiments where increased matching accuracy is demonstrated although other papers [86,87] demonstrate this is generally not the case in terms of overall performance.
2.3.5 SIFT with Irregular Orientation Histogram Binning

SIFT with Irregular Orientation Histogram Binning [23] claims to improve the accuracy of the SIFT descriptor with respect to changes in scale by altering the histogram sampling strategy used to generate the final vector. A square irregular grid is introduced that reduces the effect that the difference in scale between two identical areas has on matching. The descriptor vector is calculated by sampling a square region after the orientation has been resolved in the same manner as the original SIFT algorithm. The sampling strategy removes the need for the Gaussian weighting as it inherently gives a higher weighting to the central area of the patch. It samples regions from the outside edge inwards always including the centre of the region increasing the impact of the centre pixels (Figure 2.13). The paper claims that this method gives an increase in accuracy with respect to changes to scale with no significant computational cost.

![Image: The sampling strategy used by SIFT with Irregular Orientation Histogram Binning. All regions extend to the centre of the feature. The inner region is consequently sampled several times by different bins. Image from [23].]

The reason that the paper gives for the increased performance is that the overlap, the area that remains overlapping between descriptor bins, reduces through scaling for the SIFT more substantially than it does for the proposed descriptor. The increased overlap will reduce the Euclidean distance between the vector produced at different scales.

2.3.6 DAISY

The DAISY [132,133] descriptor has been designed for the purpose of dense wide baseline matching between images. It is inspired by the SIFT and GLOH but
can be computed faster and attempts to maintain the robustness of the sparser methods. Unlike SURF, which can also be computed efficiently at every pixel, it does not introduce artefacts that degrade the matching performance when used densely.

The algorithm calculates orientation maps for an image in $H$ directions. The default values of $H$ is 8 which results in orientations calculated across the grid in each of the cardinal and ordinal compass directions. The $H$ orientation maps are then convolved several times with Gaussian kernels to obtain convolved orientation maps. To obtain higher levels of smoothing, large Gaussian kernels are obtained by consecutive convolutions with smaller kernels. The number of convolved orientation layers is controlled by the variable $Q$ and the default is 3.

The descriptor is calculated at each pixel using the sampling strategy shown in Figure 2.14. Each orientation and scale is selected from the convolved orientation maps which allows each histogram bin to be filled quickly. For the default descriptor the vector contains 264 values.

Figure 2.14: The sampling strategy used by DAISY [132, 133]. The different regions are sampled from the convolved orientation map at the appropriate scale in all directions to create a 264 dimensional vector. Image from [133].
2.3.7 Other Notable Descriptor Techniques

There are many other feature detectors created either for speed benefits or for other properties which allow them to perform well under various circumstances.

BRIEF [15] is a feature descriptor which uses binary strings to encode the feature patches. It takes a smoothed image patch and computes the result of a binary test where a pixel is chosen within the patch and its intensity compared to another. If the value is greater than the result is 0 if not then it is 1. The locations can be randomly selected or user set and then the values can be compared to another image patch. The feature is defined as a vector of $n$ binary tests where $n$ is empirically defined as 256 in the paper. This is a very simple and fast descriptor.

ORB (Oriented FAST and Rotated BRIEF) [112] is based on BRIEF and is rotation invariant. It uses the BRIEF technique and improves on it through the calculation of an intensity centroid. This assumes that the centroid of a corner’s intensity is offset from its centre, and this vector may be used to calculate an orientation and add invariance.

LESH (Local Energy based Shape Histogram) [113] is a feature detector and descriptor originally introduced for the purpose of facial pose estimation. It uses a local energy model and postulates that features are perceived at points in an image where the local frequency components are maximally in phase. It is used to acquire a description of the underlying shape of a feature and encodes this shape by accumulating local energy of the underlying signal along several filter orientations. Local histograms from different parts of the image patch are generated and concatenated together into a 128-dimensional vector.

HOG (Histogram of Orientated Gradients) [25] starts by calculating gradients across an image using a 1D centred, point discrete derivative mask in one or both of the horizontal and vertical directions. A histogram binning strategy is then used in a similar way to the SIFT or GLOH. An image patch is divided into a set of cells, rectangular or radial, and for each cell a histogram of gradient directions is calculated. These histograms represents the descriptor vector. The blocks are normalised for variation in lighting. This is generally used for dense feature matching rather than the SIFT like techniques which only encode sparsely detected features and are more complicated and consume more computational resources.
2.3. VARIANTS OF THE SIFT ALGORITHM

SIFT Adaptations

GDOH (Gradient Distance and Orientation Histogram) [156] is based on the SIFT and is invariant to image scale, rotation, illumination and partial viewpoint changes. The purpose of GDOH is to reduce the dimensional size of the descriptor while maintaining the distinctness and robustness of the SIFT. It uses the same detection method as the SIFT for localisation, scale and the dominant orientation for each keypoint. The changes are in the final stage where the histograms are binned. Gaussian weighed concentric rings at set distances from the centre of the features are used as distance bins and for each 8 orientation bins over 360 degrees are calculated from the gradient magnitude and direction of the pixel intensities. Each gradient point to the descriptor centre is calculated to determine which distance bin it is in. Bilinear interpolation is used to alleviate boundary effects between bins. For 8 distance bins and 8 orientation bins the descriptor will be 64 dimensional.

A-SIFT (Affine-SIFT) [90] is a framework for generating fully affine invariant features by warping the angle of the image patch and using a novel sampling strategy. The strategy has been calculated to do this efficiently and cover the required views with as few samples as possible. Two extra parameters control latitude and longitude allow the algorithm to simulate all distortions caused by a variation of the camera optical axis direction. A two-resolution scheme reduces the A-SIFT complexity to about twice that of the SIFT. The algorithm uses the same SIFT descriptor technique and generates more feature descriptors for each feature location. The technique has been shown to increase the affine invariance and has been shown to outperforms the SIFT and MSER for large changes in viewing angle.

MI-SIFT [73] is the mirror and inversion invariant generalisation for the SIFT descriptor. This uses the same technique as the standard but includes extra invariance to mirror images and greyscale inverted images. The paper outlines the process to perform a flip operation and invert operation and a flip-invert operation of the descriptor after an initial standard SIFT vector has been extracted.

FLD-SIFT [121] uses Fisher’s linear discriminant analysis (FLD) to compactly represent image patches. This is similar to principal component analysis and allows high dimensional data to be mapped to lower dimensions. In the first stage, the information associated with keypoints is extracted using the standard SIFT technique. In the second stage, Fisherspace is built using the $39 \times 39$
image patches at the given scale, centred over the keypoint, and rotated to align its dominant orientation. This Fisherspace is used to derive a compact feature vector for all the images used in training.

3D-SIFT [117] is a 3D SIFT descriptor implementation for volumetric imagery such as MRI data, CT data or video sequences. As the data has a volumetric aspect the descriptor has been modified to calculate the gradient magnitude and orientations in 3D. Sampling of the training videos is carried out either at spatio-temporal interest points or at randomly determined locations, times and scales. The next step is to construct a weighted histogram similar to that of the SIFT for the 3D neighbourhood around a given interest point to generate a 3D orientation. The descriptor is calculated using $4 \times 4 \times 4$ bins and results in a 2048 dimensional vector. The original algorithm [117] is not fully rotationally invariant but this has subsequently been corrected [5, 37].

Binary SIFT [100] analyses the space of the SIFT and concludes that only a small proportion of the feature space is being populated by SIFT features. Based on this analysis the paper proposes reducing the descriptor to a 128 dimensional binary version rather than using 256 values for each dimension. The descriptors are converted by finding the median values for each descriptor and setting values to 1 or 0 based on whether they are higher or lower than that value. The median value is used so that the vector will be split equally. The results show a strong correlation with a very small error bound between the new descriptors and the original SIFT descriptors for the number of matched features on a set of test images. Thus the reduced feature space does not affect the matching process or the uniqueness of the features. The feature space is reduced from $256^{128}$ to $2^{128}$ which is equivalent to 128 bytes to 16 bytes of storage, an 8 fold reduction. Binary features allow for fast and efficient indexing and matching times are reduced by using XOR and Hamming distance which are faster than calculating Euclidean distance.

SURF Adaptations

MSURF [116] is an altered version of SURF to eliminate the boundary effects and artefacts in the original algorithm by extending the subregions in the descriptor so that they overlap. USURF [139] also makes some changes to SURF to speed it up by removing its rotation invariance which may be unnecessary for circumstances where the images are known to be at the same angle such as for stereo camera
2.4 Feature Matching Strategies

The simplest matching strategy is to use the minimum Euclidean distance and a threshold to decide if two feature vectors are close enough to match but this lacks robustness as the threshold value may not be valid for all images and over fit to training data. The following section outlines other strategies that can be used for more robust or faster feature matching.

2.4.1 Nearest Neighbour Feature Matching

To match features for the SIFT the Euclidean, nearest neighbour, distance between two feature vectors is used. To increase robustness over just having a threshold to decide if points match, matches are rejected for those features for which the ratio of the nearest neighbour distance to the second nearest neighbour distance is greater than 0.8. This has been shown to mean that 90 percent of the false matches are discarding while losing less than 5 percent of the correct matches [70]. The Euclidean distance $D_e$ is given as follows where $u$ and $v$ are two feature vectors of length $n$:

$$D_e(u, v) = \sqrt{(u_1 - v_1)^2 + (u_2 - v_2)^2 + \ldots + (u_n - v_n)^2} \quad (2.17)$$

To reduce the time spent searching for the nearest neighbours a Best-Bin-First (BBF) algorithm is used [69]. This finds an approximate solution to the nearest neighbour problems sacrificing accuracy for speed. This generally returns the nearest neighbour but will sometime return a neighbour which is very close. It uses a $k$-dimensional tree (kd-tree) to index the 128 dimensional space but modifies the search ordering so that bins in feature space are searched in the order of their distance from the query feature.

Fast approximate nearest neighbours with automatic algorithm configuration [92] has been investigated which expands on the original kd-tree work and looks into optimal approximation to linear feature search. There is a trade off between accuracy and speed and has shown that multiple, randomised, kd-trees provide the best performance where trees are built by choosing the split dimension randomly.
Once the nearest and second nearest neighbours are found, the Euclidean distance is measured between the two vectors and the target feature which is being matched. If the ratio of the distance from the target feature to the nearest and second nearest neighbours is less than 0.8 the match is deemed positive. To identify an object Lowe suggests the use of 3 or more features to recognise an object as matches with fewer may be anomalous.

2.4.2 Approximation of Euclidean Distance

Using a Euclidean distance as a measurement between two vectors is the most accurate way of calculating the distance but it can be slow for large vectors (128-dimensions) as for an \( n \) size vector \( n \) multiplications are computed and one square root. This can be reduced by using approximated distance measurements.

Computing the city-block (sum of the absolute differences of two vector’s coordinates) or chessboard distance (the greatest of two vectors differences along any coordinate dimension) is simpler than computing the Euclidean distance and can therefore speed up the matching process.

\[
D_{\text{city}}(u,v) = |u_1 - v_1| + |u_2 - v_2| + \ldots + |u_n - v_n| \tag{2.18}
\]

\[
D_{\text{chess}}(u,v) = \max (u_1 - v_1, u_2 - v_2, \ldots, u_n - v_n) \tag{2.19}
\]

Strategies combining city-block and chessboard distance based matching [144, 160] have been shown to be more efficient yet still effective.

2.4.3 Earth Movers Distance Based Matching

Earth Movers Distance based matching has also been suggested [64,101] as a way of making matching more robust. The problem with using the Euclidean distance, where bins are compared directly to other bins after the dominant orientations have been aligned, is that the same feature in two images can be misaligned. This can be due to shape deformation, lighting, occlusion or heavy noise in one or both of the images. The earth movers distance (EMD) based techniques aim to alleviate this alignment problem by sampling data across bins.

The earth movers distance treats the two vectors as \textit{supply} and \textit{demand}. The first vector has to be moved to fill the second vector using the shortest distance
from bins that are closest that contain a supply. Bins that match closely can easily be used to fill the empty bins in the second vector. Vectors which do not match well will require a larger move, i.e. across bins, to fill the empty bins in the demand vector. The amount (histogram value) moved times the distance by which it is moved is the earth movers distance. This means that vectors that are slightly misaligned will still have a low score, but would be treated by Euclidean distance measurements as mismatches.

Some different strategies have been suggested for matching image feature histograms including a drastically simplified tree based method [64] and an algorithm designed to work in linear time between non normalised histograms [101]. Both show an increase in robustness over Euclidean distance for solving matching tasks with large deformation, noise and lighting change.

### 2.4.4 Other Matching Strategies

An analysis of the SIFT descriptor using 8.5 million features [135] found that there are 8 large symmetrical peaks aligned with the features orientation. These also have a large standard deviation so it is unlikely that these will appear in all features but various combinations of the eight peaks are likely. The study suggest that this can be used as an initial comparison to discard matches quickly as if these key large peaks do not match then the feature is unlikely to match when the full descriptor is compared. The study shows that using a hierarchical Euclidean distance matching model where the vector is compared in stages and discarded if it fails at any particular stage gives a 15 times increase in speed over a linear matching system and is four and five time faster than PCA-SIFT and SURF respectively where the descriptors are smaller than for the SIFT.

VF-SIFT: Very Fast SIFT Feature Matching [4] extends the SIFT feature by a few pairwise independent angles, which are invariant to rotation, scale and illumination changes. The SIFT features are clustered based on these angles into different clusters and stored in a multidimensional table. In the feature matching stage only the SIFT features that belong to clusters so that when the features are being matched only those in areas of the table which are expected to match are compared. The experimental results show that the feature matching can be accelerated about 1250 times when compared to an exhaustive search without losing a significant amount of correct matches.

A probabilistic approach to feature matching [53] is demonstrated for stereo
views where the cameras are fixed and the rough disparity is known. Regions are estimated for each feature as to the likely area in the second image where it will match. It is compared to each feature in that area and each possible match is weighted based on the similarity of the feature vectors. This is repeated for all features in the image pairs. The weightings are then adjusted based on how the features near each other behave and how their disparities compare assuming that features near each other will have similar disparities. Probabilistic estimates of which features match are iteratively improved by a relaxation labelling technique. The algorithm is shown to be effective for matching 3D stereo pairs, it converges quickly and can be applied to images having wide disparity range.

2.5 Feature Matching with A Contrario Techniques

A contrario means “by or from contraries” and is used to describe an argument based on contrast. This is relevant to image processing in that positive correlation can be found between two samples by looking at the chances of that correlation occurring in random data. For example, two regions are meaningfully similar if the probability that the heuristic finding such similarity regions in pure noise or random data is very low. It is based on the Helmholtz principle and comes from the assumption that “an event is meaningful if its number of occurrences is very small in a random model”.

The Null Hypothesis, an assertion that is capable of being proven false, is used as the measure of an event. An example of the Null Hypothesis could be “two regions are similar” which can use a similarity measure to calculate this. A meaningfulness threshold is calculated given the probability of the hypothesis with an input of random data. This random input is known as the background model. Given the extent to which a hypothesis is likely to be true for the background model a threshold is generated for discerning true positives.

Desolneux et al. [28,29] introduced this concept in the area of computer vision. The papers present mathematical formalisation of a perceptual grouping principle and results in an a contrario detection scheme. They look at images where multiple edges align in structured images and that significant geometric structures in an image correspond to very low probability events when compared to randomly generated images. This allows the detection of the structure by contrasting the
2.6. FEATURE FILTERING AND POSE ESTIMATION

This method has been used for image processing and computer vision techniques including those related to the SIFT feature matching. Securing SIFT with A Contrario Techniques in A Theory of Shape Identification [16] uses the technique to make the SIFT more robust when matching features. Instead of taking the second nearest neighbour feature from the same image as the one that is being matched too the chapter suggests that a background model should be used for this. A random image is used as the background model and is shown to increase the number of matches in self-similar images and allow multiple instance of the same object to be detected in a scene.

A Contrario Matching of SIFT-like Descriptors [104,105] uses the a contrario methodology to adaptively threshold the earth movers distance (EMD) between two descriptors. The dissimilarity measure using the EMD is calculated and rather than using a static global threshold their a contrario framework adaptively generates a threshold on a probability of false detections. It is generated for each query descriptor and is adapted based on the diversity and size of the database of features being matched.

2.6 Feature Filtering and Pose Estimation

This section outlines methods used to work out how groups of features match to another set of feature and the relative position or pose of these features. This allows the overall transform between the two sets of matching points to be estimated and highlights any outliers (mismatches) which do not fit with the overall trend. This can be used to decide if a set of matching feature are matched correctly based on their geometric relationship to others in the set and the other set of features to which they are matching.

2.6.1 RANSAC Based Match Filtering and Pose Estimation

RANdom SAmple Consensus (RANSAC) [36] is one of the main ways that features are checked to see if the matches that have been made between a pair of images are valid. This is done by attempting to fit one set of points to the other so that they align in point space and a homography can be calculated to warp
one set to the other. Often there are outliers which match but do not fit the pose that is supported by most features and RANSAC can be used to highlight these matches as unlikely to be correct.

RANSAC works by refining matches by imposing global geometric constraints by means of robust estimation of the epipolar geometry and the incorrect matches are rejected as outliers. For general feature matching [158] it randomly selects $n$ (a predetermined number) sample feature matches and then for each sample estimates a model hypothesis, the parameters of a transformation from one set to the other. It then calculates the number of inlier features which support this hypothesis from the total set of features. A feature match is deemed to be an inlier if it is within a threshold $t$ distance of the features it is matched to. The hypothesis with the largest support is then chosen as a model and then the full set of inliers are used to refine the model parameters.

Of the $n$ samples randomly chosen the hope is that at least one sample will be the true correspondence allowing accurate parameters to be obtained. Hence in order to achieve confidence $\rho$ that one such sample is obtained, the required number of samples $n$ can be computed as follows, where $\epsilon$ is the fraction of outliers (false correspondences):

$$n = \frac{\ln(1 - \rho)}{\ln(1 - (1 - \epsilon)\rho)}$$ (2.20)

RANSAC has been shown to be successful in many cases [17, 24, 66, 114, 118, 148, 154, 155] as a way of refining the feature matches and reducing outliers.

2.6.2 Hough Binning Based Match Filtering and Pose Estimation

RANSAC can perform badly when the number of inliers falls below 50 percent which may happen if an object is occluded. Therefore it is suggested [1, 70, 119] that a Hough binning technique is used for outlier rejection and pose estimation. The technique estimates transform parameters from one set of points to another. This is done by using different hypothesis parameters for a transformation and using each feature to vote for all object poses that fit it. The poses can then be sorted by the number of votes and the best selected as the most likely correspondence between the images. The parameters that are used initially are the scale, rotation and 2D position and are binned with large margins of error. Each bin
2.7. EVALUATION OF FEATURE DETECTORS/DESCRIPTORS

covers 30 degrees of rotation and 2 scale factors. The similarity transform implied by these 4 parameters is only an approximation to the full 6 degree-of-freedom pose space for a 3D object. The bins with 3 or more entries are used in a second refinement stage where a least squares solution is used to generate the best affine projection parameters relating the two sets of features.

2.7 Evaluation of Feature Detectors/Descriptors

From reviewing literature and experimentation the adequacy of the SIFT in its current state as a general object recognition has been ascertained. Under certain conditions the algorithm performs well; generally when there is only a small variance between the source and target images. The combined affects of motion blur, scale changes and viewpoint changes that occur constantly within videos of scenes obtained from moving cameras reduce accuracy when attempting to match a single object within a cluttered scene. Although often the object is highlighted by a cluster of points, it becomes difficult to distinguish these from false positive matches that may arise. A method of distinguishing the false positives from the actual matches is required. Methods could consist of using the objects colour, shape or the spatial relationship between the points help increase the classification power of the SIFT algorithm.

2.7.1 A Performance Evaluation of Local Descriptors

The paper by Mikolajczyk and Schmid entitled *A Performance Evaluation of Local Descriptors* [86] highlights that the SIFT is one of the best local descriptor algorithms. In situations with changes in scale, rotation, viewpoint, lighting, image compression and blur it outperforms most, often all, of the other local descriptors.

The paper displays the performance of feature detectors and descriptors using receiver operating characteristic (ROC) curves. These plot the recall against $1 - \text{precision}$ while the distance between two vectors, or the ratio between the nearest and second nearest neighbour, required for a match is varied. Recall is the number of positive matches divided by the total number of expected matches. The total number of expected matches is estimated using the overlap error between images which measures how well images correspond under a transformation. The correspondences are obtained using a transformation. $1 - \text{precision}$ is the number...
of false positives divided by the total number of matches. The images used were a selection of relatively flat surfaces with repeated textures such as walls and 3D structured scenes with distinctive scene boundaries such as buildings and walls with graffiti. The fact that the scenes used are either planar in the case of the walls or cityscapes/buildings which are captured from a distance, means the images are effectively planar even with changes in view point means that the use of 3D objects is not covered in this paper.

The paper uses several region detectors with the descriptors, not just the DoG method suggested by Lowe, to choose areas from which the descriptor is created. These are Harris points, Harris-Laplace regions, Hessian-Laplace regions, Harris-Affine regions and Hessian-Affine regions. The point and region detectors are selected within the paper for their properties with respect to a specific test. For example; Hessian-Laplace regions are invariant to scale and rotation changes and as such the paper uses this region detector for the tests where the scale is altered against other detectors with similar invariance. The paper concludes that a Hessian based region detector generally perform better. 10 descriptors are used, the most notable of which are the SIFT, GLOH and PCA-SIFT.

The paper primarily investigates three descriptor matching strategies. Threshold based matching where the two regions are matched if this distance is below a threshold, nearest neighbour matching where the descriptor which best matches is chosen as long as it is below a threshold and nearest neighbour distance ratio matching where the match is confirmed if the ratio between the nearest and second nearest neighbour is above a threshold. As ascertained by Lowe, the nearest neighbour distance ratio matching is the most successful.

The results of the study show that for viewpoint changes of between 40 and 60 degrees both the SIFT and GLOH perform best. The SIFT is marginally better for textured scenes and GLOH is marginally better for structured scenes. The descriptors are computed for Harris-Affine and Hessian-Affine regions.

The second investigation focuses on scale change and image rotation. The scale change is of a factor of 2-2.5 times the original images scale and the rotation is between 30 and 45 degrees. The descriptors are computed for Harris-Laplace and Hessian-Laplace regions. In this case the SIFT performs well along with GLOH, depending on the scene the optimal choice varies, but both always perform well. For rotation alone, of 30-45 degrees, the paper again indicates that the SIFT and GLOH perform best along with the shape context descriptor. In this case
2.7. EVALUATION OF FEATURE DETECTORS/DESCRIPTORS

the descriptors are computed for Hessian-Affine regions.

For blurred images, introduced by changing the camera focus, the results for structured scenes and textured scenes vary. The descriptors are computed for Hessian-Affine regions. For a structured scene the highest scoring descriptors are PCA-SIFT and GLOH followed by the SIFT and for a textured scene the SIFT performs best followed by GLOH. Blur greatly affects textured scenes with the recall rate substantially reduced from that of structured scenes. This appears to be because the blurring make the detected regions nearly identical and therefore introduces many false positives.

For JPEG compressed images, with the quality reduced to 5 percent of the original, the SIFT and GLOH are again the best two descriptors with the SIFT marginally better as the false positive rate increases. The descriptors are computed for Hessian-Affine regions. For illumination changes, obtained by adjusting the exposure of the camera. The descriptors are computed for Hessian-Affine regions. GLOH and Harris-Affine GLOH performed best followed closely by the SIFT.

Overall the paper indicates that GLOH performs best followed very closely by the SIFT. Shape context is the next best descriptor and performs well on scenes with clearly defined and reliable edges. Although the results of the paper are promising they do not mean that GLOH and the SIFT are suitable for robust object recognition in their current state. It does indicate that they are the most suitable candidates for study and improvement but there are limitations in the data-set chosen for evaluation. The paper focuses on matching images where the target takes up a large percentage of the scene in which it is being located. It does not test for attempts to locate a small object in part of a cluttered scene which could be the case in real world situations. As such the results of object recognition may not be as successful as indicated in this paper as recognising a smaller area within a scene the number of features that are available to perform the match will be limited. Also, in real-world situations, the object may be affected by several of the variables at once, e.g. scale, rotation and blur, and as such, failings of the algorithms may be compounded. As indicated previously the planar nature of the scenes used is also an issue due to the lack of 3D tests.
2.7.2 Evaluation of Features Detectors and Descriptors based on 3D Objects

A paper by Moreels and Perona entitled *Evaluation of Features Detectors and Descriptors based on 3D Objects* [87] also compares the SIFT with several other algorithms. It uses 3D scenes where changes in viewpoint have a substantial impact on what can be viewed within the scene due to the closeness of the object to the camera and self-occlusion. The investigation features images generated by placing complex, non-planar, objects on a turn-table and taking pictures at various rotations between 5 and 60 degrees and in three different lighting conditions. One hundred different objects were used and a selection of feature detectors and descriptors were used.

The experiment consisted of matching the object rotated on the turntable to itself over the range of angles and lighting conditions and seeing how the results compare to the expected number of matches. The corresponding regions that are visible in the test view, and therefore could possibly be matched, are calculated using epipolar constraints between triplets of calibrated views of the objects. By knowing the location of a feature in two images of the scene using two cameras placed above each other with a difference in viewing angle of 10 degrees the estimated position of the feature in a third image with an object rotation of a known angle can be calculated. This roughly indicates the number of matches that could be possible and can be compared to the performance of the descriptors.

The Hessian-Affine and difference-of-Gaussian consistently performed best overall with the selection of descriptors. The combination of Hessian-affine with the SIFT was the overall best performing pair, a very similar conclusion to that reached by Mikolajczyk and Schmid [86]. PCA-SIFT again did not out-perform the SIFT, as in the paper by Mikolajczyk and Schmid, even though this was found to be the case in the paper which introduces it by Ke and Sukthankar [58]. The difference-of-Gaussian detector, the method used in Lowe’s paper, performs constantly almost as well as the Hessian-affine detector and is much faster to compute. Thus the paper states that this could be used in situations where speed is required without sacrificing much accuracy. The paper indicates that for viewpoint changes over thirty degrees a very small number of features can be matched (less than three percent) and as such attempts to match when the angle is greater than this should be avoided.
2.7.3 Local Features and Kernels for Classification of Texture and Object Categories

Local features and kernels for classification of texture and object categories: A comprehensive study [138] compare the SIFT, SPIN [54], a rotation invariant histogram of intensities, and RIFT [61] a gradient orientation histogram descriptor computed from circular normalised patches divided into concentric rings of equal width. This paper demonstrates the merit of using combinations of complementary feature detectors and descriptors in order to generate a set of features for matching which take advantages of the invariant qualities of more than one type. The idea is based around the use of one set of features to fill the performance inadequacies of another. This evaluates the descriptor on four texture datasets and five object category datasets. The tests match the data using the descriptor and combinations of the descriptors. The results show that the SIFT performs best on its own when compared to RIFT and SPIN individually. When they are combined the SIFT and SPIN and RIFT and SPIN show greater performance than any single feature detector and overall, the combination of Harris-Laplace and Laplacian detectors with the SIFT and SPIN performs best.

2.7.4 Evaluation of Image Features Using a Photorealistic Virtual World

Evaluation of Image Features Using a Photorealistic Virtual World [56] compares a variety of feature descriptors (DAISY, HOG, SIFT and GLOH) for dense reconstruction in photorealistic 3D simulations of a virtual world. The scene lighting and viewpoint are controlled precisely and the ground truth is known due to the images being simulated. The experiments show that DAISY performs best for both illumination and viewpoint changes and that the organisation of the descriptor bins makes a performance difference for various lighting conditions and the number of bins is more important for changes in viewpoint. Low dimensional descriptors such as HOG are shown to perform badly due to the lack of distinctiveness. GLOH and the SIFT perform marginally worse than DAISY. The descriptor for the SIFT and GLOH have been reduced to 8 or 16 dimensions for these experiments for speed due to the dense matching process and the large numbers of points which are matched. As such the experiments are not indicative of results expected for sparse matching with higher dimensional vectors and
DAISY has been specifically designed for dense correspondence tasks.

2.7.5 Performance Evaluation of Local Colour Invariants

Performance evaluation of local colour invariants [14] compares CSIFT with the SIFT and other feature detectors. The work includes colour information in the SIFT descriptors, and proposes three colour SIFT methods each having different characteristics with respect to photometric variation. It shows that \textit{c-colour}, a colour invariant method for encoding the SIFT descriptor which is invariant to local intensity level, shadow and shading, performs better than the original SIFT algorithm. It has significantly increased discriminative power when matching points and claims to retain all the advantageous properties of the SIFT within these tests.

2.7.6 A Comparison of SIFT, PCA-SIFT and SURF

A comparison of SIFT, PCA-SIFT and SURF [55] compares the three algorithms using the Euclidean nearest to second nearest neighbour ratio (0.5) and RANSAC for the purpose of object recognition. The paper compares the algorithm on an image dataset, which includes the general deformations, such as scale, view, illumination and rotation changes. The SIFT is shown to be the most stable and is consistent across most circumstances. SURF is the fastest but performs worse than the SIFT in all cases except illumination. PCA-SIFT performs worst overall but is better than the SIFT for illumination change.

2.7.7 An Evaluation of Image Feature Detectors and Descriptors for Robot Navigation

An Evaluation of Image Feature Detectors and Descriptors for Robot Navigation [116] compares FAST, Harris, Shi-Tomasi, SURF and STAR as they are all fast detectors which can be used in real time and thus are suited to the real time problem of robot navigation. Various descriptors were paired with each of the descriptors; SAD (sum of absolute differences), SSD (sum of squared differences), NCC (normalized cross-correlation), MSURF36, MSURF64, MUSURF36 and MUSURF64. The experiment compared consecutive frames of robot video.
sequences for five thresholds of matching and for all possible combinations of detectors and descriptors. RANSAC was used and the results show that the best descriptor overall is MSURF and the different dimensionality makes little difference. The multiscale detectors behave in general worse than Harris, FAST and Shi-Tomasi detectors while paired with the same descriptor and Shi-Tomasi performed best. The next experiment matching was performed between every 10th frame and MSURF performed best again with the single scale detectors.

2.7.8 Probing the Local-Feature Space of Interest Points

Probing the Local-Feature Space of Interest Points [62] and A Hierarchical Approach to Practical Beverage Package Recognition [157] use a local sensitivity hashing to analyze the space of interest point features. The first paper compares the SIFT descriptor on patches generated from DoG and Hessian-affine detection to randomly chosen patches. The second paper uses the techniques to compare the SIFT and SURF and to identify that SIFT features produces a more informative and diverse feature space for their dataset; beverage packaging. The technique is based on Locality-Sensitive Hashing Scheme Based on p-Stable Distributions [26] and works by hashing the feature vectors so that similar items are mapped to the same buckets and the number of buckets being much smaller than the universe of possible input items. This allows for the proximate distribution of the features within the space to be analyzed.

2.7.9 Other SIFT Evaluations

Two papers which analyse the SIFT are concerned with its consistency and its scale invariance. On the consistency of the SIFT Method [89] and Is SIFT scale invariant? [88] look at the mathematical properties of the algorithm. On the consistency of the SIFT Method investigates the different invariants of the descriptor to see how invariant the SIFT method is to each. The conclusion is that the SIFT achieves perfect performance with, zoomed, rotated, translated versions of two images but not projective transformations. Is SIFT scale invariant? investigates further whether the SIFT is scale invariant and it proves that it is scale invariant only if the initial image blurs is exactly guessed. In cases when the initial blur is misestimated the scale space representation of the descriptor attenuates the error.
An analysis conducted by Pavlidis [99] covers some image types left uninvestigated to highlight some specific areas where the SIFT performs particularly badly. A small selection of images consisting mainly of dogs in different poses with backgrounds of repeating textures was used. The small sample shows that the SIFT is unable to cope with variation in pose between the same dog. The large areas of highly textured background lead to a high number of false positives due to the large number of features they produce and the similarity between them.

2.8 Parametrization of Feature Detectors and Descriptors

The choice of parameter values of the SIFT effect the response of the algorithm but exactly how changes in their values vary the result and accuracy of feature matching has not previously been studied in sufficient detail. Table 5.1 shows a list of the main intrinsic parameters which control the response of the algorithm and Lowe’s default parameters [70]. A subset of these have been selected as the focus of the parameter analysis in Chap. 5.

Often experiments use the original Lowe algorithm parameters without specifically tuning them for the task [1, 11, 71, 85] and these may not provide the best results.

Other papers have varied the parameters for their work. Jagadish and Sinzinger [52] selected a match ratio of 0.6 for their work comparing the SIFT to Radial Feature Descriptors on tone mapped images without explanation as to why this value was selected. This is also the case in the paper by Battiato et al. [9] who justify the change of the match ratio through experimentation. They also find that adjusting the contrast threshold to extract fewer points results in a smaller set of more stable features. The paper by Park et al. [97] uses the SIFT for fingerprint identification and chooses to use 4 octaves with 5 intervals and a Gaussian sigma of 1.8. A paper by Tang et al. [130] shows that increasing the Gaussian smoothing reduces the number of features generated from an image. A paper by Cesetti et al. [18] automatically adjusts the contrast threshold value based on the properties of the images. An equation calculates a contrast threshold based on the intensity and size of the image and the image is not processed at scales where this value becomes too small as it proposes that there is a low probability
of finding useful features in a low contrast image.

Some papers focus on techniques for tuning parameters for feature detectors and descriptors including the SIFT, DAISY [132] and GLOH [86] using various methods. *Learning Local Image Descriptors* [151] assess descriptors using a framework for choosing the parameters from a training set of features descriptors extracted from a multi-image 3D reconstruction dataset where accurate ground-truth matches are known. The parameters that are trained on a set of $64 \times 64$ patches where corresponding patches are known. The parameters of the descriptors that were varied for the experiments were the transformation of the image patch into a space suitable for histogram binning, the histogram binning technique and the normalisation of the vector. 7 different transforms were used including the DoG method used by the SIFT and others inspired by other techniques. This is followed by 4 possible binning techniques including that used by GLOH and the SIFT and then a vector normalisation stage with variable parameters. The experiments use various combinations of filters and binning techniques and varies the parameters for each. The best descriptors were those with log polar histogram regions and feature vectors constructed from rectified outputs of steerable quadrature filters and show a improvement over the SIFT.

*Picking the Best DAISY* [150] tests the DAISY descriptor on a set of match/non-match image patches and test for a large selection of gradient and steerable filter based configurations and optimise over all parameters to obtain low matching errors for the DAISY descriptors. The filters used, the number of rings and segments and the overlap of these segments and the normalisation of the descriptor are varied to effect this. The paper also focuses on reducing the size of the descriptor while maintaining distinctiveness to increase the speed of the matching process using PCA. The process shows that the DAISY descriptor can be parametrized to outperform the SIFT using the default parameters and results in a lower error rate using fewer bytes of storage for the descriptor. This only focuses on the descriptor side of the algorithm and detection is not considered.

*Discriminant Embedding for Local Image Descriptors* [49] addresses the problem of parametrizing the dimensionality of descriptors and reducing the number of dimensions required for accurate matching while maintaining or improving match performance. This is done by training the feature descriptors on matching and non-matching image patches. As with the previous two papers, a filter is applied to the patches, in this case they are non-linear such as bias-gain normalisation
or rectified gradient filters. Next descriptors are applied and linear discriminant embedding, a technique for reducing the dimensionality of the descriptors, is used to map the feature vectors to a subspace. These are then matched using nearest neighbour classification. 10,000 images were used for training and another 10,000 for testing. The experiments demonstrate that they can exceed the performance of the SIFT with far fewer dimensions, and with virtually no parameters to be tuned by hand.

*SIFT Based Automatic Tie-point Extraction for Multitemporal SAR Images* [65] optimises two SIFT parameters for synthetic aperture radar (SAR) image tie-points, the edge points which overlap between two images. The number of orientation bins (NOB), the number of spatial bin (NSB) for each row of the descriptor and the size of the bins for the descriptor given by $m$ multiplies the feature scale $\sigma$. For the SAR images tested, which are planar and with little angular change as they are taken at high altitude of the ground, show that the larger values of $m$ improve the results and by using a value of 12, rather than Lowe’s values of 3, is preferable for this image set. This value $m$ increases the size of the descriptor area that is binned. Increasing the NOB and NSB are shown to have a negative impact as they do not increase the data that the descriptor captures but rather the number of bins that it is put into making the descriptor more specific and therefore less likely to match. Lowe’s defaults (NSB = 4 and NOB = 8) are shown to provide the best results.
Chapter 3

Overview of Other Related Work

3.1 GPGPUs and CUDA

One important aspect of the thesis is the use of General Purpose Graphics Processing Unit (GPGPU) technology to greatly improve the speed of the algorithms for real-time testing. The inherent parallelism of many parts of computer vision algorithms means they lend themselves to being implemented on GPGPUs. The fact that they often involve operations that run independently across an image, such as a convolution operation, means that an image can easily be broken up into subsections and the subsections processed in parallel.

The reason why the GPU is so powerful and can be utilised for this project stems from the large amounts of money being invested in improving their performance for the games industry. They are mass produced and relatively cheap and have the ability to perform highly parallel floating point calculations, for example the NVIDIA 9800 GT2 has 256 processing streams operating concurrently [95].

The large amount of memory available on the GPU and the speed at which it can be accessed is an advantage for the GPU although transfers from main memory to GPU memory are slow. As an example, the NVIDIA 9800 GT2 has 1GB of memory and a memory bandwidth of 128GB/sec peak [95] which can be utilised in tasks such as pattern matching where it is beneficial to store reference data so that it can be matched at high speed [38]. Another benefit of the use of GPU is that multiple GPUs can be used in parallel. For this project two NVIDIA 9800 GT2 GPUs are available which provides significant processing power.

NVIDIA’s CUDA [96] is a general purpose parallel computing architecture that provides the tools required for the coding of parallel code for a GPU and
facilitates its execution in a fraction of the time it would take to execute on a
CPU. It includes the CUDA Instruction Set Architecture (ISA) and the parallel
compute engine in the GPU. Currently the code is written in C (with other
languages soon to be available) using a small set of CUDA extensions to enable
parallelism. The code allows homogeneous execution on both the CPU and GPU
so all the resources of the system can be taken advantage of and code which is
suited to serial execution can still be executed on the CPU. The architecture also
allows the use of multiple GPUs in parallel.

CUDA provides function type qualifiers (that are not in C) to enable a pro-
grammer to define where a function should run:

- \_host\_: This specifies the code should run on the host CPU. It is the
default.

- \_device\_: This specifies the code should run on the GPU, and the function
can only be called by code running on the GPU.

- \_global\_: This specifies the code should run on the GPU, but be called
from the host. This is the access point to start multi-threaded code running
on the GPU.

Beyond these initial declarations grids and blocks of parallel code can be
defined which contain the parallel threads that will be executed on the GPU.
The threads are grouped into thread blocks which are in turn grouped into a
grid. This defines a memory hierarchy, where memory is shared between threads,
which is important for execution speed. The blocks and grids define the number
of threads executing the same, concurrent, task on different data.

The SIFT has been shown to be successfully parallelised on the GPU. These
include the use of CUDA; CudaSIFT [12] and SiftGPU [153] and the use of
OpenGL textures to store and process the images [48]. All cases show significant
speed up. Also of note is GPUCV [34], a library written using CUDA, designed to
provide seamless acceleration with the OpenCV interfaces. This is in development
and will contain a large number of important image processing and computer
vision algorithms that will allow a user to more easily take advantage of the
power of the GPU.
3.2 High Dynamic Range Images

Dynamic range is the ratio between the brightest and darkest pixel values in a scene. The human eye can see the low levels of light from a star at night to the sunlight at noon on a bright day. The eye cannot see both at the same time and has to adapt for each scenario. Adaptation allows for the eye to have a dynamic range that is nearly 10 orders of magnitude. High dynamic range (HDR) images have a dynamic range that is designated to be large enough to store the high range of radiance we experience in real world scenes.

Standard low dynamic range (LDR) images often use 8-bit per channel to store the RGB values that make up a pixel. The 8-bit representation gives 256 possible values per channel often with 0 representing black and 255 representing the full luminosity of the channel. This allows a dynamic range of about two orders of magnitude [106]. This is not sufficient to represent the dynamic range of the real world and information is lost.

HDR images use a format with 32-bits per channel. The radiance values for red, green and blue are stored as floating-point numbers. A IEEE floating-point number is made up of two parts, the mantissa and the exponent; 23-bits make up the mantissa which is a non-zero value used to represent the precision of a number, and 8-bits make up the exponent which indicates how many times the base, in this case 2, is used as a factor. The value of the floating-point number is calculated by first calculating the value of the base to the power of the exponent.
and then by multiplying the result by the mantissa. One bit is used to indicate whether the number is positive or negative [50].

The use of the mantissa and exponent allow the 32-bits to represent a far greater range than if the data was stored in the conventional binary manner as is the case with LDR formats. This representation can store numbers with a range up to 79 orders of magnitude, depending on the exact implementation details of the format chosen [106], although this does reduce the precision of the values stored. This large range can then easily encompass the dynamic range of real life scenes that can be viewed by human eyes.

Consider a scene with a bright spotlight shining towards the viewer above the entrance to a dark cave. The contrast between the two is large yet with standard LDR representation all that can be shown is that the spotlight is white and the cave as black. Any values above the format’s range will be mapped to the highest value so information about just how much brighter the light is than the cave is lost. A HDR image allows the difference between the two to be stored due to the large range allowed by the floating-point representation. This therefore allows more detail to be encoded about a scene which in turn should allow any algorithm which uses information from the scene to take advantage of this. Figure 3.2 shows a series of LDR images and the tone mapped HDR image generated from them. It is clear that the HDR image contains more detail than any individual LDR image. The creation of tone mapped HDR images, a technique for displaying a HDR image on an LDR display, is discussed later in this chapter.

![Figure 3.2: The top right image is a tone map of a HDR image created from the three other, LDR, images. Note that in the tone mapped image details from the scene are clearly visible both inside and outside the window.](image)

### 3.2.1 Generating High Dynamic Range Images

Although the HDR image formats can hold much more information about a scene current cameras do not generally capture HDR data, although there are some
exceptions but these are very expensive [127]. The capacity of CCD sensors in most digital cameras is limited to between 8- and 12-bits per colour channel which is not always enough to capture the full dynamic range of a scene. This means that the photographer has to adjust the exposure time to decide what part of the range he wishes to capture within the image. Radiance levels outside this range will be either over- or under-exposed and detail will be lost. One way to generate a HDR image is to take multiple images of the same scene at different exposures and composite them to incorporate the correctly exposed parts of each.

3.2.2 Debevec and Malik

A regularly used and recommended method [106] for generating HDR images was invented by Debevec and Malik [27] and has two stages which are as follows and gives a general idea of how the HDR images are generated:

1. **Generate the camera response function.** HDR images are scene-referred which means that the values stored for each pixel directly represents the radiance within the scene. LDR formats which are output-referred meaning that the values stored are associated with a target output device. There is a non-linear mapping from the scene radiance to the value stored in an LDR image. This is due to the way the camera’s sensor responds to the light and how it maps the information received to the LDR file. A pixel value that is twice that of another within an image does not necessarily correspond to double the radiance in the scene. The function that describes this non-linear mapping is called a response function and it must be found for the creation of a HDR radiance map. This requires multiple, aligned, images of the same scene with varied and known exposure times.

   By sampling the images and comparing the values at the same points for the different exposures the response of the camera to variation in scene radiance can be calculated. If doubling the exposure equates to doubling the scene radiance one can see the effect of that change in radiance on the value of a pixel.

   The quadratic objective function below is used to derive the logarithmic response function $G(Z_{ij})$. Linear optimisation is used to find a smooth curve that minimizes the mean squared error over the derived response
curve. The first term is the data fitting term and the second term ensures the smoothness of the response function.

\[
O = \sum_{i=1}^{N} \sum_{j=1}^{P} (W(Z_{ij})[G(Z_{ij}) - \ln E_i - \ln \Delta t_j])^2 \quad (3.1)
\]

\[
+ \lambda \sum_{z=Z_{\text{min}}+1}^{Z_{\text{max}}-1} [W(z)G''(z)]^2 \quad (3.2)
\]

where \(Z_{ij}\) is the value of the pixel at image position \(i\) and image exposure \(j\). \(\Delta t_j\) is the duration of exposure \(j\). \(E_i\) is the irradiance value at position \(i\) and \(w(Z_{ij})\) is the hat weighting function shown below where \(Z_{\text{min}}\) and \(Z_{\text{max}}\) are the least and greatest pixel values in the image. \(\lambda\) is scalar and weighs the smoothness term relative to the data fitting term and is chosen depending on the expected noise in \(Z\).

\[
W(z) = \begin{cases} 
  z - Z_{\text{min}} & \text{for } z \leq \frac{1}{2}(Z_{\text{min}} + Z_{\text{max}}) \\
  Z_{\text{max}} - z & \text{for } z > \frac{1}{2}(Z_{\text{min}} + Z_{\text{max}})
\end{cases} \quad (3.3)
\]

The weighting function minimises the effect of the pixel values that are near the extremes of the response function as these are most likely over- or under-exposed. The values given at extremes cannot be relied upon as they may have been outside the dynamic range of the original format. It weights the values closest to the middle highest.

The equation is solved to find an optimal value for \(G(Z)\) for every possible value of \(Z\) from 0 to 255 for each channel of an 8-bit image. This is done by choosing sample points from images of different exposures and applying the minimization technique. With enough sample points, 50 or more, the result is a smooth response function for each channel. The sample points must be spread over the range of pixel values within the 8-bit image for the technique to be effective. Once the response function has been obtained it can be used with any set of images taken by that camera.

2. **Construct a radiance map.** The response function is used to map the pixel values to the relative radiance values which can then be stored as floating point numbers. The equation below uses the response function
3.2. HIGH DYNAMIC RANGE IMAGES

$G(Z_{ij})$ and the weighting function given above to calculate the average radiance at a given point in the image from each available exposure.

$$\ln E_i = \frac{\sum_{j=1}^{p} W(Z_{ij}) (G(Z_{ij}) - \ln \Delta t_j)}{\sum_{j=1}^{p} W(Z_{ij})}$$ (3.4)

The weighting function suppresses extreme values which may be inaccurate. The highest weight is given to the exposure where a pixel is best exposed; pixels with values towards the middle of the LDR image’s range. The equation preserves the details from all the original images and discards the areas which are badly exposed with low detail. The output image has a range of radiance values much greater than any of the original images could produce alone.

The number of photos and level of exposures required to create the HDR image can be varied depending upon what is available and what information is required in the final photograph. The minimum number of photos required to acquire the camera’s response function is two. To construct the radiance map a minimum of photos required is $r_f$, where $r$ is the dynamic range that the photographer wishes to capture in the final HDR image and $f$ is the dynamic range of the camera. In both cases more photos will provide a better result and reduce the disruptive effect of noise.

3.2.3 Other Methods of Generating HDR Images

The above method has been shown to perform well but there are also other HDR generation methods of note. Mann and Picard [75], Mitsunaga and Nayar [93], Robertson et al. [109] and Ward [146] have all developed methods for generating HDR images from a series of exposures.

Mann and Picard [75] developed the first method for the generation of HDR images from multiple aligned exposures. The technique involved generating a response curve in the form $M = a + \beta I^\gamma$ where $a$ is a basis parameter to measure dark noise, found by taking a picture with the lens cap on, and $\beta$ is set to an arbitrary constant. $\gamma$ is found by solving a regression sequence $M(I), M(RI) ... M(R^nI)$. $M(I)$ in the first image generates $M(RI)$ in the second image. A pixel with brightness $M(RI)$ is found in the first image that produces the brightness $M(R^2I)$ in the second image where $I$ is unknown. $R$ is the ratio
of exposure values between the two images and $I$ the ratio of the light that is reaching a sensor between two images. $I$ times more light exposing a pixel in image one would make it have the same value as the same pixel in image two. By selecting pixels and iteratively refining the values of $M = a + \beta I^\gamma$ a response function is generated. This model is restrictive as it cannot generate accurate curves that do not fit well to the equation.

To generate the HDR image Mann and Picard used the properties of the response curve to weight the pixel values in images. Pixels which corresponded to low flat areas of the curve are considered to be unreliable and either over or under exposed. As such, these areas are weighted less than the steep parts. The corresponding radiance value for each pixel for each of the images is weighted accordingly as they are combined in the final HDR image.

Mitsunaga and Nayar [93] have developed a technique for both the creation of the response curve and the creation of the HDR images from it. The technique is similar to the Debevec and Malik approach in that a response function is generated in the form of a polynomial approximation. The algorithm does not require precise estimates of the exposures used as it estimates the ratios between consecutive aligned exposures. The response function of a camera is related to this exposure ratio. The images are ordered in order of exposure, lowest to highest but the exact exposure is unknown. With a known ratio the response function is recovered by determining the coefficients of an N-dimensional polynomial by minimizing an error function.

Once the response curve is recovered, a preprocessing step which uses temporal and spatial averaging to obtain robust pixel measurements is applied. This makes the solution suitable for video where it is easier to adjust aperture than exposure time as it reduces vignetting effects (gradual darkening of the image towards the corners caused by a wide aperture) and temporal changes. Mitsunaga and Nayar construct the final HDR image using the same method as Debevec and Malik.

Robertson et al.’s method [109] is similar to Debevec and Malik’s. The HDR image is generated from a response curve their algorithm requires an initial calibration which determines the camera response function if necessary. Once the response curve is recovered, an HDR image is constructed from the Gaussian weighted average of images with different exposures.

Ward’s method [146] constructs HDR images with translational alignment of the images within the algorithm. This means that the camera doesn’t have to be
perfectly steady when the images are taken as the alignment phase will attempt to correct for small discrepancy between frames. To generate the response curve Ward’s method uses one of the previous algorithms such as that described byDebevec and Malik. The alignment uses median threshold bitmap (MTB). This is a binary matrix which is 0 where the input pixels are less than or equal to the median value and 1s where the pixels are greater. The median is determined using a low resolution histogram over the image. When these values have been calculated for consecutive exposures the offset between the two MTBs are obtained and the minimum distance found. If a pixel is close to the median value a threshold is used to reject its influence. Pixels close to the median may be inaccurate for aligning purposes as in a corresponding image the value may be on the other side of the median. This offset is then used to align the images before the construction of the HDR image using one of the previous methods.

3.2.4 HDR Cameras

There are cameras coming onto the market that can take HDR images directly. CCD Sensors with higher dynamic range than are currently available cheaply are being produced. One example is the SpheronHD [127] camera a 32-bit RGB line scanning camera with a 360 degree field of view. It rotates on a tripod and captures the image line by line slowly building up a HDR image of the scene using a specially designed sensor. The process is slow and to capture the full scene even at a low resolution can takes over 2 minutes. The time increases substantially as the resolution increases. A second scan of the scene on a calibrated tripod by raising the camera up a set distance can be used to obtain 3D information from the scene at millimetre resolution up to ten meters away at its full resolution of fifty megapixels. This is not ideal for the purposes of mobile platforms but demonstrates that the technology to directly capture HDR images is becoming available.

Another interesting development is Fuji’s Super CCD S3 Pro camera that has a chip with high and low sensitivity sensors per pixel location to increase dynamic range. This is not a full HDR but shows one possible solution which could be developed further to solve the problem of capturing HDR images.
3.2.5 Tone Mapping

It is impossible to display HDR images on most displays as the dynamic range of the average monitor is only 2 orders of magnitude [106]. Tone-mapping has been developed to convert a HDR image into a regular 8-bit LDR format so that they can be viewed on a conventional display.

Global Tone Mapping Operators

Global tone mapping techniques are the simplest form of tone mapping and apply the same transform to all areas of the image. The most basic form is a direct mapping from the HDR values to the 256 bit range of a LDR image. The log of the HDR value is taken and mapped linearly. This does not always generate a realistic image.

There are other, more complex, global operators which convert the HDR image to an LDR image in a way which better preserves the information in the image. A method developed by Tumblin and Rushmeier [137] does so by creating a human visual system (HVS) based colour model and converts the HDR luminance values to perceived image brightness. It attempts to preserve the overall brightness in terms of how it is observed by the viewer. They created two other techniques [136] which are perceptually guided by further analysing the human visual system and it’s response to light. The first method creates layers based on the lighting and surface properties of the scene to generate the LDR images. The lighting layers are compressed reducing contrast while preserving much of the image detail. The second method is an interactive foveal method which has higher detail in the area that the viewer is looking and reduces detail at the edges of the eyes view.

Ward [145] developed a global tone mapping method which preserves the perceived contrast of the HDR image. The operator uses the ratio between the HDR and display luminance to generate a linear scaling factor to apply to the HDR pixel values. A histogram adjustment method developed by Larson et al. [60] takes advantage of the eyes sensitivity to relative rather than absolute changes in luminance. The method incorporates models for human contrast sensitivity, glare, spatial acuity, and colour sensitivity to generate the mapping between the pixels.

A method by Drago et al. [31] presented another method for the generation of tonemaps based on human perception and can be run in real-time. The method is based on logarithmic compression of luminance values. A bias power function
is introduced to adaptively vary logarithmic bases, resulting in good preservation of details and contrast. A gamma correction method is used to improve contrast in dark areas.

**Local Tone Mapping Operators**

Local tone mapping operators are those which apply different transformations to different areas of the images based on the image’s properties. One issue with local operators is the presence of halo effects where dark areas are created around bright areas in the images and vice-versa.

Chiu et al. [19] developed a non-uniform scaling function to map HDR values to LDR values. It is based on the eye’s greater sensitivity to reflectance than luminance so that gradual changes in luminance over the image can be discarded. The technique uses a first order polynomial function to map the values. This method is effected by halos.

A method created by Tanaka and Ohnishi [129] uses a Gaussian low-pass filter to reduce the image contrast. Pattanaik et al. [98] used a multiscale representation of pattern, luminance, and colour based on the human visual system. The method has two parts, the first processes the input images and calculates the perceived contrast using a visual model and the second stage uses a display model to create an LDR from the contrast data. These methods are affected by halos.

A method where the input image is separated into large features and fine details was created by Tumblin and Turk [136]. The large features are compressed and the fine details are preserved. This method is not affected by halos but often results in low contrast image and takes a long time to compute.

The method invented by Fattal et al. [35] also works by reducing the gradient magnitude in the areas of high gradient while preserving the areas of low gradient. The human visual system is not very sensitive to absolute brightness but responds to local contrast meaning that global differences in brightness can be reduced so long as the darker parts of the image remain darker and the brighter parts remain brighter. Reducing the gradient magnitude of the whole image uniformly would remove texture caused by small gradients so to maintain these a weighting is used.

Reinhard et al. [108] have a method which applies a technique based on dodging-and-burning to the digital image. Dodging-and-burning are analog photograph techniques where dodging decreases the exposure for areas of the print
that the photographer wishes to be lighter and burning increases the exposure to areas of the print that should be darker.

Ashikhmin’s [7] method works using a multipass approach which calculates local adaptation luminance, applies a tone mapping function using a threshold versus intensity function. The final pixel values are calculated to preserve details throughout an image.

Durand and Dorsey [32] developed a bilateral filtering method which preserves edges. It reduces contrast and preserves details of an image. Choudhury and Tumblin [20] developed a trilateral filtering method which is similar. It operates in a single pass of the image and preserves the edges while reducing contrast.

3.2.6 Exposure Fusion

A method has been developed to generate a tone mapped image from a set of exposures bypassing the HDR creation stage called exposure fusion. The technique fuses a bracketed exposure sequence into a high-quality, tone-map like image. The result of this process is comparable to existing tone mapping operators. Its advantages over tone mapping include the fact that no HDR image needs to be computed often making the process faster and simpler. Also the process is more robust as the exposure values are not needed and a flash can be used with the camera.

System and process for improving the uniformity of the exposure and tone of a digital image is a patent for an exposure fusion technique [128] which averages the pixel intensities of the exposure images to generate the new values for an average image. The average intensities are binned in a histogram and a histogram equalisation process is applied to them where the average intensity of the image does not change. The values are mapped back to the average image positions to generate the final fusion image. This is one of the first simpler examples of exposure fusion.

Later more robust methods have been developed using measures which assess the pixels likelihood to be from a well exposed area of an image. Fusion of multi-exposure images [42] uses a Shannon’s entropy based measure [120] to weight the pixels in each images and generates a weighted average for the final exposure fusion image. The process divides the image space into uniform blocks and for each block selects the image that contains the most information within that block based on the entropy measure given by:
3.2. **HIGH DYNAMIC RANGE IMAGES**

\[
H(p) = - \sum_{i=0}^{n} p_i \log_2(p_i) \quad p_i > 0
\]  

(3.5)

where \( p_i \) is the probability that an arbitrary pixel in the image has intensity \( i \).

To estimate \( p_i \) for each intensity a histogram of the image is computed and for each intensity value the number of pixels with that intensity \( n_i \) is divided by the total number of pixels \( n \) (\( p_i = n_i/n \)). For a colour image the values are binned and clustered in 3D-space and the 256 most dominant clusters are selected. For each colour in the images the closest of the 256 clusters is used and the entropy is calculated. Blocks are chosen from each image and where the entropy value is highest and the blocks are blended together to generate the final exposure fusion image.

A Novel Fusion Approach of Multi-exposure Image [59] uses a genetic algorithm (GA) to improve the Shannon’s entropy based technique for exposure fusion [42]. It employs the GA to optimise the block size and width of the blending function to improve the fusion results using entropy, PSNR and average gradient of the fusion image for the fitness function. Real-time exposure fusion on a mobile computer [8] shows that the entropy method of exposure fusion can be implemented in real-time using the GPU on a laptop computer at a minimum of 20 frames per second for images of size \( 1360 \times 1024 \). The author identifies the practical advantages of the system for a human operator or surveillance system viewing a high contrast scene when compared to a single exposure which contains less information.

Exposure Fusion [81, 82] uses weighted averages of the images where the weightings are calculated based on certain properties of the image: Contrast, saturation and well-exposedness. These are each weighted, combined and normalised and then used to calculate a weighted average of the exposure images’ pixels to create a fusion image. A multi-resolution fusion is used to reduce the appearance of seams in the final image. Each of the input images is decomposed into a Laplacian pyramid and the corresponding weight map is decomposed into a Gaussian pyramid. The Laplacian pyramid of the fusion image is determined by the weighted average of the input Laplacian pyramid, where the weights are given by the corresponding scale in the Gaussian weight map. Finally the fused output image can be reconstructed from its Laplacian pyramid by using an inverse transform.
Subband Architecture Based Exposure Fusion [159] process the images in a similar way but utilises subband decomposition instead of a Laplacian pyramid. The exposures are split into non overlapping frequency subbands using Quadrature Mirror Filters (QMFs) which consists of a combination of a high-pass and a low-pass filter. The weighted averages are calculated for each subband image using the measures from Exposure Fusion [81, 82]. Gain control maps are calculated according to the simulation mechanism of gain control in the human visual system in order to remove the halo artefacts caused by the nonlinear distortion introduced by the subband decomposition stage. The subband images are then summed to reconstruct the final fusion image.

Motion-blur-free exposure fusion [131] removes the blur in exposure fusion images that is introduced by movement during long exposure image capture which are often used to capture the low light areas of high contrast scenes. The technique generates the exposure fusion image [81, 82] and utilises the shortest exposure as a sharpness reference to improve the result as the shortest-exposed image is usually the least affected by motion blur. This image fusion procedure combines a photometrically calibrated short-exposed image and the result of the exposure fusion algorithm. The images are converted to YUV and the colour and intensity channels are fused separately. Intensity is fused from the two images in the wavelet domain. Colours are captured better in the blurry fusion image and are taken from that in all areas except those deemed to be blurry using the difference between the result of the intensity fusion and the intensity component of the blurry fusion image. The pixels are weighted based on their blurriness and selected for the final image from the fusion image or short exposure image based on this. The results show that areas blurred through motion in standard exposure fusion images are sharper.

Exposure fusion based on steerable pyramid for displaying high dynamic range scenes [143] addresses the problem of slightly misaligned exposure images resulting in artefacts such as haloing. In order to reduce the influence induced by the misalignment, a novel shift-invariant and rotation-invariant steerable pyramid-based exposure fusion (SPBEF) algorithm is proposed. The R, G and B channels are not processed separately but the chrominance information is obtained by the average of the median two images in a set of input exposure images. Next, the luminance values for each pixel are calculated from the source images are then transferred to frequency domain using a combination of high-, low- and band-
pass filters. They are fused incrementally to the subimages with adjacent exposure time. The fusion uses different fusion rules in different frequencies. This is repeated until a single set of frequency domain images remain and they are transformed back to image space. The final colour image is generated by combining the data of the fused luminance images and the chrominance information. Experiments show that SPBEF can give comparative or even better results compared to other algorithms.

### 3.2.7 HDR Video

The generation of HDR video is a difficult problem to solve as to process multiple video stream of different exposure and combine them into a HDR representation is computationally expensive. Problems also include the alignment of the images, adjusting exposure length on a camera means that frames will be misaligned temporally and if illuminance reaching the sensors is varied by adjusting the aperture rather than exposure time the depth of field will vary between images and some areas may be blurred. Motion blur in a moving camera system could also cause problems. Some work has been done in the area of creating HDR video.

One method developed by Kang et al. [57] uses an off the shelf video camera and rapidly adjusts the exposure of each frame. The exposure settings alternate between two different values and are continuously updated to best fit scene changes. The next stage generates a HDR image from consecutive frames using image warping and motion estimation to align the frames. The response curve is generated using Mitsunaga and Nayar’s [93] method due to it not requiring exact exposure values. The images are tone mapped using Reinhard et al.’s [108] method and a HDR video constructed. This is all carried out off-line. The results show that the method generates a wider dynamic range than a standard video camera and the resulting images contain more detail in light and dark areas simultaneously. The method is limited by the use of a single camera which therefore reduces the frame rate as each frame is compiled from consecutive images. Also, the HDR is only built from two exposures whereas to capture HDR images with high detail the use of five or more exposures is generally recommended [106].

Another method [142] uses a split aperture camera to capture the HDR video which contains three CCD sensors that capture data through the same lens using a prism. The sensors have different exposures due to a thin film neutral density filter placed in front of each with different transmittance values so the light reaching
each varies. The images are combined into a HDR images using a technique such as proposed by Debevec and Malik [27]. They are aligned so as to receive the same view of the scene. The tone mapping operator is based on the gradient domain technique proposed by Fattal et al. [35].

### 3.2.8 HDR Feature Matching

There is not much work on detecting, extracting and matching features from HDR images even though the benefits are clear for computer vision. The larger dynamic range allows objects to be seen and therefore detected in areas of high contrast images where a LDR exposure could not capture information in all areas. This has an advantage in surveillance or for robots exploring an environment. Two papers stand out in:

*Image Matching Using High Dynamic Range Images and Radial Feature Descriptors* [52] uses HDR images to match HDR images of buildings constructed from multiple exposures. A junction detector algorithm is used for detecting the features in the image which looks for certain types of joins on the buildings e.g. a T-junction. The features are described using the wedge descriptor which is modified to adapt to high dynamic range images. It shows an improved performance over the SIFT for this specific task. The key problem identified in the paper is that the shadows can change throughout the day, but without detection of the shadows themselves it is difficult to suppress the effects of the shadows on the feature matching.

*Robust point matching in HDR image through estimation of illumination distribution* [22] addresses the problem of the highlights and shadow in a HDR image which can cause instability if feature matching as they are not real world edges. The method uses a Gaussian mixture model estimation of the illumination of the HDR image and introduces an algorithm to select its parameters. From this the shadow and highlight parts of the image are recovered and their position is identified in the image from the illumination distribution. From this a “material colour” image is generated which is a novel type of tone map and when compared to standard tone mapping operators for the SIFT feature matching it is shown to perform better.
3.3 3D Reconstruction

The addition of 3D information to the features extracted from multiple images will help differentiate between different objects in a scene. The data will allow clustering of points to be achieved and objects to be defined as 3D point clouds. There are two complementary methods of generating this data, described below that are being researched.

3.3.1 Stereo Vision

Stereo vision is the generation of 3D information using the difference between two or more cameras viewing the same scene from different viewpoints. The principal behind stereo vision is that by calculating the disparity between known points in two aligned images and by calculating the relative position of the cameras that took the photographs the distance of the points from the cameras can be calculated. The stages of calculating the 3D information from two images are as follows:

1. The primary stage of the process is the removal of distortions from the image. Pin cushion and barrel distortion are caused by the optical properties of the camera and need to be corrected to ensure that the observed image is purely projectional. This is done by knowing the properties of the camera that can be calculated by taking pictures of a uniform planar checker board style grid. When viewed through the camera lens, lens distortion causes the straight lines at the edges of the squares to appear curved. From a small set of images of the grid the intrinsic camera parameters (focal length, principal point, distortion coefficients) as well as the extrinsic parameters (3D position of the pattern for the image) are calculated [91, 123, 149]. This allows a transformation to be calculated and applied to further images to correct for distortions. The procedure is fully automatic as the corners of the checkerboard pattern can be easily detected.

Methods have also been created which attempt to calculate the intrinsic parameters of the camera in scenes by making assumptions about the scene. Man-made scenes such as those containing buildings contain a lot of straight edges and by detecting these edges one can calculate the parameters required to remove the distortion in a similar manner as to that with the
grid [30]. This is less reliable as the assumption that lines are straight may not always be true.

2. The images are then projected back to a common plane to allow comparison of the image feature pairs. This is known as image rectification. The fundamental matrix defines the relationship between the cameras and the transformation to one view from the other. This is required so the images can be aligned and the measurements taken. It can be calculated by finding features which match between the images using an algorithm such as the SIFT with little impact on the accuracy compared to knowing the matrix from calibration of the cameras in advance [39]. Once at least seven pairs of corresponding points have been found between the images a $3 \times 3$ fundamental matrix which relates corresponding points in stereo images is calculated [45, 72].

3. The displacement of relative features in the aligned images is measured to calculate a disparity. Using the disparity relative depth of the features can be calculated using the following equations:

$$
\begin{align*}
  x &= \frac{uz}{f} \\
  y &= \frac{vz}{f} \\
  z &= \frac{fb}{d}
\end{align*}
$$  

(3.6)

$x$, $y$ and $z$ are the 3D world coordinates, $f$ is the focal length of the camera, $u$ and $v$ are the image coordinates in the first image, $d$ is the disparity between the points in the two images and $b$ is the baseline between the two images calculated using the fundamental matrix.

### 3.3.2 Structure from Motion

This is the generation of 3D information from the motion of the camera and the difference in the position of points in consecutive frames of a moving camera. All the input information is calculated that a structure from motion (SfM) technique would require in a system using features for matching while it is moving. SfM could be used in conjunction with the stereo techniques to generate more accurate information. The basic idea behind SfM is the same as that of stereo vision but instead of having two cameras a displaced image is generated from the movement of a single camera. The process is fundamentally the same as the calculation described for stereo vision the only difference being that the second image is
taken by the same camera and the fundamental matrix will change for every pair depending on the movement of the camera.

A recognition system on a mobile platform will already have single or multiple cameras in place for generating image data and corresponding features are likely to be calculated for each frame for feature matching. This means the building blocks for adding 3D information to features are in place and as such should create little overhead while providing valuable information to assist object discrimination and classification.

3.3.3 Bundler and Bundle Adjustment

Bundler is a system developed to generate structure from motion for unordered image collections. It extracts features from sets of unordered images of a scene and uses them to simultaneously calculate the parameters of the cameras their relative motion and the 3D coordinates describing the scene geometry. This is done using a technique known as bundle adjustment which is almost always used in 3D reconstruction from feature sets.

The system calculates the relative position from which each camera took the photograph and allows the photographs to be viewed from the correct position relative to others. This creates a pseudo 3D view of the scene by navigating between the photographs in their correct relative 3D position giving different perspectives of the scene [67,125,126].

Unsupervised 3D object recognition and reconstruction in unordered datasets by Brown and Lowe [13] worked on generating a 3D cluster of SIFT points for image matching. The paper presents a system for fully automatic recognition and reconstruction of 3D objects in image databases. It builds a model using the subset of features from a set of images of a rigid object from different views by calculating the transformations between the images using RANSAC and calculating which points correspond between views. Bundle adjustment is used to generate the structure and location of the object.

Bundle adjustment finds the parameters that most accurately predict the relative locations of the observed points in the set of available images. For two images where features have been matched the correspondence between features is known. The algorithm attempts to adjust the estimated position of one camera and its estimated intrinsic parameters so as to minimise the distance between its set of features and the other camera’s set of features. The term bundle refers
to rays of light emanating from each feature towards the camera centre and the problem is solved when these lines each cross the corresponding ray from another bundle.

The parameters that need to be found are the 3D orientation (three parameters), the camera centre (three parameters), and the focal length (one parameter). This is done by refining a set of initial parameter estimates. The aim is to find the projection matrices \( \hat{P}_k \) and the 3D points \( \hat{M}_i \) for which the mean squared distances between the observed image points \( m_{ki} \) and the reprojected image points \( \hat{m}_{ki} \) is minimized. For views and points the following criterion is minimized:

\[
\min_{\hat{P}_k, \hat{M}_i} \sum_{k=1}^{m} \sum_{i=1}^{n} D_e(m_{ki}, \hat{P}_k \hat{M}_i)^2
\]  

(3.7)

where \( D_e(x, y) \) is the Euclidean distance between the image points represented by vectors \( x \) and \( y \).

This optimisation is generally carried out using the Levenberg-Marquardt Algorithm (LMA) \([63,67]\) to find the minimum values. This is due to its robustness to choice of starting parameters and its ability to converge quickly due to its effective damping strategy. One disadvantage of this technique is that it finds local minima and therefore sometimes does not find the optimal solution.

The technique linearly and iteratively adjusts the parameters; \( \hat{P}_k \) and \( \hat{M}_i \) to find a solution. The linear solution is found by second-order Taylor series expansion around \( \hat{P}_k, \hat{M}_i \) and an approximation of the second-order derivative matrix (the Hessian) by Jacobian products.

The resultant linear system is formed by stacking the variables \( \hat{P}_k, \hat{M}_i \) into a vector \( \mathbf{x} \), and the error functions into a vector \( \mathbf{e} \), let

\[
J \equiv \frac{\delta \mathbf{e}}{\delta \mathbf{x}}
\]  

(3.8)

\[
H \equiv J^\top J
\]  

(3.9)

The linear system therefore defined as:

\[
H \Delta \mathbf{x} = J^\top \mathbf{e}
\]  

(3.10)

The LMA method augments \( H \) by adding \( \lambda \text{ diag}(H) \), where \( \lambda \) is a small positive multiplier. This is the damped version of the equation which cause larger
movement along the directions where the gradient is smaller thus leading to fast convergence:

\[(J^T J + \lambda \text{diag}(H)) \Delta x = J^T e\]  

(3.11)

### 3.3.4 Using 3D Information for Feature Matching and Object Recognition

Attempts have been made to incorporate 3D information into feature matching and object recognition to improve the robustness of the algorithms. Work in this thesis focuses on using 3D data to enhance the feature matching process and use the depth information to weight the feature matches so that the more accurate matches can be selected automatically. This section summarises work which use 3D information for detection of objects in scenes.

Integrating stereo vision with appearance based recognition to increase accuracy and efficiency has been researched. *Multi-camera Object Detection for Robotics* [21] introduces a method for using multiple cameras to simultaneously view an object from multiple angles and combine the camera views using a probabilistic method. This can significantly improve accuracy for detecting objects. The technique is trained on a “single image single view” of positive and negative images of a target object and learns and encodes object for multiple poses and backgrounds using various histogram patches including HoG [25]. The classifier is trained on this data and has different classifiers for various poses of the object. The system works by using this classifier with a sliding window for each view and then determine the correspondence between pairs of detections. It outputs the probability that the target object is contained in the sub-region at the pose for which the classifier was trained. The detections are then combined to make more confident predictions about the label of the object by combining the probabilities output by the classifiers for the corresponding regions. The results show an advantage over using the classifier in a mono-view scenario.

*Using stereo for object recognition* [47] uses the depth information to generate a scale prior which is used to reduce the areas searched by the matching algorithm. By knowing the approximate scale of an object the window used to detect the object can be better fit to the object which reduces the number of false positives. A silhouette classifier and the chamfer distance, a measurement of the distance
between curves, is used as the objects being identified are generally untextured. It uses depth to infer at what scales to sample, allowing a reduction of samples by up to eighty percent. The results indicate that a prior on scale can be utilised to increase the accuracy, efficiency, and robustness of object localisation and it can is shown to work for a variety of classification models.

Multiple Viewpoint Recognition and Localization [46] is a method for labelling objects in a scene and takes aligned stereo inputs to determine a set of 3D object locations. The paper identifies objects in each image using a classifier and treats every detection as a ray emanating from the camera. Where objects are detected multiple rays should align and mutually agree upon position, location and scale of a single 3D object. It uses a likelihood from the combined information to indicate if the object is likely to be in a position based on appearance and pose from multiple images. The results show a significant improvement in performance when compared with single image labelling and the system can be used as a framework for stereo labelling using a wide variety of classifiers.

The above three papers relate most to the work in this thesis but the use of stereo information for object detection is widespread. Robust Multi-person Tracking from a Mobile Platform [33] shows how a two stage process can be used for robust pedestrian tracking in a cluttered urban environment. The process estimates possible detections using scene geometry initially and then reduces this set based on temporal information across frames. Multi-cue Pedestrian Detection and Tracking from a Moving Vehicle [40] also detects pedestrians with stereo cameras by integrating techniques for sparse stereo-based ROI generation, shape-based detection, texture-based classification and dense stereo-based verification to filter out false detections which contain too much of the background.

Combining data across multiple frames of a video to obtain depth information for detection and tracking has been studied. Monocular 3D Pose Estimation and Tracking by Detection [6] uses a single camera to recover human pose. The algorithm makes an initial detection and estimation of a person’s pose in a single frame. This is repeated in subsequent frames and tracklets are generated between the frames following the persons movement and pose. Finally a third stage uses the estimated positions of the person in the set of frames to recover 3D pose. Monocular 3D Scene Modelling and Inference: Understanding Multi-object Traffic Scenes [152] takes a single mobile camera as input and uses that to infer depth from a scene. This is used to improve the reliability of weakly detected objects
and to prune false positive detections by using information obtained from low level scene labelling and object detection for object tracking.

*High-accuracy 3D Sensing for Mobile Manipulation: Improving Object Detection and Door Opening* [103] shows how using how high-resolution depth information obtained from a laser can be combined with visual imagery to improve the performance of object detection beyond what is achievable with single images alone. *Integrating Visual and Range Data for Robotic Object Detection* [43] also augments a 2D object detector with 3D information from a laser range scanner to produce a multi-modal object detector”.

### 3.4 Summary

The previous two chapters have covered the main areas relating the work in this thesis and may be summarised as follows:

- Feature Detectors.
- Feature Descriptors.
- Feature Matching.
- *A contrario* methods.
- Feature Detector Parameter Optimisation.
- High Dynamic Range Images.
- Tone Mapping.
- Exposure Fusion.
- 3D Reconstruction.
- 3D Object Detection.

### 3.5 Focus of Thesis

The following chapters outline the work conducted and present the results of the research:
• **Chapter 4.** An overview of some of the key principles used in this thesis and a background of the functionality of the SIFT.

• **Chapter 5.** Analysing the SIFT parameter space using sweeps and a genetic algorithm to assess the effect of the many parameters used to control the SIFT.

• **Chapter 6.** Using multiple images of different exposures to generate a feature set which encompasses a larger dynamic range than a standard set of features typically generated from a single low dynamic range image.

• **Chapter 7.** Using stereo views to generate feature match confidence weightings based on the *a contrario* methodology to provide an indication of whether a match is correct.
Chapter 4

Preliminary Analysis of the SIFT

This chapter gives an overview of the techniques and an analysis of the algorithms which should be used when analysing feature descriptors. The focus is on the Scale Invariant Feature Transform.

4.1 The SIFT Descriptor

The number of possible unique, non-matching, features that exist are given by the number of bins and the size of the pixel depth. For a standard SIFT descriptor vector of length of 128 encoding values in the range 0 to 255 the size of the features space is:

\[ 256^{128} \approx 1.8 \times 10^{308} \]

This is a very large number. However, in normal use the space is not utilised completely and a much smaller area is actually populated by SIFT features. Figures 4.1 to 4.3, 4.5 and 4.7 show various methods used to analyse large sets of SIFT features in order to understand the feature space and how the Lowe’s nearest neighbour matching technique functions (Section 2.4.1).

Figure 4.1 shows the mean, mode and standard deviation of two million features extracted from images randomly selected from Flickr. Each dimension of a feature is a bin where local gradients of a given direction are accumulated relative to the orientation of the feature. The orientation is chosen by selecting the direction across the feature patch in which the gradient is highest and as such it can be seen that the central bins corresponding with this direction have the largest

95
values. The Gaussian weighting of the feature means that the outer regions of the feature have lower values. Overall the most common values for the feature descriptor are close to 0 apart from the four central values aligned with the orientation shown by the modal values in Figure 4.1(b). The figure shows that the features have a tendency toward low values and as such it is clear in this study that the full feature space is not fully utilised.

Figure 4.1: This shows the mean, mode and standard deviation for two million feature vectors extracted from over a thousand random Flikr images. The red ticks on the tops of the graphs represent the dimensions which correspond with the feature orientation and recur every eight bins.

Figures 4.2 and 4.3 show the distribution of the features for the same two million features. They also show that for each dimension the tendency is for the values to be low. Figure 4.3 show the common distribution shapes for the feature dimensions. The most common look like bin 42 where there is a large peak at the 0 and the frequency values drop to very small values after that. For these cases 0 is by far the most likely single value to occur. Only a few vary from this toward the centre of the descriptor. Bin 41 still shows a tendency towards
4.1. THE SIFT DESCRIPTOR

lower values but the effect of the orientation and Gaussian weighting makes the
distribution so that the modal value is 7. The only bins which show a vastly
different distribution are 40, 48, 72 and 80. These show a peak near the middle
of the distribution due to the combination of high Gaussian weighting and that
they are aligned with the orientation of the feature. The example bin 40 has a
modal value of 123.

Overall these graphs show that the distribution of the SIFT descriptor is
uneven and that features with high values are far less likely to occur. Given these
input feature it can be seen that the realised feature space appears to be a much
smaller subset of the total space.

![Figure 4.2](image-url)

Figure 4.2: This shows the frequency distribution for each dimension of two
million SIFT feature vectors. The graphs are log scale and each sub graph shows
the distribution for 8 dimensions. They are colour coded to show the direction
each line refers to. The grid represents the same $4 \times 4$ used to construct the
vector as shown in Figure 2.10.

4.1.1 Distance Distributions

Given that the distribution shows that the SIFT descriptor has a tendency for
lower values it is important to determine the distance between the features. The
maximum distance between two features is $\sqrt{128 \times 255^2} = 2885$
CHAPTER 4. PRELIMINARY ANALYSIS OF THE SIFT

Figure 4.3: This graphs shows a sub set of Figure 4.2. It shows bins 40 to 47 the areas marked by a red box without the log scaling. The red line is the mean value and the green line is the modal value. Most distributions of the SIFT features dimensions look like that for 44. The dominant direction of the feature and the Gaussian weighting affecting bin 40 and its neighbours cause these bins to have a median greater than zero.

Figure 4.4: A flow chart demonstrating the steps of the SIFT algorithm. It shows the stages between the initial image to the final set of 128 dimensional vectors.
4.1. **THE SIFT DESCRIPTOR**

By taking a set of randomly selected features and calculating the distance between each feature and every other the space has been shown to be much smaller. Figure 4.5 shows the distribution of the features and it can be approximated by a Gaussian distribution. Using this distribution it can be estimated that 99.7% of the feature distances lie within the region between 364 and 716 of each other. This requires a difference of between 32 and 63 per dimension (out of a possible 255) with a mean distance of 48. This shows that the randomly measured SIFT features are concentrated in a smaller area of the space as the measured maximum distance that occurred is 708 (explained by the small fit error between the normal distribution and the data) which is 4.1 times smaller than the total distance space. Similar observations have been independently validated by *Binary SIFT: Fast Image Retrieval Using Binary Quantized SIFT Features* [100] for the squared feature distance.

![Figure 4.5: The Euclidean distance distribution for 100,000,000 random feature pair combinations. Gaussian fit with a mean of 540 and $\sigma$ of 58. The $R^2$ error for the fit is 97% and the variance is low for the main peak of the distribution. The error is introduced by the fact that there are no features that have a distance above 708 and more have lower values. The lower distance features are those which match. The graph shows that with a high likelihood (3$\sigma$) two random feature will have a distance between 364 and 716 of each other.](image)

Figures 4.6 and 4.7 show the distributions of all feature distances and the matching feature distances. The first is for the random set of features and the second for a set generated from images which are known to have corresponding areas. The blue line shows the distances between each feature and every other in the data set. Both distributions have the same peak at approximately 540. This is explained for the images which match by the fact that most of the compared features are from non-matching parts of the images.

The red line shows the features which match using Lowe’s nearest neighbour
method. This is where the main difference between the two graphs is visible. As expected the features from the images which contain corresponding areas have a larger number of features within the match distribution (the red line). Overall it contains more features with smaller distances between them. This is why the SIFT is successful as the algorithm is unlikely to generate very close matches by chance.

The most likely features to occur by chance will be at a distance of approximately 540 from each other so close matches can be identified as corresponding regions within the images with a high confidence. False positives do occur as is shown by the occurrence of matches in the random data but this is a thresholding problem and the user can favour more true positives at the expense of more false positives. Areas that disagree with the Gaussian distribution hypothesis for the full set of feature distances are also clear. The random data fits to the Gaussian distribution better as the distribution does not have as many lower value matches.

The matches also form an approximate Gaussian distribution and are shown in Figure 4.8 which shows the relationship between the matched features more clearly. Due to the random data being less similar to other features the matches that are caused by this data make up a subset of the matches from the corresponding image data.

The green lines show the distribution of all features divided by 0.8 as is done with the nearest neighbour matching. This shows that as the chance of a nearest neighbour with a distance ratio of 0.8 increases the number of matches decreases until no matches occur. The closer the match is to another feature the less likely it is that the feature has a nearest neighbour which is greater than 0.8 times the feature distance.

The SIFT is successful because the width of the distance distribution of the random features is smaller and does not contain as many lower values (less than 250 in Figure 4.8). It shows that the SIFT features are distinct enough to distinguish between random noise and real matches and close matches are unlikely to occur by chance.

### 4.1.2 Why the Feature Space is Not Filled

A main reason why the actual features do not fill the whole feature space is because of the Difference of Gaussian (DoG) scale space pyramid and the Gaussian blurring during the feature creation stage. The DoG subtracts the neighbouring
4.1. **THE SIFT DESCRIPTOR**

Figure 4.6: The matching distance for 100,000,000 combinations of random features as shown in Fig 4.5 with a log scale for the y-axis. The blue line is the distance distribution of all the feature distances. The red line represents the distribution of the features which have matched and the green line represents Lowe’s nearest neighbour threshold where the distribution has been multiplied by 0.8.

Figure 4.7: The matching distance for 858,802,832 features from corresponding pairs of images taken from the Oxford Robotics feature matching dataset used by Mikolajczyk et al. [86]. The colours represent the same thing as in Figure 4.6 and it has a log scale for the y-axis.
Gaussian blurred images within a scale space pyramid. Due to much of the image being similar after Gaussian blurring has been completed the subtraction reduces much of the DoG image to a value of zero. Zero values will not be affected by any multiplication within the SIFT process and will remain as zero when normalisation takes place whereas non-zero values will be scaled. Figure 4.9 shows the features extracted from an image and the distribution of their values with a high peak at zero. Figure 4.10 shows a part of the DoG stage of the SIFT algorithm. The resulting image from the subtraction of the two Gaussian blurred images is sparse in detail and only contains non-negative values where there are edges. The histogram shows the distribution of pixel values for the DoG image and the high count pixels set as zero. Figure 4.11 shows the features extracted from that image at approximately that scale. The zoomed in area shows that much of the features’ cover an area of zero gradient resulting in the low values in the feature space.

The second reason for the low values is the Gaussian blurring at the feature vector generation stage of the SIFT algorithm. This reduces the magnitude of the values at the edge of the image patch used for the descriptor thus reducing the values in the final vector.
4.1. THE SIFT DESCRIPTOR

Figure 4.9: This shows a set of SIFT features extracted from a test image. The histogram on the right shows their distribution, note the peak at zero.

Figure 4.10: This shows two blurred images from the scale space used to extract the SIFT features. The result of the Difference of Gaussian is shown on the right and the histogram shows the high count for low pixel values.
4.2 Image Ground Truth: Homography and Annotation

Two systems are used for measuring the precision of matches between images. This is necessary as to test the algorithms as the ground truth must be known to decide if matches are correct or incorrect.

4.2.1 Homography Based Match Checking

A homography between two images can be estimated. This is obtained in one of three ways; obtaining a dataset with the ground truth homography already estimated, using feature correspondences generated from an algorithm like the SIFT, or manually indicating corresponding points between the images. For the latter two methods, eight corresponding points are required and the results of each are checked by overlaying the warped images to confirm that the homography is accurate. Once a homography has been estimated any points that are matched between the pairs of images can be multiplied by the homography matrix to check the position that it should match to in the other image. This can therefore be used to identify correct and incorrect matches. Given two images, $I_a$ and $I_b$, and
a homography $H_{ab}$ that relates the first image to the second image the position and scale of a feature, $p_a$, in $I_a$ can be found in $I_b$:

$$p_b' = H_{ab}p_a$$ (4.1)

As the homography is an estimate of the relationship between the images there is a chance that two features which are the same may not align exactly. This is compensated for by allowing some deviation in position. Given a point $p_b'$ which has been projected from $I_a$ to $I_b$, the following overlap error, $\epsilon$, defines if it is similar enough in position and scale to $p_b$ for it to be deemed a match. This is the ratio of the intersection and union of the feature regions, $R$.

$$\epsilon = 1 - \frac{R_b' \cap R_b}{R_b' \cup R_b}$$ (4.2)

The features are deemed to be a match if $\epsilon < 0.5$. 6$\sigma$ is the approximate pixel radius occupied by the SIFT feature descriptor [70, 140] and is used to calculate the overlap. A study of matches and mismatches has shown that using this measure consistently demonstrates near perfect results [86].

Figure 4.12: Examples of pairs of feature regions with various amounts of overlap and their approximate overlap error, $\epsilon$. The image is from [85] and demonstrates the overlap error for Harris-Affine features.

### 4.2.2 Annotation Based Match Checking

The second method that is used is an annotation system where corresponding areas across pairs of images are marked up using boxes. This is less accurate as mismatches may occur within the annotated regions. It gives an estimate of the success of a matching algorithm and is sometimes necessary, instead of the preferred homography method, when the images are non-planar and an accurate homography cannot be calculated.
4.3 Metrics for Measuring Feature Matches

Various metrics are used to analyse the results of the matching algorithms used in this thesis. For a set of points matched between two images the recall, precision, $1 - \text{precision}$ and the correspondence ratio give measurements of performance.

Recall is the ratio of correctly matched features to the number of possible matched features between the images. A correspondence is any feature from the first set which if matched correctly has a corresponding feature in the second image. These are all the features that would align if the first image was warped and overlaid on the second image. This number represents the proportion of true positives which are correctly identified and highlights miss matches (false negatives). This value should be maximised.

\[
\text{recall} = \frac{\# \text{ true positives}}{\# \text{ true positives} + \# \text{ false negatives}} \tag{4.3}
\]

Precision and $1 - \text{precision}$ are commonly used. The former reflects the ratio that a positive match is correct and the latter gives the ratio of incorrect matches from a set of matched features. In terms of feature matching these represent how precise the matches between two images are as a pair of images may have many correct matches between them but this is only worthwhile if they can be identified and trusted.

\[
\text{precision} = \frac{\# \text{ true positives}}{\# \text{ true positives} + \# \text{ false positives}} \tag{4.4}
\]
4.3. METRICS FOR MEASURING FEATURE MATCHES

\[ 1\text{-}\text{precision} = \frac{\# \text{ false positives}}{\# \text{ true positives} + \# \text{ false positives}} \quad (4.5) \]

The correspondence ratio is a novel feature matching metric and identifies how many of the extracted features from an image match to another image. This gives an indication of the similarity of the two images given a known false positive rate for an algorithm. If the ratio is above that of a known false positive value then the images may be similar and this is a computationally cheap way of identifying images likely to correspond. The ratio can give an estimate of more accurate metrics such as \(1 - \text{precision}\) and \(\text{recall}\) but cannot identify which matches are inaccurate.

\[ \text{correspondence ratio} = \frac{\# \text{ true positives} + \# \text{ false positives}}{\# \text{ features extracted from target image}} \quad (4.6) \]

The correspondence ratio normalises the results of matching features. An image with a high number of matches to a scene may appear to be a good match but this is not the case unless it is a significant percentage of the total features in that original image. If there are more possible features (possibly due to a larger image size) there is a higher chance of false positives occurring so the ratio adjusts for this and allows corresponding images to be identified. Table 4.1 demonstrates how the correspondence ratio outperforms counting the number of matches for object detection. A high ratio depends on the target image having a high percentage of matches and does not depend on the number of features in the scene image only the number of matches to the scene. This is useful for detection as a scene with one feature will have one possible match and result in a low ratio as will a scene with a high number features with a low number of matches. Figure 4.14 shows an example of object detection in a video sequence using the correspondence ratio.

The correspondence has been shown to be correlated with \(\text{recall}\) and \(1 - \text{precision}\) for a set of images used in Chapter 6. From 308 sample sets of features matches the Pearson product-moment correlation coefficient (PPMCC) between the correspondence ratio and \(\text{recall}\) values is 0.93. Between the correspondence ratio and \(1 - \text{precision}\) the PPMCC is −0.74 compared to −0.79 for the \(\text{recall}\) and \(1 - \text{precision}\). This example shows high correlation between the metrics and the related graphs are shown in Figure 4.15. The metric works well above
Figure 4.14: This shows a graph of the correspondence ratio on a video sequence where an image of a target object is being matched to each of the frames. The spike correctly identifies the location of the object in the sequence.

<table>
<thead>
<tr>
<th></th>
<th>Matched</th>
<th>Total</th>
<th>Correspondence</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image 1</td>
<td>132</td>
<td>2324</td>
<td>0.05</td>
</tr>
<tr>
<td>Image 2</td>
<td>51</td>
<td>211</td>
<td>0.24</td>
</tr>
</tbody>
</table>

Table 4.1: Table showing examples of matches and the resulting correspondence ratio. If positive correspondence between the images was identified between the images based on the number of matched features Image 1 would be identified as the best match. Image 2 is actually the correct match and based on the correspondence ratio this is clear. The reason for the higher number of matches in Image 1 is that there is a higher number of possible matches that can occur.

A baseline of noise as matches will occur when matching to a random image with no corresponding points. Thus, above this baseline a strong correlation between the more precise metrics and the correspondence ratio can be shown. A high correspondence ratio between two images is likely to relate to more correct matches when using Lowe’s nearest neighbour matching as the ratio for false positive matches remains fairly consistent and low. A very low value will not be reliable as is shown by Figure 4.14 as this is when noise and false positives occur.

The graphs in Section 4.1.1 show the distance distribution for matched features and explain why the correspondence ratio is successful. Close matches are more likely to be from images which correctly match then from random noise. The feature distribution for correctly matching images is larger than for that of small images so if the correspondence ratio rises it is more likely that the increase is caused by a correct match from a corresponding image rather than by noise. Figure 4.8 shows this as the distributions are normalised by the number of matches than a larger increase in matched features is likely with corresponding images. The correspondence ratio for the true matching data is 0.068 whereas it’s 0.015 for the random data, which is an identifiable difference and useful for discrimination.
Increasing the number of random features will increase mostly the non-matching area of the distribution and these features generally do not match. So to increase the ratio of total matches to all features the features have to have a low distance to features in the target image otherwise they are unlikely to effect the distribution and match to the target. These low distance features more commonly occur in images with matching regions as is shown in Figure 4.7.

Figure 4.15: Example correlation graphs between correspondence ratio and other metrics. (a) is the correspondence ratio and recall (b) correspondence ratio and $1 - precision$ and (c) is the recall and $1 - precision$. The values are the Pearson product-moment correlation coefficient.

4.4 Detection of Objects In Video by Noise Subtraction - A Contrario Methodology

This section outlines how the a contrario methodology, described in Section 2.5, has been adapted for the use of detecting which sets of SIFT features matches are likely to be correctly identifying a target object. This is a novel contribution as the a contrario matching as it has the advantage of not requiring a threshold for the correspondence ratio to be determined in advance.

4.4.1 Background Model

The a contrario methodology is based on using a background model, a random noisy input, which can be used in comparison to the actual target to determine when an event is meaningful. An event is meaningful if its number of occurrences is very small in a random model. Therefore when matching features the number of feature matches that occur between a target and a scene can be deemed significant if the matches to the scene by the target are greater than the matches to the
scene by the background model. In the case of feature matching the background model has been generated by automatically selecting random images from Flickr and extracting features until a predetermined number has been reached. These features are essentially random but there may be bias towards more commonly photographed and shared scenes; people, landscapes etc. A comparison can be made of the correspondence ratio between the randomly selected features and the target image to determine if the correspondence ratio relating to the target features is meaningful or if it was likely to have occurred at random. This means that a threshold can automatically be determined for each scene image by comparison rather than setting it statically. If the image is matched to an image with a particularly high feature count which has a higher chance of matching then the background model will also account for this.

Figure 4.8 shows that there is a likelihood of incorrect matches with random noise. If the matches from the background model and the target image have a similar correspondence ratio it is likely that they have the same distribution of matches and therefore the matches are likely to be incorrect. Whereas if they are different they are less likely to be noise as the distribution for matches from corresponding images is larger. If the random distributions in Figure 4.8 is subtracted from the distribution of matches from corresponding images then a significant area remains which shows the difference in response between correctly matching images and random noise.

To determine the number of random features required for the background model a study was carried out by randomly extracting SIFT features from Flickr and storing between one thousand and one million. These were then matched to 1062 scene images and the average and standard deviation of the correspondence ratio was calculated. The results of this experiment are shown in Figure 4.16. One to many feature matching is used for this, which is explained in the next section. The results show that for matching the average and the standard deviation become stable for a background model of 5000 features or greater. The maximum random correspondence ratio when using a sufficiently large background model (greater than 5000) is 0.033 (using a statistical significance of 1σ) assuming a Gaussian distribution. Based on this a target image matching to these scenes should have at least a greater correspondence ratio than 0.033 to be considered a possible match. The approximate Gaussian distribution of the correspondence ratio for noisy matches is shown in Figure 4.17 and as such using the standard
deviation is a valid method of setting a threshold.

![Graph showing average results of matching random sets of features extracted from Flickr to 1062 images. The number of features extracted is given by the x-axis and the y-axis show the correspondence ratio. The graph shows that given larger sets of random features the correspondence ratio for the SIFT is fairly constant and that a background model of at least 5000 features should be used as this is where the graph starts to stabilise. A larger number may be preferable.](image)

Figure 4.16: The average results of matching random sets of features extracted from Flickr to 1062 images. The number of features extracted is given by the x-axis and the y-axis show the correspondence ratio. The graph shows that given larger sets of random features the correspondence ratio for the SIFT is fairly constant and that a background model of at least 5000 features should be used as this is where the graph starts to stabilise. A larger number may be preferable.

### 4.4.2 One to One and One to Many Feature Matching

When matching features there are multiple possible techniques described in Section 2.4 but for any method a decision can be made about whether a multiple feature in a target image can match to the same feature in the scene image. The commonly used technique proposed by Lowe [70] using the nearest neighbour technique will allow this to take place meaning that the number of features that can match to the scene image is bounded by the number of feature that have been extracted from the target image. The alternative is to identify features which match to the same feature and select only the feature which has the closest descriptor. The number of matches is therefore bounded by the number of scene features. Throughout this thesis these matching techniques will be identified as **one to many** and **one to one** respectively.

With respect to *a contrario* image noise subtraction the selection of these techniques can make a significant difference. The **one to many** techniques allows for a background model to be used with the *correspondence ratio* whereas the
CHAPTER 4. PRELIMINARY ANALYSIS OF THE SIFT

Figure 4.17: The distribution of the correspondence ratio for the million features matched to the video sequence used in Figure 4.16. The distribution has an approximate Gaussian distribution with an $R^2$ fit of 0.93. The mean of the fit is 0.024 and the standard deviation is 0.018 compared to the measured mean of 0.024 and standard deviation of 0.22 indicating that the distribution can be approximated by a Gaussian distribution. There are higher values which are visible to the right of the graph but have a low likely hood of occurring in noise as this distribution suggests.

one to one method does not. As the background model is shown to require at least five thousand features to be statistically stable and provide a good baseline the correspondence ratio that is generated from one to one matching will be limited by the number of features extracted from the scene image (see Figure 4.16). Matching so many features will, in most cases match to all the features in the target and therefore using one to one matching this will be limited and the correspondence ratio will be meaningless. Using one to many matching a significant correspondence ratio from the total features in the background model can be derived as the percentage of those which match erroneously.

This differentiation between one to one and one to many is less significant for the actual target image, which is likely to have a similar number of features as an informative scene image of the same resolution. Therefore, the selection of one to one may be preferable as this means that duplicate matches are removed and only the matches with the closest correspondence in feature space remain.

4.4.3 Examples of A Contrario Feature Matching for Object Detection

Figures 4.19 and 4.20 show examples of the a contrario methodology being used to identify which frames in a video sequence are likely to contain the target image. The process works by subtracting the correspondence ratio of the background
4.5 CONCLUSION

Figure 4.18: Examples of some test images matched to video frames. (a) and (b) are visible in video 1 and 2 respectively. (c) to (g) are not visible in the video sequences and are used as false target images to test the a contrario technique.

model, in these cases 10,000 random features, from the correspondence ratio generated by matching a target image. The standard deviation of correspondence ratio of the background model matches is iteratively adjusted as the sequence is processed using the previous frames and this is subtracted to get an upper bound for the noise and to make the system robust to the possible variation in the noise. This thresholds the matches and identifies where in the sequence is most likely to be a match to the object. The system is shown to identify the false positives, identifying the areas where the actual target object does not match to the scene. The advantage of this process is to limit the search area for any further higher level processing.

4.5 Conclusion

This chapter has covered important aspects of the functionality of the SIFT which relate to the rest of the work in this thesis. It investigates how the SIFT algorithm functions. This includes an analysis of the SIFT feature descriptor space using millions of features downloaded from Flickr. This has shown that only a small area of the feature space is filled due to the Gaussian weighting of the descriptor and the high number of zero values in the Difference of Gaussian images. Despite the distribution, the likelihood of features matching by chance is shown to be low.

Based on the descriptor observations this section introduces the correspondence ratio and a novel user of the a contrario methodology. The noise properties of SIFT matching are analysed and a methodology is introduced for detecting images which are likely to match based on the rate of false positive matches which occur for non-corresponding images. The work is shown to allow the differentiation of images which are likely to correspond within image sequences by comparing the correspondence ratio to that which occurs by chance.
CHAPTER 4. PRELIMINARY ANALYSIS OF THE SIFT

Figure 4.19: Thresholding by subtracting the a contrario correspondence ratio and $1\sigma$ from results of various target images matching to a video sequence. The first graph is for image (a) which is present in the scene. The other graphs are for images (c) to (g) respectively, which are not present at any point in the scene. Note that they generally have either very low or negative values. The green areas represent the areas where none of the target object is present.

Figure 4.20: A contrario thresholding for a second video sequence. The first graph is for image (b) which is present in the scene. The other graphs are for images (c) to (g) respectively, which are not present at any point in the scene. Note that they generally have either very low or negative values. The peaks are larger for (b) than (a) in Figure 4.19 because the image of the can of beans has clear text with corners and this is ideal for SIFT feature matching.
Chapter 5

SIFTing Through Parameter Space

Following the previous chapter which introduced some concepts related to the functionality of the SIFT this chapter analyses the parameters of the algorithm to determine what effects they have and make recommendations on how they should be chosen.

The parameter space of the SIFT is large; 18 variables are listed in Table 5.1. Selecting a set of parameters which provide good results to match images under different circumstance can be difficult and there is a lack of understanding on the subject. Good results are defined from here on as having a high precision, recall and correspondence ratio and a sufficiently high feature count to preform matching for a particular task.

Section 2.8 described a selection of parameterization attempts for different feature types. These cases indicate that adjustment of the parameters can be beneficial to the results and that the defaults are not always optimal for all situations. They do not provide a full overview of how to intelligently choose the best parameters for a scenario nor do they cover all the available parameters for their respective feature types. This chapter looks further at the parameters of the SIFT descriptor and investigates methods for analysing and selecting values which provide good results for various images types.

David Lowe has specified a set of parameters for the SIFT and empirically justifies some of these default values [70]. The contrast threshold and the curvature threshold are selected but are not justified within the work. The values are 0.03 and 10 respectively. These parameters are important as they form the
main rejection process for candidate features and the choice of these parameters
are included in this investigation.

The choice of sigma, octaves and intervals control the coarseness of the scale
sampling from which features are selected. These again are important as they
control which and how many features that are selected for further stages of the
algorithm. Lowe justifies some of his choices.

The choice of 1.6 for $\sigma$ is demonstrated using a database of 40,000 image
matched features and using a measure of repeatability. 32 images were used
to test this using a range of transformations, including rotation, scaling, affine
stretch, change in brightness and contrast, and addition of image noise. The
results show that for a value of $\sigma$ between 1 and 2 the higher the higher the value
the better the performance for both matching to the database and repeatability.
The value of 1.6 is chosen as a compromise between the efficiency, repeatability
and the number of features generated and, as such, it is not optimal.

The number of intervals, 3, is also selected based on the repeatability of the
features when an image was warped as well as the number of matched features
to a database of 40,000. 3 gave the best repeatability and best response to the
database retrieval. The number of features generated was measured and increases
with change to the number of intervals but this results in many more local extrema
being detected which are less stable and therefore less likely to be detected in the
transformed image.

The number of octaves is not specified in the original paper but the patent
filed for the algorithm suggests that the process should halt when the image has
been reduced to below 30 x 30 pixels. Lowe’s implementations specify 3 or 4 as
the default number of octaves [141].

For the descriptor magnitude threshold the paper states that the value of 0.2
was determined experimentally using images containing differing illuminations
for the same 3D objects. The value of 0.8, chosen to select if a second feature is
generated, is not justified.

Lowe justifies his choice for the size of the descriptor and the number of bins.
By varying $n$ and $r$, the descriptor width and the number of orientation bins, the
results shows that 4 and 8 are the best values respectively and values greater then
this show little improvement. After that, adding more orientations or a larger
descriptor can actually reduce the success of matching by making the descriptor
more sensitive to distortions in the image. The data is computed for images with
affine viewpoint change of 50 degrees and addition of 4% noise and the percentage of correct nearest neighbour matches to a database is shown.

0.8 is selected as the nearest to second nearest neighbour match ratio by using the same database of 40,000 features; as the experiments show it eliminates 90% of the false matches while discarding less than 5% of the correct matches.

Although many of the parameter selected by Lowe to be used as the default values appear to be justified the values appear to be chosen dependant on previous choices made. For example the experiments which suggest that 1.6 should be chosen for \( \sigma \) in turn effect the results of any subsequent parameter decisions. This chapter looks at large sweeps where multiple parameters are varied independently. This multivariate analysis investigates a far larger number of possible combinations than have been previously considered.

A second issue is that the choice of parameters are made on a database of 40,000 features and 32 images with various changes and are not specific to any image properties. They are a general case where are results are averaged. This may result in a set of parameters which are not optimal for any specific case. This chapter’s main aim is to highlight the inadequacy of selecting a single default set of parameters for this algorithm and show that adjustment is necessary to achieve better results.

### 5.1 GPGPU Techniques

This section describes the techniques used and results of sweeps across the SIFT parameter space for various images in an effort to find the best parameter selection for differing scenarios. A semi-exhaustive search has been completed by utilising the speed-up provided by a cluster of general purpose graphical processing units (GPGPUs) over CPUs. The large amount of data produced has then been analysed using parallel coordinate graphs [51], scatter plots and histograms to uncover patterns to indicate how individual parameters effect the algorithm’s precision.

#### 5.1.1 GPGPUs and CUDA

The parameter sweep is a computationally expensive task and is too time consuming to be carried out on a single CPU in a reasonable time, so general purpose graphics processing units (GPGPUs) have been used to implement the SIFT. The
### Table 5.1: The main parameters of the SIFT algorithm and Lowe’s default values. The number in brackets refers to the stage of the SIFT algorithm where the parameter is applied and relates to Section 2.2.2.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
<th>Default Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Octaves (1)</td>
<td>The number of octaves.</td>
<td>3</td>
</tr>
<tr>
<td>Intervals s (1)</td>
<td>The number of sampled intervals per octave.</td>
<td>3</td>
</tr>
<tr>
<td>Sigma $\sigma$ (1)</td>
<td>The sigma value for initial Gaussian smoothing.</td>
<td>1.6</td>
</tr>
<tr>
<td>Image doubled (1)</td>
<td>Whether to double the image size before pyramid construction?</td>
<td>Yes</td>
</tr>
<tr>
<td>Scale factor $k$ (1)</td>
<td>Constant factor separating images in scale space.</td>
<td>$2^{1/s}$</td>
</tr>
<tr>
<td>Initial sigma (1)</td>
<td>The assumed Gaussian blur for input image.</td>
<td>0.5</td>
</tr>
<tr>
<td>Contrast threshold (2)</td>
<td>The threshold on feature contrast $</td>
<td>D(x)</td>
</tr>
<tr>
<td>Curvature threshold $\tau_{th}$ (2)</td>
<td>The threshold on feature ratio of principle curvatures (maximum).</td>
<td>10</td>
</tr>
<tr>
<td>Orientation histogram bins (3)</td>
<td>The number of bins in histogram for orientation assignment.</td>
<td>36</td>
</tr>
<tr>
<td>Orientation sigma factor (3)</td>
<td>This determines the Gaussian sigma for orientation assignment.</td>
<td>1.5</td>
</tr>
<tr>
<td>Orientation radius (3)</td>
<td>This determines the radius of the region used in orientation assignment.</td>
<td>6$\sigma$</td>
</tr>
<tr>
<td>Orientation peak ratio (3)</td>
<td>The magnitude relative to maximum resulting in multiple orientations.</td>
<td>0.8</td>
</tr>
<tr>
<td>Descriptor histogram width (4)</td>
<td>The height and width of the descriptor histogram array.</td>
<td>4</td>
</tr>
<tr>
<td>Descriptor histogram bins (4)</td>
<td>The number of orientation bins per histogram array.</td>
<td>8</td>
</tr>
<tr>
<td>Sample array (4)</td>
<td>The height and width of the sampled area.</td>
<td>16</td>
</tr>
<tr>
<td>Descriptor magnitude threshold (4)</td>
<td>The threshold on the magnitude of the elements of the descriptor vector.</td>
<td>0.2</td>
</tr>
<tr>
<td>Feature vector (4)</td>
<td>The dimensions of the feature vector.</td>
<td>128</td>
</tr>
<tr>
<td>Match ratio</td>
<td>The ratio of the nearest to next nearest feature during matching.</td>
<td>0.8</td>
</tr>
</tbody>
</table>
inherent parallelism of many parts of the SIFT algorithm means it lends itself to being implemented on GPGPUs resulting in significant speed-ups. Tests on a GPU have shown a speed increase for the SIFT of up to twenty times over a CPU\(^1\).

The reason why the GPU is so powerful and can be utilised for this project stems from the large amounts of money being invested in improving their performance for the games industry. They are mass produced and relatively cheap and have the ability to perform highly parallel floating point calculations. NVIDIA’s CUDA is a general purpose parallel computing architecture that provides the tools required for the coding of parallel code for a GPU. The code allows homogeneous execution on both the CPU and GPU so all the resources of the system can be taken advantage of and code which is suited to serial execution can still be executed on the CPU. The architecture also allows the use of multiple GPUs in parallel.

The SIFT has been shown to be successfully parallelised on the GPU in several cases. These include the use of CUDA in the cases of CudaSIFT [12] and SiftGPU [153], and the use of OpenGL textures to store and process the images [48].

5.1.2 Methodology

To perform the parameter sweep a pair of images are required. The areas which match between the images are either annotated or the homography is known so the system can tell where the scene should show correspondence. This is shown in Figure 4.13. For some cases a homography is computed so that exact correspondences can be measured. The system is based on CudaSIFT by Marten Bjorkman [12] and extracts the features from each of the images in parallel. The extracted features are then matched using the GPU and the number of correctly and incorrectly matched features can be calculated using the annotation points or the homography. The parameters are then changed and the process is repeated.

5.1.3 Choice of Parameters for Investigation

The choice of the parameter sweep bounds has been chosen through a mix of empirical investigation and through the constraints of the hardware. Even with the increased computational speed provided by the GPU sweeping of large swaths

\(^1\)Using an AMD Athlon64 FX-70 CPU and an NVIDIA GeForce 8800 GTX GPU.
of the parameter space can be temporally prohibitive using a naive approach. As such the size of the sweep and the step size have been carefully chosen.

An initial investigation, the results of which are shown in Figure 5.1, involved using Lowe’s suggested parameters and varying each parameter individually relative to the other set parameters to gauge the effect of each and identify the likely required range and resolution needed for each. This is not a complete investigation, hence the multivariate sweep that follows, as varying one parameter will effect the response of each of the others. It does provide some understanding about the effect of each parameter and so allows for a more educated selection of the sweep parameters.

The test involved 16 pairs of images and all parameters were set as Lowe’s defaults and then each of the eight shown in Figure 5.1 were varied in turn. The focus is on the detection stage parameters. The chosen parameter ranges for the final sweep are shown in Table 5.2.

The first five are the main parameters used for the scale space detection and selection; sigma, contrast threshold, curvature threshold, intervals and octaves. Orientation peak ratio which controls whether two or one features are generated from a location with two possible dominant directions is included. The match ratio is also swept, which affects the difference between the second and second nearest neighbour required for a match to be confirmed as a positive.

The parameters can affect the number of features less as they occur later in the algorithm from the initial scale space detection (sigma, octaves and intervals) to the creation of secondary features (peak ratio). As such, the choice of some parameters are much more important than others. The results of the behaviour in Figure 5.1 can be explained as follows:

**Sigma.** The initial choice of the scale space. There is a compromise between selecting a high and low value. Too high and the small scale features will be removed and few features will be detected. Too low and the subtraction for the DoG stage will mean that the images are not different enough as the later stages of the scale space are a function of the initial choice of sigma (that difference is $k\sigma$ where $k = 2^{1/s}$ and $s$ is the number of intervals, see Table 5.1). As the intervals are multiples of the choice of sigma a low choice will result in small difference between the images. This may result in fewer extrema and those that do occur will have low contrast due to the reduced DoG response.

**Intervals.** The is similar to sigma as the values are directly related through the
value \( s \), the number of intervals controls the difference between and number of smoothed images in the DoG pyramid. Given that the difference between the DoG images are calculated using \( k = 2^{1/s} \) a high value means that the level of smoothing between the pyramid levels is low so subtracting images does not result in many features are they are too similar and the resulting extrema are of a low contrast. For a low number of intervals the number of images in the pyramid is low so this also generates fewer features. Although the two options may produce the same number of features the specific quality and stability of the features may differ for different values.

**Octaves.** The graph shows an increase in the number of features and then it flattens out. This is because the number of octaves results in the halving of the image size and increasing the scale of the features generated by 2 for each. This shows that for these cases that the graph flattens out for values larger than 4 octaves. There is still variation after this but it is relatively small compared to the initial increase. This is because the larger scales are created by halving the image size and reducing the number of possible feature locations.

**Contrast Threshold.** The graph shows that as the contrast threshold increases the number of features decreases. The graph is approximately linear until a threshold of 9. A higher threshold requires higher contrast extrema and with the DoG being generated from subtraction very high extrema are unlikely to occur increasing the value decreases the point count.

**Curvature Threshold.** This is a threshold that removes features that generate a value greater than it. The curve shows a steep increase from 0 to 10 and then the number of added features starts to flatten out. This shows that of the features which have reached this thresholding point the majority have a fairly high curvature response and that a value of 10 includes most of them into the final selection.

**Orientation Peak Ratio.** For this parameter the graph shows a linear change in the number of features as the value is increased. If a second peak in a feature histogram is within this ratio of the dominant direction peak then a second feature is generated. As expected decreasing this value gradually increase the number of features to double the number for when the ratio is set to 1. This is expected as a low values will mean that most features will generate a secondary feature. Whether these features are unstable in matching or not will be uncovered in the parameter sweep.
**Match Ratio.** As expected, the graphs shows that by increasing the value more features will match until the number of matches equals the number of features extracted.

![Graphs showing variation of Lowe’s parameters](image)

Figure 5.1: This graph shows variation of Lowe’s parameters one parameter at a time. The variation is shown for the number of features extracted and matched features for the first of a pair of matching images. Each graph shows the variation of a single parameter while the rest of the parameters remain constant. These are the mean results of 16 image pairs and been used to help determine limits for multivariate parameter sweep.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
<th>Step size</th>
<th>Samples</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sigma</td>
<td>0.1 - 5.1</td>
<td>0.5</td>
<td>11</td>
</tr>
<tr>
<td>Intervals</td>
<td>1 - 6</td>
<td>1</td>
<td>6</td>
</tr>
<tr>
<td>Octaves</td>
<td>1 - 4</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Contrast threshold</td>
<td>0 - 25</td>
<td>2.5</td>
<td>11</td>
</tr>
<tr>
<td>Curvature threshold</td>
<td>0 - 25</td>
<td>2.5</td>
<td>11</td>
</tr>
<tr>
<td>Orientation peak ratio</td>
<td>0.1 - 0.9</td>
<td>0.2</td>
<td>5</td>
</tr>
<tr>
<td>Match ratio</td>
<td>0.2 - 0.8</td>
<td>0.2</td>
<td>4</td>
</tr>
</tbody>
</table>

Table 5.2: The sweep input parameters.

A subset of the parameters in table 5.1 has been used within these experiments. The parameters chosen are shown in Table 5.2 along with their starting values, the range over which they are varied and the step size of each iteration. This set of parameters results in up to 638,880 iterations of the algorithm, depending on the features generated, and can take up to 45 hours for an image pair.

The images used are varied so that different features are matched in various

---

2Using an Intel Core i7-920 2.66GHz CPU and an NVIDIA 9800 GX2 GPU.
5.2. GENETIC ALGORITHMS FOR PARAMETER INVESTIGATION

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Points 1</td>
<td>The number of points extracted from the first image</td>
</tr>
<tr>
<td>Points 2</td>
<td>The number of points extracted from the second image</td>
</tr>
<tr>
<td>Total matches</td>
<td>The total number of matches between the images</td>
</tr>
<tr>
<td>Correct matches</td>
<td>The number of correct matches between the annotated regions</td>
</tr>
<tr>
<td>Annotated matches</td>
<td>The total number of matches from the annotated regions in the first image</td>
</tr>
<tr>
<td>Precision (Annotation version)</td>
<td>The ratio of correct matches to the number matches from within the annotated regions</td>
</tr>
<tr>
<td>Precision (Homography version)</td>
<td>The ratio of correct matches to the total matches</td>
</tr>
<tr>
<td>Recall (Homography version)</td>
<td>The ratio of correct matches to the total possible matches</td>
</tr>
</tbody>
</table>

Table 5.3: The sweep output parameters.

scenes with changes in scale, rotation and viewpoint so that many different possible SIFT usage scenarios are covered. This helps indicate which parameters effect the algorithm under different circumstances, provides information about the optimal parameters for various scenarios and helps indicate any trends and correlation across the parameters. The data produced for information visualization analysis has up to 14 dimensions which relate to each of the input and output parameters. There are six output parameters that are generated during the parameter sweep and these are described in table 5.3. The precision value that is generated is calculated differently depending on whether the homography or annotation is used to indicated correct matches as explained in Section 4.2.

The selected parameters for investigation are a subset of the total parameters. The study has focused on the detection stage of the algorithm and the descriptor parameters have been set as Lowe’s defaults for this study.

5.2 Genetic Algorithms for Parameter Investigation

Further parametrization has utilised multi objective genetic algorithm (GA) to create sets of Pareto optima for a multi-objective minimisation or maximisation. The technique randomly generates a number of sets of initial parameters within certain bounds and iterates, combining and mutating the parameters, to find better values. As opposed to the sweep which uncovers the distribution of the space the GA optimises the choice after each round to attempt to locate the best parameters.

The process works by using an objective function which takes a set of parameters as an input and returns values which the algorithm can interpret as a
score to evaluate the success of that set of parameters. In this case the objective function calls the GPU based SIFT implementation and returns the *precision* and the *number of correct matches*. The objective of the GA is to maximise both of these values finding a set of parameters which return high precision matches with the largest number of accurate matches. The process uses the following steps:

**Initialise Population.** Multiple sets of random parameters are generated as starting points for the genetic algorithm. These are the first set of parents and the number generated is defined by a user selected population size. The objective function is used to score and rank each of these initial values and assign a fitness score. The initialisation for the SIFT is done by generating a random population within the ranges shown in Table 5.4. The initial population size for these experiments is 20.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Range</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sigma</td>
<td>0.1 - 4 (5 in some tests)</td>
</tr>
<tr>
<td>Intervals</td>
<td>1 - 6</td>
</tr>
<tr>
<td>Octaves</td>
<td>1 - 4 (5 in some tests)</td>
</tr>
<tr>
<td>Contrast threshold</td>
<td>0 - 25</td>
</tr>
<tr>
<td>Curvature threshold</td>
<td>0 - 25 (50 in some tests)</td>
</tr>
<tr>
<td>Orientation peak ratio</td>
<td>0 - 1</td>
</tr>
<tr>
<td>Match ratio</td>
<td>0.2 - 0.9</td>
</tr>
</tbody>
</table>

Table 5.4: For the GA initialisation a uniformly random starting population is generated with parameters values within the above ranges. These are similar ranges to those from the sweep but have varied between experiments because of increasing of knowledge of the space.

**Fitness scaling.** The scaling function converts raw fitness scores returned by the fitness function to values in a range that is suitable for the selection function. The initial fitness function is based on a *pareto score* generated from the precision and the number of correct matches. A pareto front is a way of choosing between two or more choices among conflicting goals. Maximising the *precision* may minimise the number of correct matches and vice-versa. A pareto front plots the best possible combinations of the objectives that have been found so far by the GA. This tends to be more than one point as the objectives are conflicting and a front can be generated which shows the combinations of the two objectives which produce the best results. Figure 5.2 shows this.

Raw fitness score is calculated using the pareto dominance. If solution one (a set of parameters) is better than solution 2, i.e. both of the objectives values are deemed to be superior (or one equal and the other superior) then solution 1 is said to dominate solution 2. If neither of the solutions dominates both are regarded as
5.2. GENETIC ALGORITHMS FOR PARAMETER INVESTIGATION

The raw fitness score of an individual within the population depends on the number of individuals dominating that individual.

Rank scaling is a technique which then scales the raw scores based on the rank of each individual, rather than its fitness score. The rank of an individual is its position in the sorted scores. The rank of the fittest individual is 1, the next fittest is 2, and so on.

Selection. A subset of the existing population is selected to breed a new generation. Stochastic uniform selection lays out a line in which each parent corresponds to a section of the line of length proportional to its scaled fitness. The algorithm moves along the line in steps of equal size, one step for each parent. At each step, the algorithm allocates a parent from the section it lands on. The first step is a uniform random number less than the step size. Figure 5.3 shows this technique. Stochastic uniform selection means the parents with higher scores are more likely to have their information included in the next generation. This corresponds to survival of the fittest in biology.

Crossover and Mutation. The next stage is to generate a new sets of parameters (children) from the previous sets of parameters (parents). Using this
CHAPTER 5. SIFTING THROUGH PARAMETER SPACE

Figure 5.3: Stochastic uniform selection selects the next generation for the genetic algorithm. The population, labelled A to G, is sorted by size and the total fitness is divided by the number of children required for the next round (N) to generate the step size. The algorithm selects a random start position less or equal to the step size from the beginning and chooses N individuals corresponding to each step position.

A new generation is generated using crossover and mutation. Crossover is a technique for selecting various properties of two or more parents and combining them to make a child. In terms of parameter selection for the SIFT this has been done by using scattered selection. This creates a random binary vector. It then selects the genes where the vector is a 1 from the first parent, and the genes where the vector is a 0 from the second parent, and combines the genes to form the child. Mutation functions make small random changes in the parents in the population to create children. These mutations provide genetic diversity and enable the genetic algorithm to search a broader space.

The population is maintained by keeping at least the two highest ranked individuals from the old population of parents and selecting new individuals from the set of children. More are kept if too few children are created from crossover and mutation. 80 percent of the children are generated from crossover alone and 20 percent use mutation to create children with wholly new attributes. For these experiments the crossover and mutation are independent of each other and applied to different individuals separately.

Termination. The selection, crossover and mutation stages are repeated until a termination criterion is met. There are various termination criteria that can be used but for the SIFT parameter experiments the process is terminated when successive iterations no longer produce better results. This is called stalling. The results are judged to have stalled if the weighted average change in the fitness function is less than a threshold (0.000001) for a set number of iterations (50). The process halts regardless after it has iterated for 100 generations.
5.3. DATA VISUALIZATION AND ANALYSIS TECHNIQUES

Results. The results are given in the form of a pareto front. The best results from all the iterations of the GA are displayed as the maximisation of the two objectives but as the two objects are conflicting multiple choices are possible so a series of points is produced providing a front along which various best possible solutions lie. The GA provides a front but does not provide information about the distribution of the space. The sweep provides the distribution but a less accurate front as the choice of values are unlikely to be as well optimised as those generated by the GA.

5.3 Data Visualization and Analysis Techniques

This section outlines the interactive analysis process for analysing results of the parameter sweeps. Parallel coordinate graphs [51], scatter plots and histograms have been used to visualize and analyse the data. Parallel coordinates is a common way of visualizing multivariate data such as that produced by the parameter sweep. Each parameter has its own parallel axis and a polygonal line with vertices on the parallel axes representing a point in n-dimensional space. This visualization method allows correlation between parameters to be viewed with the careful use of brushing.

An example of a parallel coordinate graph is shown in Figure 5.4. It has been brushed to display parameter combinations with the highest precision and a number of correct matches greater than 5. Brushing is an interactive process of reducing the data to a subset. The red lines display the parameters which meet a criteria. A parameter must produce a high percentage precision and a number of correct matches greater than a minimum to be deemed a good selection. The reason that a minimum number of correct matches is required is that a single correct match could give a precision of 1 but would be useless for confirming matches between images as a cluster of points are required and a single data point could be erroneous. Setting this minimum eliminates parameter combinations which give a high precision without enough data to be confident of an image match.

This is one method of selecting a set of features with desirable properties but brushing can be used also to select data with other properties. Examples include selecting data points which give high precision and high recall or high recall and a low number of matches. Metrics can be varied depending on what is deemed
Figure 5.4: Parallel coordinates displaying a 8 dimensional dataset with 90525 elements. The data is brushed to exclude elements with a very low number of correct matches ($\leq 5$) and low precision ($\leq 0.5$). The lines show that this selection excludes higher and low sigma values, high contrast threshold values and low curvature threshold values for this set of images.

to be a good result for a specific task. From the parameter sweep data a pareto front can be generated in the same manner as for the GA. This is shown in Figure 5.5 and can be used as the set of data for further analysis.

On the parallel coordinate graph multiple overlaid lines on a parameter point cannot be distinguished from a single line through a point. The use of histograms allows each parameter to be plotted individually showing the distribution of values that pass through each parameter point. This indicates which parameters contribute most to the results with the highest precision as shown in Figure 5.6. This histogram binning can also be used to visualize the distribution of the parameter values for the points which lie on a pareto front. Box plots can also be used to summarise this data for many samples as shown in Figure 5.7.

5.4 Results, Analysis and Recommendations

The following section outlines the data used and the results generated from these experiments.
5.4. RESULTS, ANALYSIS AND RECOMMENDATIONS

Figure 5.5: The pareto front (right) generated from parameter sweep data (left). This techniques allows for the selection of a set of points which can be used for further analysis.

Figure 5.6: Histograms of six parameters after the data from Figure 5.4 has been brushed. The peaks indicate the values which contribute most to the data selected by brushing.
CHAPTER 5. SIFTING THROUGH PARAMETER SPACE

Figure 5.7: This shows examples of box plots which are used to summarise the histogram data. This example shows the distribution of part of the sample for the match ratio on a series of images. They present the information in a more easily compatible way when viewing multiple samples. The blue area shows the interquartile range, the horizontal red line is the median value, the red dot is the mean value, the whiskers show the extremes of the distribution and the red crosses show outliers in the sample.

5.4.1 Data Sets Used for Analysis

Two sets of data have been analysed for these experiments. The first is the set used by Mikolajczyk et al. for A Performance Evaluation of Local Descriptors [86] from the Oxford Robotics Research Group. This will be referred to as the Oxford data set. This data consists of eight subsets of images of different scenes ranging between 0.6 and 0.7 megapixels (MP) in size. For each subset the images have properties varied so as to determine the effects of that change on matches between image pairs. The images are shown in Figure 5.8 and each is labelled to show the variable that is altered in each. The homography is known for images pairs such that correct and incorrect matches can be accurately identified.

The second set that is used is a set of trial images provided by MBDA which will be known as the MBDA data set. It consists of a pick-up truck captured with different cameras and angles as the car drives in a circle and the images range from 0.3 to 0.5 MP in size. This data set has been annotated by hand to indicate the corresponding areas of the truck between different images. Examples of the images are shown in Figure 5.9. The five images types are as follows. The SIB is an infra-red camera and the others capture from within the visible spectrum:
5.4. RESULTS, ANALYSIS AND RECOMMENDATIONS

(a) Bark - Zoom & Rotation  (b) Bikes - Blur  (c) Boat - Zoom & Rotation  (d) Bricks - Viewpoint
(e) Cars - Lighting  (f) Graffiti - Viewpoint  (g) Trees - Blur  (h) UBC - JPEG Compression

Figure 5.8: Images from the Oxford test data set.

1. **HAWK** 8-bit image from a Raptor Hawk EM247 monochrome camera.
2. **SIB** 8-bit image from a Segmented mask Infrared Bolometer (SIB).
3. **SIBHDR** 16-bit image from a Selex Segmented mask Infrared Bolometer (SIB).
4. **WAT232** 8-bit image from a Watec 232 Colour camera.
5. **WAT902H** 8-bit image from a Watec 902H2 ULTIMATE monochrome camera.

(a) HAWK  (b) SIB and SIBHDR  (c) WAT232  (d) WAT902H

Figure 5.9: Images from the MBDA test data set.

For the Oxford data eight data sets were analysed. Each set shows images of the same scene with some properties varied. The first image is matched to all the other images in a set resulting in five sweep and GA results for each set. This totals forty sweep results and forty GA results.

For the MBDA data, eight sets of the five image types, one from each camera, were used. The pose of the target object is the same across each set and the
corresponding regions were annotated. Matches were generated from each image
to every other in the same set resulting in ten combinations per set and eighty
results data sets for both the sweep and GA. This supplies information on matches
between various camera types.

5.4.2 Results

This section shows some of the results from the sweeps. The majority of the
graphs are from the GA from the Oxford data set to demonstrate examples of
the data and how the summaries of the data sets were generated. For the GA the
plots use the points which are on the pareto front; a selection the best parameters
which have been found by the algorithm.

Figure 5.10 shows typical sweep and GA results. The graphs have been gen-
erated from matches between HAWK and SIB images in the MBDA dataset.
The graph on the left shows the distribution points in terms of precision and the
number of correct matches for the sweep and the GA. It can be seen that the GA
generates a smaller set of data points from the same range of parameter values
as it can refine the values and act at higher resolution then the sweep which has
a set number of possible parameter values. This is typical among the results.
The sweep provides information about the structure of the space and allows for
other types of analysis to be applied afterwards. For example, the histograms on
the right show the distribution of a set of parameters which contribute to the 500
points with the highest precision which have more than 5 correct feature matches.

Figures 5.11 to 5.17 show the distribution of the parameters for each of the
Oxford images. The distribution is based on the values in the pareto front. When
the front of the sweep was analysed in the same way a similar yet less refined set of
points were generated. The GA and the Oxford sets are the most comprehensive
range set of parameters that have been swept in these experiments. The sweep
has taken place for 40 pairs for the Oxford data and 80 pairs for the MBDA data.
Further graphs produced from the data are available in Appendix A.

5.4.3 Analysis - General Observations

The first section of analysis of the results gives an overview of the general trends
observed for each of the parameters. Following this a more specific analysis of
specific results is given. The inferences form the data stated below focus on the
5.4. RESULTS, ANALYSIS AND RECOMMENDATIONS

133

(a) Sweep Results

(b) Parameter Histogram

Figure 5.10: An example of the result data from the sweep and the histogram analysis. The data is from MBDA Set 3 for the HAWK matching to the SIB. This is typical as the GA values are forward of the sweep results and they show that the GA is more successful at finding optimal values. The sweep however provides information about the rest of the space and how it is populated. These graphs have been generated for all of the data and provide insight into how well matching a pair of images can perform if the parameters are well chosen. The histograms show the distribution of the parameters when the data is brushed to selected the 500 highest precision data points which have more than 5 correct matches. Using the histogram peaks as the parameters provides a better result than Lowe’s selection but it is far from optimal.

Figure 5.11: Oxford GA Pareto Front - Sigma.

Figure 5.12: Oxford GA Pareto Front - Contrast.
CHAPTER 5. SIFTING THROUGH PARAMETER SPACE

Figure 5.13: Oxford GA Pareto Front - Curvature.

Figure 5.14: Oxford GA Pareto Front - Intervals.

Figure 5.15: Oxford GA Pareto Front - Octaves.

Figure 5.16: Oxford GA Pareto Front - Peak.
Oxford set as this is more accurately measured and is therefore a more accurate
gauge of the effect of the parameters but all data is considered.

**Sigma.** Sigma has strong peaks across the full range of values meaning that
the choice of sigma is very specific to an image type and can greatly affect the
results. Sigma therefore needs to be chosen carefully and a single value cannot
guarantee precision across all image types. The values tend towards having a
strong response between 1 and 3.

**Contrast threshold.** Contrast threshold, like sigma, consists of strong peaks
across the full range of values and therefore must also be chosen carefully for each
image pair that is used. A bad choice can cause low precision and the best value
varies from image to image so a single, universal, value will not suffice. Values
close to 1 (0.004) appears regularly across the image pairs as the best choice.

**Curvature threshold.** The maximum curvature threshold tends to have peaks
towards the higher end of the swept range. Nearer 25 or 45 depending on the
ranges used in these experiments. This is a consistent occurrence. The peaks that
occur in the histogram binning are not as strong as for the previous parameters
indicating that its importance may be lower. The number of features remaining
increases as the parameter is increased so setting it too low will generate fewer
features. It indicates that Lowe’s choice of 10 may be too low and that such a
strict measure of curvature is not required for stable feature extraction. There
appears to be some correlation with the contrast threshold where higher contrast
threshold results in a lower curvature threshold.

**Intervals.** The optimal number of intervals varies with variable strong peaks in
the histogram binning as with sigma. The two variables, sigma and the number
of intervals, are correlated, strongly in some cases as is shown in Figure 5.18. This is due to the relationship of the distance in scale distance between intervals calculated as \( k\sigma \) where \( k = 2^{1/s} \) and \( s \) is the number of intervals. This means that for optimal performance the value of sigma or the number of intervals cannot be adjusted without taking the other into account.

![Graph showing correlation between mean sigma and intervals for pareto front of the Oxford sweep.](image)

Figure 5.18: This shows the correlation between mean sigma and intervals for pareto front of the Oxford sweep. The red line is the line of best fit and it highlights the correlation. This is the most obvious instance of correlation between the two variables but it is also observable in the other data sets.

**Octaves.** The optimal number of octaves is 4 or less in nearly all cases. A higher number of octaves and intervals appears to be unnecessary as they include the features generated when the parameter is set to a lower value. The mean and median values for both the GA and the sweep of each image for all the data sets tends to be between 3 and 4. The size of the images used dictates the number of octaves which can be used to generate extra features.

**Orientation peak ratio.** The results have shown that a low orientation peak ratio, closer to 0.2 performs better than the 0.8 suggested by Lowe. This may be because the generation of a second feature in a location that has passed the previous threshold criteria will make extra matches more likely increase the precision value within the tests. Smaller secondary peaks in the feature histogram are more stable between views then previously assumed. This effectively means that stable locations are encoded as features twice on more occasions than with Lowe’s parameters.

**Match ratio.** Strong and medium histogram peaks indicate that the match ratio should be high, generally larger than 0.6, which supports Lowe’s conclusions. The
lower the match ratio value is the more discriminative it is reducing the number of matches. Extremely low numbers of correct matches have been brushed out of the data as they do not provide sufficient information to reliably indicate correspondence between images. This explains why the match ratio value tends to be high and why it is better to select a larger value.

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sigma</td>
<td>Variable for each image, strong histogram peaks.</td>
</tr>
<tr>
<td>Intervals</td>
<td>Variable for each image, linked to sigma.</td>
</tr>
<tr>
<td>Contrast threshold</td>
<td>Histogram peaks at lower values.</td>
</tr>
<tr>
<td>Curvature threshold</td>
<td>Histogram peaks at higher values. Correlation with contrast threshold.</td>
</tr>
<tr>
<td>Octaves</td>
<td>Flatter histograms, histogram peak tends towards 3 or 4.</td>
</tr>
<tr>
<td>Match ratio</td>
<td>Consistently a high value, around 0.8.</td>
</tr>
<tr>
<td>Orientation peak ratio</td>
<td>Consistently a low value around 0.2.</td>
</tr>
</tbody>
</table>

Table 5.5: Observed parameters hierarchy. This shows the order of precedence for varying values when conducting a sweep on a subset of parameters. Parameters lower in the table are less important for tuning the SIFT algorithm.

When the parameter selection results in a high number of features being extracted this is generally detrimental to *precision*. High numbers of features are more likely to result in mismatches as there are more opportunities for the features to match incorrectly. Even if it does not affect the *precision* it is beneficial to avoid unnecessary feature extraction as this results in more computation. A balance must be found between generating enough features to match the target area within the image and too many such that mismatches become more likely to occur and become computationally expensive.

It should also be noted that while these sweeps took up to 45 hours per image pair to complete the time can be reduced. By removing two parameters from the sweep and setting their values to values within the recommendations above the execution time can be reduced. The parameters which should be focused on are sigma, intervals, contrast and curvature as their histograms have the strong peaks and large variability between images. These are the parameters responsible for the initial selecting and thresholding of the features and make the largest difference in which features are selected. The other parameters analysed tend to form flatter histograms, or are generally don’t change much between images. This means that it is possible for a variety of values, or to always select a similar value, to generate very good results for a pairs of images and as such tuning these parameters is less important. Table 5.5 shows the hierarchy for the parameters and provides a guideline for selecting a subset for a smaller sweep. Examples are shown in
Figure 5.19: This shows typical examples of the ‘best 500’ sweep histograms. Sigma, contrast and intervals have strong peaks and the location of the peaks varies for different image pairs. The other parameters are more consistent between images and either tend to have a peak at a similar value or are flatter and selection of that parameter for optimal results is less important.

5.4.4 Validation of the Choice of Analysis Data

Figures 5.20 and 5.21 show the precision and number of correct matches for the results generated by Lowe’s parameters and compares them to the values generated using the peaks of the histogram analysis. The peaks, in this case are selected using the ‘best 500’ method where the 500 results with the largest precision with more than 5 correct matches are chosen and binned in a histogram.

The results show that in many cases the results for the Sweep out perform using Lowe’s parameters and Figure 5.20, the results for all of the MBDA sets, shows that for precision 64% of the values are greater than the Lowe values and for the number of correct matches 79% of the matches are have a larger value. For the Oxford set it shows that for precision 90% of the values are greater than the Lowe values and for the number of correct matches 40% of the matches are have a larger value. The full tables of results are available in Appendix A.

This indicates that analysing the results in the way described above and looking at the parameters which contribute to the best results is a valid focus of investigation. The peaks can equate to an improved response of the SIFT algorithm and the distribution of this subset of data can provide information on how to select better parameters for an image type. It shows that the points are not just individually better but combined they can show a trend to indicate better parameter choices.

5.4.5 Further Analysis of the Sweep Data

This section analyses more specific instances of the data generated to gain further insight into the effect of the parameters.
5.4. RESULTS, ANALYSIS AND RECOMMENDATIONS

Figure 5.20: Graphs of Lowe’s values plotted against the histogram peak values for the MBDA set using the ‘best 500’ values. The red line shows the middle point between the two axes. The results shows a larger count for each on the sweep sides of the graph. For precision 64% of the values are greater than the Lowe values and for the number of correct matches 79% of the matches are have a larger value.

Figure 5.21: Graphs of Lowe’s values plotted against the histogram peak values for the Oxford set using the ‘best 500’ values. The red line shows the middle point between the two axes. The results shows a larger count for each on the sweep sides of the graph. For precision 90% of the values are greater than the Lowe values and for the number of correct matches 40% of the matches are have a larger value.
CHAPTER 5. SIFTING THROUGH PARAMETER SPACE

Oxford

The Oxford dataset shows results that are more precisely measured than for the MBDA data as the homography method was used to measure the matches and this is shown by the tighter distributions of results and the higher precision generated by the histogram results. The images are scenes with various changes applied to different degrees; lighting, viewpoint, zoom, blur, rotation and JPEG compression.

Change in zoom and rotation occurs in the bark and boats datasets. The images gradually increase in the amount of rotation and zoom. Figure 5.22 shows the graphs produced for each of the pairs that have been analysed for the boats dataset. As zoom and rotation become more exaggerated and the difference between the two images is increased the area that the sweep covers is reduced and the pareto front does not push as far forward. The difference is not particularly detrimental to the algorithm and the SIFT is shown to be robust to these changes. The negative performance is greater as the zoom and rotation in increased for the bark dataset. This is due to the changes in the bark dataset being more extreme and the image has self-similar elements which results in a greater number of mismatches.

![Figure 5.22](image)

Figure 5.22: Pareto fronts and sweep distribution for the Oxford boats dataset which shows increasing change in zoom and rotation.

Change in blur occurs in the bikes and trees datasets. The images gradually increase in the level of blur and Figure 5.23 shows the decrease in the area of the
sweep as the maximum possible precision and correct matches decreases. The worst results are not as poor as when the increase in other properties, such as viewpoint, and high precision matches can be achieved even with high levels of blur. It shows that the SIFT can be parametrized to perform well under changes in blur as this is a fundamental aspect of the algorithm.

![Figure 5.23: Pareto fronts and sweep distribution for the Oxford bikes dataset which shows increasing changes in the level of blur.](image)

Change in viewpoint occurs in the bricks and graffiti datasets. The change in viewpoint in angle is gradually increased to a larger angle from the initial image. This reduces the number of parameter choices which produce good results and the sweep results area becomes very small for the final image pair shown in Figure 5.24.

Change in lighting occurs in the cars dataset. The images gradually increases in the difference in lighting from the initial image by exposing each image for a longer period of time. Figure 5.25 shows how this effects the sweep results and that SIFT matches become poorer as the difference widens but good performance can still be achieved.

Change in JPEG compression occurs in the UBC dataset. The images gradually decrease in the level of compression. Figure 5.26 shows that high precision matches are possible in all cases even with very high JPEG compression. This is due to the blurring that occurs within the SIFT algorithm reducing the effect of the artefacts introduced by the JPEG compression.
CHAPTER 5. SIFTING THROUGH PARAMETER SPACE

Figure 5.24: Pareto fronts and sweep distribution for the Oxford bricks dataset which shows increasing changes in viewpoint in angle.

Figure 5.25: Pareto fronts and sweep distribution for the Oxford cars dataset which shows increasing change in lighting.
5.4. RESULTS, ANALYSIS AND RECOMMENDATIONS

Figure 5.26: Pareto fronts and sweep distribution for the Oxford ubc dataset which shows decreasing levels of JPEG compression.

These experiments highlight that the SIFT’s response to changes in viewpoint are poorest which supports the findings in Mikolajczyk and Schmid’s paper entitled *A Performance Evaluation of Local Descriptors* [86]. It shows also that it can be parametrized to be robust to blur and compression and that if the parameter space is explored poor responses can be improved. This is more important in images where good matches are less likely such as changes in viewpoint as the number of parameter choices which produce accurate matches is reduced. The results show that the GA is generally better at obtaining these values as the front is often forward of the sweep data.

MBDA

The first thing to note is that the MBDA data set uses the annotation method to calculate correct matches. This is an approximation and is not as precise as the measurement made for the Oxford set. Expected trends can still be seen and the data still indicates which parameters combinations provide better matches albeit with less precision.

The MBDA data set results shows the best parameter combinations for matches between the various images types used. Some example GA and sweep results from Set 1 of the MBDA data set are shown in Figure 5.27. These show various matches between images types and indicate which types match well to each other. This is
indicated by the size and shape of the distribution plotted for the sweep and the location of the pareto front. Images which with smaller difference between them will have more correct and higher precision matches.

The best match combination is from WAT232 to WAT902H and HAWK to WAT902H followed by HAWK to WAT232. This is expected as these are all matches between visible spectrum images. Examples of the results which demonstrate levels of precision which are achievable within the scope of the sweep are shown in Figure 5.27. Matching from visible light to infrared images are shown for HAWK to SIB and the number of matches that occur within the swept parameter space and the maximum values for correct and precision is fewer. The experiment does show that they can be successfully matched using the standard SIFT algorithm but the parameters will have to be more carefully chosen then for pairs of standard images.

For the matches between the full HDR SIB infrared images and the 256 bit SIB image the results show that they match well despite the difference in the source images. The SIFT descriptor encodes the information in the patch such that this is not a problem and the pareto front distribution for the parameters shows a different shape to the others where more matches occur at a high precision.

The consistency of the fronts is clear across the same images type combinations between each of the eight sets. For example, the HAWK to SIB front looks the same for each of the eight pairs of images matching between these types. This means that there is a maximum possible performance for these image pairs that can be achieved within the sweep space and it varies for the images combinations. The parameter values don’t shows the same consistency. For the GA front, the sweep front and the ‘best 500’ the values of the parameters used to obtain the results is not consistent between across all the image pairs of the same type. This implies that parameter tuning is a necessary to achieve the best results for any image pair but may also be a result of the less accurate annotation matching technique. This said, the parameter values do fit to the observations that were made in Section 5.4.3.

5.5 Conclusion

It is proposed that this interactive technique should be used to fine tune the SIFT in situations where the images are of a known type. The results show that some
5.5. CONCLUSION

Figure 5.27: GA and sweep results from Set 1 of the MBDA data set.
parameters such as sigma, the number of intervals and the contrast threshold have strong peaks over the range of values meaning that a single selection will never suit all situations. Some of the other parameters are quite flexible and robust which means that non-optimum selection may not be detrimental. Table 5.5 shows the hierarchy for the parameters and provides a guideline for which parameters are more flexible.

Further work will look into how to use this technique to create an intelligent means of parametrizing the SIFT based on the properties of an image pair. The aim of this is to allow a SIFT user to reliably set parameters based on the image properties such as the size of the object, viewpoint or the object type without having to apply this parameter sweep technique themselves. Other areas of interest include the effect of other parameters such as the number of bins in the descriptor, test other ranges and step values and other images types such as high dynamic range (HDR).

Overall, the parameters of the SIFT cannot make an improvement by adjustment if the data is not within the image in the first place so there is an inherent best match precision based on the image data. However, selecting the wrong parameters can reduce the precision of the SIFT. Tuning has been shown to make improvements which can be beneficial to an application with constrained bounds and where the task will be repeated many times to justify the computationally expensive sweep process.

This chapter has outlined that to successfully use the SIFT algorithm a user must carefully consider the parameters as the values can greatly effect the performance and no single set of parameters performs optimally for all image types.
Chapter 6

High Dynamic Range Fusion Features

The previous chapter looked at improving the performance of the SIFT through analysis and selection of the parameters which control the algorithm. This chapter focuses on improving the performance of the SIFT in conditions where there may be contrast lighting and a single image will contain over or under exposed areas.

As described in Section 3.2 High Dynamic Range (HDR) images can increase what is visible in a scene. Using this extra information image features can be made more effective for all stages of a feature matching and object detection and identification. This chapter focuses on novel techniques that use HDR data to improve the likelihood of accurate matches between images. This focus is driven by the observation that the use of HDR has become more prevalent as cameras have become cheaper and better equipped and this data type will be more common.

Consider a scenario where an automated vehicle is travelling in an indoor environment lit by natural light through windows as demonstrated in Figure 6.1. A single Low Dynamic Range (LDR) camera will be unable to correctly expose the objects both inside and outside concurrently and may incorrectly select an exposure value which results in the target object being hidden from the camera’s view by being either over or under exposed. Using HDR data will negate this issue as the large dynamic range will allow the acquisition of data from both the indoor and outdoor environments at the same time. This work has many applications including security, surveillance, robotics and recognition tasks in environments where the lighting cannot be controlled.
CHAPTER 6. HIGH DYNAMIC RANGE FUSION FEATURES

Figure 6.1: A example where a single LDR exposure cannot capture all the information in the scene due to its high dynamic range. The left image shows is well exposed for the indoor environment. The middle image is well exposed for the outdoor environment. The right hand image show a HDR exposure fusion image [81, 82] generated from the first two images and has the advantage of containing more data within the scene possibly allowing a better match to a target object.

This chapter introduces the concept of Fusion Features. This is a novel technique for the combining of features from multiple exposure images using a variety of measures to select only those from the best exposed areas of each image. These are compared to other methods for extracting features from HDR images and are shown to outperform the alternatives. This chapter then explores the benefits and the compromises that occur when using 3D and HDR data. This is implemented using pairs of stereo cameras with exposure varied by filtering the camera lens reducing the amount of light reaching it using a Neutral Density (ND) filter.

6.1 High Dynamic Range Fusion Features

Feature matching is a common computer vision application. In high contrast lighting conditions it can be difficult to extract features in all areas of a scene with a single exposure image as important areas can be over or under exposed. As such, vital information about a scene can be missed. The problem that this section solves is how to best utilise multiple exposure images to match features in scenes with a large dynamic range. The main contribution of this section is a feature fusion process using the scale invariant feature transform (SIFT) within sets of images taken of the same scene with varied exposures. These features cover a large dynamic range in a scene and are extracted in a way which improves match accuracy when compared to extracting features directly from single high dynamic range image types. FAW defines the recommended order for extracting fusion features; Features extraction, image Alignment then pixel Weighting.
This is opposed to AWF, the order for generating tone mapped and exposure fusion images and extracting features from them; Alignment of the images, pixel Weighting and image merging and then Features extraction.

The concept proposed is based on exposure fusion [81,82] and its purpose is to create an improved set of features which represent a higher dynamic range than a set of features extracted from a single image. A key component is that areas which contain information unseen in one exposure can utilise the features from a differently exposed image. The process selects from the best exposed areas of each exposure image using four different measures given in Section 6.2. This generates a new set of features which cover a larger dynamic range. This process can be applied to aligned images, as with exposure fusion, but can also be extended to misaligned and stereoscopic images as shown in Section 6.3.

Figure 6.2: An example of two aligned input images taken at different exposures. The arrows represent the scale, orientation and position of the SIFT features. The bounding box in each shows the areas within which SIFT features have been matched between the images using RANSAC during the alignment process [134].

6.2 Fusion Feature Selection

The process of selecting fusion features utilises the main measures of exposure fusion [81,82] and an entropy measure [42]. A set of images of varying exposures are taken and for each of these images a set of features are extracted using the SIFT as shown in Figure 6.2. These features are then used to accurately align the images using RANSAC [134]. Then the feature locations are transformed to match the alignment transformation. For each pixel in the aligned images weightings are generated using some or all of the four measures outlined below. The final weight values for each pixel indicate the exposure image in which each
pixel is best exposed. This is then used to select which features are added to the set of fusion features using a Gaussian weighting at the scale and radius of the feature.

**Contrast Measure** $C$: The contrast measure $C(x, y)$ is calculated across the image, $I$, for each greyscale pixel location:

$$C(x, y) = \sqrt{(I(x + 1, y) - I(x - 1, y))^2 + (I(x, y + 1) - I(x, y - 1))^2} \quad (6.1)$$

This gives larger values for textured areas and this indicates if an area of the image is well exposed. It is postulated that over or under exposed areas will have small values.

Using the absolute values returned by a Laplacian filter as suggested by Mertens et al. [81, 82] has been replaced by the gradient magnitude. Using a zero crossing, second derivative, function to calculate the weighting means that the edge peaks will return a value of zero. Thus, two edges, one with a large magnitude and one which is much smaller in magnitude will both have a value of 0 at their apex and a weighting based on this will weight both pixel values equally. If they are slightly misaligned then one edge pixel will get the full weighting in its favour at a point when the other image may have a larger edge. A first derivative function returning the gradient magnitude and allows edge gradient values to be compared and weighted accordingly.

**Saturation Measure** $S$: This follows from the fact that as an image is exposed for a longer period of time it becomes desaturated. The less saturated the image, the more washed-out it appears until finally, when saturation is at zero, the image becomes a monochrome or greyscale image. This is used as another measure of how well exposed the image is. The standard deviation of the three RGB values is calculated at each pixel to generate this measure. Given the vector of RGB intensities for the pixels in image $I_{RGB}(x, y)$ the saturation $S(x, y)$ is calculated as follows:

$$\sigma(I_{RGB}(x, y)) = \sqrt{\frac{(I_R(x, y) - \mu)^2 + (I_G(x, y) - \mu)^2 + (I_B(x, y) - \mu)^2}{3}} \quad (6.2)$$

where:
6.2. FUSION FEATURE SELECTION

\[ \mu = \frac{I_R(x, y) + I_G(x, y) + I_B(x, y)}{3} \]  \hspace{1cm} (6.3)

\[ S(x, y) = \sigma(I_{RGB}(x, y)) \]  \hspace{1cm} (6.4)

**Well-exposedness Measure** \( E \): This is a measure to weight the value based on its closeness to the maximum or minimum pixel values. Well exposed parts of an image will consist of pixel values close to 0.5 and as values get closer to zero or one they indicate under and over exposed areas. A Gaussian function is used to calculate a value \( g \) for each colour channel intensity \( i \) independently at each pixel and the values are multiplied to generate the final weighting \( E \). A \( \sigma \) value of 0.2 is used as suggested by Mertens et al. [82].

\[ g(i) = \exp \left( -\frac{(i - 0.5)^2}{2\sigma^2} \right) \]  \hspace{1cm} (6.5)

\[ E(x, y) = g(I_R(x, y)) \times g(I_G(x, y)) \times g(I_B(x, y)) \]  \hspace{1cm} (6.6)

**Entropy Measure** \( H \): Entropy is a measure of the uncertainty associated with a random variable. For an image patch which is uniform the entropy is zero as that information obtained from that patch indicates that the only likely prediction that can be made about future pixels is that they will be the same uniform value. For an image patch to have high entropy it requires that values are varied and contain many different pixel values. The prediction of the value of a future pixel based on this would be uncertain. This allows entropy to be used as a measure of exposure as a well exposed area is likely to have more variation in pixels values.

Shannon’s entropy [120] is used for this measure. For an image patch the entropy is calculated by binning the intensity values of that patch in a histogram \( p \), normalising it using the number of pixel values and then using the following equation:

\[ H(p) = -\sum_{i=0}^{n} p_i \log_2(p_i) \quad p_i > 0 \]  \hspace{1cm} (6.7)

where \( p_i \) is the probability that an arbitrary pixel in the patch has intensity \( i \) in the range of 0 to 255 for an 8-bit image. For the following experiments the patch is selected based on the scale of the feature.
A subset of all of the image measures can be used to select a preferred set of features. If more than one measure is used they are combined by multiplying and each can be scaled to vary the effect of each measure using the powers $c$, $s$ and $e$. The entropy measure is dependent on the feature size and is calculated for each feature independently. As such, it is incorporated later in the process. For this thesis all measures are scaled equally using the value 1.

$$W(x, y) = C(x, y)^c \times S(x, y)^s \times E(x, y)^e$$  \hspace{1cm} (6.8)

Each aligned exposure image will then have its own set of pixel value weightings. The resulting values are normalised to the range of 0 to 1 for the corresponding pixels in each exposure image as shown in Figure 6.3. Given a set of weightings images $W$, where $k$ denotes a specific image in the set, the images are normalised as follows:

$$W_k(x, y) = \frac{W_k(x, y)}{\sum_{i=0}^{n} W_i(x, y)}$$  \hspace{1cm} (6.9)

where $n$ is the number of images.

To select the features for the final set the weightings at each feature location are used, only the features from the best exposed locations will be preserved. The selection takes place over the area and scale that the feature was originally extracted. At each location at the scale of the feature, $\sigma$ is used to calculate an approximate radius of the feature; $6\sigma$ [70]. A Gaussian weighting of that radius with a standard deviation corresponding to the scale of the feature $\sigma$ is then applied to the weights centred on the feature position. The resultant values are
averaged across the total feature area and used to select the feature. The score for a patch of weights \( W_k(x, y) \) are calculated as follows:

\[
Score(x, y) = \sum_{x,y}^n W_k(x, y) \times G(x, y) \frac{n^2}{n^2 (6.10)}
\]

where \( n \) is the patch size \( (6\sigma) \) and \( G(x, y) \) is a Gaussian kernel of the same size. A feature is selected if the averaged value is greater than that for the same location in all the other images.

At this stage the entropy measure can be considered. As the entropy measure is patch based and cannot be calculated in advance and is dependent on patch size and location it can only be incorporated here. After calculating the entropy measure \( H \) for a patch the value is adjusted using the following equation:

\[
AH(x, y) = H(x, y)^{\frac{h}{m}} \quad (6.11)
\]

where \( m \) is the number of other measures used excluding the entropy measure. This value reduces the influence of this final measure to that of the others but this is only valid if the original measures use the same power value. The adjusted entropy \( AH \) is then normalised and can then be multiplied by the \( Score \) value to generate a final \( Score \) value. A feature is selected if the final \( Score \) value is greater than that for the same location in all the other images. If the entropy measure is used alone then a feature is selected if the entropy for that feature patch is greater than that for the same location in all the other images. An example of a final set of features selected using FAW is shown in Figure 6.4.

### 6.2.1 Feature Blending

The image alignment process uses RANSAC \[134\] to register matched features and calculate a transform to align the images. The features which are aligned between images can be merged for the final fusion feature set by averaging their vectors as they both must be in well exposed areas for both of them to match. The alternative is to treat these features like any other and select one based on their weightings.

### 6.2.2 Evaluation Methodology

The scenario for testing the feature fusion process is as follows:
Figure 6.4: The set of fusion features displayed on a rough exposure fusion image on the left and on a binary fusion image on the right. The binary image shows which areas are best exposed in each image. For iterative use of the system colours are used; yellow arrows indicate features selected from image 1 and turquoise indicate features that are selected from image 2. Blue and green arrows are from the features which match between the images and have been blended for the final feature set.

Figure 6.5: Example set of high contrast images used for testing.
6.2. FUSION FEATURE SELECTION

![Feature matching examples](image)

Figure 6.6: Feature matching examples represented by the parallel lines. The number in brackets gives the number of matches. Note that there are more fusion features matches.

A high contrast scene is obtained by using a spotlight in a darkened room or locating an area of shadow. Two aligned exposures of the scene are captured, each exposed correctly for the different parts of the scene. A third, target image, is captured. This is done by taking a picture of the scene after the scene lighting has been changed by turning on a larger brighter light source (the camera flash or ceiling light) which allows the whole scene to be captured in a single LDR exposure. Neither exposure image will match to all of the areas of the target image but a high dynamic range image created from both images should. This scenario relates to a real world scenario in which a well-lit target image has been captured under controlled circumstances and an attempt is being made to locate an object or scene where the dynamic range is now larger.

The two aligned exposure images are used to create a tone mapped image using Reinhard’s [107] techniques and an exposure fusion [82] image is also generated as shown in Figure 6.5. The parameters for these algorithms have been adjusted to generate images which are well exposed. If the exposure images are misaligned they are aligned first to get the best possible results [134]. A set of SIFT features are extracted from each resultant HDR representation. These processes represents the AWF paradigm as they are ordered; alignment, weighting and then feature extraction.

The two exposure images are used to create a set of fusion features. The three sets of features are matched to the target image using the nearest and second nearest neighbour technique as described by Lowe [70]. All the features from both LDR exposure images are also matched for comparison as shown in
CHAPTER 6. HIGH DYNAMIC RANGE FUSION FEATURES

6.2.3 Results and Analysis

Table 6.1 shows the mean results for each of the fusion features measures, tone mapped images, exposure fusion and the single exposures. The experiment consists of 28 image pairs matched to well-lit images images.

Figures 6.7 and 6.8 shows the difference distributions in the recall and $1 - \text{precision}$ when compared to the first input image. This is generated by subtracting the recall and $1 - \text{precision}$ from the exposure fusion results to show the difference. This gives a baseline for performance by comparing the data to the results of a single exposure LDR image. This is more useful for comparison than Table 6.1 as the absolute values can be effected by the quality of the image and the variability within the results can be caused by the image rather than the quality of the features.

Figure 6.9 shows a histogram of the recall and $1 - \text{precision}$ for all the 420 extracted fusion features for the 28 test cases generated using variations of the measures. The results of the first exposure images are subtracted from the results and comparative distribution is shown. Figures 6.10 and 6.11 show similar data with using the tone mapped and exposure fusion results instead of the first exposure image.

Figure 6.7: This graph shows the recall results used in Table 6.1. The values are generated from the difference between matched features and the features extracted from the first exposure image. Each bar shows the distribution of the recall. The horizontal line is the median and the dot is the mean. The box shows the interquartile range. The extrema for each bar are the maximum and minimum values.
Table 6.1: The mean and median results for twenty eight test exposure image pairs showing the number of features extracted, the matched features, the correspondence ratio, the *recall* and 1−*precision*. The best values for each metric are highlighted in bold. The method which perform best have the highest *recall* and lowest 1−*precision*. The label are as follows; **EF** - exposure fusion, **TM** - tone map, **C, S, E, H** - various combinations of the fusion features contrast, saturation, well-exposedness and entropy measures. **I1** and **I2** - the first and second exposure image.
CHAPTER 6. HIGH DYNAMIC RANGE FUSION FEATURES

Figure 6.8: This graph shows the $1 - \text{precision}$ results for the same images used in Table 6.1. The bars present the same information as in Figure 6.7 but for $1 - \text{precision}$.

Figure 6.9: The distribution of the feature fusion measure when compared to the first exposure image results. The histogram shows the distribution of 420 cases of various combinations of exposure fusion measures for 28 sets of images for fusion features and displays the difference in the recall and $1 - \text{precision}$ when compared to the first exposure image from the same sets.

General Comparison of FAW with Other Image Types

The first observation that can be made is that for all cases the mean and median of the fusion features performance is better than that for the tone mapped images and the exposure fusion features. The improvement in recall for fusion features is less consistent (Figures 6.7 and 6.9) then for $1 - \text{precision}$ (6.8). This is less significant as recall is a measure of missed potential matches whereas $1 - \text{precision}$ represents the accuracy of the matches that have actually occurred which is likely to be more important in a practical application. The results show that the $1 -$
6.2. FUSION FEATURE SELECTION

Figure 6.10: The distribution of the feature fusion measure when compared to tone mapped results. This is the same data as used in Figure 6.9 except with a tone mapped image used for comparison.

Figure 6.11: The distribution of the feature fusion measure when compared to exposure fusion results. This is the same data as used in Figure 6.9 except with a exposure fusion image used for comparison.
precision is consistently better for the fusion feature over the other feature types whereas the recall varies but is better on average.

The distribution of the results is approximated in Figures 6.7 and 6.8 where it is compared to a standard single exposure image, the first image used to generate the synthetic images. This case shows an improvement in recall 55% and an increase in $1 - precision$ 90% of the time.

For the fusion features variations when compared to the tone mapped images the recall is improved 80% of the time and the $1 - precision$ is improved 98% of the time. For the fusion features variations when compared to the exposure fusion images the recall is improved 84% of the time and the $1 - precision$ is improved 96% of the time. Figures 6.10 and 6.11 show these distributions when compared to the HDR and exposure fusion images respectively. The distributions show the advantage of using fusion features over both.

The results show that the feature set for feature fusion is generally smaller than for the other HDR techniques and there are fewer superfluous features. For example, exposure fusion generates, on average, 43% more features but only generates 18% more feature matches therefore the extra features provide a disproportionate increase in matches.

In Figures 6.7 and 6.8 it can be observed that there is a large variation for image 2. This is due to the fact in some cases, when the exposures are combined, one or other of the exposures will contain more features that correspond to the scene. This leads to the large range when the results are compared to the other exposure image. It shows the problem with having a single exposure in high contrast matching scenarios. The mean and median are consistently less than the fusion feature results and for $1 - precision$ the maximum is less than all of the fusion features.

The results show the advantages of using the fusion features and FAW over the synthetic images and AWF for these test cases. This is due to the artefacts, compression and changes in luminance which occurs when the synthetic images are created. Any slight misalignment can affect the resultant SIFT features whereas the fusion features are more robust to these errors. The fact that the fusion feature process relies on features which have been extracted from scene results in fewer processing stages and avoids the selection of good parameters which is required for the tone mapping process. The weighted pixel averaging that takes place in the exposure fusion and tone mapping processes effects the quality of the
pixel values as poorly exposed areas can still negatively affect the final, average, pixel values.

The reason that the performance improvement is less significant when compared to the single exposure is that no artefacts are introduced and that the areas that are well exposed within the image are likely to match. In cases when the well-lit image is not greatly different from the single exposure image or where the main features of the scene exist in the single exposure it will match well to the scene and not contain extra unnecessary features that may be added by a second exposure. The single image is close enough to the well-lit image and extra information provided by the second image does not improve the feature set and introduces extra unnecessary information which increases the likelihood of incorrect matches. In the cases where the dynamic range of the scene contains a more even spread of features then a single exposure can accommodate then the fusion features will outperform a single exposure. Also a large dynamic range scene requires the correct exposure to be selected and selecting one exposure may not be the best one as is shown by the large range for $I_2$ in Figures 6.7 and 6.8.

**Best Performing Measures**

Given that in many case the fusion features have been shown to out-perform the other image types consistently the next question then arises of how to select the fusion features and which measure or combination of measures are best. The mean results are shown in Table 6.1. Twenty eight aligned exposure pairs were used and they show that fusion features perform better than the synthetic images generated from exposure fusion and tone mapping in high contrast scenarios. The combinations with the highest recall and lowest $1 - precision$ are SH, CSE and CSEH for both mean and median results with CH also performing significantly well with a slightly higher median $1 - precision$.

Figure 6.8 supports these choices. It shows that the range and the skew along with the median and mean of these measures also outperform the other measures and combinations displaying a more negatively shifted distribution for $1 - precision$.

For the cases when a single measure is to be used the best is C. The H measure has the highest recall but has the lowest $1 - precision$. Combined measures out-perform the individual measures. The combinations show that the saturation
measure \( S \) is in all the best performing measures combinations but has poor individual performance with a very large range. \( E \) when combined with another single measure is shown to not significantly improve the performance, and sometimes decrease, the performance of the measure it is combined with.

On the exclusion of \( E \) as it provides minimal advantage and can be detrimental the best remaining combinations are \( SH \) and then \( CH \). This shows that the entropy measure \( H \) is important for the best performance and that both \( C \) and \( S \) perform well with it. A combination of the saturation measure and the entropy measure appears to be a recommended measure combination.

**Analysis**

It is difficult to determine why some measures perform better than others due to the number of variables that are present in these experiments. Some insight into the reasons behind the performance can be uncovered based on how the measures function.

The reasons why the well-exposedness measure \( E \) performs badly is that it is potentially based on a unsound assumption. It assumes that pixels that are closer to the centre of the possible range (0.5) are better exposed than pixels with a value closer to the edges of the range. This is not always true as the value of the pixels are not only dependant on the exposure but also the scene. The assumption only applies to pixels close to the edge of the range and therefore this method cannot be reliably used to weight the pixels across the whole of the range of possible values. A pixel with a value of 0.5 cannot be assumed to be better exposed than one with a pixel value of 0.75. Only extreme cases should be deemed unreliable and a Gaussian scaling may not be suitable as shown in figure 6.12.

![Figure 6.12: The result of the well-exposedness (E) measure returned for a pixel value.](image)

The saturation measure \( S \) gives an estimate of how much light intensity is distributed across the spectrum of different wavelengths. This is related to exposure;
as exposure increases the saturation decreases as the scene becomes washed out. This is because with the RGB model as colours get more saturated one channel becomes more likely to hit the maximum possible value and start clipping. For under exposure the channel values will all be low. Unlike the Gaussian distribution of the well-exposedness the values of the saturation can be high even with values at extremes of the range as long as all the channels are not close together.

Saturation and well-exposedness measure a similar effect; the pixel channel distribution. The standard deviation is shown to be a better measure for this type of exposure estimation rather than associating it with an artificial Gaussian distribution. The entropy and contrast measures are similar in that they measure the amount of change in a patch and whether there is information contained within it. This is less directly tied to exposure in that the weighting is not based on whether the pixel channels deviate but whether local pixels are different.

The contrast measure gives a value for a patch describing the changes between neighbouring pixels. This works well as it is based on the change in a patch. When two patches are compared the one with the most change is more likely to be a well exposed area and this is the best performing single measure. The SIFT selects the features as the original features locations based on extrema (the highest contrast points) and that a contrast measure which uses a Gaussian weighted patch to threshold the features. This can explain why the contrast performs best individually as it directly relates to how the features were primarily selected.

Entropy gives a value that describes the distribution of values within a patch. It is independent of how far apart the values are as with the contrast measure, just that the values differ. This is successful in determining if there is information within a patch but does not indicate to the same extent as the contrast measure the value of that information. Two patches with different contrast values may return the same entropy measure value. This measure is successful in conjunction with another measure and improves the performance of the saturation and contrast measures.

Aside: Results for Specific Images

Some pairs of images when combined into a set of fusion features match less successfully than a single exposure image. For the images used in these experiments the average performance for the various fusion measures can be seen in Figure 6.13 for each test case. It can be seen from this that for $1 - \text{precision}$ the cases
Figure 6.13: The mean values of the FAW methods for $1 - \text{precision}$ when compared with the results for image 1 for each of the test images. The results show a mean improvement for 25 out of 28 cases for $1 - \text{precision}$. The green line shows the mean.

3, 19 and 27 do not improve on the single image. For all other cases the median and mean show improvement. By looking at these cases and the images used the following observations can be made:

**Test Case 3:** Figures 6.14(a) to 6.14(c). These two exposure images could not be mapped to each other by a single homography. As such the measurement of the $1 - \text{precision}$ cannot be relied on as it is only correct for the first of the two images. The slight misalignment means that the features from any of the synthetic images cannot be relied on.

**Test Case 19:** Figures 6.14(d) to 6.14(f). This is a poor result as the flash did not brighten the scene sufficiently to vary the target image significantly from the first exposure image. The target image looks very much like the first exposure image. This means that the fusion features will include extra features from the second exposure and can increase the false positive count.

**Test Case 27:** Figures 6.14(g) to 6.14(i). The fusion features results show little improvement in this case as the second exposure image mainly contains a repeated brick pattern. Repeated patterns perform badly when using the ratio matching technique as the second nearest neighbour will be close to the best match. The second exposure therefore provides few extra matches and increased false positives.
These examples show that it may not be a good idea to use fusion features in a scenario where HDR scenes are not expected although even in these test cases the results show that performance is close to that of the single exposure. The best cases for fusion features, such as case 20, have a balance of features from both scenes.

![Exposure images and fusion examples](image)

Figure 6.14: Test case 3, 19 and 27: Examples of the test cases which performs worse than the single exposure for $1 - \text{precision}$. The show the first exposure image, the exposure fusion image and the target image. The exposure fusion shows how the high dynamic range scene compares to the target image.

### 6.3 Stereo Fusion Features

Stereoscopic systems are common in computer vision applications. To utilise this and extend the dynamic range of such systems it is proposed that the two cameras have different exposures values (EVs) resulting in a lower quality 3D reconstruction but increasing the dynamic range for feature matching. This can
be achieved using a neutral density (ND) filter on one camera so that the exposure time can remain consistent between the cameras. This may be preferable in some circumstances where an increased feature matching range is desirable over high quality 3D. Stereo fusion features is the process of generating fusion features from misaligned stereo images of varied exposure.

When using stereo images to create tone maps often, after warping, the images do not align correctly. This is due to the absence of a homography which will correctly warp all areas of the image and leads to ghosting and edge effects which means that features extracted from a synthetic image generated from these pairs may contain errors. Figure 6.15 demonstrates the problem but it is proposed that because the fusion feature process does not generate new images or features then this problem is negated.

A compromise can be made between good 3D and good HDR images by varying the exposure difference and baseline of the stereo images. A stereo pair with a small baseline will generate a poor 3D representation but will allow the images to more easily registered for HDR. A large baseline has the opposite effect. The exposure difference between the stereo cameras has an effect as a large difference will make the dynamic range of the features increase but make it more difficult to match features between the images as is shown in Figure 6.16.

Bundler [125], a structure from motion tool which utilises bundle adjustment, can be used to generate a 3D model of the features and indicate which features can be aligned, Figure 6.17. This subset can be used for the projective transformation from one image to the other. If the 3D data is not required RANSAC alone [134] can be used for alignment. The second image is transformed to align with
6.3. STEREO FUSION FEATURES

Figure 6.16: A graph showing the number of feature that are used to create a set of 3D points from stereo pairs at various angles and exposures. The exposure axis values represent the change in EV between the image pair. The data has been generated from 12 pairs of images, similar to those in Figure 6.17, using Bundler [125]. As the exposure and angle difference increases the number of features that can be matched to create the cloud decrease. This demonstrates the trade-off between the number of reliable 3D features and the dynamic range captured.

Figure 6.17: The set of features selected from two stereo input images in Figure 6.15. A lower quality 3D point cloud is generated than if the EV values were the same but the dynamic range of the feature set is higher. Features have been selected from the second image on the toe area of the shoe which is over exposed in the first image because of the light shining on it.
Figure 6.18: A graph showing the correspondence ratio for fusion features and features extracted from exposure fusion images generated from 28 pairs of stereo images. The x-axis shows the stereo pair disparity in degrees (plus or minus refers to left or right of the first image) and the EV of the two images. The features are all matched to a single target image taken at $0^\circ$ and 0 EV at approximately 1 foot away from the shoe. The images used are the same as those used for Figure 6.16 and resemble those shown in Figure 6.15.

The features produced using stereo images will provide information about the presence of that object in a scene. Features outside the bounding box are unreliable in their exact location due to the lack of a projective transformation which will accurately transform all the feature locations from the second image to the first.

### 6.3.1 Results

The evaluation has been conducted in a similar manner to the standard fusion feature tests. A set of twenty eight stereo images have been used with two sample images shown in Figure 6.15. They consist of the stereo pairs taken at measured exposures and angles. The second image and its feature positions are warped to best align to the first before exposure fusion takes place. The results are shown in Figure 6.18. In all cases the greater correspondence ratio for feature fusion demonstrates the advantages over the exposure fusion and tone mapped techniques.
6.3. STEREO FUSION FEATURES

6.3.2 Analysis
The difference between the feature fusion and other results for the stereo test cases is because of the substantial ghosting effects which are exaggerated as the stereo baseline is increased. The advantage of the stereo tests is more useful in the lower baseline examples where the images align well with a projective transformation and as such the use of the feature fusion technique is valid. As the angle increases the 3D object cannot be satisfactorily aligned with a projective transformation and as such aligned areas of the images which represent the same positions in space become smaller thus the fusion feature technique becomes less reliable.

6.3.3 Conclusion
The process introduced in this chapter allows sets of features to be generated which allow matching to take place in high contrast environments. This is advantageous as it allows objects to be detected using features which may otherwise be hidden in a single exposure image. The performance advantage of using the fusion feature technique has been demonstrated over extracting features from exposure fusion or tone mapped images. This is due to the artefacts and changes that are introduced to these synthetic images which create features that do not always match to features taken from images captured directly from a scene. The advantages of FAW over AWF are clear as FAW reduces artefacts introduced in the image processing stages.

Other advantages of using the process include the robustness to misaligned 3D images at small changes for non-projective scenes. Misaligned images will make noisy tone maps and exposure images but using the fusion as a way of selecting features is better than trying to generate new ones. Fusion features do not require a HDR image to be generated therefore does not require as many resources to be consume and intermediate steps.
Chapter 7

A contrario SIFT Feature Weighting

The previous chapter focused on improving SIFT matching in high contrast environments. This chapter introduces novel techniques for object detection and feature weighting. The process is called FEWER; Feature Extraction and Weighting for Enhanced Recognition. The process solves the problem of matching a target stereoscopic image pair of a 3D object to a hand-held stereoscopic video sequence to locate the object.

The process relies on the novel a contrario feature matching methods outlined in Section 4.4.1 for calculating weightings from false positives. It provides the means to select the best matches from a set of matching features by applying a confidence weighting to each match. This is a problem as often features in a set are matched incorrectly. By learning which features do not match correctly we can predict the matches which are most likely to be correct using the a contrario methodology.

This chapter focuses on selecting a set of measures which when combined can help identify a correct feature match. SIFT matches can be identified using the RANSAC geometric consistency method [70] but this does not weight the matches. This is an experiment to validate the use of a background model to identity the best feature matches in the domain of feature matching. The setup used is one possible method but other metrics could be identified for generating the weightings. This research uses stereo pairs to generate extra information for each feature to create weights. The system can be used as an added layer to extend standard matching techniques.
To generate the weights stereo pairs of images are used for both the target object and the scene in which the object is being located. Lowe states in his paper “once the keypoint is repeatedly located, it is likely to be useful for recognition and matching tasks” [70]. This is one of the key aspects of this work. Once features have been identified that match between stereo views of the same object then it is more likely that they are more stable and reliable when matching to new views.

7.1 FEWER: Feature Extraction and Weighting for Enhanced Recognition

Following the \textit{a contrario} methods outline in Section 4.4 for subtraction of SIFT noise a second process has been developed which utilises the 3D stereoscopic image pairs of the target and scene to specify weighted feature matches to indicate confidence in their accuracy. This is called FEWER; Feature Extraction and Weighting for Enhanced Recognition. A pair of target images of the object that is being detected and a pair (or stream of pairs) of stereo images of a scene are used. Simply put, if a feature does not match well to its counterpart in a stereo pair the chances of it being stable are lower. The process has nine stages:

\textbf{Extract SIFT Features} Extract the features from the target and scene stereo pairs as shown in Figure 7.1.

![Figure 7.1: A stereo pair of target images displaying the SIFT features extracted from them. There are 2176 in the left image and 2087 in the right image.](image)

\textbf{Calculate 3D Positions} For both the target and scene pairs a 3D point cloud is generated from the features as shown in Figure 7.2.

\textbf{Cluster 3D Data} The 3D matched features are then spatially clustered in 3D space (using k-means [74]). Clusters help differentiate between foreground and
CHAPTER 7. A CONTRARIO SIFT FEATURE WEIGHTING

Figure 7.2: The set of 3D feature positions generated from the stereo pair in Figure 7.1 using the Bundler API [125]. The first two images show two different angles for the same data and the curvature of the shoe is clearly visible. This is a subset of the total features extracted from the original images and consists of 885 features. The right hand image shows the type3 features spatially clustered.

background objects.

Figure 7.3: The final set of clustered type2 and type3 features for the left and right images in a stereo pair. There are 1802 in the left image and 1782 in the right image.

Feature Labelling Three different feature types are defined depending on their 3D and cluster properties. Type3 features are labelled by mapping the 3D features back to their 2D image locations for each image. Type3 features are those which have 3D information associated with them and therefore match to the other stereo images. To define type2 features a distance threshold is used to find other features near each of the type3 features. These features are likely to be part of the same object as they are nearby but as they do not match to the other stereo image they can be considered less reliable. The remaining features are then labelled as type1 and they do not have any cluster information relating to them.

Target to Scene Matching Feature matching is performed for each target to scene combination; left target to left scene, left target to right scene, right target to left scene and right target to right scene. This is done using the nearest neighbour technique described by Lowe [70].

Initial Weighting Each target image has its own set of weighting for matches to both of the scene images. Thus four sets of weightings are calculated. The
7.1. FEWER

Figure 7.4: The set of clustered features in a scene input image. The advantages of spatial clustering are clearer here as various objects have are roughly separated by the different clusters so as to provide more information when matching features.

initial weightings for each feature are given by which type they are and which type they match too. A \textit{type}3 target to \textit{type}3 scene match will have a larger initial weighting than a \textit{type}3 target to \textit{type}1 scene match. There are therefore nine possible combinations of matches each with their own weighting.

\textbf{Type 3 Mismatches} The weightings are then adjusted by checking if matching pairs of \textit{type}3 features from each target image match to similar positions in the scene images. Figure 7.5 illustrates these cases. If the same \textit{type}3 feature in both of the target images matches to different points in the scene the weighting is reduced. The weighting is effected differently if the single scene feature is \textit{type}3 or not \textit{type}3.

Figure 7.5: This shows the two cases of \textit{type}3 mismatches. Case \textit{a} shows the \textbf{correct} (lighter) and \textbf{incorrect} (darker) matches from \textit{type}3 features in the target images to any type of scene feature. Case \textit{b} shows the \textbf{correct} and \textbf{incorrect} matches from the target image to the \textit{type}3 scene features.

A secondary check is carried out for each target feature which matches to a \textit{type}3
scene feature. If the feature matches to both corresponding type3 scene features then the weighting is increased. If a target feature matches a type3 scene feature and also matches a different feature in the other scene image then the weighting is reduced. There is no effect if the target feature matches one scene but not the other. Again the weighting is affected differently if the single target feature is type3 or not type3.

Cluster Weightings The next stage is to adjust the weightings based on the 3D spatial cluster that a feature is in and how groups of features in the same cluster match. The basic hypothesis is that as more features in a target cluster match to a specific scene cluster the more likely it is that there is a correspondence between these areas of the scenes. The confidence weighting is calculated as follows:

$$\text{confidence} = \frac{\text{signal}}{\text{noise}} \times \sqrt{\text{sample size}} \quad (7.1)$$

where signal is the correspondence ratio from a target cluster to a scene cluster, noise is the correspondence ratio from the target cluster to every other scene cluster and sample size is the total correspondence ratio from the target cluster to all of the scene clusters. Multiplying by the square root of the sample size reduces the effect of the small samples. A confidence value is calculated for each target cluster to every scene cluster. This equation means that the sample size and the signal both have to be significantly large to generate a high confidence thus a low numbers of matches will not be statistically significant when calculating a feature’s weighting. This confidence value is thresholded so that a high confidence cluster pair will result in a higher weighting for features which match between them. The boundaries and distribution of the clusters can affect the performance of this technique and as such there is no negative weighting for low confidence.

Threshold Matches The weighting is normalised transforming its value into the range of 0 to 1. A threshold can now be applied to select a subset of the weighted feature matches.
7.2. CALCULATING FEATURE WEIGHTS

<table>
<thead>
<tr>
<th>Stage</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>Extract SIFT Features</td>
<td>Sets of SIFT image features.</td>
</tr>
<tr>
<td>Calculate 3D Positions</td>
<td>Relative 3D positions of matched features.</td>
</tr>
<tr>
<td>Cluster 3D Data</td>
<td>Index of features indicating the cluster they are contained in.</td>
</tr>
<tr>
<td>Feature Labelling</td>
<td>Features labelled by type.</td>
</tr>
<tr>
<td>Target to Scene Matching</td>
<td>Indices indicating how features match to the scenes.</td>
</tr>
<tr>
<td>Initial Weighting</td>
<td>Weightings for each feature match.</td>
</tr>
<tr>
<td>Type 3 Mismatches</td>
<td>Updated weightings based on a disparity in matches.</td>
</tr>
<tr>
<td>Cluster Weightings</td>
<td>Updated weightings based on matches between clusters.</td>
</tr>
<tr>
<td>Threshold Matches</td>
<td>Set of matched features with weightings above a threshold.</td>
</tr>
</tbody>
</table>

Table 7.1: The stages used for extracting and weighting features with FEWER.

7.2 Calculating Weights Using the *A Contrario* Methodology

Values for the FEWER weighting adjustment stages described above have to be calculated to weight various characteristics of a matched feature. This is done by studying the noise properties for each stage using a set of stereo features known not to match correctly. By looking at the level of false positives for various feature match types, ratios can be calculated which indicate how much more reliable one type of match is than another. The data describes how each type of match is affected by false positives.

For the initial weighting stage the correspondence ratio for false positives for each match combination is calculated using large sets of random features. They are matched to videos which are known to contain no correspondence to the scene image. By obtaining the average correspondence ratio across a number of frames and adding the standard deviation it can be seen for the test data that type3 to type3 feature matches have a correspondence ratio 16 times less (0.64 / 0.04 from the full set of example data listed in Table 7.2) than type1 to type1 thus the weighting reflects this directly. The weighting \( w \) is calculated as follows:

\[
w = k \frac{1}{\bar{x} + \sigma} \times \frac{\text{relevant matched features}}{\text{total matched features}} \tag{7.2}
\]

where \( \bar{x} \) is the mean noise value across a sample, \( \sigma \) is the mean standard deviation of the noise and \( k \) is a scaling factor to make the numbers more convenient for human evaluation. The *relevant matched features* are the subset of the *total matched features* actually involved in the particular weighting process so that the weightings are scaled accordingly.
### Table 7.2: An example of weighting values calculated from experimental data for different aspects of the weighting process.

<table>
<thead>
<tr>
<th>Type</th>
<th>Target</th>
<th>Scene</th>
<th>Type 3 Target</th>
<th>Type 3 Scene</th>
</tr>
</thead>
<tbody>
<tr>
<td>Type 1 T</td>
<td>0.04</td>
<td>0.07</td>
<td>0.12</td>
<td>0.11</td>
</tr>
<tr>
<td>Type 2 T</td>
<td>0.06</td>
<td>0.07</td>
<td>0.09</td>
<td>0.06</td>
</tr>
<tr>
<td>Type 3 T</td>
<td>0.11</td>
<td>0.06</td>
<td>0.12</td>
<td>0.07</td>
</tr>
</tbody>
</table>

The same process is used to calculate the weightings for the type 3 mismatches where the number of false positives matches are used but as only the type 3 features are involved the relevant matched features value reflects this. This incorporates a negative weighting for mismatches which have a relatively high cost as seen in Table 7.2.

For the cluster weightings, analysis has provided data on how well false positive matches cluster and what is the minimum level of cluster matching confidence required to occur beyond random chance. This allowed a cluster confidence threshold to be calculated using the same equation and a weighting for values greater than the threshold to be defined. This only relates to type 2 and type 3 features as type 1 features are not clustered. The threshold was calculated to be 0.00015 and the weighting value added to matches greater than this threshold is 0.4 when using six clusters. This is calculated from the mean summed with the standard deviation of the cluster response from the background model images.

After these three stages the maximum possible weighting that can be achieved using this experimental data weightings is 1.36 and this value is used for normalisation.

#### 7.3 Evaluation Methodology

The experiments use three videos of 1537, 2500 and 1062 frames labelled as A, B and C. Each stereo video has a target object located within part of the sequence which is to be detected. Stereo images of the target objects are matched to each frame using the techniques described. The system outputs the four match images for each combination of target to scene matches with the matched features drawn using a heat map style colour coding. The colour changes linearly through RGB space from blue to green to red as the weighting increases.
7.3. EVALUATION METHODOLOGY

The background model used consists of six stereo pairs unrelated to the video sequences. The weightings for the features are trained using these negative results on the twenty frames of the sequence previous to the current frame. This means that the weightings calculated for the various feature types are based on the current scene context. For each scene video the object is matched and the matches are weighted based on the average weightings of the six background model images. The weightings have been tested with and without the use of the clustering stage of the algorithm to analyse if this provides any advantage.

The tests are compared with result of using RANSAC to select features based on a geometric constraint by calculating the homography between a target and scene image and selecting only the features which fit with the homography. The analysis focuses on the left image of both the scene and the target images. To evaluate the matches a random subset of the images have been selected and analysed individually.

![Figure 7.6: The left hand view of each of the target image pairs.](image)

![Figure 7.7: Stereo background model images for the a contrario technique in the following test. Each is one of a pair of images used. They were chosen as non-relevant 3D image pairs which are very unlikely to occur within the test scenes and provide a measure of the likelihood of matches occurring by chance.](image)
7.3.1 Examples of Matching and Weighting

Figures 7.8 and 7.9 show examples of the coloured weightings as feature matches and the images are consistent with the other frames in the sequence showing that incorrect matches are weighted lower.

Figure 7.10 shows the correspondence ratio across the 2500 frames and the large peak indicates the location of the target. By adjusting the weighting threshold it is shown that the false positive count is reduced leaving many of the most reliable features. The weighting threshold could be computed adaptively by analysing a set of known false positive feature matches in a similar manner to Section 4.4.1 and adjusting the weighting to minimise them.

![Figure 7.8: A typical example of weighted feature matching displaying matches from the left hand target image to the left scene image. Some of the correct matches are green and red indicating higher weightings. The mismatched features in this scene have received low weightings and are coloured blue. The feature matches with low weightings can be removed by adjusting the weighting threshold which is set at 0 in these cases. The graph below shows the weightings for each of the 33 matched features and whether they match correctly.](image)

7.3.2 Selecting a Threshold

The selecting of a weighting threshold determines the size of the resulting feature set. For these experiments an initial threshold of 0.3 was used followed by using 0.7. These values were chosen by analysing the frames in the sequences which
7.3. EVALUATION METHODOLOGY

Figure 7.9: A typical example of weighted feature matching displaying matches from the left hand target image to the left scene image. This shows false positives matches successfully being weighted with lower values.

Figure 7.10: This shows the correspondence ratio before and after applying a threshold on the feature weightings. The graphs are the mean of the four possible match scenarios (each target to each scene). The peak indicates the location of the object. The left graph shows the correspondence ratio when no threshold has been applied and the right graph shows what happens when a threshold of 0.9 is applied. This reduces the remaining correspondence ratio substantially but the features remaining are of a higher quality and fewer false positives are present across the video sequence. The threshold for these graphs have been generated from the background model applied to all of the frames and the results averaged.
Figure 7.11: These are the mean correspondence ratio graphs for the three feature types for matches from both target to both scene images. It can be seen that the type 1 feature matches have fewer peaks and troughs and the green (lighter) areas, where the object is not present are harder to distinguish than for the type 3 feature matches. They therefore resulted in lower weighting (see Table 7.2). For the non-matching data used for calculating weightings in Section 7.2 these graphs are flatter with lower correspondence ratios. They display the random noisy correspondence ratio and give a minimum baseline for noise for each feature type.

were known not to contain the target image and assessing the number of false positive matches for a given threshold.

Figures 7.12 and 7.13 show that there are diminishing returns in threshold values beyond 0.3 and the next significant drop in the graph is for the value 0.7. After 0.3 the number of false positives that are removed by further thresholding becomes less significant.

Figure 7.14 shows the distribution of the weightings across the whole of each set of data. The weightings are based on the previous 20 images. There is a high count for 0 as this is the value given to non-matching features. The next positive bin is also high as this tends to be Type 1 to Type 1 matches. Beyond these initial peaks the distribution of weightings is much flatter. The negative weightings occur when one of the later match tests are failed. There are fewer of these in case B as most of the frames do not match to the target so there are fewer overall matches and the distribution contain a higher number of lower weighted features.
7.4. RANSAC COMPARISON

<table>
<thead>
<tr>
<th>Threshold</th>
<th>Mean Matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>5.24</td>
</tr>
<tr>
<td>0.1</td>
<td>2.57</td>
</tr>
<tr>
<td>0.3</td>
<td>0.54</td>
</tr>
<tr>
<td>0.5</td>
<td>0.47</td>
</tr>
<tr>
<td>0.7</td>
<td>0.13</td>
</tr>
<tr>
<td>0.9</td>
<td>0.03</td>
</tr>
</tbody>
</table>

Figure 7.12: This shows the mean number of matches as the weighting threshold for 4400 frames which do not contain the target object. It demonstrates how increasing the threshold decreases the number of false positive matches.

7.4 RANSAC Comparison

A selection of 20 frames have been chose at random (using a random number generator) from each sequence and analysed; A, B and C. The results have been compared to the same frame with RANSAC applied to the matched features and the features which do not fit to the calculated homography are rejected. This takes place for the left hand images from the target matched to the left hand scene image. The true negatives and false positives are the most easily identified by and eye and are counted. From these and the total number of matches the performance for each system can be evaluated.

For the FEWER technique the threshold of 0.3 has been used. The matches have been plotted on the images and each viewed to identify the correctly and incorrectly matched features. Tables 7.3, 7.4 and 7.5 contain the results of the matching. Figure 7.15 shows a graph of the matches which have been identified by each process. B has been left out from this as most of the data are where the target image is not present and as such the ideal result for this case is 0.

7.4.1 Results

It can be seen from the results that on average there are a higher number of false positives and a lower number of true negatives for the FEWER technique when compared to RANSAC. Also, in the two cases where the object is present, the number of identified features is lower. For case A the difference between RANSAC, on average, for the number of false positives is 0.44 and for the true negatives is 0.61. For case B the differences are 0.7 and 0.7 and for C the values are 0.05 and 0.05. Overall RANSAC performs better than FEWER at identifying
Figure 7.13: The number of features remaining in test set B after various thresholds have been applied; 0, 0.1, 0.3, 0.5, 0.7 and 0.9. This uses the last 20 frames in the sequence to generate the background model, causing the peaks after some of the larger gaps in the sequence, as the weighting values are not generated from sufficient data.
7.4. RANSAC COMPARISON

Figure 7.14: Histograms with logarithmic y axes of the feature weightings across each of the test sets generated by matching the target images.
correct matches and rejecting incorrect matches.

The reason for this is that RANSAC uses a geometric transformation to calculate the correct point matches whereas FEWER is probabilistic based on previous data. Due to this errors will occur but by increasing the threshold this can be reduced as shown by Figure 7.12.

However, the tests do show that the concept does work and is an improvement over just utilising the set of features as they are first matched and show that the a contrario methodology has utility in identifying the features which have matched correctly which is demonstrated by Figure 7.13. If the same number of features were chosen at random the number of false positives likely to occur by chance is higher than for FEWER.

Table 7.3: The results for test case A for RANSAC (R) and the left pair of images using the FEWER (F) technique. For each of the images selected at random from the sequence the total number of matches and those which have been identified as correct are shown for both methods. With this the number of false positive matches (FP) and the number of correctly rejected matches or true negatives (TN) are given.

Table 7.4: The results for test case B for RANSAC and the left pair of images using the FEWER technique. The labels are the same as those for Figure 7.3.

### 7.4.2 Thresholding Using 0.7

Table 7.6 shows the number of identified features and the number of false positives used for the three sets of 20 selected image when the threshold was set at 0.7. The results show a reduced feature count with a near 0 error count (1 false positive
7.4. RANSAC COMPARISON

| C Frame | 57 96 193 316 336 373 375 488 555 655 678 716 724 906 966 1023 1035 1048 1058 1062 | Mean 173.05 |
| Total   | 109 70 21 242 68 20 37 51 16 81 966 1048 1058 1062 | 142.30 |
| R Identified | 80 50 0 195 45 0 17 21 0 62 160 72 275 286 133 67 340 341 372 352 |
| FP      | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |
| TN      | 10 18 21 6 17 16 18 24 16 18 2 9 2 7 18 20 2 1 2 2 |
| F Identified | 14 16 0 47 21 0 2 2 1 12 53 14 88 58 23 13 97 111 114 74 |
| 0.3 FP  | 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 |
| TN      | 10 18 21 6 17 16 18 24 15 18 2 9 2 7 18 20 2 1 2 2 |

Table 7.5: The results for test case C for RANSAC and the left pair of images using the FEWER technique. The labels are the same as those for Figure 7.3.

Figure 7.15: Scatter plots of the number of selected matches in test cases A and C. They show the number of features selected using the RANSAC technique plotted against the number of matches selected by FEWER for a threshold of 0.3. The results show a strong correlation between the two with the RANSAC implementation generally selecting a greater number of matches.

across the 60 selected images). This shows that the threshold can be change to select a highly accurate, reduced set of features where only the matches which the algorithm is most confident about are selected.

| A Frame | 56 157 312 327 554 649 703 760 798 874 1021 1088 1204 1222 1252 1274 1312 1324 1348 1489 | Mean 8.1 |
| Identified | 28 0 6 29 0 0 0 0 18 6 0 3 11 13 8 3 3 0 n/a 26 |
| FP       | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 n/a 0 |
| B Frame | 350 361 585 619 686 815 904 988 1087 1100 1230 1287 1416 1590 1701 1881 2098 2144 2168 2192 |
| Identified | 0 0 4 3 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 |
| FP       | 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 |
| C Frame | 57 94 193 316 336 373 375 457 555 655 678 716 724 906 966 1023 1035 1048 1058 1062 | Mean 0.4 |
| Identified | 8 0 2 18 0 1 0 0 3 3 13 40 40 14 12 53 65 69 43 |
| FP       | 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 |

Table 7.6: This shows the results for the selected frames with a threshold value of 0.7. The false positive rate is lower than for 0.3 and the number of identified features have been reduced.

### 7.4.3 Error Cases

Some of the examples Table 7.3 show errors which occur in the weighting process. For example, case 554 is an edge case where the target object is at the left
hand edge of the left hand image. This means that the object is not visible in the right hand image and the feature weightings cannot be generated. This is especially prevalent when the cameras are close to the scene objects and the disparity between the views increases.

Figure 7.16: An example of a FEWER edge case. All of the valid matches have been given low weightings as they do not exist in both scene images.

A second error occurs in case 1348 where motion blur in the left hand image has caused a discrepancy between the two scene images. This results in no weightings being generated and a failure to identify matches.

Figure 7.17: An example motion blur causing an error in FEWER. This has resulted in too few matches between the scene image to generate any weighting data as such FEWER fails in this case.

Errors also occur when there is a break in the sequence of usable frames for a period of time greater than 20 frames. When the system is reintroduced to frames which contain stereo information the training takes place on only a single frame as that is the only valid scene pair in the sequence from the previous 20. This results in incorrect weightings being applied and the resulting peaks in feature matches are shown in Figure 7.13.
Another source of error is the clustering. The k-means clustering is naive as it assumes certain things about the cluster which may be false. It uses a Gaussian model for the cluster shape and requires a number of centres to be predefined. This set number of centres means that a weighting can be calculated as both the background model and target will use the same number. This can introduce errors as the objects in the scene may not fit to the Gaussian cluster distribution and background features may be included. Test have shown that the process performs better with the clustering then without. Improved clustering methods could be introduced in future work.

Finally, the probabilistic nature of the system introduces and inherent chance of error occurring as the weightings are an estimate based on unrelated images and the matches that will occur.

7.5 Evaluation

The FEWER process has been shown to weight the features with a variable success rate based on the chosen threshold. It relies on the probability of a feature type being a mismatch therefore, in some cases, incorrect matches can be weighted highly and vice-versa. The weighting threshold provides a sliding scale between a small number of highly reliable matches and a large number of features including more unreliable matches.

The reason FEWER works is that type3 features are likely to be more stable than the other features as they correspond between the stereo images and are therefore known to match to a different view of the object. The stereo depth process [125] could be removed and normal SIFT matching used instead to generate type3 features but these are likely to be less reliable. The stereo depth process has its advantages for clustering and background separation and is more discriminative when matching than using the SIFT as the matched features have to fit correctly to a 3D model not just match. The type2 features are more stable than type1 as the features are likely to exist on the objects that have been matched between the stereo objects due to their proximity to the type3 features and less likely to be background features. Type1 features are the least stable and have no extra properties associated with them. The difference between them is highlighted in Figure 7.11.

Table 7.2 shows examples of the normalised weightings applied to features.
The example shows that the initial weightings are highly favourable towards the matches which include the Type 3 or Type 2 features. Lower feature types can be weighted more highly if they correspond to a more reliable type e.g. Type1 to Type3. Type1 to Type1 will mainly rejected as they have a low score and cannot gain further weighting increases later in the process. Other features matches, Type1 to Type3 or Type2 to Type2 may begin with a lower weight but can increase their weightings through the mismatch checking or clustering stages to increase the possibility of their selection.

FEWER allows the system to select a subset of features which are higher in confidence rather than just thresholding using the noise properties in Section 4.4.1 which has no indication of which features are likely to be correct. A combination of the noise thresholding for detection and FEWER could be used so that the computationally expensive weighting process is only applied to frames which are likely to contain the object to select the best matches.

The results have shown that the weighting system in its current form does not perform as reliably as RANSAC due to it’s probabilistic nature. It does do something different to RANSAC in that it attempts to weight the matches. It has been shown that the results can be improved by altering the threshold. It does however validate the possibility of weighting features using a contrario techniques and the correspondence ration to select features which are more likely to be correctly matched.

### 7.6 Conclusion

The results of this work are promising and provide a technique for identifying and selecting the best feature matches. The results have shown examples of features being weighted to indicate which matches are correct and which are incorrect. The advantages of FEWER is that the weighting process provides a selection of matches with higher confidence than the standard SIFT matching alone. The system could result in lower data transmission rates as fewer matched features are selected.

Further development of the algorithm can use the 3D information gathered from the process to allow the process to determine if the object is flat (a drawing of the object) or an actual 3D object. Comparison could be made to other methods for reducing the number of incorrect matches such as the Hough binning used by
Lowe [70].

The system contains many parameters to vary and analyse in order to determine which combinations of measures and values produce the best results. Further distinguishing measures could be introduced such as added views of the object or including images with different exposure to determine if features in the target are robust to changes in lighting and weight them accordingly. Temporal weightings are also a possibility with the feature weights generated from previous frames in the sequence rather than stereo cameras.

This system is not currently a practical system for detecting features but this chapter is an experiment to determine if the use of a weighting system based on the \textit{a contrario} methodology could be successful. This system may not be ideal but it shows that in principal the technique could be used to successfully weight the feature matches.

This chapter has introduced the concept of FEWER and has shown that it can successfully weight features to indicate if a match is likely to be correct. The focus is in the use of an \textit{a contrario} methodology and it has proven that this technique can be used to increase understanding and discriminate between feature matches.
Chapter 8

Conclusion and Future Work

This chapter summarises the work in this thesis, re-emphasises the contributions and highlights possible areas of future work.

8.1 Summary

In this work, current feature detection methods and descriptors were reviewed and certain limitations were examined. The area of focus is based on the framework defined in Chapter 1 which defined the scope of the work. The problems focused on were outlined in the introduction to this thesis and are as follows:

- The Scale Invariant Feature Transform has many parameters that effect the results of detection and matching. It is not well understood how varying these parameters effect the matching of different image types and how best to select the parameters.

- High dynamic range environments pose a problem when matching features as an 8-bit low dynamic range camera often cannot capture all the information in a scene.

- False positive matches often occur when matching images due to feature vectors being generated from different areas of images which correspond well enough to be matched mistakenly. Identifying which features matches are correct when matching images is important.
8.2 Contributions

This thesis has made the following main contributions:

8.2.1 Preliminary Analysis of the SIFT.

This work includes an analysis of the SIFT feature descriptor space using millions of features downloaded from Flickr. This introduces an investigation of the noise properties of SIFT matching and introduces a novel use of the *a contrario* methodology for detecting images which are likely to match based on the rate of false positive matches which occur for non-corresponding images. The correspondence ratio has been introduced in this thesis as a metric for the assessment of how well images match and its use is supported through experiment.

8.2.2 SIFTing through Parameter Space.

Work has focused on optimising the SIFT by adjusting the parameters for specific image types. The parameter space for the SIFT is substantial; there are over twenty variables. This makes selecting the best parameters for matching between a pair of images difficult. Image pairs have been collected from multiple sources including infrared cameras. Parameter sweeps and genetic algorithm based optimisation have been used to explore the parameter space to better understand how different image types effect the algorithm and that the default parameters often require adjustment. This work has outlined recommendations for which parameters to focus on when tuning and demonstrated that these methods of analysis, using sweeps or genetic algorithms, can improve the performance of the SIFT when compared with the default parameters suggested by Lowe and others.

8.2.3 Multi-exposure, High Dynamic Range, Fusion Features.

Work within this thesis has introduced and evaluated an improved process where fusion features assist matching SIFT image features from high contrast images. FAW defines the preferred order for extracting features: First Feature extraction, then Alignment of images and finally pixel Weighting. The process uses up to four quality measures to select features from a series of differently exposed images and weights the features in favour of those areas that are defined as well exposed.
The results show an advantage in using these features over features extracted from the common alternative techniques of exposure fusion and tone mapping.

8.2.4 A Contrario SIFT Feature Weighting

An object detection and feature match weighting utilising stereoscopic image pairs, the scale invariant feature transform and 3D reconstruction has been developed. The object detection technique is based on a contrario noise subtraction utilising false positive matches from random features. The feature weighting process utilises 3D spatial information generated from the stereoscopic pairs to divide these into three different types. The weightings are automatically computed by analysing a large number of false positive matches. The techniques described provide increased precision, reduces the occurrence of false positives and can create a subset of highly relevant features. The performance show the feasibility of the a contrario methodology for such a weighting process.

8.3 Future Work

The work in this thesis has begun to address some of the challenges within the scope of the framework. Extending the work to other feature detection and matching algorithms is an area of future work which can be applied to all the topics that have been covered in this thesis. For each of the main areas future work has been identified as follows:

8.3.1 SIFTing through Parameter Space.

This section has provided an investigation of the SIFT parameter space but further areas can be investigated. The parameters that were studied were a subset of the total parameters available and the descriptor parameters have not been analysed. Future work can look at these parameters and apply the same techniques here to consider their effect on the SIFT algorithm under various conditions. More specific analysis of the parameters that have been analysed here is another area of further work with more focused sweeps within the ranges of values.

The sweep and the GA are time consuming and future work should focus on ways of reducing this time and developing practical ways of adjusting the parameters quickly. Automating the process should be another focus such that
the parameterization can be estimated for images where the correspondence is unknown possibly through the inclusion of the use of RANSAC to estimate if the matches are correct. Varying the parameters of the sweep and the GA, such as the step sizes, the ranges and the random number generator used for the GA could also be investigated to determine if better results can be achieved.

8.3.2 Multi-exposure, High Dynamic Range, Fusion Features.

The first area of future work for this chapter is that the number of images used and the number of exposure images used for each could be expanded. Comparisons to other HDR image techniques could be looked at and direct comparison to features extracted directly from 16 bit HDR images rather than the tone mapped images generated from them.

Examination of the compromise between good stereo reconstruction and HDR features generated by exposing the stereo cameras differently is an area of interest. Future work can look at the use and stereo cameras utilising neutral density filters on the cameras to alter the exposure and allow frames from both cameras to be exposed at the same rate yet still capture different areas of the dynamic range of a scene.

8.3.3 A Contrario SIFT Feature Weighting

Further work in this area will look at the parameters and the measures used within the weighting process. The system contains a lot of parameters to vary and analyse to determine which combinations of measures and values produce the best results. Automatically determining a threshold of the weightings is a possible area of research.

Further distinguishing measures could be introduced such as added views of the object or including images with different exposure to determine if features in the target are robust to changes in lighting and weight them accordingly. Temporal weightings are also a possibility with the feature weights generated from previous frames in the sequence rather than stereo cameras. Another area of focus could be to utilise the depth information to distinguish between flat and 3D objects to determine if the match is an image or the actual object and weight accordingly.
Furthermore, other possible uses of the *a contrario* methodologies applicability to features should be investigated as the method provides a robust way of determining feature matching thresholds. With the increasing computational resources available, especially those provided by GPGPUs, the computational expense of matching known false positives is becoming less prohibitive.
Appendix A

A.1 Oxford Test Images

Figure A.1: The Oxford bark images; variation in zoom.

Figure A.2: The Oxford bikes images; variation in blur.

Figure A.3: The Oxford boat images; variation in zoom and rotation.
Figure A.4: The Oxford bricks images; variation in viewpoint

Figure A.5: The Oxford cars images; variation in lighting

Figure A.6: The Oxford graffiti images; variation in viewpoint.

Figure A.7: The Oxford trees images; variation in blur.

Figure A.8: The Oxford ubc images; variation in jpeg compression.
A.2 Sweep/GA Graphs

A.2.1 Oxford Sweep

Best 500

Figure A.9: Oxford Sweep Best 500 - Sigma.

Figure A.10: Oxford Sweep Best 500 - Contrast.

Figure A.11: Oxford Sweep Best 500 - Curvature.
Figure A.12: Oxford Sweep Best 500 - Intervals.

Figure A.13: Oxford Sweep Best 500 - Octaves.

Figure A.14: Oxford Sweep Best 500 - Match Ratio.

Pareto Front

Figure A.15: Oxford Sweep Pareto Front - Sigma.
A.2. SWEEP/GA GRAPHS

Figure A.16: Oxford Sweep Pareto Front - Contrast.

Figure A.17: Oxford Sweep Pareto Front - Curvature.

Figure A.18: Oxford Sweep Pareto Front - Intervals.

Figure A.19: Oxford Sweep Pareto Front - Octaves.
### A.3 Lowe vs Sweep Histogram Values

|        | Bark | Sweep 500 | Lowe | Sweep 500 | Bike | Sweep 500 | Lowe | Sweep 500 | Bricks | Sweep 500 | Lowe | Sweep 500 | Graffiti | Sweep 500 | Lowe | Sweep 500 | Trees | Sweep 500 | Lowe | Sweep 500 | Lowe | Sweep 500 |
|--------|------|-----------|------|-----------|------|-----------|------|-----------|--------|-----------|------|-----------|----------|-----------|------|-----------|-------|-----------|------|-----------|
|        | Correct Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision |
| 1to2   | 6    | 0.15      | 7    | 1         | 29   | 0.22      | 10   | 1         | 20    | 0.12      | 29   | 1         |
| 1to3   | 1063 | 0         | 3    | 0.43      | 32   | 0.29      | 7    | 1         |
| 1to4   | 1064 | 1         | 12   | 0.86      | 18   | 0.15      | 7    | 1         |
| 1to5   | 1065 | 2         | 23   | 0.85      | 8    | 0.09      | 18   | 0.95      |
| 1to6   | 1066 | 2         | 2    | 0.29      | 13   | 0.14      | 12   | 1         |
|        | Boat | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision |
| 1to2   | 38   | 0.11      | 37   | 0.97      | 20   | 0.12      | 29   | 1         |
| 1to3   | 30   | 0.1       | 20   | 1         | 14   | 0.07      | 1    | 1         |
| 1to4   | 26   | 0.08      | 7    | 1         | 6    | 0.03      | 4    | 0.07      |
| 1to5   | 17   | 0.05      | 5    | 1         | 1    | 0.01      | 0    | 0         |
| 1to6   | 11   | 0.03      | 9    | 0.9       | 0    | 0         | 1    | 0.17      |
|        | Cars | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision |
| 1to2   | 13   | 0.28      | 22   | 0.96      | 17   | 0.06      | 1    | 1         |
| 1to3   | 6    | 0.15      | 9    | 0.82      | 9    | 0.04      | 10   | 0.1       |
| 1to4   | 3    | 0.07      | 29   | 0.53      | 0    | 0         | 1    | 0.05      |
| 1to5   | 4    | 0.07      | 0    | 0         | 1    | 0.01      | 0    | 0         |
| 1to6   | 3    | 0.05      | 0    | 0         | 0    | 0         | 1    | 0.13      |
|        | Trees | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision | Correct | Precision |
| 1to2   | 12   | 0.03      | 9    | 0.9       | 28   | 0.13      | 7    | 1         |
| 1to3   | 13   | 0.03      | 8    | 1         | 59   | 0.26      | 18   | 1         |
| 1to4   | 5    | 0.01      | 27   | 0.77      | 66   | 0.28      | 15   | 1         |
| 1to6   | 9    | 0.02      | 17   | 0.61      | 90   | 0.38      | 47   | 1         |

Table A.1: The precision and number of correct matches for feature matching when using the Lowe’s default SIFT parameters and the parameters obtained in the sweep from for the Oxford set using the ‘best 500’ values. The values in **bold** are highest.

Figure A.20: Oxford Sweep Pareto Front - Match Ratio.
A.4 MBDA Test Images

(a) 1 HAWK  (b) 2-3 SIB  (c) 4 WAT232  (d) 5 WAT902H

Figure A.21: The MBDA set 1 images.

(a) 1 HAWK  (b) 2-3 SIB  (c) 4 WAT232  (d) 5 WAT902H

Figure A.22: The MBDA set 2 images.

(a) 1 HAWK  (b) 2-3 SIB  (c) 4 WAT232  (d) 5 WAT902H

Figure A.23: The MBDA set 3 images.

(a) 1 HAWK  (b) 2-3 SIB  (c) 4 WAT232  (d) 5 WAT902H

Figure A.24: The MBDA set 4 images.
Figure A.25: The MBDA set 5 images.

Figure A.26: The MBDA set 6 images.

Figure A.27: The MBDA set 7 images.

Figure A.28: The MBDA set 8 images.
A.4. MBDA TEST IMAGES

A.4.1 MBDA GA

Pareto Front

Figure A.29: MBDA GA Pareto Front - Sigma.

Figure A.30: MBDA GA Pareto Front - Contrast.

Figure A.31: MBDA GA Pareto Front - Curvature.

Figure A.32: MBDA GA Pareto Front - Intervals.
Figure A.33: MBDA GA Pareto Front - Octaves.

Figure A.34: MBDA GA Pareto Front - Match Ratio.

A.4.2 MBDA Sweep

Best 500

Figure A.35: MBDA Sweep Best 500 - Sigma.

Figure A.36: MBDA Sweep Best 500 - Contrast.
A.4. MBDA TEST IMAGES

Figure A.37: MBDA Sweep Best 500 - Curvature.

Figure A.38: MBDA Sweep Best 500 - Intervals.

Figure A.39: MBDA Sweep Best 500 - Octaves.

Figure A.40: MBDA Sweep Best 500 - Match Ratio.
Pareto Front

Figure A.41: MBDA Sweep Pareto Front - Sigma.

Figure A.42: MBDA Sweep Pareto Front - Contrast.

Figure A.43: MBDA Sweep Pareto Front - Curvature.

Figure A.44: MBDA Sweep Pareto Front - Intervals.
A.5. **LOWE VS SWEEP HISTOGRAM VALUES**

Figure A.45: MBDA Sweep Pareto Front - Octaves.

Figure A.46: MBDA Sweep Pareto Front - Match Ratio.

A.5  **Lowe vs Sweep Histogram Values**
<table>
<thead>
<tr>
<th>Set1</th>
<th>Lowe</th>
<th>Correct Precision</th>
<th>Sweep 500</th>
<th>Correct Precision</th>
<th>Set2</th>
<th>Lowe</th>
<th>Correct Precision</th>
<th>Sweep 500</th>
<th>Correct Precision</th>
</tr>
</thead>
<tbody>
<tr>
<td>HAWK-SIB</td>
<td>2</td>
<td>0.28</td>
<td>4</td>
<td>0.19</td>
<td>0</td>
<td>0</td>
<td>25</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>HAWK-SIBHDR</td>
<td>4</td>
<td>0.44</td>
<td>9</td>
<td>0.52</td>
<td>3</td>
<td>0.18</td>
<td>34</td>
<td>0.19</td>
<td></td>
</tr>
<tr>
<td>HAWK-WAT232</td>
<td>7</td>
<td>0.35</td>
<td>22</td>
<td>0.37</td>
<td>4</td>
<td>0.26</td>
<td>17</td>
<td>0.3</td>
<td></td>
</tr>
<tr>
<td>HAWK-WAT902H</td>
<td>21</td>
<td>0.72</td>
<td>14</td>
<td>0.82</td>
<td>7</td>
<td>0.43</td>
<td>10</td>
<td>0.52</td>
<td></td>
</tr>
<tr>
<td>SIB-SIBHDR</td>
<td>10</td>
<td>1</td>
<td>28</td>
<td>1</td>
<td>1</td>
<td>0.2</td>
<td>14</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SIB-WAT232</td>
<td>1</td>
<td>0.2</td>
<td>5</td>
<td>0.2</td>
<td>0</td>
<td>0</td>
<td>6</td>
<td>0.1</td>
<td></td>
</tr>
<tr>
<td>SIB-WAT902H</td>
<td>1</td>
<td>0.33</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>26</td>
<td>0.13</td>
<td></td>
</tr>
<tr>
<td>SIBHDR-WAT232</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0.11</td>
<td></td>
</tr>
<tr>
<td>SIBHDR-WAT902H</td>
<td>1</td>
<td>0.16</td>
<td>6</td>
<td>0.42</td>
<td>0</td>
<td>0</td>
<td>21</td>
<td>0.36</td>
<td></td>
</tr>
<tr>
<td>WAT232-WAT902H</td>
<td>6</td>
<td>0.5</td>
<td>18</td>
<td>0.5</td>
<td>19</td>
<td>0.7</td>
<td>2</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>HAWK-SIB</td>
<td>0</td>
<td>0</td>
<td>4</td>
<td>0.08</td>
<td>5</td>
<td>0.5</td>
<td>2</td>
<td>0.14</td>
<td></td>
</tr>
<tr>
<td>HAWK-SIBHDR</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0.1</td>
<td>0</td>
<td>0</td>
<td>27</td>
<td>0.17</td>
<td></td>
</tr>
<tr>
<td>HAWK-WAT232</td>
<td>2</td>
<td>0.5</td>
<td>5</td>
<td>0.55</td>
<td>3</td>
<td>0.3</td>
<td>18</td>
<td>0.52</td>
<td></td>
</tr>
<tr>
<td>HAWK-WAT902H</td>
<td>5</td>
<td>0.62</td>
<td>17</td>
<td>0.77</td>
<td>4</td>
<td>0.66</td>
<td>10</td>
<td>0.9</td>
<td></td>
</tr>
<tr>
<td>SIB-SIBHDR</td>
<td>10</td>
<td>0.76</td>
<td>198</td>
<td>1</td>
<td>6</td>
<td>0.6</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SIB-WAT232</td>
<td>2</td>
<td>0.28</td>
<td>7</td>
<td>0.12</td>
<td>3</td>
<td>0.75</td>
<td>8</td>
<td>0.38</td>
<td></td>
</tr>
<tr>
<td>SIB-WAT902H</td>
<td>0</td>
<td>0</td>
<td>5</td>
<td>0.06</td>
<td>0</td>
<td>0</td>
<td>7</td>
<td>0.38</td>
<td></td>
</tr>
<tr>
<td>SIBHDR-WAT232</td>
<td>1</td>
<td>0.2</td>
<td>5</td>
<td>0.08</td>
<td>2</td>
<td>0.33</td>
<td>7</td>
<td>0.43</td>
<td></td>
</tr>
<tr>
<td>SIBHDR-WAT902H</td>
<td>1</td>
<td>0.25</td>
<td>9</td>
<td>0.12</td>
<td>0</td>
<td>0</td>
<td>10</td>
<td>0.41</td>
<td></td>
</tr>
<tr>
<td>WAT232-WAT902H</td>
<td>14</td>
<td>0.6</td>
<td>9</td>
<td>0.9</td>
<td>5</td>
<td>0.41</td>
<td>3</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>HAWK-SIB</td>
<td>5</td>
<td>0.35</td>
<td>6</td>
<td>0.17</td>
<td>7</td>
<td>0.46</td>
<td>38</td>
<td>0.24</td>
<td></td>
</tr>
<tr>
<td>HAWK-SIBHDR</td>
<td>2</td>
<td>0.28</td>
<td>32</td>
<td>0.2</td>
<td>3</td>
<td>0.37</td>
<td>46</td>
<td>0.28</td>
<td></td>
</tr>
<tr>
<td>HAWK-WAT232</td>
<td>6</td>
<td>0.6</td>
<td>8</td>
<td>0.88</td>
<td>5</td>
<td>0.41</td>
<td>23</td>
<td>0.69</td>
<td></td>
</tr>
<tr>
<td>HAWK-WAT902H</td>
<td>14</td>
<td>0.77</td>
<td>16</td>
<td>0.69</td>
<td>10</td>
<td>0.62</td>
<td>15</td>
<td>0.88</td>
<td></td>
</tr>
<tr>
<td>SIB-SIBHDR</td>
<td>2</td>
<td>1</td>
<td>137</td>
<td>1</td>
<td>3</td>
<td>0.6</td>
<td>7</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SIB-WAT232</td>
<td>2</td>
<td>0.4</td>
<td>33</td>
<td>0.2</td>
<td>0</td>
<td>0</td>
<td>20</td>
<td>0.25</td>
<td></td>
</tr>
<tr>
<td>SIB-WAT902H</td>
<td>1</td>
<td>0.5</td>
<td>9</td>
<td>0.21</td>
<td>5</td>
<td>0.62</td>
<td>0</td>
<td>0</td>
<td></td>
</tr>
<tr>
<td>SIBHDR-WAT232</td>
<td>5</td>
<td>1</td>
<td>19</td>
<td>0.15</td>
<td>3</td>
<td>0.18</td>
<td>4</td>
<td>0.4</td>
<td></td>
</tr>
<tr>
<td>SIBHDR-WAT902H</td>
<td>2</td>
<td>0.4</td>
<td>10</td>
<td>0.15</td>
<td>7</td>
<td>0.53</td>
<td>2</td>
<td>0.5</td>
<td></td>
</tr>
<tr>
<td>WAT232-WAT902H</td>
<td>16</td>
<td>0.76</td>
<td>14</td>
<td>0.82</td>
<td>19</td>
<td>0.63</td>
<td>13</td>
<td>0.92</td>
<td></td>
</tr>
<tr>
<td>HAWK-SIB</td>
<td>2</td>
<td>0.08</td>
<td>8</td>
<td>0.09</td>
<td>4</td>
<td>0.15</td>
<td>12</td>
<td>0.75</td>
<td></td>
</tr>
<tr>
<td>HAWK-SIBHDR</td>
<td>1</td>
<td>0.06</td>
<td>27</td>
<td>0.17</td>
<td>3</td>
<td>0.18</td>
<td>19</td>
<td>0.63</td>
<td></td>
</tr>
<tr>
<td>HAWK-WAT232</td>
<td>8</td>
<td>0.47</td>
<td>5</td>
<td>0.25</td>
<td>12</td>
<td>0.6</td>
<td>11</td>
<td>0.55</td>
<td></td>
</tr>
<tr>
<td>HAWK-WAT902H</td>
<td>12</td>
<td>0.57</td>
<td>6</td>
<td>0.54</td>
<td>20</td>
<td>0.76</td>
<td>11</td>
<td>0.91</td>
<td></td>
</tr>
<tr>
<td>SIB-SIBHDR</td>
<td>1</td>
<td>0.1</td>
<td>6</td>
<td>1</td>
<td>16</td>
<td>0.94</td>
<td>108</td>
<td>1</td>
<td></td>
</tr>
<tr>
<td>SIB-WAT232</td>
<td>1</td>
<td>0.2</td>
<td>24</td>
<td>0.13</td>
<td>4</td>
<td>0.4</td>
<td>6</td>
<td>0.18</td>
<td></td>
</tr>
<tr>
<td>SIB-WAT902H</td>
<td>0</td>
<td>0</td>
<td>38</td>
<td>0.15</td>
<td>4</td>
<td>0.4</td>
<td>14</td>
<td>0.38</td>
<td></td>
</tr>
<tr>
<td>SIBHDR-WAT232</td>
<td>1</td>
<td>0.2</td>
<td>5</td>
<td>0.26</td>
<td>4</td>
<td>0.36</td>
<td>4</td>
<td>0.57</td>
<td></td>
</tr>
<tr>
<td>SIBHDR-WAT902H</td>
<td>1</td>
<td>0.5</td>
<td>43</td>
<td>0.16</td>
<td>4</td>
<td>0.33</td>
<td>29</td>
<td>0.35</td>
<td></td>
</tr>
<tr>
<td>WAT232-WAT902H</td>
<td>16</td>
<td>0.76</td>
<td>14</td>
<td>0.63</td>
<td>9</td>
<td>0.47</td>
<td>11</td>
<td>1</td>
<td></td>
</tr>
</tbody>
</table>

Table A.2: The precision and number of correct matches for feature matching when using the Lowe’s default SIFT parameters and the parameters obtained in the sweep from for the MBDA set using the ‘best 500’ values. The values in bold are highest.
Appendix B

B.1 Fusion Features Test Data

Figure B.1: The images used in test case 1.

Figure B.2: The images used in test case 2.

Figure B.3: The images used in test case 3.
Figure B.4: The images used in test case 4.

Figure B.5: The images used in test case 5.

Figure B.6: The images used in test case 6.

Figure B.7: The images used in test case 7.

Figure B.8: The images used in test case 8.
B.1. FUSION FEATURES TEST DATA

Figure B.9: The images used in test case 9.

Figure B.10: The images used in test case 10.

Figure B.11: The images used in test case 11.

Figure B.12: The images used in test case 12.

Figure B.13: The images used in test case 13.
APPENDIX B.

Figure B.14: The images used in test case 14.

Figure B.15: The images used in test case 15.

Figure B.16: The images used in test case 16.

Figure B.17: The images used in test case 17.

Figure B.18: The images used in test case 18.
### B.1. FUSION FEATURES TEST DATA

![Test Case 19 Images](image1)

**Figure B.19:** The images used in test case 19.

![Test Case 20 Images](image2)

**Figure B.20:** The images used in test case 20.

![Test Case 21 Images](image3)

**Figure B.21:** The images used in test case 21.

![Test Case 22 Images](image4)

**Figure B.22:** The images used in test case 22.

![Test Case 23 Images](image5)

**Figure B.23:** The images used in test case 23.
Figure B.24: The images used in test case 24.

Figure B.25: The images used in test case 25.

Figure B.26: The images used in test case 26.

Figure B.27: The images used in test case 27.

Figure B.28: The images used in test case 28.
Appendix C

C.1 Top Level Pseudo Code for the 3D SIFT Algorithm

Algorithm C.1 3D SIFT

1: procedure 3DSIFT(LeftTarget, RightTarget, LeftScene, RightScene, neighbourDistance, numberOfClusters, clusterConfidenceThreshold, weightingThreshold)
2:   for all input images (4) do
3:     features ⇐ extractSIFTFeatures(image)
4:   end for
5:   for all stereo pairs (2) do
6:     3DFeatures ⇐ calculate3DFeaturePositions(leftFeatures, rightFeatures)
7:   end for
8:   for all stereo pairs (2) do
9:     clusterIndex3D ⇐ clusterIn3DSpace(3DFeatures, numberOfClusters)
10: end for
11: for all input images (4) do
12:   type3Features ⇐ map3DFeaturesToImages(features, 3DFeatures)
13: end for
14: for all input images (4) do
15:   type2Features, clusterIndex ⇐ mapClustersTo2DNeighbours(features, type3Features, clusterIndex3D, neighbourDistance)
16: end for
for all target images (2) do
    \text{matchIndex} \leftarrow \text{matchFeatures}(\text{targetFeatures}, \text{sceneFeatures})
end for
for all matched target features indexes (4) do
    \text{featureWeighting} \leftarrow \text{initialFeatureWeighting}(\text{type2FeaturesTarget}, \text{type3FeaturesTarget}, \text{type2FeaturesScene}, \text{type3FeaturesScene}, \text{matchIndex})
end for
for all target images (2) do
    \text{featureWeightingsLeft, featureWeightingsRight} \leftarrow \text{sceneSameFeatureTarget}(\text{featureWeightingsLeft}, \text{featureWeightingsRight}, \text{matchIndexLeft}, \text{matchIndexRight}, 3\text{DFeaturesScene}, \text{clusterIndex3DTarget})
end for
for all scene images (2) do
    \text{featureWeightingsLeft, featureWeightingsRight} \leftarrow \text{targetSameFeatureScene}(\text{featureWeightingsLeft}, \text{featureWeightingsRight}, \text{matchIndexLeft}, \text{matchIndexRight}, 3\text{DFeaturesTarget}, \text{clusterIndex3DScene})
end for
for all matched target features indices (4) do
    \text{featureWeighting} \leftarrow \text{clusterWeighting}(\text{clusterIndexTarget}, \text{clusterIndexScene}, \text{featureWeighting}, \text{matchIndex}, \text{clusterConfidenceThreshold}, \text{numberOfClusters})
end for
for all matched target features indices (4) do
    \text{featureWeighting} \leftarrow \text{normaliseWeightings}(\text{featureWeighting}, \text{maximumPossibleWeighting})
end for
for all matched target features indices (4) do
    \text{featureWeighting} \leftarrow \text{normaliseWeightings}(\text{featureWeighting}, \text{maximumPossibleWeighting})
end for
for all matched target features indices (4) do
    \text{matchedFeatures} \leftarrow \text{thresholdWeighting}(\text{featureWeighting}, \text{weightingThreshold})
end for
return \text{matchedFeatures}
end procedure
C.2 Matched Test Images for FEWER

Figure C.1: The selected images from test set A used for testing FEWER. Each displays all the matches colour coded from blue to red based on the weighting.
Figure C.2: The selected images from test set B used for testing FEWER. Each displays all the matches colour coded from blue to red based on the weighting.
Figure C.3: The selected images from test set C used for testing FEWER. Each displays all the matches colour coded from blue to red based on the weighting.
Bibliography


