



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

Research Commons

<http://waikato.researchgateway.ac.nz/>

Research Commons at the University of Waikato

Copyright Statement:

The digital copy of this thesis is protected by the Copyright Act 1994 (New Zealand).

The thesis may be consulted by you, provided you comply with the provisions of the Act and the following conditions of use:

- Any use you make of these documents or images must be for research or private study purposes only, and you may not make them available to any other person.
- Authors control the copyright of their thesis. You will recognise the author's right to be identified as the author of the thesis, and due acknowledgement will be made to the author where appropriate.
- You will obtain the author's permission before publishing any material from the thesis.

ADAPTIVE VISION BASED SCENE REGISTRATION FOR OUTDOOR AUGMENTED REALITY



THE UNIVERSITY OF
WAIKATO
Te Whare Wānanga o Waikato

A thesis
submitted in partial fulfillment
of the requirements for the degree
of
Doctor of Philosophy in Computer Science
at
The University of Waikato
by
JASON CATCHPOLE

The University of Waikato

2007

Abstract

Augmented Reality (AR) involves adding virtual content into real scenes. Scenes are viewed using a Head-Mounted Display or other display type. In order to place content into the user's view of a scene, the user's position and orientation relative to the scene, commonly referred to as their pose, must be determined accurately. This allows the objects to be placed in the correct positions and to remain there when the user moves or the scene changes. It is achieved by tracking the user in relation to their environment using a variety of technology. One technology which has proven to provide accurate results is computer vision. Computer vision involves a computer analysing images and achieving an understanding of them. This may be locating objects such as faces in the images, or in the case of AR, determining the pose of the user.

One of the ultimate goals of AR systems is to be capable of operating under any condition. For example, a computer vision system must be robust under a range of different scene types, and under unpredictable environmental conditions due to variable illumination and weather. The majority of existing literature tests algorithms under the assumption of ideal or 'normal' imaging conditions. To ensure robustness under as many circumstances as possible it is also important to evaluate the systems under adverse conditions.

This thesis seeks to analyse the effects that variable illumination has on computer vision algorithms. To enable this analysis, test data is required to isolate weather and illumination effects, without other factors such as changes in viewpoint that would bias the results. A new dataset is presented which also allows controlled viewpoint differences in the presence of weather and illumination changes. This is achieved by capturing video from a camera undergoing a repeatable motion sequence. Ground truth data is stored *per* frame allowing images from the same position under differing environmental conditions, to be easily extracted from the videos.

An in depth analysis of six detection algorithms and five matching techniques demonstrates the impact that non-uniform illumination changes can have on vision algorithms. Specifically, shadows can degrade performance and reduce confidence in the system, decrease reliability, or even completely prevent successful operation.

An investigation into approaches to improve performance yields techniques that can help reduce the impact of shadows. A novel algorithm is presented that merges reference data captured at different times, resulting in reference data with minimal shadow effects. This can significantly improve performance and reliability when operating on images containing shadow effects. These advances improve the robustness of computer vision systems and extend the range of conditions in which they can operate. This can increase the usefulness of the algorithms and the AR systems that employ them.

Acknowledgements

Firstly i would like to thank my three supervisors, Mark Apperley, David Bainbridge, and Bill Rogers. Without their help and inspiration I could never have made it through this project. They also provided countless hours of proof reading and interesting discussions.

I would also like to thank my loyal proof readers, Berdine Catchpole, Kushla Haenen, Sid Guinness, Kathryn Hempstalk, and Matthew Lucky. Without these good people this thesis would not be what it is. It takes a long time to go through a PhD thesis, and yet they didn't complain.

Many thanks to Mark Billingham and Michael Haller who were the examiners for this thesis. They provided a lot of useful feedback and comments and helped to make this thesis all that it is.

During my PhD i was also lucky enough to go on an exchange to several places in Europe. I would like to thank Masood Masoodian for helping get this organised from the Waikato University end. Furthermore, thanks to Anna-Lee Mason, Jeremiah Diephuis, and Christina Huber-Beran for all their help organising visas, accommodation and many other things. Thanks to Jeremiah Diephuis and Michael Haller for providing a very enjoyable stay in Austria at the Upper Austria University of Applied Science.

Thanks to Matjaž Jogan and Aleš Leonardis for a great time in Solvenia at the University of Ljubljana, Computer Vision lab. I would also like to thank them for their help and input on computer vision matters, as well as supplying the components required to create the SloDB capture system.

Thanks to Steve Benford for putting me up at the Mixed Reality Lab in Nottingham. Many thanks to Matt Jones for allowing me to present my work at Swansea University in Wales, and for providing accommodation during my stay there.

Thanks to all the enthusiastic people i have had discussions with over the years

on computer vision and AR topics, its been a great ride. Thanks to all the other researchers who have helped as well, particularly those who have answered questions and agreed to my use of their images throughout this thesis.

This whole project would not have been possible without the financial support of the Computer Science department and the University of Waikato.

Lastly I would like to thank my parents for helping me financially during my undergraduate years. I would also like to thank my wife, Berdine, for putting up with me spending so much time on this thesis.

Thanks be to God.

Contents

List of Figures	vii
List of Abbreviations	xiii
1 Introduction	1
1.1 Overview	2
1.2 Augmented Reality Uses	3
1.3 Problems in AR	9
1.3.1 Ergonomic	9
1.3.2 Realism	13
1.3.3 Registration Problem	15
1.4 This Thesis	17
1.4.1 Thesis Contributions	17
1.4.2 Thesis Structure	18
2 Background	21
2.1 Causes of Registration Error	21
2.2 Tracking Technologies	23
2.2.1 Hybrids	25
2.3 Computer Vision	28
2.3.1 Fiducial Markers	28
2.3.2 Natural Feature Tracking	33
2.4 Issues with Computer Vision in the Outdoors	38
2.4.1 Why Natural Feature Tracking can be Difficult	38
2.4.2 Improving Performance of Natural Feature Tracking	39
2.5 Assessing the Robustness of Computer Vision	43
2.6 Summary	43

3	AR Implementation	47
3.1	2D System	48
3.1.1	Fourier-Based Technique	49
3.1.2	Using the Algorithm for AR	55
3.1.3	Image Library	58
3.1.4	Problems	60
3.1.5	Summary	61
3.2	3D System	62
3.2.1	Overview	62
3.2.2	Feature Extraction	63
3.2.3	Feature Matching	68
3.2.4	Two-View Geometry	72
3.2.5	Three-View Geometry	76
3.2.6	Multi-view Correspondences	81
3.2.7	Camera Pose Recovery	81
3.2.8	3D Reconstruction	84
3.2.9	Optimisation	91
3.2.10	Virtual Object Insertion	92
3.2.11	Live Pose Estimation	94
3.2.12	Problems	95
3.2.13	Summary	97
3.3	System Comparison	98
4	Test Data	101
4.1	Related Databases	101
4.2	WILD	104
4.2.1	Capture Setup	105
4.2.2	Ground Truth Data	107
4.2.3	Limitations	107
4.2.4	WILD Selection	108
4.2.5	Summary	112
4.3	SloDB	114
4.3.1	The Capture System	115
4.3.2	Limitations	118
4.3.3	SloDB Selection	118

4.3.4	Lessons Learned in Setting up SloDB	118
4.4	Arizona Test Set	124
4.5	Summary	124
5	Analysis	129
5.1	Introduction	129
5.2	Background	131
5.3	Algorithms Evaluated	133
5.3.1	SIFT	135
5.3.2	MSER	136
5.3.3	HARAFF	137
5.3.4	HESAFF	138
5.3.5	EBR	138
5.3.6	IBR	139
5.3.7	Algorithm Summary	140
5.4	Testing Methodology	140
5.4.1	Repeatability	141
5.4.2	Recall and 1-Precision	143
5.4.3	Practical Tests	145
5.4.4	Test Framework Summary	146
5.5	Test Results	147
5.5.1	The Shadow Effect	155
5.5.2	Cloudy Selection	157
5.5.3	Seasonal Changes	162
5.5.4	Fourier Analysis	165
5.5.5	Light Dataset	168
5.5.6	SloDB Selection	168
5.6	Descriptor Analysis	173
5.6.1	Choice of Distance Ratio	176
5.7	Summary	178
6	Dealing with Shadows	181
6.1	Improving Detection in Live Images	182
6.1.1	Relaxed Thresholds	184
6.1.2	Image Enhancement	193

6.2	The PILLAGE Algorithm	203
6.2.1	How PILLAGE Works	205
6.2.2	Performance	210
6.2.3	Clear Weather WILD Image Performance	210
6.2.4	Cloudy WILD Image Performance	214
6.2.5	Arizona Test Set Performance	217
6.2.6	SloDB Test Set Performance	223
6.3	Summary	224
7	Conclusions	229
7.1	Design Guidelines	233
7.1.1	Developers using Existing Feature Extraction and Matching Algorithms	233
7.1.2	Researchers Creating New Feature Extraction and Matching Algorithms	234
7.2	Future Improvements	235
7.3	Closing Remarks	237
A	WILD Transformation Choice	239
A.1	Performance of 3std <i>vs</i> 2std	240
A.2	Performance of 3std <i>vs</i> 4std	244
B	Matching Score Results	249
B.1	Methodology	249
B.2	Test Results	250
B.2.1	Clear Weather WILD Selection	250
B.2.2	Cloudy Selection	251
B.2.3	Seasonal Changes	252
B.2.4	SloDB Selection	252
B.2.5	Descriptor Performance	253
B.2.6	Relaxed Thresholds	254
B.2.7	Image Enhancement	255
B.2.8	PILLAGE	257
	References	262

List of Figures

1.1	Example AR scene	2
1.2	Milgram's Virtuality Continuum	3
1.3	Example AV scene	3
1.4	F-35 Joint Strike Fighter HMDS	4
1.5	ARQuake Images	5
1.6	AR Guitar	6
1.7	AR View of a Virtual Brain Overlaid on a Real Head	7
1.8	Tinmith Hybrid AR System	10
1.9	PDA Based AR System	13
1.10	AR on Deforming Surfaces	14
2.1	Conceptual Diagram of Inside-out and Outside-in Optical Tracker	27
2.2	Magic Book	30
2.3	ARTag Markers Handling Occlusion vs ARToolKit	31
3.1	ARCHEOGUIDE Augmentations	48
3.2	Phase Correlation for Determining Translation	51
3.3	Fourier-based Approach to Retrieving Rotation	53
3.4	Fourier-based Approach to Retrieving Scale	56
3.5	2D System Limitations Conceptual Images	58
3.6	Arenberg Castle Image Selection used For Reconstruction	64
3.7	SIFT Pyramid Conceptual Diagram	66
3.8	SIFT Extrema Detection	67
3.9	SIFT Descriptor Conceptual Diagram	68
3.10	SIFT Features and Descriptors for Two Images	69
3.11	Putative Two-view Matches	70
3.12	Epipolar Geometry Conceptual Diagram	73
3.13	Epipolar Geometry Baseline Conceptual Diagram	74

3.14	Final Two-view Matches Consistent with Computed Epipolar Geometry	75
3.15	Guided Two-view Matching Process	77
3.16	Putative Three-view Matches	78
3.17	Point-point-point Relation between Three Views	79
3.18	Final Three-view Matches Consistent with Computed Trifocal Tensor	80
3.19	Point Transfer under Three Views	81
3.20	One Multi-view Match Seen Over Nine Images	82
3.21	Triangulation Conceptual Diagram	84
3.22	Conceptual Diagram for Correcting 2D Points	85
3.23	Reconstruction Ambiguity Conceptual Diagram	86
3.24	Different Types of Reconstructions	87
3.25	3D Calibration Object	88
3.26	Images of Calibration Grid used to Calibrate Cameras	89
3.27	Views of Final 3D Reconstruction	93
3.28	Virtual Object Placement	94
3.29	Assigning Virtual Object Orientation	94
3.30	Augmented Images Produced by 3D System	95
3.31	Influence of Image Separation on Point Localisation	96
4.1	Mt. Wilson Image Database	103
4.2	Example WILD Images	105
4.3	WILD Capture Apparatus	106
4.4	WILD Images with Snow	108
4.5	Histogram Truncation Method for Correcting the WILD Selection .	110
4.6	Selection of WILD Images taken in Clear Conditions used for Testing	113
4.7	Panoramic View of the Scene Captured by SloDB	114
4.8	SloDB Images with Differing FOV	116
4.9	SloDB Capture System in Situ	117
4.10	Selection of SloDB Images used for Testing	119
4.11	Selection of Arizona Time-lapse Images used for Testing (1–15) . . .	125
4.12	Selection of Arizona Time-lapse Images used for Testing (16–22) . .	126
4.13	Selection of Reference Arizona Time-lapse Images used for Testing .	127
5.1	Affine Invariant Region Example	134
5.2	Detected Regions Produced by SIFT	136

5.3	Detected Regions Produced by MSER	137
5.4	Detected Regions Produced by HARAFF	138
5.5	Detected Regions Produced by HESAFF	139
5.6	Detected Regions Produced by EBR	140
5.7	Detected Regions Produced by IBR	141
5.8	Depiction of Overlap Error Thresholds	142
5.9	Detector Performance for WILD Selection	148
5.10	Matching Results for WILD Selection	149
5.11	Homography Computation on WILD Selection	150
5.12	Detector Performance for MSER using Different Reference Frames .	152
5.13	Matching Results for WILD Selection	153
5.14	Shadow Properties	155
5.15	WILD Shadow Examples	156
5.16	MSER Features Extracted from a Clear Weather WILD Selection . .	158
5.17	Selection of WILD Images Captured in Cloudy Conditions	159
5.18	Detector Performance for Cloudy WILD Selection	160
5.19	Matching Results for Cloudy WILD Selection	161
5.20	Homography Computation on Cloudy WILD Selection	162
5.21	Set of WILD Images used for Seasonal Change Tests	163
5.22	Image Combinations for Computing Seasonal Performance	164
5.23	Detector Performance Changes for Seasonal WILD selection	164
5.24	Matching Results for Seasonal WILD Selection	165
5.25	Fourier-based Algorithm Performance on WILD Selection	166
5.26	Fourier-based Algorithm Performance on Cloudy WILD Images . . .	167
5.27	Features Extracted from Light Dataset	169
5.28	Detector Performance Results for SloDB Selection	170
5.29	MSER Features Extracted from the SloDB Selection	171
5.30	Matching Results for SloDB Selection	172
5.31	Homography Computation on SloDB Selection	173
5.32	Matching Results for Different Descriptors on WILD Selection . . .	174
5.33	Homography Computation using a Variety of Descriptors	175
5.34	Matching Results using Modified Distance Ratio Thresholds for MSER	176
6.1	Example Specification of Shadow Region	185
6.2	Detector Performance for MSER using Relaxed Thresholds	187

6.3	Matching Results for MSER using Relaxed Thresholds	190
6.4	Homography Computation Results for MSER using Relaxed Thresholds	191
6.5	Detection Performance for Contrast Adjusted Shadow Regions . . .	195
6.6	Matching Performance for Contrast Adjusted Shadow Regions	196
6.7	Homography Computation on Contrast Adjusted Shadow Regions .	197
6.8	Detection Performance for Gamma Adjusted Shadow Regions	198
6.9	Matching Performance for Gamma Adjusted Shadow Regions	199
6.10	Homography Computation on Gamma Adjusted Shadow Regions . .	200
6.11	An Image Enhanced with Homomorphic Filtering	200
6.12	Detection Performance for Homomorphic Filtering on Shadow Regions	201
6.13	Matching Performance for Homomorphic Filtering on Shadow Regions	202
6.14	Homography Computation on Homomorphic Adjusted Shadow Regions	203
6.15	Mockup of PILLAGE algorithm	205
6.16	Final PILLAGE feature set	211
6.17	Detector Performance using PILLAGE and the Clear WILD Selection	212
6.18	Matching Performance using PILLAGE and the Clear WILD Selection	213
6.19	Homography Computation on PILLAGE and the Clear WILD Selection	214
6.20	Detector Performance using PILLAGE and the Cloudy WILD Selection	215
6.21	Matching Performance using PILLAGE and the Cloudy WILD Selection	216
6.22	Homography Computation on PILLAGE and the Cloudy WILD Selection	217
6.23	Detector Performance using Arizona Test Set	218
6.24	Matching Performance using Arizona Test Set	219
6.25	Homography Computation on Arizona Test Set	220
6.26	Detector Performance Between Arizona Reference Images	221
6.27	Matching Performance between Arizona Reference Images	222
6.28	Homography Computation Between Arizona Reference Images . . .	223
6.29	Detector Performance using PILLAGE and the SloDB Image Selection	224
6.30	Matching Performance using PILLAGE and the SloDB Image Selection	225
6.31	Homography Computation on PILLAGE and the SloDB Image Selection	225
A.1	Detection Performance between WILD Images using 3σ and 2σ . . .	240
A.2	Histogram Truncation Results when Utilising 2σ	242
A.3	Matching Performance between WILD Images using 3σ and 2σ . . .	243
A.4	Homography Computation between WILD Images using 3σ and 2σ .	244
A.5	Detection Performance between WILD Images using 3σ and 4σ . . .	245

A.6	Histogram Truncation Results when Utilising 4σ	246
A.7	Matching Performance between WILD Images using 3σ and 4σ . . .	247
A.8	Homography Computation between WILD Images using 3σ and 4σ .	248
B.1	Matching Score Results for WILD Selection	250
B.2	Matching Score Results for Cloudy WILD Selection	251
B.3	Matching Score Changes for Seasonal WILD Selection	252
B.4	Matching Score Results for SloDB Selection	253
B.5	Matching Score Results for Different Descriptors on WILD Selection	254
B.6	Matching Score Results using Relaxed Thresholds	255
B.7	Difference in Matching Score Results using Contrast Enhancement .	256
B.8	Difference in Matching Score Results using Gamma Enhancement . .	257
B.9	Difference in Matching Score Results using Homomorphic Enhancement	257
B.10	Matching Score Results for PILLAGE using Clear WILD Selection . .	258
B.11	Matching Score Results for PILLAGE using Cloudy WILD Selection .	259
B.12	Matching Score Results for Arizona Test Set	260
B.13	Matching Score Results for SloDB Image Selection	261

List of Abbreviations

AR	Augmented Reality
ARCHEOGUIDE	Augmented Reality Based Cultural Heritage On-Site Guide
AV	Augmented Virtuality
BA	Bundle adjustment
CAD	Computer Aided Design
CALTECH	California Institute of Technology
CCD	Charge Coupled Device
CMU	Carnegie Mellon University
CT	Computed Tomography
DOF	Degrees of Freedom
DOG	Difference of Gaussian
DGPS	Differential GPS
EBR	Edge Based Region
FOV	Field of View
FPGA	Field Programmable Gate Array
fps	frames <i>per</i> second
GIS	Geographic Information System
GLOH	Gradient Location and Orientation Histogram
GPS	Global Positioning System

GPU	Graphics Processing Unit
HARAFF	Harris Affine
HESAFF	Hessian Affine
HDD	Hard Disk Drive
HDR	High Dynamic Range
HMD	Head-Mounted Display
HMDS	Helmet Mounted Display System
HUD	Head-Up Display
IBR	Intensity Based Region
IC2	Inertia Cube 2
KLT	Kanade-Lucas-Tomasi
LED	Light Emitting Diode
LM	Levenberg-Marquardt
LoG	Laplacian of Gaussian
MIT	Massachusetts Institute of Technology
MRI	Magnetic Resonance Imaging
MSER	Maximally Stable Extremal Regions
PCA	Principle Component Analysis
PDA	Personal Digital Assistant
PIE	Pose, Illumination and Expression
Pillage	Pyramid of Illumination Images
PTU	Pan Tilt Unit
RANSAC	Random Sample Consensus
RSC	Rockwell Science Center

RTK GPS	Real-Time Kinematic GPS
SC	Shape Context
SCAAT	Single Constraint At A Time
SDR	Standard Dynamic Range
SfM	Structure from Motion
SIFT	Scale Invariant Feature Transform
SLAM	Simultaneous Localisation And Mapping
SloDB	Slovenian video Database
VASC	Vision and Autonomous Systems Center
VR	Virtual Reality
WILD	Weather and Illumination Database

Chapter 1

Introduction

“The most profound technologies are those that disappear. They weave themselves into the fabric of everyday life until they are indistinguishable from it”

Mark Weiser (1952–1999)

The field of computer graphics has made giant strides in recent years, and now approaches technology capable of rendering photo-realistic images in real time. In the real world, people are becoming more accustomed to information traditionally only available at a desktop computer to be available wherever they are. The concept of Augmented Reality (AR) provides a medium for presenting this data to the user in an unobtrusive way, by merging the data directly into the view of their surroundings. It does this by leveraging advances in computer graphics, allowing realistic virtual content to be rendered inside the user’s view. Information on where these objects are positioned is provided by sensors capable of tracking the user’s position and orientation. The current level of technology means systems are bulky and provide inconsistent performance. The final goal is systems that seamlessly blend virtual content into the real world with a level of realism such that the user is unable to distinguish the real from the virtual.

By way of introduction, in this chapter AR is defined and described, then the problems of AR implementations are discussed, the most fundamental being the registration problem. Finally the contribution of this thesis and its structure is outlined.

1.1 Overview

AR combines the real world with the virtual. It merges computer-generated graphics into real-world environments by inserting 2D or 3D objects into a live video feed from a digital video camera, presenting this augmented view to the user through a Head-Mounted Display (HMD) or on some type of computer screen. The combination of the real and the virtual allows the enhancement of the real world through the addition of content that can aid the user. Augmentations can range from simple annotations overlaid on real objects, to realistic virtual objects. Figure 1.1 shows a city scene that has been augmented with virtual advertisements such as the “City Dog” ad attached to the Sony Center building. Here the augmentations are used to attract the user’s attention.



Figure 1.1: Example AR scene (Image reproduced from [MHU⁺03])

When considering the spectrum of environments possible, Milgram [MK94] defined a continuum of real to virtual environments. This continuum is shown in Figure 1.2. The key element of Milgram’s continuum is that it is a measure of how much of the user’s environment is computer generated. Milgram places AR between the real world and Augmented Virtuality (AV), which is the addition of real objects into a virtual environment.

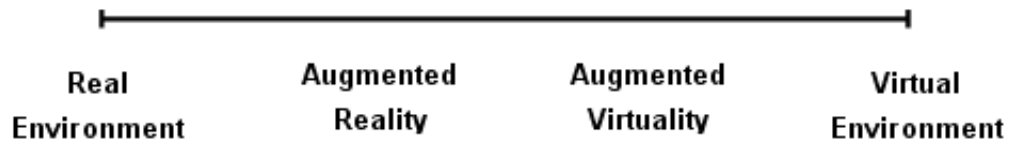


Figure 1.2: Milgram's Virtuality Continuum

An example of an AV scene from a system presented by Simsarian and Åkesson [SA97] is shown in Figure 1.3. Here a virtual environment is augmented with the textures of real objects to enhance the sense of realism while maintaining the flexibility of the virtual world. The images rendered on the windows are captured from the real world as is the image shown on the virtual monitor and the contents of the white-board on the right of the image. Both AR and AV are considered to be types of mixed reality, as they combine elements from both real and virtual environments into one space. These differ from the more well known Virtual Reality (VR), where the user's entire environment is generated by the computer.



Figure 1.3: An example AV scene. (Image courtesy of Karl-Petter Åkesson – Swedish Institute of Computer Science. Reproduced from [SA97])

The usefulness of AR has been explored by Tang *et al.*. They performed a comparison [TOBM02] of the effectiveness of an AR system which aids in an assembly task compared with other more traditional media. Results showed that errors were reduced by 82%. They also found that the mental workload was reduced when using the AR system and that this workload was offloaded to the computer.

1.2 Augmented Reality Uses

Even in its infancy researchers have found many applications for AR and as the technology improves will doubtless find many more. One of the first uses for



Figure 1.4: F-35 Joint Strike Fighter HMDS

AR was in enhanced vision systems for pilots to “help the pilot navigate, avoid obstacles, maintain situational awareness in reduced visibility, and interact with avionics instruments without looking down” as described by Foxlin *et al.* [FANH04]. Recently, similar technology has been utilised in a Helmet Mounted Display System (HMDS) for the F-35 Joint Strike Fighter. The HMDS replaces typical Head-Up Displays (HUDs) and provides highly accurate head tracking allowing target tracking information to be rendered inside the pilots view. General aircraft information such as navigation data and system status are also included. This allows pilots to remain focussed on their surroundings rather than having to look down at instrument panels. The system operates at night by superimposing an infrared image of the world onto the pilots view. Figure 1.4 shows a pilot wearing this HMDS.

Other military applications of AR include mechanisms for delivering more situational data to ground troops without taking their attention away from the battlefield in the way that traditional media such as maps do. One such project is described by Livingston *et al.* [LRJ⁺02]. AR is also used for simulation of battlefield exercises, where virtual enemies are inserted into real environments.

Numerous projects have implemented AR systems designed for entertainment purposes. The ARQuake project [TP02] is one such example. ARQuake is an AR version of the popular first person shooter game “Quake”, by id software. In ARQuake virtual enemies appear to move about in a real environment. The user navigates the real environment and uses a gun interaction device to shoot virtual



Figure 1.5: ARQuake; the image on the left is a mock-up, the image on the right is the view produced by the system. (Images used courtesy of Wayne Piekarski – University of South Australia)

characters; images from the system are shown in Figure 1.5.

The Magic Book project [BKP01] is another example. Here a normal book is augmented so that virtual 3D objects pop out of the page. This system allows the user to switch from an AR to a VR view of the scene, where the user becomes one of the characters in the book and views the world from a first person perspective.

Gandy *et al.* have presented a system which allows users to act in selected movie scenes with virtual actors [GMP⁺05]. The AR interface provides cues, on stage directions, and lines for the user to follow.

AR is also used in sports broadcasting. In American Football games the “first down” line is digitally rendered onto the field. When players run over this virtual line they correctly occlude it as though the line were actually painted on the field. Race car broadcasts are also augmented, with annotations added to the cars in NASCAR broadcasts. In Formula One racing, the statistics of the fastest driver for the previous lap are shown seemingly on the track with cars driving over the top of them.

AR has also been used for educational purposes. Fjeld and Voegtli have investigated the use of AR to assist in chemistry education, in a project called Augmented Chemistry [FV02]. Augmented Chemistry is a tangible user interface which allows a user to compose molecular models from basic elements of the periodic table. The user ‘grabs’ the 3D model of any periodic element, and adds them to a composite molecular structure—building their own molecular model.

Motokawa and Saito have utilised AR in learning to play the guitar [MS06]. The



Figure 1.6: An example of the support system for guitar playing. The virtual hand shows the finger positions for the indicated chord. (Image used courtesy of Hideo Saito – Keio University)

system aids the user by overlaying a virtual hand onto the fret board, showing the correct position of the fingers for a given chord. The player can then place their hand over the virtual one. A screen-shot of this is shown in Figure 1.6.

Many medical projects have also been undertaken. The University of North Carolina at Chapel Hill have developed a system for aiding in ultrasound-guided needle breast biopsies[SLG⁺96]. In this system ultrasound information is rendered directly onto the patient under the ultrasound probe in relation to the needle. Using this system an expert successfully performed a needle biopsy on an artificial tumour on a training phantom.

The Artificial Intelligence group at MIT have also investigated the uses of AR to aid in medical visualisation [GLPW⁺96]. MRI or CT scans are overlaid in real-time on the patient, effectively giving the surgeon X-Ray vision. Figl *et al.* have established a system designed to provide an AR visualisation to assist in computer aided surgery [FBE⁺02]. Here, previous work on providing additional data to the surgeon inside the operating microscope is extended to support AR. Figure 1.7 shows an example of AR's use in a medical context. The figure shows a head with a computer generated model of the brain overlaid.

An AR based forceps delivery training system has been presented by Sielhorst *et al.* [SBN05]. The system seeks to synchronise the movements of an expert and a trainee to assess the trainee's performance quantitatively. Trainees are able to use

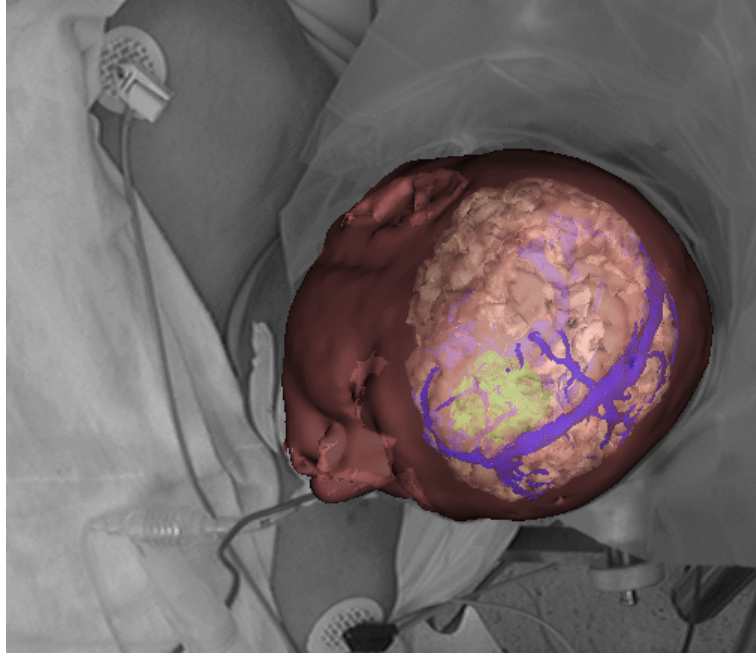


Figure 1.7: An image of a head with a virtual model of the head and brain overlaid (Reproduced from [Bro])

a HMD based AR system in order to follow the movements of an expert during the birth simulation. Figure 1.7 shows an example of a medical use for AR, where a virtual model of a head and brain are overlaid directly onto a patient.

Many other systems have been developed as tools to assist in a range of diverse fields. King *et al.* set up a system designed to assist viticulturists in understanding the parameters that affect their yields and the quality of grapes in their vineyards [KPT05]. This was achieved through visualising existing viticulture Geographic Information System (GIS) data in the field using AR.

In the automotive industry, a system under development by Tonnis *et al.* assists drivers by informing them of dangerous situations around the car by presenting this information on an AR system. The project is especially novel in that the car’s windshield is used as the display device [TSLB05].

Benko *et al.* developed a tool designed to aid in the visualisation of an archaeological dig [BIF04]. The system allows users to see a recreation of the dig site, visualise 3D models of excavated objects, as well as see more traditional 2D data which is typically gathered from a site—such as drawings, pictures and notes.

AR has also been used to enhance cultural pursuits. The “Augmented Reality Based Cultural Heritage On-Site Guide” system [VIK⁺02], called “ARCHEOGUIDE”, provides a guide for visitors to an archaeological site. It places virtual reconstruc-

tions of ruined buildings in the correct positions so visitors can see what the site may have looked like in ancient times. It also adds other items of interest; for example, virtual characters moving about in the environment as they would have done in the period in question.

Schmalstieg and Wagner propose a system to be used by visitors to a museum [SW05]. The system implements an electronic tour guide on a Personal Digital Assistant (PDA). The augmented exhibits are linked together into a story which leads the user through the museum.

In industrial applications, Boeing have experimented with the use of AR to assist their engineers in assembling wiring harnesses for planes [CM92]. In fact, it was these engineers who coined the name “Augmented Reality.”

Gordon *et al.* have designed a system for the visualisation of oil and gas field information [GBB⁺02]. The system is designed to replace traditional paper based maps which show the earth model at various depths. Such maps are used at meetings where different oil and gas exploration and production companies come together to agree upon the design of a new oil or gas field as the companies seldom work alone. The system allows each participant to have an independent view of the model and collaboratively discuss it while minimising costs compared with the more traditional immersive projection displays commonly used in this industry.

An early test case for the usefulness of AR was demonstrated by Feiner *et al.* [FMS93]. The system was used to assist in simple printer maintenance for end users. A key component of the system was that it only rendered virtual representations of real objects if the view of the object was obstructed. More recently Tschirner *et al.* made use of AR to assist in gas arc welding [THG02]. Welders have trouble seeing fine detail associated with the material being welded because of the extreme brightness from the welding process and the protective eyewear. This adversely affects the quality of the process. Through the use of the AR system a welder can see a far more detailed and clear picture during the welding, and is shielded from the radiation and light intensity by the helmet and HMD.

Navab presented a project undertaken by Siemens corporate research [Nav03]. This system was designed to aid in the revamp, monitoring, and maintenance of industrial plants. Navab and Zokai *et al.* later extended this system by taking advantage of a subset of AR called “diminished reality”. In diminished reality the AR system is used to remove real elements in the scene, effectively blanking them

out. The implementation of this by Zokai *et al.* produced an application for removing the clutter of pipelines, pumps, and valves from an industrial factory. Regions of clutter were replaced with an image of the scene behind the clutter—rendering those objects invisible. By doing so, the addition of new, replacement pipes *etc.* could be visualised, allowing factory designers to see what the changes would look like. This approach also has many potential applications for removing unsightly, distracting or irrelevant data from a user’s view.

AR systems for product design have also been implemented. Klinker *et al.* [KDB⁺02] designed a system to allow product designers to demonstrate a product to end users and to allow them to assess it. The test application they chose was car design and they collaborated with BMW to produce a test system that was evaluated by experts.

Ishii *et al.* developed a system to assist in urban design. A designer can combine multiple forms of more traditional design mediums such as sketches, physical models and computational simulation into one information space. This improves the design process and reduces the mental load required for a designer to form a mental construct from many different media, allowing the designer to concentrate more on the design. A system similar to this was presented by Piekarski and Thomas [PT02] which allows for modelling of large outdoor ground features interactively.

1.3 Problems in AR

There are numerous problems that must be addressed before an AR system can be considered ‘perfect’. A perfect system is one that is lightweight and capable of blending virtual objects seamlessly into real environments. Ideally the user should be unable to distinguish the real from the virtual. Alternatively, if the objects by their nature do not belong in a scene, a system should make them look like they at least could be real. It is important to note there are many applications of AR that are useful without requiring the technology to be perfect. Of the many problems encountered when developing AR systems three are discussed here and fall into three broad categories: ergonomic issues, registration issues, and realism issues.

1.3.1 Ergonomic

Ergonomic issues are caused by the physical size and weight of the apparatus. Current and previous AR systems are cumbersome due to the amount of equipment



Figure 1.8: Tinmith; a hybrid AR system. The image on the left shows an old version of the system while the other images show the most recent implementation. (Images used courtesy of Wayne Piekarski – University of South Australia)

the system requires. Batteries are needed to power these components, allowing the system to run for extended periods. In addition, these systems require significant computational resources, meaning some sort of computer must be included. Several types of sensors and interaction devices are also required so the user can control the system. All these components add size, weight and clutter to the apparatus. Recent advances in technology have led to smaller sensors, along with batteries and computers that are more powerful than their larger counterparts. Despite these advances current systems are still bulky but are improving. It should be noted that the size and weight of AR systems is less of an issue for indoor systems as the user does not have to carry around the computer and tracking hardware. Therefore, these issues mainly effect outdoor systems.

Figure 1.8 shows the Tinmith [PT01] system developed at the University of South Australia. The image on the left shows an older version of the system, while the other two images show the most recent version and can be considered state of the art. The recent version has a belt attached unit containing a computer, and the helmet has a built in Global Positioning System (GPS) receiver, orientation sensor, headphones, HMD and video camera. The system uses gloves for interaction and these communicate with the computer *via* Bluetooth. Wireless communication with sensors and interaction devices cuts down on the number of wires used in the system, compared with previous versions of the Tinmith system. The weight of the system shown in Figure 1.8 is 4KG not including the batteries.

The heavier the system is, the more unwieldy the system becomes. Heavy

systems can tire the user, especially when there is a lot of weight attached to the user's head from large HMDs and sensors. An optimal HMD would be no more than the size and weight of a pair of normal glasses, thereby being more discrete and lighter than current prototypes. Some types of displays such as Myvu's Crystal displays [Cor] are approaching this level of ergonomics. The glasses are approximately the size and weight of a normal pair of glasses, however, their low resolution (640×480) and narrow Field of View (FOV) (22.5°) means they are not ideal for AR applications.

Other factors greatly influence the quality of the experience for the user. Surveys conducted by the HITLabNZ [Hum] from a demonstration indoor system at SIGGRAPH 2000 found that most users identified technological problems with HMDs as the most significant, such as the resolution, FOV, and weight and size of the unit. The FOV is influenced by the choice of display. Two techniques can be adopted for AR displays, video see-through or optical see-through.

Video See-Through

Video see-through systems render a live camera feed onto a display, typically an HMD. This means the user is seeing what the camera is. This has the advantage that the images a user sees are exactly what any computer vision algorithms are also using. Any results computed using the camera's position and orientation—commonly referred to as the pose—are directly applicable to the user's view. The drawback of this approach is that the user's FOV is limited to that of the particular camera used. Generally, the FOV of a camera—which can range from approximately 50° – 90° —is considerably less than human vision which has a horizontal FOV of approximately 170° . The resolution of the camera and HMD is also considerably less than that of the human eye. This adversely affects the user as the system is automatically detracting from reality before any augmentations are added—the antithesis of AR's intent.

Azuma [ABB⁺01] identifies problems with current HMDs as: brightness, resolution, FOV, contrast levels, size, weight and cost. The parallax error caused by the cameras being mounted away from the true eye location is also identified as a problem with video see-through displays.

Optical See-Through

Optical see-through utilises a transparent display which allows the user to see the virtual objects rendered onto the display, and yet still see the real world unhindered. This means the user's FOV and the resolution of the scene is the same as without the system. The disadvantage of optical see-through displays is the difficulty in rendering virtual objects which are completely opaque. Recent projects, however, have implemented optical see through displays capable of rendering completely opaque virtual objects [KBCW03, CHR04].

Unfortunately, in their current form both video and optical see-through displays involve large and cumbersome HMDs. Other types of display medium have been used, often using a video see-through paradigm, however they do not use immersive HMDs. This allows the user to continue to see their surroundings while using the AR system.

Other AR Display Techniques

Other methods of displaying AR include the use of projectors. Ehnes *et al.* built an innovative system using projectors [EHH05]. In it, the virtual objects are projected onto real objects in the scene, eliminating the need for the user to carry any display or tracking apparatus at all. Olwal *et al.* also use projectors, however their system operates more like an optical see-through approach where two or more projectors project an image onto a transparent holographic optical element which the user looks through [OLG⁺05]. These systems are less obtrusive than other systems as they do not require the user to wear HMDs or other such devices.

Some projects have used Tablet PCs to view the augmented world [KD04], while others have utilised PDAs [GFM03, PW03]. Figure 1.9 shows an AR system developed by Pasman and Woodward for PDAs [PW03]. Henrysson *et al.* made use of mobile phones for AR as they have many advantages [HBO05]. Firstly, their ubiquitous nature means there is a large base of potential AR users. Secondly, their computational power is ever increasing as is the quality of the displays used. In addition, their displays are designed to be visible in direct sunlight. Typically they also have a built-in camera that can be used for computer vision tracking. Siemens corporation implemented an AR game for mobile phones called "Mosquito Hunt" in which the user shoots virtual mosquitos which were rendered over the real world [Sie03]. This work took advantage of computer vision techniques implemented



Figure 1.9: A PDA based AR system (Image courtesy of Wouter Pasman – Delft University of Technology. Reproduced from [PW03])

on the phone which tracked natural features and worked like a simple orientation sensor.

One advantage of these non-immersive approaches is that if the virtual object alignment is not stable and the virtual objects are moving about in a way inconsistent with the rest of the environment, they are less likely to induce motion sickness than a system making use of an immersive HMD.

1.3.2 Realism

Another problem encountered when developing AR systems is producing realistic looking virtual objects. Several issues must be addressed to accomplish this. The first of these issues is being able to correctly light the virtual objects so they appear consistent with the illumination of the scene. To do this, a model of the lighting in the real scene must be computed, and the location of light sources determined. Both Kanbara and Yokoya [KY02] and Agusanto *et al.* [ALCS03] term this process “Photometric Calibration.”

Shadows are an important element in making a virtual object appear more real and can aid the user in 3D depth perception of the scene. It was discovered by Sugano *et al.* that shadows were beneficial even when they were rendered in a manner

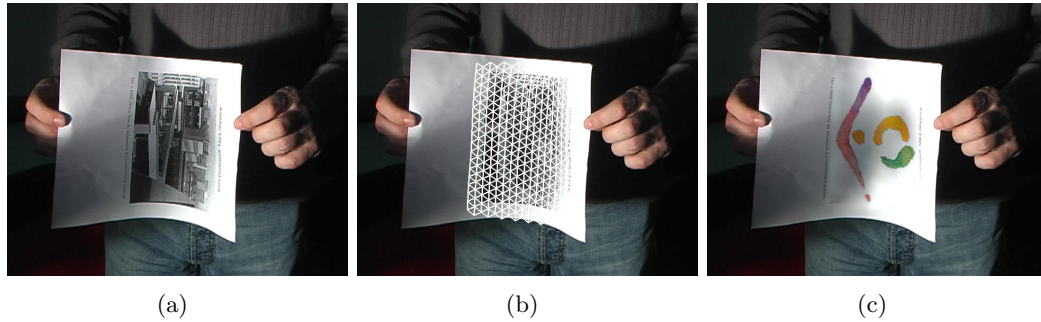


Figure 1.10: AR on deforming surfaces. (a) shows the surface to be augmented. (b) shows the estimated deformed mesh. (c) shows the printed image replaced with a virtual object correctly deformed to match the paper, and illuminated to be consistent with the scene. (Images courtesy of Julien Pilet – EPFL - Computer Vision Laboratory. Reproduced from [PLF05])

inconsistent with the actual lighting in the scene [SKT03]. In order to achieve accurate shadowing, real shadows must be applied to virtual objects should they fall under the shadow of a real object. Virtual shadows must also be applied to real objects so that the rendered shadow is consistent with the 3D structure of the object in its shadow. Good knowledge of the 3D structure of the scene and the objects that make up the scene is required in order to apply these shadows.

Knowledge of the 3D structure of the scene—or at least the depth of objects from a given viewpoint—is also required to be able to correctly create occlusions in the scene. This means that real objects can correctly occlude virtual objects, if the virtual object is positioned behind a real one. This problem is considered by Kim *et al.* [KjYS03], where depth maps of the scene are created so the real and virtual objects can interact.

Pilet *et al.* considered the problem of accurately rendering virtual objects onto deforming real objects [PLF05]. Generally when surfaces are augmented, they are assumed to be planar. In Pilet’s system, deformable surfaces are supported by determining the deformation of the surface with respect to some reference image. This deformation was then applied to the virtual object so it appears consistent with the surface of the real object. The shading of the real object was also determined. This was applied to the virtual object to increase the realism of the final scene. This process is shown in Figure 1.10.

It is also noted that there are some instances where it might be desirable to have non-photorealistic objects so the user can easily distinguish between what is really there and what is not. A user walking down a street littered with photorealistic

virtual billboards may attempt to dodge objects which are not actually there.

1.3.3 Registration Problem

One of the most fundamental requirements of AR systems is accurate and stable registration. Registration refers to the process of accurately aligning virtual objects to real ones. AR systems typically work in real-time, meaning they provide a live view to the user. Therefore, visual updates must be provided fast enough, and with minimal levels of delay, to produce smooth motion during movement of the object or user. A rate of 30 frames *per* second (fps) is sufficient for this.

In some situations it may be satisfactory to obtain only limited accuracy, where the virtual object's actual location may be misaligned from the desired location by a few pixels. A user navigating around a campus may not notice or care if a virtual label above a building is a few pixels off, corresponding to perhaps a few metres misalignment. There are many situations, however, where very high accuracy is needed. In the case where an AR system is used to aid in performing needle biopsies [SLG⁺96, GLPW⁺96], even a small misalignment could cause a failed biopsy. In the case of the pilot navigation system [FANH04] where pilots might be relying on the system to fly in conditions with reduced visibility, imprecise alignment could result in the plane crashing.

Errors in these measurements manifest themselves in several ways: a measurement error that is consistently off by a fixed amount results in virtual objects appearing to stay in the same location with respect to the real world, but being displaced from the desired location. More serious errors result from unstable measurements, where small variations in successive measurements result in the virtual objects swimming about, or worse, jittering. Such objects can distract the user, breaking any illusion that they actually exist in the real world, which is the ultimate goal of AR systems. If the objects are prominent enough or the user is relying on them to navigate they might induce motion sickness.

In order to achieve accurate registration, the user's position and orientation relative to the scene must be precisely determined for the duration of the system's operation; a process known as tracking. Once this pose is known, the virtual object can be rendered accordingly, appearing as it would be seen from this position. Other factors can also prevent accurate registration, such as mechanical misalignments in the HMD or incorrect camera calibration.

Many different technologies exist to aid in tracking a user's location and orientation. Of these, computer vision has been identified as a technique capable of providing accurate registration. Computer vision involves a computer system gaining some understanding of an image through an analysis of its various properties and make-up. One of the advantages of computer vision is that it is cost effective—all that is required is image information from a video camera. This is especially true in video see-through applications where the same camera that captures the images the user sees, also serves as the tracking device, as noted by Park *et al.* [PJN99]. Furthermore, when using video see-through displays the video of the real world can be delayed to match the system delay induced by the trackers, thus enabling precise final registration.

Although capable of producing accurate registration, the performance of current vision techniques are highly dependent on imaging conditions, making their performance unreliable. The algorithms are often computationally expensive, and require a clear line-of-sight. Occlusions of the camera may interfere or prevent the system from successfully tracking.

The ultimate aim of AR, as identified by Azuma [AHIS99], is “accurate operation indoors, outdoors, and everywhere else.” To achieve this in a computer vision based AR system robust computer vision algorithms are needed. The system may be expected to operate in a large range of possible outdoor scenes, from snow covered mountains and forests, to suburban streets and city centres, through to open fields and sunny beaches. Environmental changes in the scene itself such as fog, rain and snow, as well as lighting changes from dawn to dusk and between seasons all compound to make a difficult problem worse, and can render an already brittle vision algorithm useless. The algorithms must also cope with dynamic changes in the scene due to cars or people coming and going, objects being moved, occlusions and varying viewpoints.

Environmental changes impact the images of the scene, thus any computer vision stages that rely on the images directly are at risk. The majority of existing work presents algorithms working either on images taken under ideal conditions, or under the assumption of uniform changes in illumination. Uniform illumination changes apply an equal change in illumination across an entire image. Many applications for computer vision however, such as AR or video surveillance, require algorithms to operate under a wider range of conditions and cannot make this assumption. Vision

techniques have mostly been designed to cope with uniform changes in illumination and assume a diffuse ambient light source, however the presence of non-uniform changes due to shadows from scene structure, reflections, and point light sources must be considered in order to achieve the level of robustness desired.

1.4 This Thesis

This thesis investigates whether the robustness of computer vision systems operating in the outdoors can be improved through an in depth analysis of the effects of adverse environmental conditions. It is important to note that the results of the analysis are applicable to any application of computer vision that operates in the outdoors not just AR. Some examples of other fields that could benefit from this work include surveillance, robot navigation and 3D reconstruction.

The analysis is based on commonly used methods for evaluating the performance of computer vision algorithms. Six state-of-the-art feature detectors and five matching algorithms are tested, each developed by other researchers in the computer vision field. The algorithms are tested on several data sets, one of which was captured during the course of this project specifically for testing the robustness of vision algorithms. An important element of data sets applicable for such tests is their inclusion of ground truth data that enables results from the vision algorithms to be confirmed based on actual measurements.

In the context of this analysis several novel methods are proposed to deal with non-uniform illumination changes. The methods attempt to lessen the impact of shadows on several fronts and an analysis into their effectiveness is included.

1.4.1 Thesis Contributions

The major contributions of this thesis include:

- A video collection with ground truth data *per* frame, suitable for testing computer vision algorithms under a range of weather and illumination conditions.
- An analysis of the effects of non-uniform illumination changes on a variety of detection and matching algorithms
- Two techniques for improving feature detection in regions of shadow and an evaluation into their effectiveness

- The introduction of a new algorithm designed to minimise the impact of shadows on reference data

In addition several minor contributions include:

- Reports of lessons learned in the implementation of a video collection capture system that incorporates ground truth data *per* frame. This is useful for anyone contemplating setting up such a system.
- A proposed image library extension to the ARCHEOGUIDE AR system to increase the number of view points it can operate from.
- A modification of Skrypnik and Lowe’s [SL04] 3D AR system to instead use Hartley and Zisserman’s [HZ04] procedure for building 3D reconstructions. A detailed explanation of this 3D reconstruction procedure is included.

1.4.2 Thesis Structure

The structure of this thesis is as follows: Chapter 2 examines the history of AR and the diverse range of sensors available for tracking, along with their advantages and disadvantages. Computer vision is presented as a potential tracker, and existing approaches to its use in AR are described. Relevant developments in the history of computer vision based AR systems are presented, up to current state-of-the-art approaches.

In Chapter 3, two computer vision based approaches for achieving registration in AR systems are described in detail. One approach uses a global 2D based algorithm. The other, a more state-of-the-art technique incorporating 3D models of the scene being augmented, built using automated methods utilising local features. These systems give contextual information on how computer vision is used to build complete AR systems, and identify stages in the computer vision system affected by adverse imaging conditions, thus indicating stages requiring more in depth analysis.

To enable an analysis into the performance of computer vision algorithms in the presence of adverse environmental conditions a suitable dataset is required. Chapter 4 presents two datasets designed to evaluate the algorithms under a wider range of conditions than generally tested. Both capture a spectrum of weather and illumination effects representative of potential situations an AR system may need to operate in. The second dataset was collected as part of this project. Details of

how the capture system was built, and the lessons learned in its construction are also presented.

Chapter 5 presents the performance of six feature detectors—utilising a range of different image properties—when operating on the test data of Chapter 4. Any of these methods could be used for the first stage of the 3D AR system discussed in Chapter 3; that is the feature extraction stage. In addition, the performance of several descriptor algorithms are evaluated. Non-uniform illumination changes are the major focus of this analysis, as they are the most common adverse imaging condition a system encounters in the outdoors. Changes in illumination occur every day while environmental changes such as fog and snow occur with lesser frequency.

Chapter 6 investigates several methods of improving performance in the presence of a non-uniform illumination change shown to significantly impact performance: shadows. Several approaches are identified that fall into two categories: lessening the impact of shadows in the live data or in the reference data. Reducing the impact of shadows in the reference data is achieved through the introduction of a novel algorithm called PILLAGE that was developed as part of this project. Finally Chapter 7 has closing remarks and directions for future research.

Chapter 2

Background

“If we knew what it was we were doing, it would not be called research, would it?”

Albert Einstein (1879–1955)

Since its initial inception, AR has come a long way. The ultimate goal, however, of photorealistic virtual objects augmenting any scene in any situation is still some way off. Computer graphic systems provide excellent capabilities at rendering objects in a photorealistic manner. Enabling AR systems to work in any situation is an area requiring further development. One of the most fundamental aspects in this regard is ensuring pixel level registration accuracy and maintaining this accuracy under changes in the scene induced by point-of-view, movement in the scene, variable illumination or weather conditions. The following section details existing literature which examines the types of registration error in AR systems as well as their causes.

2.1 Causes of Registration Error

Holloway [Hol95] examined the causes of registration error in an example AR system, and the impact each source of error had on registration accuracy. Four sources of registration error were documented: acquisition/alignment error, head-tracking error from delays and measurement error, display error and viewing error. Holloway created a test system that allowed him to systematically introduce errors of each type into the system and measure the registration error. It was found that error due to delay is greater than all other sources of error combined. Holloway concluded

that the only way of achieving accurate dynamic registration was through the use of predictive head tracking.

Azuma [Azu95] investigated the use of predictive head tracking to reduce dynamic registration errors. Azuma broke registration errors into two categories: static errors, which are present when the user’s viewpoint and the objects in the environment remain stationary, and dynamic errors that have no effect until the viewpoint or the objects are moved. Dynamic errors were caused by the end-to-end system delay; that is, the time difference between when the tracker measured the position and orientation to the time the generated image was shown in the HMD. Azuma demonstrated that predicting future head locations could be an effective method of reducing dynamic registration errors. However, in order to evaluate the effect of prediction, accurate static registration was required, therefore a custom optoelectronic head-tracking system was developed to facilitate this. The system predicted future head positions with the aid of inertial sensors. Azuma discovered that, on average, prediction using inertial sensors is two to three times more accurate than without, and five to ten times more accurate than without any prediction at all. The main sources of static errors were identified as optical distortion, errors in the tracking system, mechanical misalignments, and incorrect viewing parameters.

Twelve years later, the processing power available in modern computers has seen a dramatic increase, by approximately a factor of 20. This increase in computing power means that trackers making use of a CPU to perform all their calculations (such as computer vision based trackers) can now provide pose updates at an increased rate with less delay between each measurement. Conversely, the added processing power has permitted previously slow algorithms to be performed in real-time.

Many tracking sensors, however, have not improved much in terms of update rate. For example, when comparing Ascension’s Flock of Birds [TGW⁺95] tracker to the Intersense Inertia Cube 2 (IC2) [Int], the update rate has only increased from 144Hz to 180Hz. The Flock of Birds tracker was a popular magnetic tracker 12 years ago while the Intersense IC2 is a more recent 3 Degrees of Freedom (DOF) orientation tracker containing nine sensors; it has an accelerometer, gyro, and magnetometer for each axis (yaw, pitch, and roll). In contrast to this relatively minor improvement in update rate, Welch and Foxlin [WF02] identify that an ideal tracker would have an update rate of 1000Hz with less than 1ms of latency between when the measurement

was taken and when that measurement is available to the rest of the tracking system. The IC2 has 2ms latency, “The Flock’s” latency is not advertised. This means that dynamic registration errors still comprise a significant portion of the error associated with current AR systems. More detail on tracking technologies is provided in the following section. The area where these sensors have improved the most is in terms of their size.

Despite this, static registration is still an important consideration as these errors are present both when the user is stationary and during movement. This thesis, by Azuma’s definition, deals with static alignment errors induced by the tracking system. From this point on the term ‘registration’ will refer to static not dynamic registration.

The custom optoelectronic head-tracker was used by Azuma to provide accurate registration. This limited the usefulness of the system due to the size of the apparatus, and the limited range imposed by the need for a prepared, active environment. Fortunately, accurate registration is now possible without the need of a custom solution. Numerous more generic technologies exist to track the location and pose of a user. The following section describes many types of trackers and discusses their use in existing AR projects.

2.2 Tracking Technologies

In order to obtain accurate registration the AR system must precisely determine the position, orientation and optical properties of the real camera (in the case of video see-through) so that virtual objects can be rendered as if seen by the real camera. Monitoring the user’s pose must occur the whole time the system is in operation and is a process known as tracking. Many technologies exist to aid in tracking including: GPS, inertial sensors, dead-reckoning, magnetic sensors, acoustic sensors and active and passive computer vision based tracking. Table 2.1 summarises some of the technologies discussed in this section.

Azuma [Azu99] identified GPS as a technology available for tracking in the outdoors. GPS has an accuracy of 10–30 metres down to centimetre level, depending on whether standard (GPS), Differential GPS (DGPS), or Real-Time Kinematic GPS (RTK GPS) is used. One disadvantage of both DGPS and RTK GPS is that they require communication with a reference station to achieve their improved level of accuracy. RTK GPS further requires the location of the reference station to

be precisely known, and uses the carrier phase of the signal to provide real-time corrections to the position measurements. All GPS systems require direct line-of-sight with at least four satellites in order to be able to compute its position. RTK GPS requires line-of-sight to six satellites in order to achieve optimal performance. This line-of-sight can be blocked by buildings, hills or other objects.

Inertial sensors allow position to be recovered using accelerometers, and orientation to be recovered using gyroscopes. The advantage of these sensors is that they are sourceless; they do not require any external source to work. Their main drawback is that they drift over time and the amount of error accumulates.

Passive optical uses a video feed to identify the user's location based on what is visible. These approaches are adversely affected by fast camera movements which blur the video image making it difficult to analyse. The camera also requires line-of-sight to the scene, therefore occlusions can prevent operation.

The final tracking technology discussed by Azuma is electronic compass and tilt sensors which provide orientation information. The main drawback of magnetic sensors is that they are adversely affected in the presence of large masses of ferrous metals such as transformers and vehicle chassis. Large electrical currents and permanent magnets such as electric motors also produce magnetic fields that can interfere with magnetic sensors. Azuma observed that the peak-to-peak error when using the compass was between 20–30 degrees, far too much to be used as the sole source of orientation information. Azuma concludes by stating that the only feasible approach to building a robust system is with a hybrid tracking approach as no one technology is accurate and robust enough to solve the tracking problem for all applications.

AR projects utilising virtually every tracking technology possible have been implemented. ECRCs User Interaction and Visualisation Group made use of magnetic trackers such as The Flock [TGW⁺95]. The Flock tracks multiple magnetic sensors and provides position and orientation measurements 144 times *per* second. UNC also used The Flock in some of their projects [SHC⁺96]. The Flock of Birds tracker, like all magnetic based trackers, is susceptible to magnetic fields generated by nearby metal objects and is also somewhat bulky. Hoff and Azuma [HA00] also utilised magnetic trackers and described a technique to automatically calibrate a compass in order to cope with environmental disturbances in the Earth's magnetic field.

At the HITLabNZ [Hum] a more recent outdoor AR project called “Outdoor AR”, used RTK GPS capable of providing centimetre level accuracy. The intended application for this system was the visualisation of survey data in the field.

Tracking Technology	Data Provided	Accuracy	Update Rate
1) GPS	Position	10–30m	1Hz
2) DGPS	Position	10cm–2m	1Hz
3) RTK GPS	Position	<10cm	1Hz
4) IC2	Orientation	1° RMS at 25°C	180Hz
5) Flock of Birds	Position & Orientation	Position: 1.8mm RMS Orientation: 0.5° RMS	144Hz

Table 2.1: Summary of tracking technologies discussed

2.2.1 Hybrids

Indeed most trackers are insufficiently accurate to be used alone at the best of times, let alone in difficult circumstances. In addition, some types of sensors only provide certain information. For example, GPS only provides positional information, meaning it would need to be combined with some type of orientation sensor in order to form a complete solution. These factors necessitate the need for hybrid AR systems.

Welch and Foxlin have performed an extensive comparison of a large number of tracking technologies [WF02]. They conclude that no one technology is robust enough to form a complete solution in every circumstance, and that the choice of tracking technology should be made based on the application it will be used for. Many other researchers also concur with this assessment [Azu97, AHIS99, YNA99].

Azuma *et al.* built a hybrid system utilising both inertial (gyro) and magnetic (compass) sensors [AHIS99]. They assume the scene being augmented is distant, thus the accuracy of DGPS is sufficient to provide positional information adequate for rendering virtual objects in approximately the correct position in the user’s view. State *et al.* choose to combine vision-based landmark tracking with magnetic sensors [SHC⁺96].

The Tinmith project [PT01] from the Wearable Computer Laboratory at the University of South Australia is a hybrid system combining GPS, magnetic and gyroscopic sensors for orientation, as well as computer vision to provide accurate tracking. This system was used for ARQuake [TP02] described in Chapter 1, Section 1.2, and also an outdoor modelling system [PT02].

Feiner *et al.* from Columbia University developed an AR system called the Mobile Augmented Reality System (MARS) [HFT⁺99]. Using the MARS system, several different applications were tested, one of which was the “Touring Machine Scenario” [FMH97]. In this scenario, users were able to query the system for information about places and buildings in the Columbia campus. They also experimented with extending the Touring Machine to provide additional media such as videos, sound clips, newspaper articles and other information, all within the spatial context of the environment. The MARS system uses a see-through HMD tracked by RTK GPS, providing centimetre level accuracy. Orientation information is provided by a combined inertial and magnetometer sensor. In addition to the HMD, the user also carries a hand-held display showing a map of the environment including the virtual information. In a recent update to the project [HHTF01], the group combined multiple tracking technologies to allow the system to operate in a variety of different environments. An ultrasonic tracker is used when indoors, an RTK GPS tracker is used when outdoors, and when neither is available a tracker combining pedometers and an orientation sensor is used. The RTK GPS tracker has position accuracy between 1–2cm, however if less than six satellites can be seen this drops to 10cm or even metre level accuracy. If the GPS base station cannot be contacted, normal GPS is used with accuracy between 10–20 metres.

A user interface which adapts to the level of tracking accuracy was also implemented [HHTF01]. When indoors and in range of the accurate ceiling tracker, a 3D wireframe model is overlaid on top of the real world. When out of range of the ceiling tracker the less accurate dead-reckoning system takes over. Dead-reckoning systems provide relatively better orientation estimates compared to their position measurements. Therefore, when the system switches to dead-reckoning the interface switches from the overlaid 3D wireframe model to a “World in Minature” view, which displays a small-scale model of the environment centred on screen but with the correct orientation.

The pilot navigation system presented by Foxlin *et al.* integrates both inside-out and outside-in optical measurements [FANH04]. That is, measurements from the user’s perspective viewing the scene, and outside-in where fixed mounted cameras in the scene track the user’s location. This concept is shown in Figure 2.1. Inertial data from both helmet and aircraft mounted sensors are also used to provide accurate and robust low-latency registration. The system implemented in this dissertation

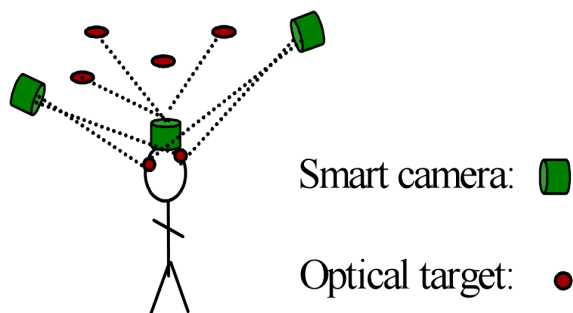


Figure 2.1: Conceptual diagram of inside-out and outside-in optical tracker. (Image courtesy of Eric Foxlin – Intersense Inc.)

follows the inside-out paradigm, where the system establishes its pose by making observations of the world, rather than the world telling us our position as is the case in an outside-in system.

Welch determined that inertial-vision hybrids are a good combination due to their complementary nature [Wel95]. The reason for this is that computer vision performs best during slow movement—when the image quality is at its best and it has sufficient time to process the frames. During slow movements, the bias and drift associated with inertial based trackers becomes the most noticeable. Inertial trackers on the other hand, perform best during fast movements when the signal-to-noise ratio is high, whereas fast movements produce the worst results in computer vision as images are often blurred to the point where the algorithms no longer work. This complementary nature covers the weaknesses of each individual technology and has been exploited in many projects since [AHIS99, KFTY00, YN01, FN03].

Inertial sensors also benefit computer vision systems as they can provide an estimate of the camera’s motion. For example, when performing feature matching with a rough estimate of inter-image motion, the search space when matching a given feature goes from the entire image to a small region. This has two benefits: reduced processing requirements as there are fewer potential points to match; and a decreased likelihood of obtaining a false match. By reducing the search region to that which must contain the correct match, a significant number of potential false matches are eliminated. This reduces the likelihood of obtaining a false match to a feature more similar in appearance than the correct match.

In summary, a range of different sensor technologies exist to provide registration information. Unfortunately, no one technique provides the level of accuracy and robustness required for an AR system, therefore hybrids are formed by combining multiple sensor types, covering the weaknesses of a single approach. Combinations

involving computer vision based techniques have proven to be successful as vision approaches are capable of providing the pixel level registration required in AR.

The following section describes various approaches in the use of computer vision for AR. Projects that employ these methods are also discussed.

2.3 Computer Vision

Computer vision has become popular in many fields including AR. Bajura and Neumann identify a problem with registration techniques that rely solely on information from the tracking system, in that the system has no feedback on how closely the real and the virtual objects are [BN95]. By using techniques based on computer vision, the images themselves can be used to aid in registration, providing feedback to the system.

There has even been a project using active computer vision. Active vision was described by the Robotic Systems Lab [Rob] as: “one that is able to interact with its environment by altering its viewpoint rather than passively observing it.” Davison *et al.* [DMM03] present a wearable collar-mounted robot with a camera utilising traditionally robotics based techniques such as Simultaneous Localisation And Mapping (SLAM), to determine the location of the camera with respect to the scene. The robot is intended to be able to assist the wearer in various actions, and being autonomous is able to track objects independently of what the user is looking at.

The more common computer vision technique is passive computer vision, where the system itself has no control over what it is seeing and takes what images it is given and processes them. The following section separates computer vision down into projects which employ tracking aids such as passive or active fiducial markers, which are artificial objects introduced into the scene to assist the computer in tracking its location, or those which make use of natural features; that is, tracking existing elements in the scene.

2.3.1 Fiducial Markers

Accurate registration has been achieved in AR through the use of fiducial markers. Fiducial markers provide a known point of reference inside a scene. Virtual augmentations can then be geometrically calibrated by defining their location in the real world with respect to these reference points. Establishing the user’s pose in

relation to these points, together with knowledge of the augmentation's calibration, allows us to establish the relationship between the user's pose and that of the virtual object. This permits a correct rendering of the virtual objects as seen from the user's viewpoint.

Markers are designed to be clearly identifiable inside a view of the scene. They stand out in order to be easily extracted by the computer vision system. Types of markers include retro-reflective markers, printed markers with unique patterns allowing the system to determine which marker it is seeing, or active markers which emit light to stand out more from the rest of the scene. The more distinctive and easily identifiable a marker is, the more robust the system will be.

Passive Fiducial Markers

Vogt *et al.* implemented a system which utilises clusters of disc shaped retro-reflective markers [VKSN02]. The camera utilises an infrared illuminator to be able to sight the markers. The system is applied to the problem of tracking a needle when performing a guided needle biopsy on an abdominal training phantom.

A prominent marker based approach in AR is presented by Billingham and Kato from the HITLabNZ and Hiroshima City University [KB99]. Their system called the "ARToolKit" provides the tools necessary to create AR applications. The library provides the ability to track square planar fiducial markers in a video see-through AR system, as well as all the functionality to calculate camera position and orientation relative to physical markers in real time. The virtual objects can then be rendered attached to the markers. It also assists in setting up custom markers with arbitrary patterns. A marker is a black square surrounding a white interior with a black pattern in the centre used to identify an individual marker. Figure 2.2 shows an example of such a marker on the open page, as well as the augmentation associated with it. The computer vision system works by thresholding the image and identifying square regions which could potentially be markers. Once markers have been identified, the pattern in the centre of the marker is extracted and matched to a database which determines the corresponding 3D model to be attached to this marker. The 3D model and its geometric calibration relative to the marker are established *a priori*.

Through the use of markers, projects such as the Magic Book [BKP01] have been possible. The Magic Book is similar to a normal story book except it has markers



Figure 2.2: Magic Book. The logo with stars is the augmentation added to this scene. (Image courtesy of Mark Billingham – HITLabNZ)

printed on one of the pages where a picture would normally be. Figure 2.2 shows an example of a magic book. When the book is viewed using an AR display, 3D models appear to pop out of the page. The user can even go into the book and immerse themselves in the VR version of the world. Many other projects have also taken advantage of these fiducial markers [LBC04, SW06, TP02]. In particular, Thomas and Piekarski utilised fiducial markers in ARQuake [TP02].

ARTag [Fia04] is based on the ARToolKit and improves on it by utilising digital encoding methods rather than the correlation matching method of ARToolKit. This results in lower error rates with few false positives and reduced instances of confusion between markers. ARTag uses an edge detection method to find markers as opposed to the threshold approach of ARToolKit. This change enables the system to cope with large occlusions as shown in Figure 2.3.

Zhang *et al.* performed a comparison between various systems utilising square planar fiducial markers [ZFN02]. The comparison evaluates the usability, efficiency, accuracy, and reliability of each system. The usability is concerned with how easily users can integrate the AR system into their application. The efficiency is based on how long it takes each system to locate and decode a marker inside an image of the scene. This factor influences how fast the final system can run and the frame rate possible. The measured accuracy is based on the error in locating the markers in the image while the reliability is tested under less than ideal situations, for example when

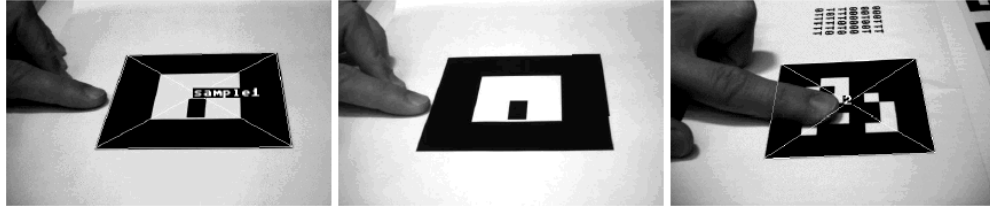


Figure 2.3: The left most image shows an ARToolKit marker with no occlusion, the middle image shows the same ARToolKit marker with a small amount of occlusion failing to match. The rightmost image shows an ARTag marker correctly detected even in the presence of occlusion (Image courtesy of Mark Fiala – NRC Institute for Information Technology. Reproduced from [Fia04])

using a poorly focused camera. The results indicate that the ARToolKit performed well and is easy to incorporate, however its robustness and accuracy mean it is more suitable for prototyping rather than systems which require a large number of markers.

Naimark and Foxlin presented another approach to fiducial design [NF02, FN03]. The design is a circular shape with a bar-coded pattern in the centre. These markers are placed on the walls and ceilings, and improved on previous projects which use simpler round, coloured fiducials with concentric rings [NYC⁺99]. The system using these markers is designed to operate in potentially large areas, meaning a large number of fiducials is used. This required a more computationally efficient approach to marker identification. They solve this using bar-coded patterns rather than the arbitrary patterns used in the ARToolKit. Performing matching on arbitrary shapes can be computationally expensive when using large databases. A belt based unit houses a processing unit and DSP to perform the fiducial detection, while the head mounted unit consists of an upwards facing camera and an inertial sensor.

Baratoff *et al.* utilise multiple markers simultaneously to improve the pose estimation in “out-of-the-laboratory” situations, because pose estimates from a single marker are insufficiently stable [BNR02]. To use multiple markers simultaneously, the relative marker poses had to be properly calibrated with respect to each other and a semi-automatic technique is proposed for this.

Many groups using fiducial markers have achieved accurate registration in indoor AR systems. The use of passive fiducial markers, however, is inappropriate for most outdoor systems as the scale of the environment would require large numbers of giant markers be placed all over the scene. In contrast, active optical based techniques make use of markers which emit light, making them less intrusive yet more distinctive than passive markers. This makes them easier to detect, increasing their robustness

and allowing them to operate over longer distances. Three such projects utilising active markers are listed here.

Active Fiducial Markers

One indoor project which makes use of active rather than passive markers was developed at the University of North Carolina at Chapel Hill and is called HiBall. Welch *et al.* [WBV⁺01] describe the latest system by the group, which uses a small optical based unit mounted on top of the users head. The unit provides position and orientation information and contains six lenses that track Light Emitting Diodes (LEDs) mounted in ceiling panels in a large room. The project works towards developing a wide area tracking system to be used for experimentation with virtual and augmented reality applications.

The system takes advantage of previous work such as the Single Constraint At A Time (SCAAT) algorithm [Wel96], which generates new tracking reports whenever a new measurement is made by the system. This approach allows pose estimates to be generated more frequently, with less latency and improved accuracy. A Kalman filter [Kal60]—which is an algorithm used to calculate an optimal estimate of a past, present, or future state of a system using a time sequence of measurements—fuses the measurements into an estimate of the pose of their system.

The authors report that the HiBall tracking system generates over 2000 pose estimates *per* second, with less than 1 ms latency, 0.2mm position resolution and 0.03° orientation resolution. For future work, the group identifies the need to progress from dense, active landmarks such as LEDs, to fewer passive and preferably natural features that would not require any “landmark infrastructure.” They also state that their long-term goal is to have a system that works outdoors; they believe that such a system “will involve a clever and careful combination of multiple complementary technologies” [WBV⁺01].

Matsushita *et al.* [MHU⁺03] developed a system utilising a high speed camera which detects optical beacons in a scene. The system determines which beacons it is seeing by decoding the individual blink pattern produced by each beacon. These beacons can be seen in any lighting condition and up to 40m indoors and 20m outdoors. The system can recognise and track up to 255 beacons simultaneously.

Naimark and Foxlin [NF05] developed a system that improved upon this technique. LEDs are used as markers, with their benefits described as having a larger

viewing angle range, being able to work in the dark and allowing small scale applications. The benefits of this project over others utilising optical beacons is that they do not use blinking codes to determine which beacon they are seeing, but rather use amplitude modulation codes which, due to their constantly on nature, can be continuously tracked after it is decoded.

Accurate and robust systems have been developed using marker based computer vision techniques. However, modifying the environment through the addition of any type of marker may not be possible in sensitive environments such as the archeological site used in the ARCHEOGUIDE system [VIK⁺02]. In addition, the use of markers requires an increased amount of maintenance. This means the preferred approach in many applications is to track natural features or rely on other types of sensors to establish the position and orientation of the viewer.

2.3.2 Natural Feature Tracking

Natural feature tracking, often called ‘marker-less’ tracking, makes use of existing elements of the scene to determine and track the user’s pose (also referred to as the camera pose). Natural features could be anything from points such as corners or intersections between high contrast regions, lines, curves, planes, to entire objects. The disadvantages of natural features compared with fiducial markers is that they are less distinctive, meaning detection can be more intermittent with features coming in and out of view. Being less distinctive also brings a higher chance of getting false positives, therefore calculations performed on any matches must be robust to outliers.

Point features are often utilised as they support higher levels of occlusion than other structures, and are common place in almost every scene. In contrast to the aforementioned planar ARToolKit markers, camera pose estimation is impossible from one point, therefore a number of features are required to estimate pose. A reasonable number of features are also required to accommodate intermittent drop out. This occurs either because the feature is no longer visible or because they were not detected in a particular frame for some reason.

Some projects exist which bridge the gap between marker based techniques and natural feature tracking approaches. One such project by Subbarao *et al.* takes advantage of both fiducial markers and natural features [SMG05]. The system has two stages: one for initial scene learning, and the second for tracking. The first

stage takes advantage of fiducial markers for the initial scene setup phase, where the system creates a 3D model of the scene using natural features from the scene. In Subbarao’s system the fiducials are used to determine the initial camera poses thus allowing computation of the scene model. Then in the tracking phase, markers are not needed and the system can be tracked utilising the learned scene features. This system also differs from others through its use of the HEIV estimator [MM00, Mat01] as opposed to the more common Levenberg-Marquardt (LM) algorithm. HEIV, as described by Subbarao *et al.* [SMG05], is used to “solve the nonlinear regression for camera pose and structure recovery”; in essence, determining the camera pose and the 3D model from a set of tracked features. The method also differs from others in that it employs the projection based M-estimators [CM03] for outlier detection rather than the more widely used Random Sample Consensus (RANSAC) method [FB81].

Kanbara *et al.* also use both fiducials and natural features [KYT01] together with an inertial sensor. They use a combination of both types and when markers are not visible they switch to only using natural features until a marker is visible again.

There are a number of differences that set apart projects which solely use natural features. These differences include the type of algorithm they use together with the type of features they track such as lines, points, and shapes.

The Mixed Reality Lab at the National University of Singapore takes the idea of matching to reference images and extends it by including information from previous frames. In a project called “Camera Motion Tracking from Natural Features” [CCP02], the group implement a system to recover the 6DOF camera pose in an AR system, using natural features rather than fiducial markers. Corner features are detected in each image then the features are matched. Once corner matches have been determined the change in camera pose must be calculated based on the change in feature locations in the images when compared with two or more reference images of the scene. By making comparisons with reference images rather than the previous frames, problems associated with the gradual increase in drift are avoided. One problem with matching to reference frames is that the current user pose may be a long way from the reference image’s pose—a problem called the “wide-baseline” problem. To solve this, two matches are computed: a global homography between the current frame and the previous frame, as well as an affine transformation between the previous frame and the reference frame. The RANSAC algorithm is then used

to get the closest matching corner points between the images.

Some projects exploit natural features which are less common than points. The systems utilising these features only operate in specialised circumstances when the scene actually contains features of this type. The benefit of such approaches is that they can provide accurate results while requiring fewer features than when using points. Examples of such systems follow.

Specialised Techniques

Simon *et al.* describe a vision-based technique which can accurately solve a more specialised case of the general problem, which is when planar structures are visible in the scene [SFZ00]. Planes such as those found in urban environments like the sides of buildings, the ground, and walls can be tracked throughout a video feed and be used to accurately determine the user's pose and thus register virtual objects inside the scene. The group notes that a sequential approach, which determines relative pose changes from the previous frames, is needed for autonomous or long running applications.

Simon and Berger then extended the technique to track multiple planes in the scene thereby improving the accuracy of the system [SB02]. This system also reduces the amount of preparation, as camera calibration and scene acquisition are performed while the camera is moving. The authors note that vision-based methods are generally accurate since results are computed directly from features extracted from the images to be augmented. They also note that when tracking natural features, too few visible features results in tracking which suffers from high-frequency jitter. The system uses Harris corner features [HS88] in addition to RANSAC and previous work on image matching between two views [ZDFL94] for determining the camera motion throughout the video feed.

At the Rockwell Science Center (RSC), Behringer *et al.* have investigated other computer vision methods to aid in registration [Beh99]. Behringer describes an outdoor approach exploiting horizon silhouettes to improve orientation precision. The system uses GPS to get a rough indication of position. The horizon silhouette is then extracted from the video stream using the Sobel operator to detect the edges. This silhouette is then compared with a predicted silhouette generated from a digital elevation map based on the position reported by the GPS. The method does not work under unfavourable conditions such as when the horizon is not visible, due to

fog, or if there is no silhouette, such as when in the desert or out at sea. It also has problems if there are occlusions of the horizon.

Model-based Tracking

In another project, the RSC group looked at using visual cues from buildings in an urban environment to aid in registration [SB98b]. The technique is based on the visual servoing approach presented by Espiau *et al.* [ECR92]. The method involves matching the current view of a building to an existing Computer Aided Design (CAD) model. The algorithm works by minimising the error in the forward projection of points from the CAD model with 2D Kanade-Lucas-Tomasi (KLT) features [TK91] extracted from the live view of a video camera. Once matches are established, the camera motion producing the least amount of error is determined. Camera motion and pose estimates are provided by the method. They identify that improvements need to be made on robust feature tracking, and that an approach which automatically adapts to viewpoint changes and variable lighting conditions is needed. The project was later updated by the group [BPS02].

A similar approach by Lepetit *et al.* makes use of a rough CAD model of parts of the scene to initialise the tracker [LVTF03]. The real-time tracker then uses information from preceding frames to eliminate jitter and drift. A small number of key-frames are also utilised in order to make the system more robust.

Klein and Drummond [KD03] also use a model-based tracking method combined with gyros to cope with fast head motions. The model used for tracking is also rendered to the screen so that the registration error can be easily assessed. Wuest *et al.* implemented a model based approach [WVS05] that extracts edges from the image and matches these with lines (edges) in the CAD model. The camera pose is computed based on the most likely line matches between the model and the extracted edges.

A difficulty with traditional model-based approaches is they require a model of the scene before the system can run; this severely limits their application to the few environments where such a model exists. Recent advances have combined the process of computing 2D-3D matches between a set of extracted features and points from a model, with the field of 3D reconstruction from images, or Structure from Motion (SfM). This allows the system to automatically build a 3D model of the scene that is to be augmented, eliminating the need for an existing model.

The group from K. U. Leuven, Belgium, uses an approach developed for scene reconstruction from uncalibrated monocular video sequences to create an AR system that works offline [CPVG01]. A motion and structure recovery algorithm is used to compute the scene structure as well as the motion of the camera throughout the video sequence. This algorithm uses the Harris corner detector [HS88] to locate features, which are tracked through the course of the video. This process takes advantage of the RANSAC algorithm to determine the best features to use for computing the pose of the camera. The virtual augmentation can be positioned inside the generated model so that it can be rendered throughout the entire video sequence thus obtaining an augmented video sequence.

A project closely related to this thesis is presented by Skrypnik and Lowe [SL04]. The authors present an approach to automatically compute a 3D model of the scene. The virtual object is then positioned in this model. The scene model is built from a set of images. Scale Invariant Feature Transform (SIFT) features [Low04] are extracted from each image and matches between each image are computed. Adjacent images are determined by constructing a spanning tree, where successive image pairs with the highest number of matches are incrementally added, while ensuring no cycles develop in the tree. 2D matches through the entire image sequence can then be computed at which time a global minimisation is sought using the LM algorithm. This step minimises the reprojection error from parameters such as the internal and external camera parameters, as well as the 3D scene points that the 2D multi-view matches represent. Reprojection error is measured as the error between the location of a 2D feature point and the 2D location of the corresponding 3D point projected through the current estimate of the camera model.

During live operation Skrypnik's system extracts SIFT features from the live image and matches these to the feature descriptors of the points that make up the model. These results provide 2D-3D matches like the model based systems previously discussed. The LM algorithm computes the live camera parameters by minimising the reprojection error between the 2D points extracted from the live image and their corresponding 3D scene points projected through the current estimate of the live camera. RANSAC is used at this stage to cope with outliers in the 2D-3D match set. This process is considerably faster than the previous minimisation as the only parameters which are unknown here are the internal and external camera parameters for the live view. Once the live camera parameters are

established the virtual object can be successfully rendered from the user’s viewpoint.

The limitations of these approaches is that they are unreliable in adverse imaging conditions. They also require 3D models of the scene before the system can be used and automatically computing 3D models is a computationally expensive task.

2.4 Issues with Computer Vision in the Outdoors

With systems like the one presented by Skrypnik and Lowe [SL04], AR systems built solely by tracking natural features are possible. These vision based systems could be combined with other sensors to form hybrids capable of achieving high registration accuracy.

Achieving high registration accuracy and good reliability when operating in the outdoors is a difficult problem. One difference noted by Azuma is that in comparison with indoor systems, outdoor environments cannot be controlled [AHIS99]. For example the lighting is controlled indoors but not outdoors. Natural feature tracking in particular can be difficult.

2.4.1 Why Natural Feature Tracking can be Difficult

Natural feature tracking can be difficult due to several factors. First, for an AR system to work anywhere it will need to operate in a range of scene types, from cities, to bush, to mountains, to beaches; all of which may have only a few of a given type of feature in one instance and many in another, thus one algorithm is unlikely to suit every situation.

Another source of difficulty is that the appearance of features can change from varying viewpoints, under varying lighting conditions which change throughout the day and between seasons, and under different weather conditions. Dynamic scene changes introduced by objects coming or going in the scene or objects changing form can also introduce changes the system must be robust to.

Narasimhan *et al.* list the changes that can affect the appearance of an outdoor scene as “viewing geometry, scene structure and reflectance (BRDF or BTF), illumination (sun, moon, stars, street lamps), atmospheric condition (clear air, fog, rain) and weathering (or aging) of materials.” Black adds to this list through what he terms iconic changes [BFY00]. These are changes specific to the appearance of a given scene or a particular object; examples given are “complex occlusion events and changes in the material properties of the objects.” Adini *et al.* discovered that

in the context of face recognition, the degree of variability in facial appearance due to changes in illumination, are the same as changes in viewpoint or facial expression [AMU97].

2.4.2 Improving Performance of Natural Feature Tracking

Improvements in the performance of computer vision systems can be made in several areas. The first of these is through general improvements which benefit systems operating in normal and adverse conditions.

General Improvements

As computer vision algorithms operate on images from a camera, the choice of camera can have a significant impact on the performance of the system. Parameters such as the picture quality, frame rate, the dynamic range, as well as the FOV of the lens all affect the final performance.

Strecker *et al.* investigated the influence of the choice of lens on SfM techniques [SESK05]. They found that in the context of AR, a fisheye lens yields superior performance for SfM compared with a normal perspective lens which typically has a FOV of only 40–50°. Other research in the field of SfM has revealed that a wider FOV can help to stabilise camera pose estimation [DCK04]. Following this idea Lee *et al.* constructed an outdoor AR system utilising an omni-directional (360°) camera [LYN02].

Other projects have sought to improve robustness by taking advantage of more information in the image, in particular by attempting to track multiple types of features. Vacchetti *et al.* combine two feature types in one system [VLF04]—both edges and point features are tracked. They state “as it can exploit more of the image information, it is more stable and less prone to drift than purely edge or feature-based ones.” In contrast, Neumann and You combine region tracking with optical flow based methods [NY98]. Simon and Berger take a different approach by implementing a model-based system which tracks points, lines and freeform curves [SB98a].

Systems which combine multiple computer vision algorithms are similar in spirit to hybrid AR systems that combine different types of sensors, in that the final system covers the weaknesses of any one approach and yields more robust systems. Future computer vision systems will likely combine a number of diverse vision algorithms simultaneously. This is especially true as computational power increases, reducing

the expense of such operations. By using multiple algorithms the best algorithm for a given scene type or a particular environmental condition can be selected; thereby utilising the complementary nature of many algorithms.

Mikolajczyk and Schmid [MS04] also identify the complementary nature between their algorithm, which extends the Harris interest point detector to be invariant to scale and affine transformations, and Lowe’s SIFT algorithm [Low04]. They state that “the scale invariant Harris-Laplace approach detects different regions than the Difference of Gaussian (DOG) detector. The latter one detects mainly blobs, whereas the Harris detector responds to corners and highly textured points, hence these detectors extract complementary features in images.”

Lowe also identifies that “the local feature approach can easily incorporate novel feature types because extra features contribute to robustness when they provide correct matches, but otherwise do little harm other than their cost of computation. Therefore, future systems are likely to combine many feature types” [Low04].

Advances in Feature Detection and Matching Algorithms

Improvements in feature detectors and matchers have also resulted in features more robust to changes in viewpoint and illumination. The initial concept of local features was introduced by Moravec’s corner detector [Mor81]. Harris and Stephens later improved upon this by handling small image shifts and being less prone to nearby edges [HS88]. The Harris corner detector has since been widely used.

Zhang *et al.* demonstrated matching Harris features using a correlation function on the region around a detected point in order to obtain an initial set of matches. A robust technique was then used to discard false matches [ZDFL94].

Harris corner features however, are sensitive to image scaling, therefore more recent algorithms have sought to improve invariance to scale changes between images. The common approach utilises a pyramid of scaled images, a concept introduced by Crowley and Parker [CP83], where feature detection occurs at each scale. Rotation invariance is an important addition, allowing local features to be invariant to rotational differences.

Approaches such as SIFT [Low04] have produced more distinctive feature descriptors, while being more robust to viewpoint and uniform illumination changes. This robustness is due to the use of local gradient data, rather than image intensity data in the descriptor image patch. Ke and Sukthankar extended the SIFT

algorithm with PCA-SIFT [KS04]. PCA-SIFT improves upon SIFT descriptors by applying Principle Component Analysis (PCA) to the image patch. This results in a descriptor that is more distinctive and compact. This brings improvements in matching speed, at the expense of requiring a training phase. Other approaches have produced affine invariant features, which are features more robust to viewpoint changes induced by camera rotation or transformations of planar scenes [MS04, MCMP02, TG04].

These advances have resulted in features more robust to scale, rotation, and viewpoint changes, as well as descriptors more invariant to illumination change. However, tests conducted generally evaluate the algorithms under uniform changes in illumination, an assumption violated in most scenes due to light sources interacting with scene structure producing non-uniform illumination changes such as shadows and reflections. For this reason additional methods must be used to ensure robust performance in adverse conditions.

Improvements in Dealing with Adverse Conditions

One of the main factors which influences the performance of a computer vision system is the images themselves. The quality of the images and the conditions under which they are taken have a direct bearing on the performance of the system.

Numerous researchers have taken a number of diverse approaches to dealing with a variety of different environmental effects such as haze, fog and mist. Narasimhan and Nayar discuss an interactive method for adding and removing weather affects from images [NN03b]. They identify that images taken under bad weather conditions such as rain, snow, fog and mist suffer from poor contrast and corrupted colour. In addition, “the radiance from a scene point is significantly altered due to atmospheric scattering” where “the amount of scattering depends on the distances of scene points from the observer.” The method takes advantage of previous work by the authors in the area of removing weather effects from images. Narasimhan and Nayar have also performed extensive work on ‘deweathering’ images [NN03a, NN02]. This process uses multiple images of the same scene taken under differing weather conditions to break the ambiguities in deweathering, thereby eliminating the effects of fog and haze. Schechner *et al.* developed a similar approach [SNN01] which takes advantage of multiple images captured by varying the optics of the camera. The approach uses a small amount of input from a user to deweather the images.

Oakley and Satherley also present a method of improving image quality through the removal of fog and haze in [OS98]. Their method relies on knowledge of scene geometry and a physical model for atmospheric conditions. They note that ‘atmospheric aerosols’ such as fog, haze or mist progressively reduce image contrast with increasing distance and that traditional methods of contrast enhancement such as histogram equalisation are insufficient to account for such losses. Histogram equalisation applies uniform changes globally across the image, whereas contrast changes due to atmospheric aerosols degrade quality based on distance. Therefore, any contrast adjustment needs to take this into account in order to be successful.

Garg and Nayer provide an analysis into the visual effects of rain [GN07]. They find that “rain produces sharp intensity changes in images and videos that can severely impair the performance of outdoor vision systems.” A pose-processing algorithm was developed containing a model to describe intensity changes produced by rain streaks, allowing them to be successfully removed from videos.

Clark *et al.* considers the presence of Gaussian noise in images, and seeks to improve previous approaches which only utilise one image by taking advantage of all the frames available in a video sequence. The problem of blurring in images is also evaluated utilising current approaches to blur removal. These techniques require an initial estimate of motion blur direction; Clark *et al.* estimate this by applying optical flow techniques between frames of the video sequence.

Illumination is a strong factor in determining computer vision performance. Even in fiducial based systems which typically operate indoors, little attention has been paid to the effects of changing lighting conditions. Naimark and Foxlin identify this [NF02], stating that other techniques use colour segmentation or simple thresholding, which are prone to failure under variable illumination. Their approach uses a homomorphic image processing algorithm which is designed to eliminate the effect of non-uniform illumination in images by modelling the grayscale values as a product of reflectance and illumination parameters.

Progress has been made in reducing the effects of weather on images. However, current approaches rely on known scene geometry, models of the atmospheric conditions, or user intervention. Additional research is needed to further analyse and compensate for the multitude of possible effects.

2.5 Assessing the Robustness of Computer Vision

In order to assess how robust a given computer vision algorithm is, two things are needed: data to test on and a method of testing to establish the algorithm’s performance under various imaging conditions. Typically most researchers use both standard data sets and standard performance measures when assessing the robustness of computer vision systems. Standardised data and performance metrics (despite being informal standards) allow comparisons between the results from different researchers. Furthermore, public availability of standard data sets eliminates the need for each researcher to capture test data, as capturing suitable test data can be a time consuming task.

In addition to reviewing the test data used in this thesis, the “Related Databases” section of Chapter 4 (Section 4.1), reviews several other image databases intended for testing computer vision algorithms. In particular, databases that provide images under a range of weather or illumination conditions are examined. Furthermore, the background section of Chapter 5, Section 5.2 presents a review of work in assessing the performance and robustness of computer vision algorithms.

2.6 Summary

Achieving accurate registration in the outdoors is still a challenging topic and is considerably more difficult than indoors due to the unpredictability and uncontrolled aspects of the environment [ALJ⁺99]. Many technologies exist to provide the pose information needed to obtain registration including GPS, magnetic, inertial and dead reckoning sensors as well as many different vision-based algorithms. Unfortunately no single tracking technology exists which can solely provide the robustness necessary for the large range of potential applications. For this reason hybrids are used which combine multiple sensor types into one system. One type of sensor frequently used is computer vision.

Vision based trackers have proven useful due to their low cost and the ability to obtain accurate registration in both indoor and outdoor systems. They also benefit from tracking features in the same images that will later be augmented [SB02]. Many indoor systems make use of fiducial markers which greatly simplify determination of camera pose and provide accurate and robust results. However, in outdoor environments the scale, range, and possibly sensitive nature of the environment

[VIK⁺02] prohibit the addition or alteration of the scene with markers. For these reasons the use of natural features is desirable in many applications. These features include: edges [Can86, HSSB96, ZT98], corners [HS88], points [TK91, ST94, Low04], curves [Ber94], or planes [SFZ00]. Research into new feature extraction algorithms has yielded increasingly robust features.

Previous AR projects utilising natural features have either taken advantage of the specialised nature of some applications, for example using planar structures when they are visible (a reasonable assumption for the most part), or using existing CAD models of the scene for matching. The practicality of this approach is limited, as few scenes have CAD models readily available. Hence more recent projects have incorporated this technique with that of the SfM field which enables a 3D reconstruction of the scene to be computed solely from a sequence of images and provides the camera pose for each image. This provides an easy means of preparing the scene, because the virtual object can be positioned inside a model of the scene.

Now that accurate systems exist for outdoor AR, more emphasis needs to be placed on evaluating how these systems perform when placed into environments which are less than ideal—an assumption made by most existing projects. One of the desired outcomes of AR is the ability to work anywhere and at anytime. If this is to be possible when using computer vision, the vision system must be able to operate in any type of scene. Most AR research systems, particularly markerless outdoor systems, use urban scenes to test on, where there are clear high-contrasting features that can be more easily tracked. As accuracy improves, the robustness of the system under more varying scene types can be evaluated, including beach scenes, rural scenes, and mountains.

In addition, robustness to changing environmental conditions must also be assessed and improved upon. Changes due to illumination as well as adverse conditions such as fog, rain, and snow, affect early stages of the computer vision system, namely the feature detection and matching stage which utilises image data. Later stages are independent of the image data and thus not affected.

Recent research has gained a greater understanding of the effects of variable environmental conditions, with methods to deal with image degradation caused by atmospheric changes induced by fog, haze, mist being addressed [NN03a, NN02, NN03b, SNN01, OS98].

Further improvements can be made on several fronts. Continued research

into more robust feature detection and matching methods will yield increases in performance in every situation, or at the very least an increased level of confidence in the features. More research into dealing with weather conditions is needed in order to provide methods capable of operating on single images, or between frames from a video or camera, without user intervention. Mitigating the effects of non-uniform illumination changes also needs further investigation.

Additional research into how different yet complementary vision algorithms can be combined may yield systems able to operate in a more diverse range of environments. In addition, approaches that minimise the amount of preparation required may make these systems more easy to use and widespread.

This thesis deals primarily with improving robustness in the case of illumination variation by analysing then mitigating some of its effects. In this thesis the performance of the computer vision system is considered without any other sensors assisting in obtaining registration. It is envisaged that in a real AR system it would be combined with other sensors to form a hybrid solution, however, here it is considered alone so that the vision performance can be assessed in isolation. Furthermore, it is examined without the assistance of any other sensors so that the results are directly applicable to other applications of computer vision such as surveillance or 3D reconstruction, which typically do not make use of hybrids. The following chapter details two reference AR systems implemented during the course of this project.

Chapter 3

AR Implementation

“Good things come in pairs”

Chinese saying

In order to evaluate the performance of computer vision based AR systems in adverse imaging conditions, demonstration systems are needed for testing. For this purpose, two different approaches were implemented during the course of this project. These systems are called the “2D System” and the “3D System”. These names reflect the abilities of the underlying algorithms. The 2D system utilises an algorithm that determines the registration between two images. The estimated image differences account for only four of the six degrees of freedom needed to express both movement and rotation about all three axes in the world co-ordinate frame. The registration algorithm cannot distinguish between movements along the world coordinate frame’s X and Y axes and pitch and yaw rotations. Therefore, the 2D system must be adapted to achieve 6DOF operation. In contrast, the 3D system utilises local feature points detected in images of the scene, and obtains a geometric understanding of the scene’s structure. This enables it to obtain full native 6DOF operation.

Implementing these systems is an important step to both understand and demonstrate the role computer vision algorithms play in AR systems. This gives a clearer picture of the effects and implications adverse environmental conditions have on AR systems, and more specifically which parts of the computer vision process are impacted the most. It is in the context of these systems that Chapter 5 analyses the performance of the stages directly influenced by adverse imaging conditions.

In Section 3.1 of this chapter the 2D system is described including the image registration technique, and the limitations of the system. In Section 3.2 a more contemporary approach is described, making use of local features to build a 3D model of the scene and then tracking the user’s location with respect to it.

3.1 2D System

The 2D system implemented utilises a Fourier-based algorithm to determine registration between two images. The algorithm recovers translation, scale and rotation differences between a reference image of the scene and the current frame from a live video feed. It is based on the approach described by Vlahakis *et al.* [VIK⁺02]. Vlahakis *et al.* present a system called the “Augmented Reality Based Cultural Heritage On-Site Guide” or “ARCHEOGUIDE” for short; an AR guide for archaeological sites.



Figure 3.1: ARCHEOGUIDE augmentations. Image on the left is before augmentation. Image shown on the right is the augmentation as seen through the ARCHEOGUIDE system

ARCHEOGUIDE allows regions of interest around an archeological site to be augmented with reconstructions of ruined buildings or relevant information designed to help recreate the site as it might have once been. DGPS and a digital compass are combined with an image tracking system comprised of a Fourier-based technique described by Reddy and Chatterji [RC96]. The DGPS and digital compass are used to provide rough position and orientation estimates while the Fourier-based algorithm refines these measurements. Figure 3.1 shows images from ARCHEOGUIDE of the Temple of Hera before and after augmentation.

The Fourier-based algorithm is only a 2D approach, recovering differences between two images. This means the algorithm recovers rotation around the optical

axis (roll), translation along the X and Y axes of the image and limited scale values. Therefore, the system can only track under restricted camera motions and from predefined viewing locations for which reference images are available. As a consequence, the user can only move within a limited range of these predefined locations, determined experimentally as a 2m radius. Movement any further than 2m and the Fourier method fails to register as the images from the live feed are too different from the reference images. Vlahakis *et al.* claim that using this approach the system can cope with occlusions of up to 60%.

The Fourier-based algorithm was chosen as the basis of the 2D AR system implemented as part of this thesis, due to the reported robustness of the technique to changes between the live and reference images, particularly with regard to illumination changes. Vlahakis *et al.* list these changes as “new objects, or visitors may be present in the scene, or new lighting conditions (due, for example, to shadows or highlights in the images).” The authors also note their preference towards global algorithms stating: “we prefer to use algorithms that exploit global rather than local image properties.” Lowe independently identifies that the phase rather than magnitude data provided by a Fourier transform is “likely to provide improved invariance to illumination” [Low04].

The implementation of the Fourier-based algorithm used for the 2D system is provided by Cideciyan *et al.* [CJK⁺92] and closely follows that described by Reddy and Chatterji [RC96]. This implementation was used as the basis of the 2D system. As a novel contribution of this thesis the system was extended to support a larger range of viewing locations than possible in the ARCHEOGUIDE system. A larger range of viewing locations allows users to move around more freely and extends the usefulness of the system. These modifications are discussed in detail in Section 3.1.2.

3.1.1 Fourier-Based Technique

The Fourier-based registration technique used by the 2D system and the ARCHEOGUIDE project, takes advantage of the Fourier shift theorem used to determine the translation difference between two images.

Translation

Two images I_1 and I_2 with a translation difference (x_0, y_0) are related by Equation (3.1).

$$I_2(x, y) = I_1(x - x_0, y - y_0) \quad (3.1)$$

The Fourier shift theorem describes the translation property of the Fourier transform. The theorem states that the Fourier magnitude data is invariant to translation differences between the two images; the translation only affects phase data.

Figures 3.2(a) and 3.2(b) show two images separated by a translation, where the second image is translated up and to the right. Figures 3.2(c) and 3.2(d) show the magnitude spectra of the Fourier transforms of these images. They demonstrate no translation difference, showing the invariance of the magnitude spectrum to translation changes. Note that these spectra are not identical, as the image data in the two images is not identical. Figures 3.2(e) and 3.2(f) show the phase data for these transforms. Phase data is not invariant to translations and therefore exhibits the same translation seen in the spatial domain.

The Fourier shift equation describes the relationship between the Fourier transforms of two images shifted with respect to each other. It states:

$$F_2(\xi, \eta) = e^{-j2\pi(\xi x_0 + \eta y_0)} F_1(\xi, \eta) \quad (3.2)$$

where F_1 and F_2 are the Fourier transforms of images I_1 and I_2 . This equation shows that only the phase data $e^{-j2\pi(\xi x_0 + \eta y_0)}$ is impacted by the translation (x_0, y_0) while the magnitude spectrum is unaffected.

By performing a correlation between the phase data sets for the two transforms, the translation difference between the two images is established. This process is called “phase correlation” and was originally described by Kuglin and Hines [KH75].

Phase correlation begins by windowing the input images. Fourier transforms assume periodic data, therefore sharp discontinuities produced by the edges of the image must be removed. This can be achieved by windowing the image so that the edges drop off smoothly. In this case a Butterworth filter was applied, however Hamming, Hanning or other windowing techniques could also be used.

The next step involves performing a Fourier transform on the images yielding results as *per* Equation (3.2). In this case a fast Hartley transform is used, which is closely related to the Fourier transform, but only operates on and produces real values.

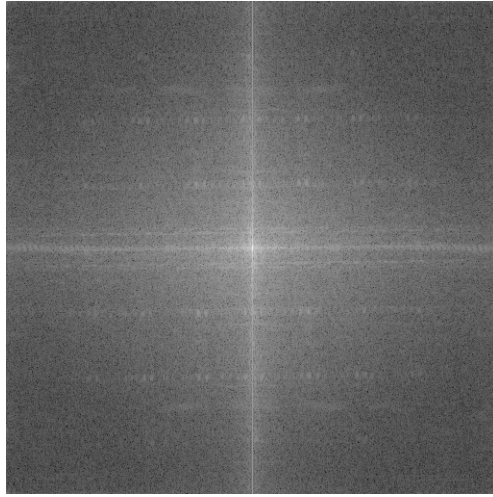
Next the cross-power spectrum of F_1 and F_2 is computed using Equation (3.3),



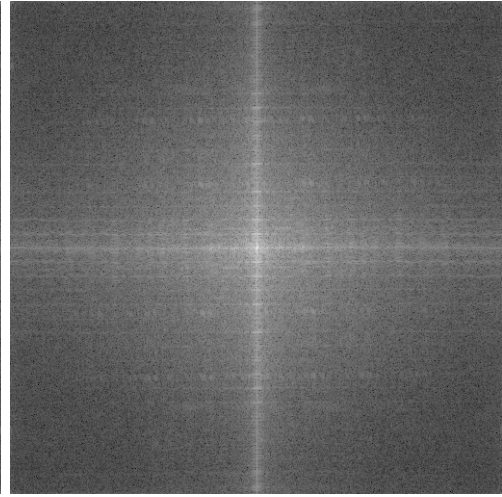
(a) Original Image



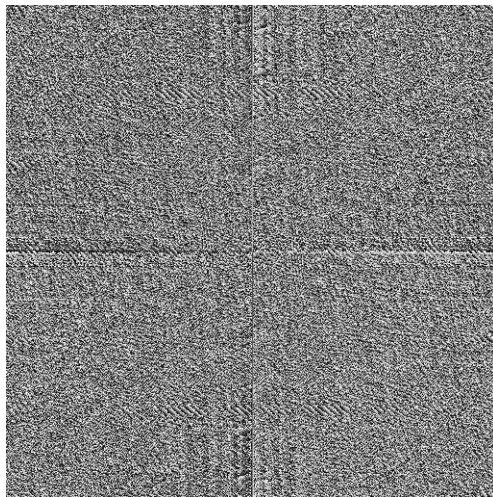
(b) Translated Image



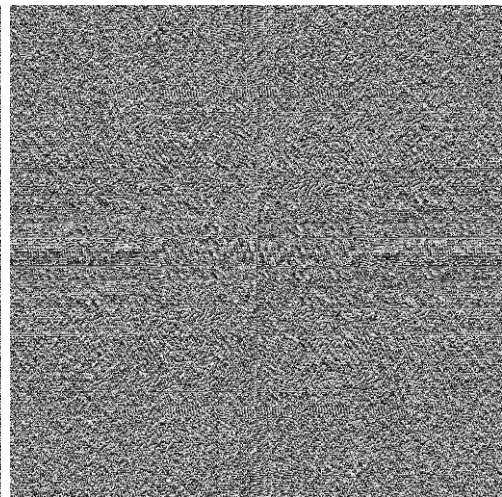
(c) Magnitude Spectra of Original



(d) Magnitude Spectra of Translated Image



(e) Phase Spectra of Original



(f) Phase Spectra of Translated Image

Figure 3.2: Images (a) and (b) show two images with a translation difference. (c) and (d) show the Fourier magnitude spectra of the two images, while (e) and (f) show the phase data

where F_2^* is the complex conjugate of F_2 . By applying an inverse Fourier transform to this result a peak at the position corresponding to the values (x_0, y_0) is obtained, where the magnitude of the peak and the value of the second closest peak can indicate the level of confidence in this result; the maximum would be a value of 1. In order for phase correlation to be successful an overlap of approximately 30% between the images is required.

$$\frac{F_1(\xi, \eta)F_2^*(\xi, \eta)}{|F_1(\xi, \eta)F_2^*(\xi, \eta)|} = e^{j2\pi(\xi x_0 + \eta y_0)} \quad (3.3)$$

Figures 3.2(a) and 3.2(b) show two images that have a translation difference between them of (107, 56) in the X and Y axis respectively. The algorithm computes this difference as (108, 55) respectively, with a correlation peak of 0.64.

Although magnitude data is invariant to translation, it is not invariant to the effects of rotation and scale; thus the presence of a small amount of rotation or scale will prevent successful computation of translation. In order to compute translational differences, the effects of rotation and scale changes must first be mitigated.

Rotation

A rotational difference between two images in the spatial domain produces the same rotation in the frequency domain. This effect is shown between Figures 3.3(a) and 3.3(b) which have a rotational difference, together with their frequency domain equivalents shown in Figures 3.3(c) and 3.3(d). Note the clear rotational difference in Figures 3.3(c) and 3.3(d), in contrast to Figures 3.2(c) and 3.2(d) which show similar images in the case of translation. Two images differing by a translation and rotation are related by:

$$I_2(x, y) = I_1(x \cos \theta + y \sin \theta - x_0, -x \sin \theta + y \cos \theta - y_0) \quad (3.4)$$

where I_1 and I_2 are the images, (x_0, y_0) is once again the translation, and θ is the rotation. Applying the Fourier transform to these images yields a result as shown in Equation (3.5). This equation shows that the translation affects the phase, while rotation affects the magnitude data.

$$F_2(\xi, \eta) = e^{-j2\pi(\xi x_0 + \eta y_0)} F_1(\xi \cos \theta + \eta \sin \theta, -\xi \sin \theta + \eta \cos \theta) \quad (3.5)$$

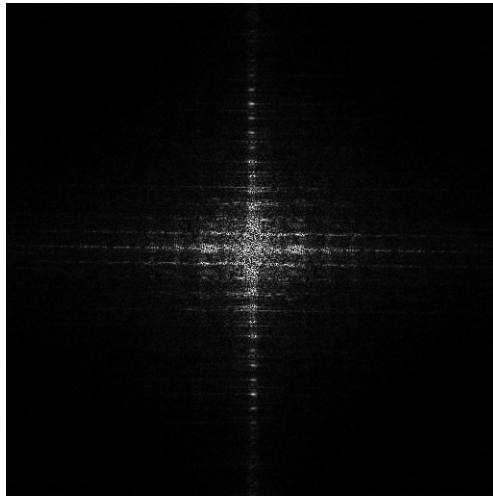
Given that rotation affects the magnitude data, the amount of rotation (θ) must



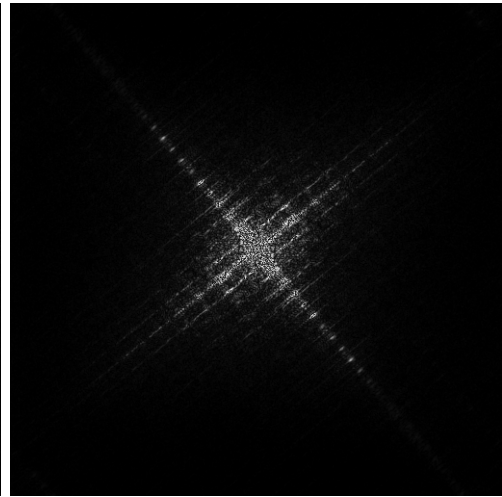
(a) Original Image



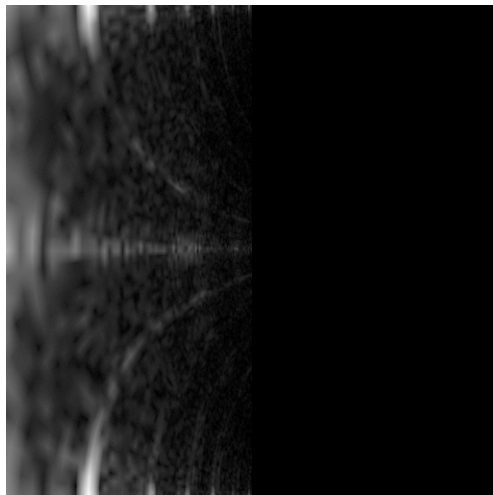
(b) Rotated Image



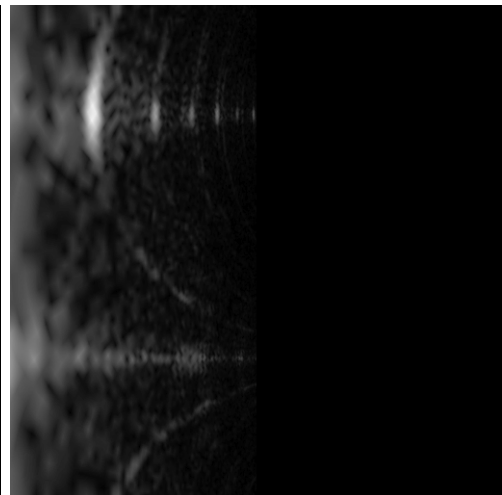
(c) Magnitude Spectra of Original Image



(d) Magnitude Spectra of Rotated Image



(e) Log-polar Magnitude Spectra of Original



(f) Log-polar Magnitude Spectra of Rotated

Figure 3.3: (a) shows the original image and (b) shows a version rotated by 40° . (c) and (d) show the respective magnitude spectra with the same rotation. (e) and (f) shows the log-polar versions of the magnitude spectrum images where the rotation is represented as a translation in the Y axis

be determined from the magnitude data. The relationship between the magnitude data for the Fourier transforms is given by:

$$M_2(\xi, \eta) = M_1(\xi \cos \theta + \eta \sin \theta, -\xi \sin \theta + \eta \cos \theta) \quad (3.6)$$

where M_1 and M_2 are the magnitude spectra for the respective Fourier transforms F_1 and F_2 . To compute the rotational difference, these magnitude images are converted into polar co-ordinates, where rotation is represented as a translation. The same phase correlation approach used for establishing translation differences can then be applied, effectively determining the rotation between the images.

Figure 3.3 shows two images, one rotated 40° clockwise from the other. The images show the Fourier rotation property at work. The Fourier magnitude spectra are the same except the Fourier spectrum of the second image F_2 is a rotated version of that of the first image F_1 , where the rotation difference is the same as that of the original images shown in Figure 3.3(a) and Figure 3.3(b). Figures 3.3(e) and 3.3(f) show the log-polar versions of the Fourier spectra which differ by a translation in the Y axis. The algorithm in this situation computes the rotation as $\theta = 40$, which is the correct value. This value has a correlation peak of 0.97.

Scale

A similar issue to that of rotation arises when a scale difference exists between the images such as the relation:

$$I_2(x, y) = I_1(\alpha x, \alpha y) \quad (3.7)$$

$$F_2(\xi, \eta) = \frac{1}{|\alpha|^2} F_1\left(\frac{\xi}{\alpha}, \frac{\eta}{\alpha}\right) \quad (3.8)$$

where α is the scaling factor. This produces Fourier spectra related according to Equation (3.8) where the scale change between the images is present between the magnitude spectra. Note that in this case an equal scaling of both the X and Y axis is assumed. In this case, representing the magnitude spectra in the log-domain can reduce the scale change to a translation, at which time the phase correlation can once again be performed. This yields a translation in the X axis corresponding to the scale change α between the images.

Figure 3.4 shows two images with a differing scale. The image in Figure 3.4(b)

has a scale difference from that of Figure 3.4(a) of $\alpha = 1.25$. Figure 3.4(c) and 3.4(d) show the Fourier spectra of these images and demonstrate the scale difference between the original images. Figures 3.4(e) and 3.4(f) show the log-polar versions of the spectra where the scale change is now represented as a translation change, computed correctly as 1.25.

Translation, Rotation and Scale

Computing all three parameters can be achieved by first converting the magnitude spectra of the Fourier transformed images into log-polar space. In log-polar space a translation along the X axis represents a scale change, while a translation on the Y axis represents a rotational change. This allows phase correlation to compute these parameters in one step. Because translation has no effect on the magnitude spectra, the rotation and scale can be isolated independently.

Once the rotation and scale have been computed, one of the original images can be rotated and scaled by the computed amounts. This reduces the difference between the images to the remaining translation difference, which can then be determined using phase correlation.

3.1.2 Using the Algorithm for AR

For the system to be used in AR, reference frames must first be captured. The number of reference frames required is dependent on the area the AR system should cover, however, each reference frame must overlap by at least 30° for the Fourier-based algorithm to be able to establish the registration. Once captured, these frames are then calibrated by specifying the pose of a virtual object inside the reference frame. Registration between each of these reference frames and a live frame is then computed using the Fourier-based algorithm. The results from the reference frame yielding the highest confidence is selected. Confidence in the Fourier-based method is based on the magnitude of the correlation peaks computed during operation. The registration information can then be combined with the knowledge of the virtual object's pose inside the reference frame to establish the pose of the object in the live image. This allows the virtual object to be correctly rendered in the user's current view.

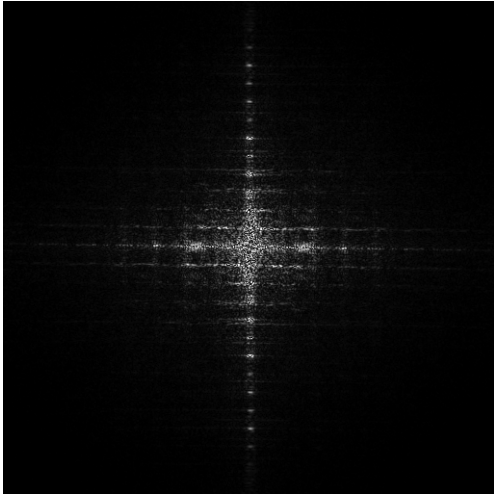
In the ARCHEOGUIDE system, DGPS and a digital compass are used to estimate which monument the user is looking at and returns the set of reference images for this



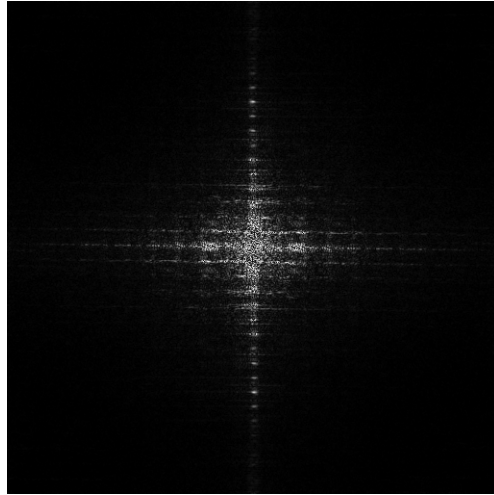
(a) Original Image



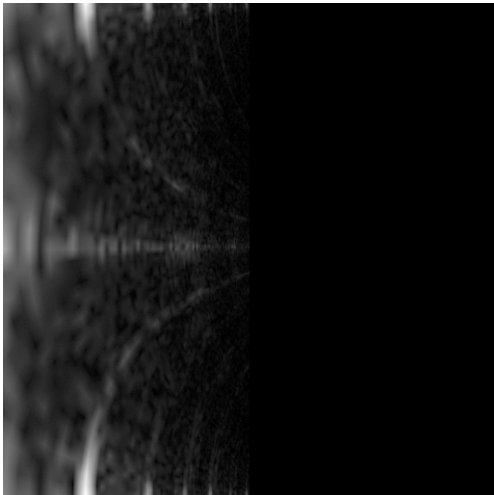
(b) Scaled Image



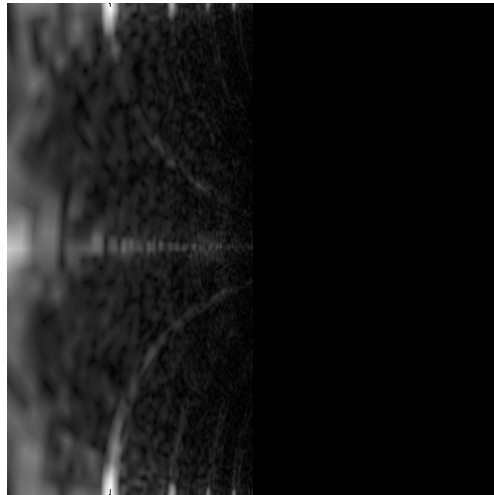
(c) Magnitude Spectra of Original Image



(d) Magnitude Spectra of Scaled Images



(e) Log-polar Magnitude Spectra of Original



(f) Log-polar Magnitude Spectra of Scaled

Figure 3.4: (a) shows the original image and (b) shows a version scaled by a factor of 1.25. (c) and (d) show the respective Fourier spectra with the same scaling visible. (e) and (f) show the log-polar versions of the Fourier spectrum images where the scaling is represented as a translation in the X axis

site. The image tracking system then compares the current view with all reference frames for this monument from the pre-defined location. The system typically stores between 5–10 reference frames for each monument in order to cover a sufficient FOV, as well as viewpoints within a 2m radius of the pre-defined location which gives the user a little freedom to move around. The reference frame yielding the highest correlation peak value, as calculated by the Fourier-based method, is determined to be the best match and the calibration information for the matching reference frame is used to render the virtual object.

The main problem with this approach is that this technique, by nature, only supports 4DOF. It can only recover translation differences between two images along the X and Y axes, as well as rotations around the optical axis and limited scaling. In the world coordinate frame translation differences between two images correspond to rotations about the X and Y axes. Rotations about the optical axis of an image correspond to rotations around the Z axis in the world coordinate frame, while scaling corresponds to a translation along the Z axis. This means the algorithm has no concept of movement along the world coordinate frame's X or Y axis, only rotation around them.

The limitations of this concept are shown in Figure 3.5. Figures 3.5(a) and 3.5(b) show two images captured by a camera rotating around a fixed position, along with a virtual crate rendered in the same relative position. When two images are captured from the same position but in a different direction, the images differ by a translation, and a 2D approach is capable of correctly estimating this difference. Given the virtual object's pose in the first image (*i.e.* given the crates pose in Figure 3.5(a)), establishing the translational difference between the images is enough to render the object correctly in the second image as is shown in Figure 3.5(b). In contrast, Figure 3.5(c) shows an image captured from a different position. Even if the 2D approach correctly established the image translation, because there is no concept of translation along the world frame's X and Y axes (*i.e.* no concept of position changes), the object is not rendered correctly. Because the position change can not be established the object's pose from this new location cannot be estimated, therefore it is rendered with an incorrect orientation as shown in Figure 3.5(c). Figure 3.5(d) shows how the object should be rendered in this image.

In order to account for these limitations and extend the ARCHEOGUIDE system to support 6DOF motion, a significantly larger number of reference images are required

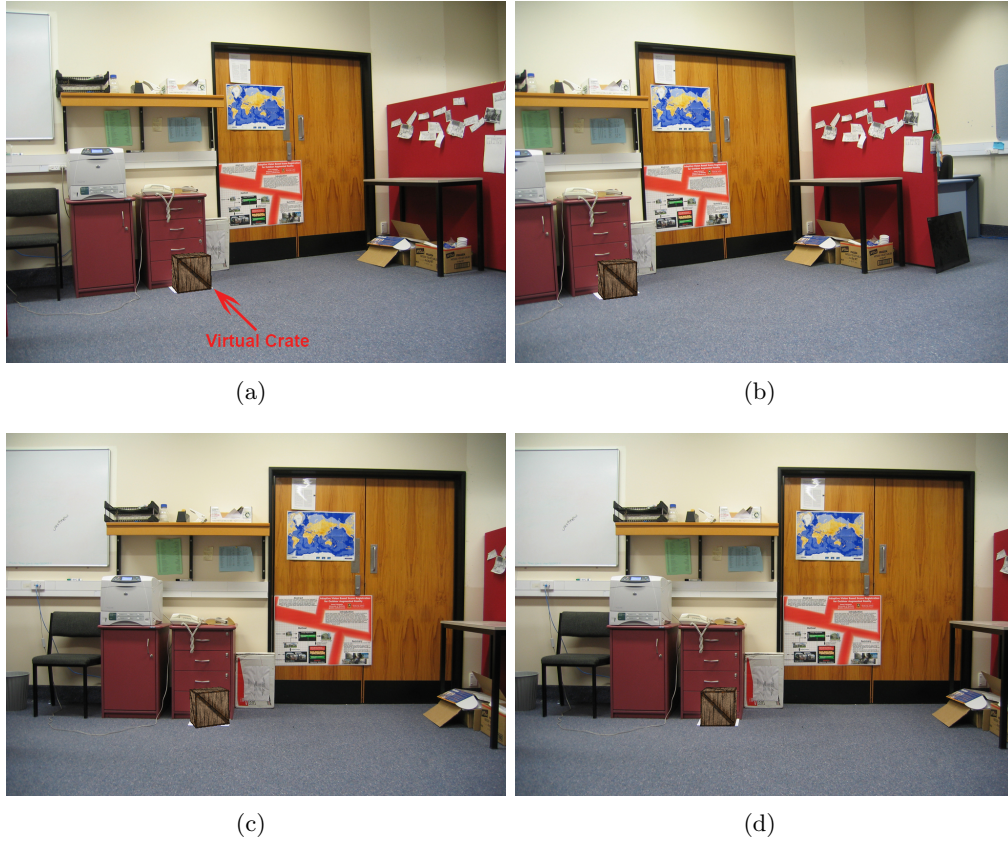


Figure 3.5: 2D System Limitations Conceptual Images

by the 2D system. Now, reference frames from many different positions are needed. A less restricted FOV than that imposed by ARCHEOGUIDE, also imposes the need for additional reference frames. The number of reference frames should be sufficient to cover the range of viewpoints so that a user can seamlessly move between them. Performing image registration is a computationally expensive operation, therefore the number of reference frames the algorithm is applied to at any given time should be minimised. For this reason an image library mechanism was constructed to store a large number of reference frames and return a small subset close to the position of the user as determined by an orientation sensor.

3.1.3 Image Library

The library stores the reference images based on the position and orientation they were captured at, and allows efficient retrieval of a small set of images, based on position and orientation estimates provided by the other sensors. The image library also takes into account the error associated with each tracker. For example, the orientation sensor used in this implementation was the Intersense IC2, which

although accurate, can, like all magnetic sensors, suffer from disturbances in the earth’s magnetic field from nearby metal objects that can significantly reduce its reliability. The image library returns all images with orientation values within a predefined distance of the actual measured orientation. This ensures the true closest reference frame from our initial reference set is included despite errors in the initial orientation estimate.

Upon receiving this initial reference set the image tracker determines the best matching frame and renders the virtual object accordingly. The use of this image library is an important distinction over the approach of ARCHEOGUIDE. ARCHEOGUIDE matches the live image to all reference frames from a given location. In contrast, the 2D system takes advantage of the orientation information to reduce the search space, resulting in less computation.

The image library was implemented using set structures to ensure that images near the position reported by the sensors could be located as quickly as possible. Each time an image is added to the library it is added to multiple set structures, one for each parameter (termed a parameter set), for example a yaw set and a pitch set. During live operation of the AR system these sets can be used to locate images close to the current readings from the tracking sensors. To do this, a subset of each parameter set is formed containing images with parameter values close to the one measured. For example, if the measured yaw was 45° a subset would be formed from all images in the yaw set containing images with yaw values between 40° - 50° . This filtering operation is performed for each parameter set using the supplied measurements. The intersection of these sets constitutes the final set where each image’s parameters are close to the current measurements provided. Each entry in the library also stores the transformation parameters required to position the virtual object correctly in the reference frame for the entry.

Having established a set of reference images captured with similar orientation values the Fourier-based algorithm is applied between each reference image and the live image. The reference image yielding the most confident result (*i.e.* the highest correlation peak) is considered the closest match. The transformation parameters for the virtual object in the closest matching reference frame are then combined with the computed difference between the reference image and the live image, effectively establishing the virtual object’s pose in the live image. The virtual object can then be rendered as seen from the live image’s viewpoint.

3.1.4 Problems

When implementing this system it quickly became apparent that there was a lot of preparation required before the system could be used. This was due to the large number of reference frames required. If the frames are separated by a small viewpoint change, the algorithm can correctly detect the closest reference frame to the image. However, depending on the distance to the virtual object, the augmentation may appear incorrectly rendered. This is due to the fact that the algorithm only recovers translation, optical axis rotation, and scale changes. When a user changes viewpoint, for instance strafing to the left or right maintaining the same orientation, a virtual object placed directly in front not only translates to the left, but also rotates a small amount in the same direction. This concept was described in Section 3.1.2 and demonstrated in Figure 3.5.

The only way to fix this is to have another reference frame at the new location, manually specifying the correct pose of the object. This means correctly rendering the virtual objects while supporting changes in position requires a significant number of reference frames for virtually every possible viewpoint. This becomes an exorbitant number for large scenes. In addition, the spacing between reference frames becomes smaller the closer the user gets to the virtual object. In the ARCHEOGUIDE system, the user is generally a reasonable distance from the augmentations, and the augmentations themselves are generally quite large. Consequently, it is more difficult to notice small errors in the virtual object’s orientation.

Problems with this approach are compounded due to issues associated with calibrating virtual objects inside reference frames. Any discrepancies in the calibration of the objects between reference frames—where there is a slight error in the relative pose of the object specified in each reference frame—manifests itself as a sudden discontinuity in the rendered pose of the object when the system switches between reference frames. This means between every reference frame the pose of the virtual object must be specified consistent with the actual change in pose between the cameras that took the images.

Lepetit *et al.* identify another problem inherent to key frame based approaches stating: “Key frame-based techniques prevent drift, but cannot provide good precision for every frame without using a very large set of key frames. Furthermore, they typically introduce jitter” [LVTF03]. Jitter is produced by small variations in the registration results computed between successive live frames. These variations

lead to discrepancies in the pose of the virtual object which, when rendered at live frame rates of 30 fps, make the virtual object appear to jitter. Iterative approaches eliminate jitter by establishing registration relative to the previous frame, but tend to drift as noted by Lepetit *et al.* [LVTF03]. Vlahakis *et al.* selected a key frame-based approach to avoid this drift, stating: “In real-time applications, however, abrupt and unpredictable motion makes sequential approaches uncertain and fragile.”

3.1.5 Summary

The invariance of phase data to illumination change is a desirable feature of the Fourier-based approach. The inherent 2D nature of this approach precludes its use for a 3D system as the number of reference frames required makes preparation too costly to support non-trivial changes in viewpoint.

The approach works well for the ARCHEOGUIDE application [VIK⁺02], and could likely be used in other similar situations where changes in viewpoint are not needed for success. The only way to use this algorithm to support full 3D operation would be to use accurate GPS (RTK GPS for instance). The GPS could provide information on position, with the image tracker effectively improving the accuracy of the orientation. Position information stored for each reference frame could then be used to correctly compute the pose of the virtual object given the position and orientation change between the reference and live video frame, allowing the object to be correctly rendered. Such a system could make use of the image library mechanism described in Section 3.1.3 to provide a small set of reference frames based on the current position and orientation provided by the GPS and orientation sensor.

Recent improvements in the accuracy of the Fourier-based registration method could also improve any systems relying on this approach. Phase correlation has been improved by Foroosh *et al.* [FZB02], where they extend phase correlation to support sub-pixel accuracy. Keller *et al.* improved on Reddy and Chatterji’s approach by utilising the pseudo-polar Fourier transform to estimate the polar and log-polar Fourier transform [KAI05]. This resulted in improvements in the accuracy, speed and robustness of the registration. An increased range of scale changes is supported, up to a factor of 4, in comparison to a factor of 2 under Reddy and Chatterji’s technique. Despite these improvements, however, the requirement for large numbers of reference frames still remains.

3.2 3D System

Given the limitations of the 2D approach presented, it became clear a native 3D approach was needed to achieve the level of flexibility required by an AR system and produce a system supporting 6DOF. The switch from a 2D to 3D approach involves a change from a global based technique to a local feature based approach. Global techniques use the whole image when determining changes between images. In contrast, local feature based approaches use specific points of interest located in an image in order to gain an understanding of the scene structure that forms the image.

The system presented in this section takes advantage of recent work in the field which is widely considered state-of-the-art. The system builds on those projects discussed in Chapter 2, Section 2.3.2. These went from somewhat restrictive systems which required an existing CAD model of the scene in order to track the user's pose, to systems which build their own models of the scene as a separate preparation step. What follows is an examination of the steps necessary in building a 3D system which constructs its own 3D model of the scene using natural feature tracking, and subsequently tracks the live view of the scene with respect to this model.

3.2.1 Overview

The approach used for this system is based on the work of Skrypnik and Lowe [SL04] and that of Hartley and Zisserman [HZ04]. The overall concept of building a 3D model from local features and the method by which the augmentation is specified with respect to the model is based on the work of Skrypnik and Lowe. The 3D system implemented in this thesis is novel in that Hartley and Zisserman's text book approach to 3D reconstruction from images is used rather than Skrypnik and Lowe's process.

The overall process is split into two phases. The first is building a 3D reconstruction of the scene and is only performed once for each scene to be augmented. This process is based on work in the field of SfM and projective geometry. The field of SfM deals with simultaneously building a 3D model and establishing the camera poses for each captured image for cameras whose calibration is known *a priori*. In contrast, projective geometry removes this assumption and deals with the case of an uncalibrated camera, where nothing is known about the internal camera parameters or the camera's pose. Camera calibration will be

discussed in detail later in this chapter.

The second phase is performed when a user is actually viewing the augmented scene. This involves determining the current camera’s pose (effectively the user’s pose) with respect to the 3D model previously computed. This process is based on previous work in AR, involving establishing the pose of the current camera utilising 2D-3D matches, as was done in projects utilising CAD models. An outline of the steps involved in each phase is shown in Table 3.1.

3D Reconstruction	Live System
1) Image Capture or Existing Images	1) Image Capture
2) Feature Extraction	2) Feature Extraction
3) Feature Matching	3) Feature Matching
4) Two-View Geometry	4) Pose Recovery
5) Three-View Geometry	
6) Camera Pose Recovery	
7) 3D Reconstruction	
8) Bundle Adjustment	

Table 3.1: 3D system algorithm outline. Steps on the left are those performed in the preparation stage, those on the right during live system operation

In testing, a sequence of images of the Arenberg Castle in Leuven, Belgium, were used. This dataset was used by Pollefeys *et al.* in their work on 3D reconstruction from uncalibrated images [PVG⁺04]. A selection of these images is shown in Figure 3.6.

3D reconstruction in itself is an important problem and is a requirement for many fields. 3D models of real scenes are used in the field of robotics for navigation, as well as computer generated graphics for movies and games, and for virtual and augmented reality. In the following sections the key steps involved in preparing the system are discussed, including how the 3D model of the scene is constructed, and then how virtual objects are positioned inside the scene. Many of the steps involved in the live system are identical to the 3D reconstruction stage, thus the discussion of many stages is applicable to both phases.

3.2.2 Feature Extraction

Feature extraction is an important step in both phases of this system. Feature extraction though often presented as one step actually constitutes two distinct parts: feature detection and feature description. Feature detection involves finding features or areas in an image which have the potential to be re-detected in another image of the same scene, possibly under different conditions. A good feature, for example,



Figure 3.6: A selection of the Arenberg castle images from the Leuven Castle sequence, used for 3D reconstruction testing

should be robust to viewpoint change and changes in appearance, allowing a feature to be re-detected from a different viewpoint.

The second step, feature description, is the process of describing these detected features. Determining that two features represent the same real world point can be established by a feature matcher. The matching process examines the descriptors in order to assess their similarity, thereby concluding whether they are a match, or not. A good descriptor is both distinctive to preclude the possibility of false matches with other similar features, and robust to viewpoint and environmental changes. This means the descriptors are similar even when a real world feature is viewed from a different location.

Many feature types exist (as discussed in Chapter 2, Section 2.4.2), each exploiting different properties of the image. Hartley and Zisserman [HZ04] make

use of Harris corner features [HS88] to perform 3D reconstruction. In the 3D system implementation, SIFT [Low04] features are used instead. As Lowe [Low04] describes, SIFT features are invariant to scale changes, robust to occlusion and illumination effects, and can handle a wide range of viewpoint changes. Skrypnik and Lowe [SL04] also make use of SIFT features in their implementation.

Other feature types could also be used here and may provide properties better suited to a particular scene type or environmental condition. For example, phase data was exploited by Carneiro and Jepson [CJ02] who then later improved the technique to support scale changes [CJ03]. They implemented an algorithm for extracting phased-based local features. They reported that this approach yielded improved performance when dealing with illumination changes, 2D rotation and sub-pixel translations. Other feature detection algorithms are explained in Chapter 5, however for the sake of simplicity only the SIFT algorithm is described here. An overview of the SIFT algorithm is now presented.

SIFT

SIFT features are detected by finding extrema points in a DOG pyramid structure. This structure is built from an image progressively smoothed using a Gaussian smoothing operator. Each smoothed image is subtracted from its neighbour producing difference images. This process is shown in Figure 3.7.

For each scale in the pyramid, the DOG image can be constructed using Equation (3.9). Here, $G(x, y, k\sigma)$ and $G(x, y, \sigma)$ are Gaussian images separated by a constant factor k , and are each convolved with image $I(x, y)$. These convolved images are subtracted from each other producing the DOG image $DOG(x, y, \sigma)$ for scale σ . Extrema points are then determined as the maxima or minima in a 3×3 region in neighbouring scales. This is shown in Figure 3.8.

$$DOG(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \quad (3.9)$$

Two properties of SIFT feature descriptors are that they are invariant to both rotation and scale. Scale invariance is achieved by a search for features at all scales. Rotation invariance can be achieved by computing the dominant orientation for each feature. To accomplish this, gradient magnitudes and orientations are pre-computed from pixel differences over the entire image. For a given pixel at position (x, y) in image L —which is an image convolved with a Gaussian such that $L(x, y, \sigma) =$

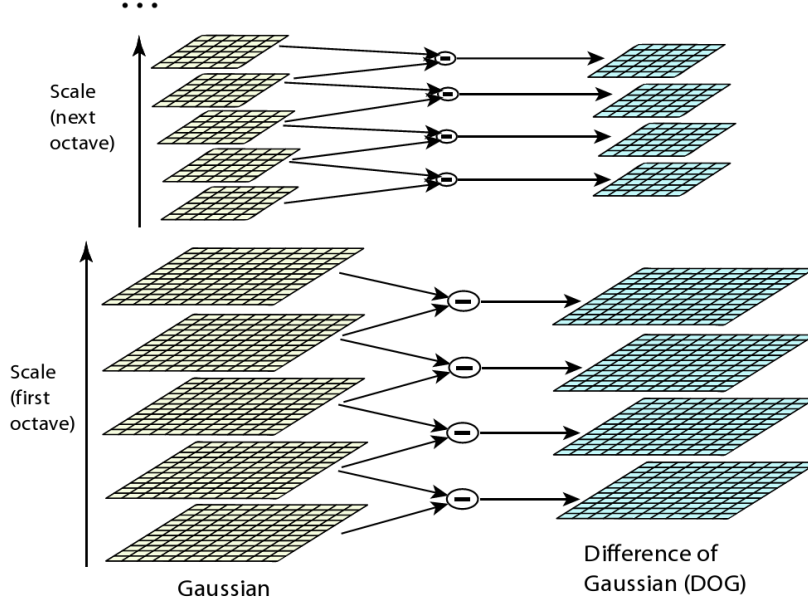


Figure 3.7: Squares on the left represent an image successively convolved with Gaussians producing increasing smoothing. Each successive octave represents the image down-sampled by a factor of 2. Squares on the right represent difference-of-gaussian images which are produced by subtracting the adjacent Gaussian smoothed images on the left. (Image courtesy of David Lowe – University of British Columbia. Reproduced from [Low04])

$G(x, y, \sigma) * I(x, y)$ —the magnitude $m(x, y)$ and orientation $\theta(x, y)$ are computed using Equations (3.10) and (3.11) respectively.

$$m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2} \quad (3.10)$$

$$\theta(x, y) = \tan^{-1}((L(x, y+1) - L(x, y-1)) / (L(x+1, y) - L(x-1, y))) \quad (3.11)$$

For a given feature a histogram is formed from the magnitudes and orientations computed in a 16×16 image patch around the feature's location. The histogram has 36 bins covering the range of possible orientations (in 10° increments). Each value is added to the bin corresponding to its computed orientation and is weighted by the calculated magnitude. In addition, each sample added is also weighted by a Gaussian-weighting window that increases the weight of samples in the middle of the image patch and reduces those on the outside. When there are discrepancies in point localisation—where a re-detected point is misaligned by a small amount—samples on the outskirts of the image patch are less likely to be in common between the descriptors, thus they produce more error in calculation of the orientation. To counter this, these samples have a diminished weighting, while samples near the centre—which are likely to be common—have a higher weighting. Peaks in the

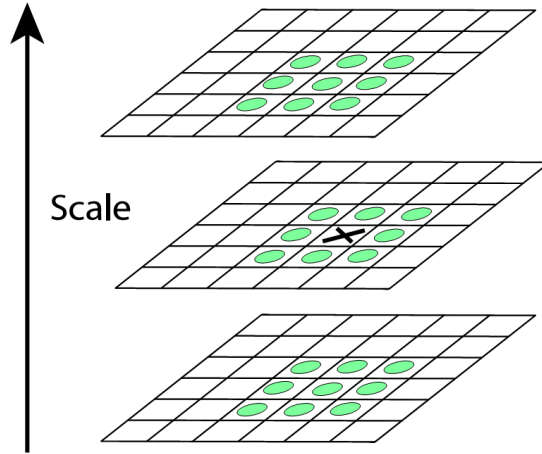


Figure 3.8: The pixel marked with an X is compared with its neighbouring points in the same scale image and adjacent scales and selected only if it is either the maximum or minimum of these values. (Image courtesy of David Lowe – University of British Columbia. Reproduced from [Low04])

orientation histogram indicate a dominant gradient orientation, thus the feature is formed using the orientation of the largest peak. Duplicate features are also created using the corresponding orientation for any peaks that lie within 80% of the highest peak.

Through knowledge of the feature’s orientation, a descriptor invariant to rotation can be formed. All that remains is to produce a descriptor designed to mitigate the remaining effects of illumination and viewpoint change as much as possible. To achieve this, a 16×16 region around the feature location is taken. The gradient magnitudes and orientations in this region are rotated relative to the feature’s computed orientation to achieve orientation invariance. 4×4 subregions are then used to form orientation histograms similar to those discussed for computing the feature orientation, however with only 8 bins. Again, the magnitudes for these bins are weighted with a Gaussian window to avoid small changes in the window location or orientation producing large changes in the descriptor. Small changes in the window also produce boundary effects, as samples abruptly change between adjacent histograms. To prevent this, an approach similar to trilinear interpolation is used to distribute the values from adjacent histogram bins.

This process is depicted in Figure 3.9. The square on the left depicts the image patch with gradient orientations shown for each sample. The direction of each arrow indicates its orientation while the length of the arrow shows its magnitude. The blue circle indicates the Gaussian weighted window applied to these samples. For the sake of simplicity the region shown is only 8×8 instead of 16×16 . The square

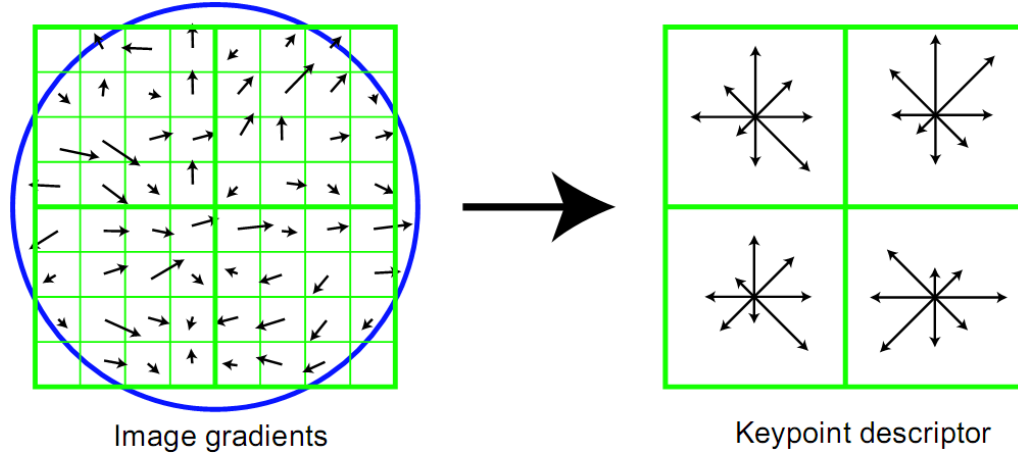


Figure 3.9: Image on the left shows a 8×8 image patch with computed orientations where the arrow lengths represent the magnitude. The image on the right represents the histograms built using 4×4 image patches from the gradients shown on the left. (Image courtesy of David Lowe – University of British Columbia)

on the right represents the histograms computed for each 4×4 subregion. The 8 arrows represent the 8 bins in each histogram while the length of each arrow shows the combined magnitude of all the samples in this subregion with an orientation in the range of this bin; each bin covers a 45° range.

Once the histograms are computed, a 128 element vector is constructed from the 16 histograms which each contain 8 bins. This vector is then normalised to unit length to reduce the effects of contrast change. Uniform brightness changes where a constant b is added to each pixel have no effect on the approach, as gradients are computed from pixel differences. Accommodations are made for non-linear changes in illumination by thresholding values in the vector to be no larger than 0.2, thereby putting more emphasis on the distribution of orientations instead of their magnitudes. This reduces the impact of camera saturation or surfaces under high illumination producing a lot of reflected light. The vector is then re-normalised.

Figure 3.10 shows a set of SIFT features detected in two images. The top images show the detected features and the dominant gradient detected for each feature. The size of the circle indicates the scale of the feature. The bottom images show the square patch used to construct the descriptor, the vectors shown in each 4×4 subregion indicate the dominant gradient detected in the 4×4 region of samples.

3.2.3 Feature Matching

Having computed descriptors for every feature detected in an image, putative matches between another set of SIFT features can be established. The word putative

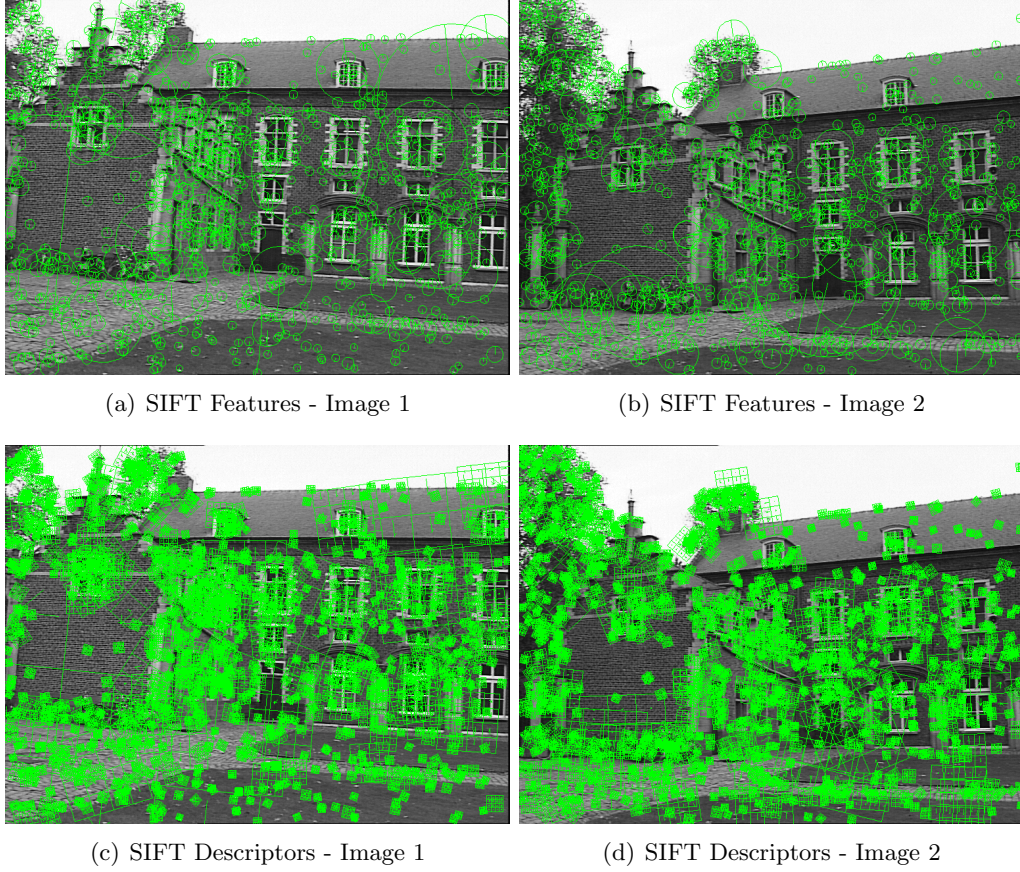


Figure 3.10: SIFT features extracted from images 1 & 2. Only 1000 features are shown for each of these images from a total of 5553 features from image 1 and 5316 from image 2. (a) and (b) show the locations of the features as well as their scale. (c) and (d) show the location of the SIFT descriptors and the area the descriptors describe

describes the fact that these matches are only true based on the inconclusive evidence obtained so far.

In order to compute putative matches for features utilising SIFT descriptors, a given feature from one image is compared with every feature extracted from another. The closest and second closest match are established for the given feature, where the metric of ‘closeness’ is the Euclidean distance between the two 128 element descriptor vectors. The ratio of the closest to second-closest distance is used as a measure of confidence in a given match. If the difference between the closest and second-closest match is too small, it indicates a level of uncertainty in the match, and therefore the match is dropped. This relation is described by:

$$\frac{dist(D_1 - D_2)}{dist(D_1 - D_3)} < t \quad (3.12)$$

where $dist$ is the distance function used, in this case the Euclidean distance between

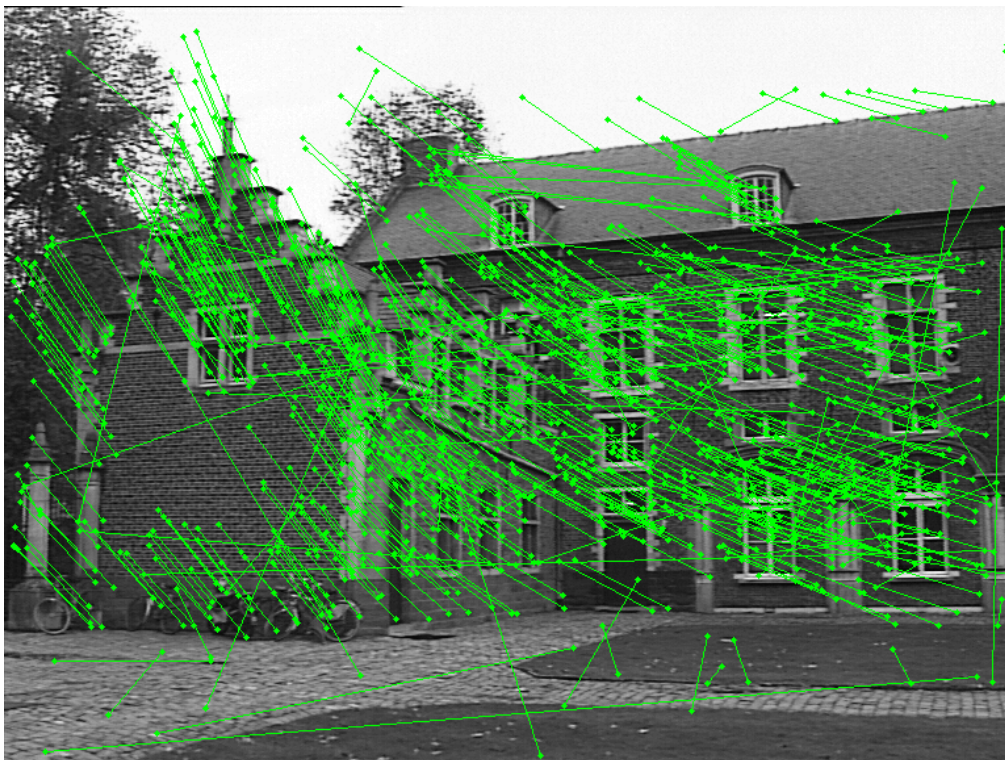


Figure 3.11: Putative matches between images 1 and 2. For the sake of clarity only 750 of the 1531 putative matches are shown

descriptors D_1 and D_2 . A ratio is computed between this distance and the distance between D_1 and D_3 where D_3 is the second closest match to D_1 in the second image, and D_2 is the closest. t is the threshold value of which the ratio test must be less than for D_2 to be considered a match.

Figure 3.11 shows a selection of putative matches computed using the features extracted in Figure 3.10. The image shows the displacement of the features between the two views. It can be seen that there are a number of false matches. These are identified as lines in the image which do not follow the pattern of the bulk of the lines. Matches in this image produce lines of roughly equal length which go up and to the left. Some example incorrect putative matches have lines which instead go across the image and are a lot longer.

K-D Tree

Performing 128 element Euclidean distance calculations on every feature is an $O(n^2)$ problem for n features. When large feature sets are used (for object recognition for instance), or when matching one image to many other images, the complexity of the matching process makes determining matches in this way infeasible in a reasonable amount of time. For this reason Lowe investigated the use of an approximate

Euclidean distance calculation algorithm called “Best-Bin-First” [Low04]. This approach searches for the closest match for a given feature in a set of features stored in a K-D tree. This reduces the complexity of the matching process to an $O(n \log n)$ problem and Lowe reports that this technique yields a speedup of 2 orders of magnitude when using a database of 100,000 features, yet only results in a 5% loss in the number of correct matches.

This approach was used in Skrypnik and Lowe’s AR implementation [SL04] to assist in establishing images which were spatially adjacent [SL04]. In order to get a rough idea of image position—*i.e.* to know the most promising image pairs to compute the epipolar geometry for—each image was compared with all other images in the set, and pairs with the highest number of putative matches were determined as being potentially adjacent. Matching between every image necessitated the use of the best-bin-first approach, otherwise computation would take too long.

The Leuven castle images, however, are in sequence—just as a video is—meaning spatial adjacency can already be assumed by image adjacency. For this reason image matching need only be computed between adjacent images rather than between every image in the set. This means use of an approximate technique for Euclidean distance calculations is not strictly necessary as these can be performed in reasonable time. In addition, time constraints are not so severe in the 3D reconstruction phase of the system, as typically the images have all been captured *a priori* and the system is only run once offline. The biggest benefit of the Euclidean distance approximation comes during live operation, where the features extracted from the current image are matched to a 3D model comprised of a potentially large number of features. In addition, live operation is when performance is most crucial, as the matching operation must be performed 30 times *per* second.

Although the K-D tree approach to matching was tested, it was deemed as unnecessary for the implemented 3D system for two reasons. The first reason is that the implementation of the 3D system was to validate the local feature based approach rather than to perform any type of user testing. Therefore ensuring real-time implementation—even during live operation—was not critical. The second reason is that using the approximate Euclidean distance approach would introduce bias into the test results, as feature matching may fail due to the approximate nature of the best-bin-first technique rather than any deficiencies in the descriptors or the detectors. For more information on the performance and accuracy of the K-D tree

approach see Lowe’s paper [Low04].

3.2.4 Two-View Geometry

Once putative matches have been established between every image pair in the set, the geometric relationships between the images can be established. In the case where images are separated from each other by a viewpoint change, the relationship between any two adjacent images can be described by their epipolar geometry.

Epipolar Geometry

Epipolar geometry is completely independent of scene structure and describes the geometric relationship between two views that is only dependent on the internal parameters of the cameras and their relative poses. In the fields of stereo vision and SfM, epipolar geometry can be used to constrain the matching problem between two views.

Figure 3.12 depicts the epipolar geometry between two views. In Figure 3.12(a) the planes on the left and right are the image planes taken by cameras with centres C and C' . Points x and x' are the images of the scene point X seen by cameras C and C' respectively. Figure 3.12(b) shows the epipolar line l' for point x . This line can be imagined as the view in the second image of the ray cast from C through x . Since X must lie on this ray it follows that its image in the second view must also lie on the image of this ray. Hence the epipolar constraint states that a matching point x' must lie on the epipolar line l' . The location of x' along l' is dependent on the location of X along the ray cast from C through x and represents the depth of scene point X which is unknown. Regardless of the true depth of X the epipolar line l' is still the same.

Figure 3.13(a) also shows the line connecting C and C' ; called the baseline. The epipoles e and e' are the intersections of the baseline with the image planes, while the intersection of the epipolar plane with the image planes forms the epipolar lines l and l' . If a ray was cast from the camera of the first image C through the epipole e it would reach the camera in the second image C' , meaning e is the image of C' with respect to C , or more generally the epipoles represent the image of the other camera’s location in the current image plane.

Figure 3.13(b) shows that the baseline is common between all epipolar planes produced by different scene points. This commonality is not surprising given that

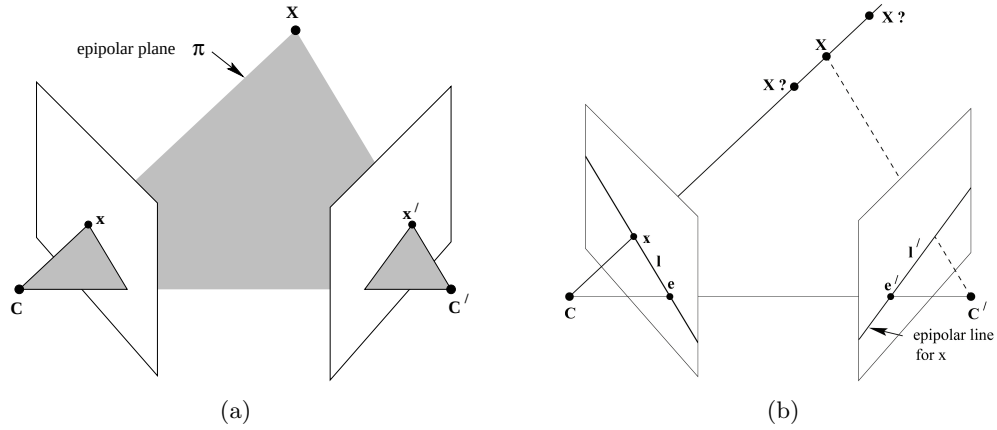


Figure 3.12: (a) shows the centres of two cameras C and C' , the 3D point X currently being imaged and the epipolar plane π formed by these points. x and x' are the images of X in the two views. (b) shows a ray cast from the camera centre C in the first image through x , the actual 3D point X lies somewhere along this line. l' represents the ray cast from C through x . e and e' are the epipoles and represent the other camera as seen in the current view. (Image courtesy of Richard Hartley and Andrew Zisserman – Australian National University and University of Oxford. Images reproduced from [HZ04])

the location of the cameras remains constant for different scene points and that each epipolar plane is formed using the baseline. This also means the epipole locations (*i.e.* the view of the other camera on a given image plane) are also constant for all epipolar planes.

Epipolar geometry does not provide the exact location of a matching feature in the second image. The location for a match x' to a feature x , is constrained to lie on a line in the second image rather than anywhere in the entire image. This reduction provides a greatly reduced search space when searching for matches, or the ability to remove matches which do not obey this constraint, *i.e.* when x' does not lie on the epipolar line generated by x in the first image. The only false matches that remain after this filtering are those which happen to fall on epipolar lines.

Fundamental Matrix

The fundamental matrix is described by Hartley and Zisserman as “the algebraic representation of epipolar geometry” [HZ04]. In essence it encapsulates the constraints imposed by projective epipolar geometry on a scene and stores all the information on camera motion and the camera’s internal parameters.

The fundamental matrix F is computed between two images using only putative image correspondences. RANSAC is used to find a fundamental matrix model

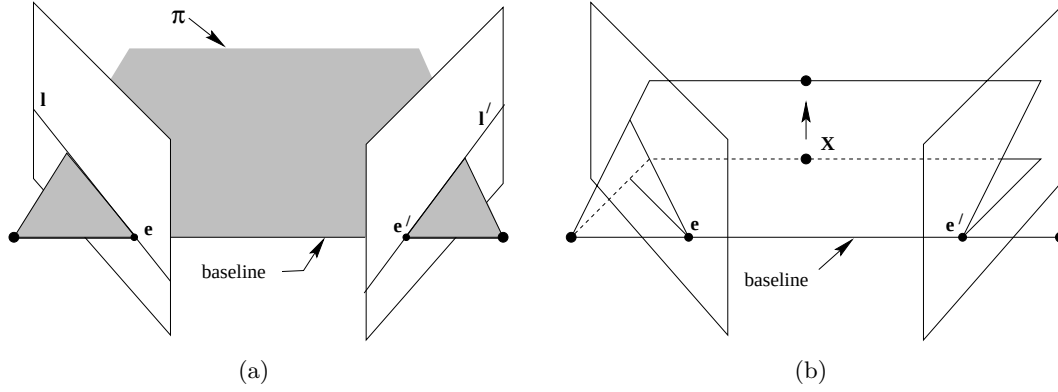


Figure 3.13: (a) shows the baseline connecting the two cameras. The baseline intersects the image planes at points e and e' . The intersection of the epipolar plane π with the image planes forms the epipolar lines l and l' respectively. (b) shows the effect of a change in the epipolar plane orientation. (Images courtesy of Richard Hartley and Andrew Zisserman – Australian National University and University of Oxford. Images reproduced from [HZ04])

yielding the most inliers; that is the largest number of matches consistent with the computed fundamental matrix. RANSAC takes samples of seven points from the putative match set, and computes the F matrix for each sample. Each matrix is assessed based on how many of the points conform to this computed model. RANSAC searches through the sample space randomly to find the model with the most support. A minimum of seven points is required to compute an F matrix. Outliers will be randomly scattered and generally do not conform to a valid model of geometry, therefore the correct model can be successfully established in the presence of a large percentage of outliers. RANSAC can even cope with a percentage of outliers greater than 50%. This approach looks for a solution representative of the inliers in the set, and is only confused when outliers correspond to an incorrect yet feasible model.

With knowledge of the fundamental matrix, the epipolar constraints can be applied to putative matches. In order for two features to be considered a match, the point in the second image must lie on the epipolar line formed using the point in the first image and the fundamental matrix, as shown in Equation (3.13).

$$l' = Fx \quad (3.13)$$

$$l = F^T x' \quad (3.14)$$

The opposite can also be computed, *i.e.* the epipolar line in the first image

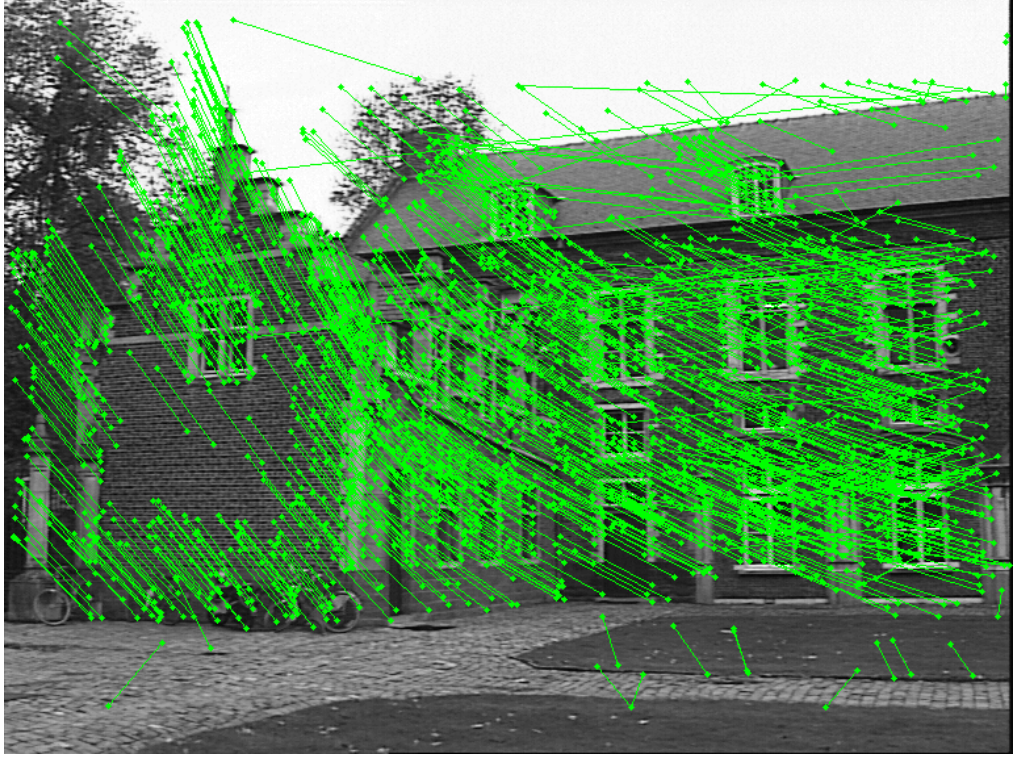


Figure 3.14: 1384 matches remain after applying the epipolar constraints to the set of 1531 putative matches in Figure 3.11

corresponding to a point x' in the second image can be computed, as shown in Equation (3.14). Thus all correspondences must obey the relation:

$$x'^T F x = 0 \quad (3.15)$$

Generally the measurement of two corresponding points $x \leftrightarrow x'$ from the feature matching stage is not completely accurate, thus $x'^T F x$ is not exactly zero. x' however should be geometrically close to l' to be considered a match.

Figure 3.14 shows the remaining feature matches after application of the epipolar constraint. As can be seen, some incorrect matches still exist, corresponding to putative matches that happen to fall on epipolar lines.

The only requirement for calculating the epipolar geometry is sufficient overlap between the images. Image overlap ensures a large enough number of correct matches to enable successful computation of the fundamental matrix.

Two-View Guided Matching

Knowledge of the epipolar geometry also allows additional matches not detected in the initial feature matching stage to be added. By reducing the search space for a

match to a given feature x in the first image to a line in the second image, more relaxed matching constraints can be applied and new matches sought. This step can help counter any previous deficiencies in the distinctiveness of the descriptors that might have prevented successful matching of re-detected points.

By multiplying F together with a given unmatched feature x using Equation (3.13), a line in the second image is obtained. This line specifies where a corresponding feature must lie. Features not within the region of this line are masked out, and a feature match is sought using the same process as that described in Section 3.2.3, but with a more relaxed threshold on the distance between the closest and second closest matches.

This process is demonstrated in Figure 3.15. Figure 3.15(a) shows a feature from the first image that was previously unmatched. Figure 3.15(b) shows the mask generated from the line computed using Equation (3.13) for the epipolar line in the second image. A region of 5 pixels on either side of the true epipolar line is shown here, accounting for localisation errors of the feature when detected in the second image. Figure 3.15(c) shows the remaining feature set from the second image after applying the mask. After performing guided matching between images 1–2 an additional 540 matches are found.

3.2.5 Three-View Geometry

When three views of the scene are available, additional constraints over the two-view case can be added, providing increased knowledge of the geometric relationships between the images and further eliminating outliers from the match set. To begin with, putative three-view matches are computed. These matches are a combination of 2 two-view match sets compliant with epipolar geometry as discussed in the previous step. When combining matches from images I_j and I_{j+1} , with those from I_{j+1} and I_{j+2} the features the match sets have in common from image I_{j+1} are used to determine which features correspond over three views. Given two matching points between I_j and I_{j+1} , $x \leftrightarrow x'$ and two matches from I_{j+1} and I_{j+2} , $x' \leftrightarrow x''$ the sequence can be extended to $x \leftrightarrow x' \leftrightarrow x''$.

Figure 3.16 shows the putative three-view matches computed between images 1, 2 and 3. This set still contains some outliers; therefore additional constraints over each image triplet are needed to eliminate or reduce the number of outliers.

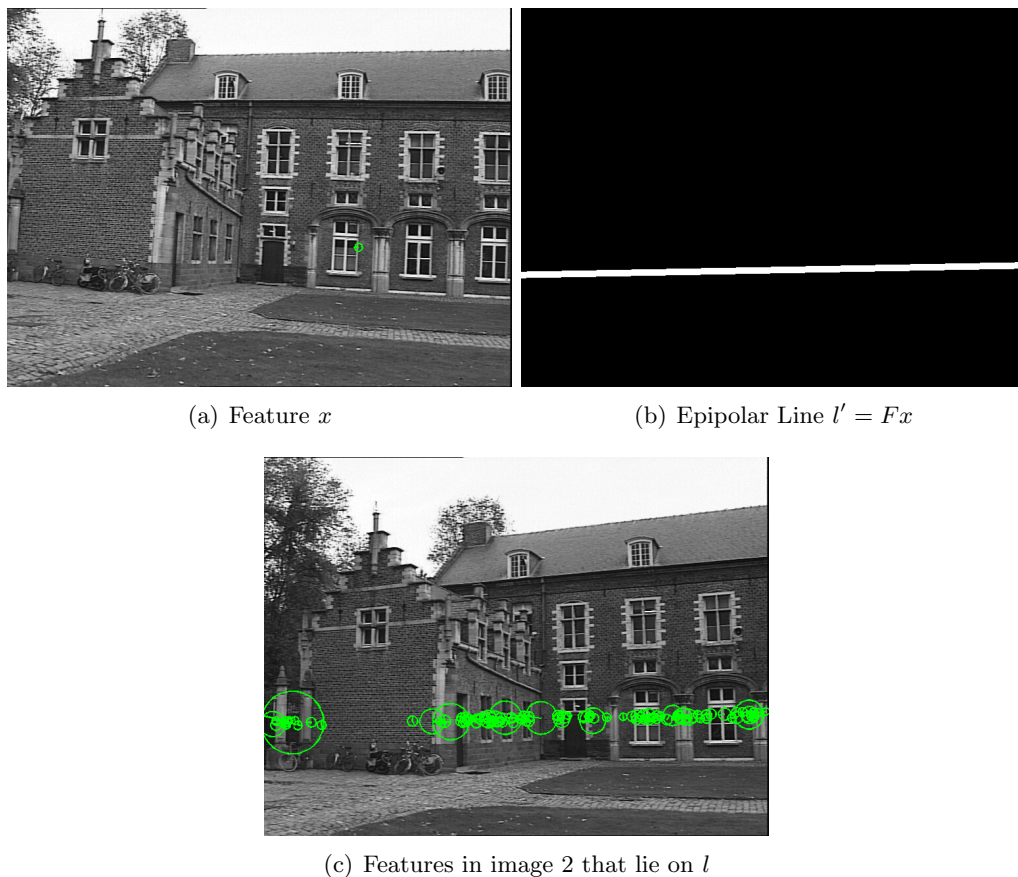


Figure 3.15: Guided matching images. (a) shows the feature in image 1, x , for which a match is sought. (b) shows the mask generated by the line computed using F and x as in Equation (3.13). (c) shows the set of features in image 2 with the mask applied, a potential match must be one of these points

The Trifocal Tensor

The trifocal tensor describes the projective geometry between three views and is analogous to the fundamental matrix for two-view geometry. Like the fundamental matrix, the trifocal tensor is independent of scene structure and can be computed from image correspondences without any knowledge of camera motion or internal camera parameters. Describing the three-view geometry has advantages over the two-view case; given a matching point in two-views, a corresponding point in a third can be obtained, and similarly for lines. This differs from two-views where a point in one image generates a line in the second image on which the corresponding point must lie.

Computation of the trifocal tensor is also similar to that of the fundamental matrix. In this case the process begins by obtaining two-view matches between the two image pairs in the same manner as that described in Section 3.2.4. Putative three-view matches are then formed by the combination of the 2 two-view sets as

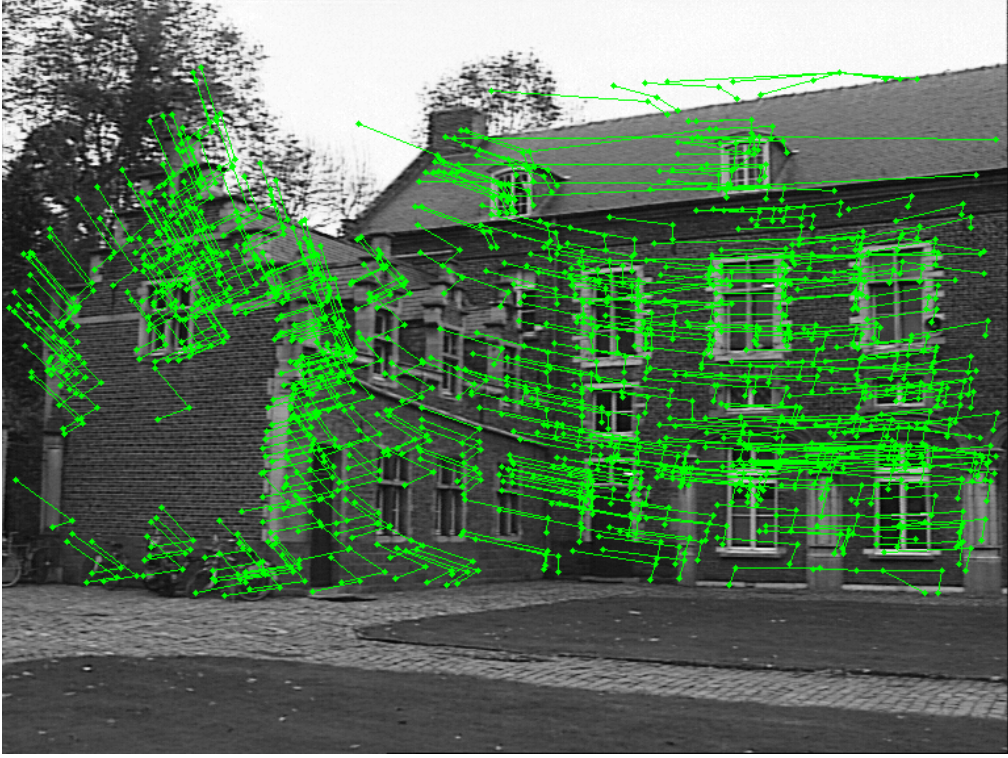


Figure 3.16: 590 Putative three-view matches between images 1–2–3

described previously.

RANSAC is then used to find the trifocal tensor T consistent with the most matches. Each RANSAC sample taken from the set consists of six putative three-view correspondences. Once the best model of T is found it is re-estimated over all matches classified as inliers.

The use of three views provides additional constraints further reducing outliers in the match set. Trifocal tensors provide a point-point-point relationship as shown in Figure 3.17, where three matching points will satisfy the point-point-point correspondence relation of Equation (3.16). Here x , x' , and x'' are the matching points over the three views, while T is the computed tensor.

$$[x']_{\times} \left(\sum_{i=1}^3 x^i T_i \right) [x'']_{\times} = 0_{3 \times 3} \quad (3.16)$$

$[x']_{\times}$ is the cross product matrix of x' where the cross product of any homogenous 2D point a is:

$$[a]_{\times} = \begin{bmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{bmatrix} \quad (3.17)$$

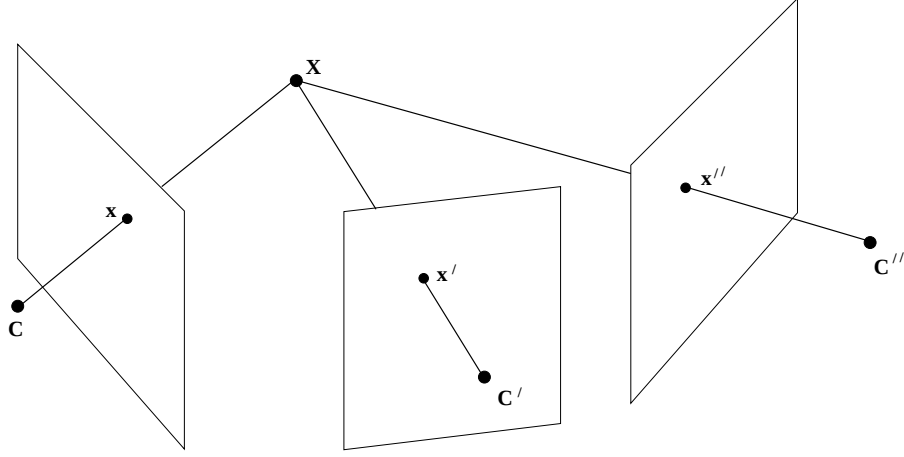


Figure 3.17: Point-point-point relation between three views (Image courtesy of Richard Hartley and Andrew Zisserman – Australian National University and University of Oxford. Image reproduced from [HZ04])

Figure 3.18 shows the inliers from the set of three-view putative matches after computing the trifocal tensor and applying the constraint in Equation (3.16).

Guided Three-View Matching

Another useful property of the trifocal tensor is that it can help guide three-view matching. This is provided by the ability to transfer points using the trifocal tensors. Essentially, given a matching point between any two views $x \leftrightarrow x'$, the corresponding location x'' in the third can be determined. This more restrictive relationship produces a smaller search space than that of the two-view case, this—assuming a correct trifocal tensor model—results in virtually no outliers. Performing the guided matching on images 1–2–3 yields 104 additional matches.

Equation (3.18) is used for transferring a point x in the first image to the point x'' in the third. Here $l'_j T_i^{jk}$ defines a homography between the first and third image with a plane being computed from a line l' in the second image. The 2D points $x \leftrightarrow x'$ are corrected so that they exactly satisfy the relation $x'^T F_{21} x = 0$. This correction is necessary as most points will not precisely satisfy this relation due to noise. These corrected points are designated $\hat{x} \leftrightarrow \hat{x}'$.

$$x''^k = \hat{x}^i l'_j T_i^{jk} \quad (3.18)$$

In order to transfer the point, a line l' passing through the corresponding point in the second x' is chosen. A good choice for l' was determined by Hartley and Zisserman as the line perpendicular to the epipolar line induced by the point x



Figure 3.18: 522 final matches between images 1–2–3

namely $F_{21}x$. The two fundamental matrices can be retrieved from the trifocal tensor using:

$$F_{21} = [e'] \times [T_1, T_2, T_3]e'' \text{ and } F_{31} = [e''] \times [T_1^T, T_2^T, T_3^T]e' \quad (3.19)$$

The epipoles e' and e'' in the two images can be computed as the null-vectors of the matrices. u_i and v_i are the left and right null-vectors of T_i

$$e'^T[u_1, u_2, u_3] = 0 \text{ and } e''^T[v_1, v_2, v_3] = 0 \quad (3.20)$$

Li *et al.* made use of the point transfer ability of the trifocal tensor to build a complete AR solution [LLR04]. Here three reference images of a scene are used; matches are extracted and a trifocal tensor between the views is computed. When performing live matching an additional tensor is formed between two of the reference views and the live camera view. The point transfer function is then used to transfer the location of a marker placed in the reference frames into the live view. The pose of the marker in the live frame is then passed to ARToolKit which renders the virtual object.

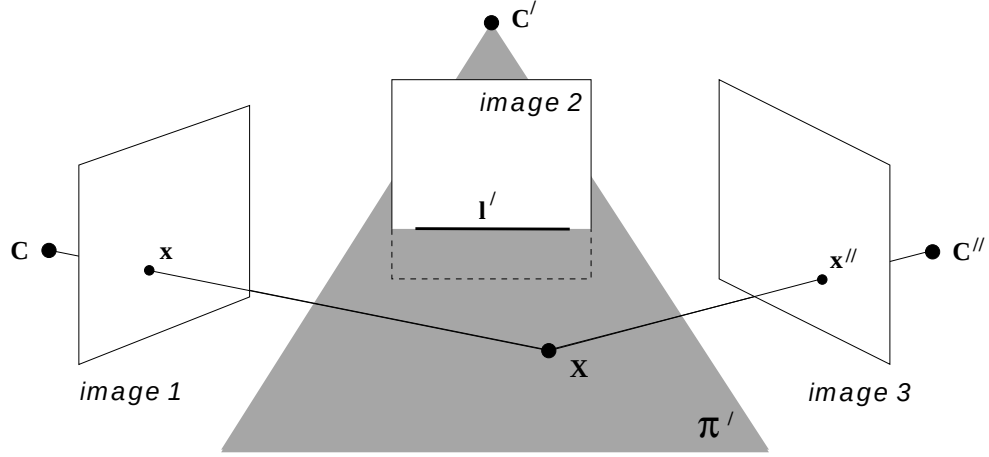


Figure 3.19: The point transfer used for guided matching under three views (Image courtesy of Richard Hartley and Andrew Zisserman – Australian National University and University of Oxford. Image reproduced from [HZ04])

3.2.6 Multi-view Correspondences

Once the final three-view matches have been computed the multi-view correspondences can be established by combining the matches for all the image triplets. For example the triplet 1–2–3 merged with 2–3–4 becomes 1–2–3–4, *i.e.* the common matches in images 2 and 3 can then be extended over all four images. This process is continued over all triplets until corresponding features over the entire sequence have been established.

An important result of this is to ensure only one instance of any real world point exists in the final reconstruction. Duplicate 3D points will result in duplicate 2D feature descriptors being used to describe 3D points which make up the model. Performing 2D-3D matching in the presence of these duplicates can result in the closest and second closest matches being very similar, thus reducing confidence in the match and potentially dropping a correct match. Figure 3.20 shows one scene point established as a multi-view match through nine images.

3.2.7 Camera Pose Recovery

In order to build a 3D model of the scene, the camera parameters need to be established so that the 3D locations for multi-view matches can be determined. Equation (3.21) shows the composition of an uncalibrated camera P . A camera is said to be uncalibrated if the internal parameters of the camera are unknown. P is



Figure 3.20: One feature detected and tracked over nine images

a 3×4 matrix comprised of the internal camera parameters K , called the calibration matrix or intrinsic parameters, and the extrinsic camera parameters R and t which are the rotation and translation relative to the world co-ordinate frame.

$$P = K[R \mid t] \quad (3.21)$$

The calibration matrix K is made up of the parameters shown in Equation (3.22). fx and fy are the focal lengths in the X and Y axes. A ratio of fy/fx —often termed the aspect ratio—that is not 1 indicates the pixels in the camera sensor are not square. s is the skew of the pixels. Most cameras have rectangular pixels thus this is set to 0. px and py are the principle points, or the optical centre of the camera.

$$K = \begin{bmatrix} fx & s & px \\ 0 & fy & py \\ 0 & 0 & 1 \end{bmatrix} \quad (3.22)$$

Any 3D scene point X is imaged through P to produce the 2D point x , as shown in Equation (3.23). Thus the reconstruction problem is obtaining a set of cameras P^j viewing a scene comprised of 3D points X_i , and x_i^j is the image of 3D point i as seen through camera j such that Equation (3.24) holds.

$$x = PX \quad (3.23)$$

$$x_i^j = P^j X_i \quad \forall i, j \quad (3.24)$$

Computing the camera matrices for any given image can be achieved in several ways. One way is to extract the camera matrices consistent with the fundamental

matrix computed between 2 images. Equation (3.25) shows the two camera matrices P and P' for the images represented by the fundamental matrix F . Here, the cameras are specified in canonical form; that is the world frame is aligned with P , while the second camera is expressed relative to the first. To achieve this, camera P is specified as the identity matrix $[I \mid 0]$. The second camera P' is computed using the cross product matrix of the epipole in the second image e' , and F .

$$P = [I \mid 0] \text{ and } P' = [[e']_{\times} F \mid e'] \quad (3.25)$$

A similar process can also be applied to extract cameras matrices consistent with a computed trifocal tensor in the case of three views. Again, the three camera matrices, P , P' , and P'' are specified in canonical form relative to P as shown in Equations (3.26) to (3.28). This time the tensor T is used rather than F .

$$P = [I \mid 0] \quad (3.26)$$

$$P' = [[T_1, T_2, T_3]e'' \mid e'] \quad (3.27)$$

$$P'' = [(e''e''^T - I)[T_1^T, T_2^T, T_3^T]e' \mid e''] \quad (3.28)$$

Establishing camera matrices for a series of cameras can be achieved by merging the cameras computed from individual pairs or triplets. Camera matrices P , P' from images 1–2, and matrices P' and P'' from images 2–3, can be extracted from the respective fundamental matrices computed between the image pairs. Each of these camera pairs is in canonical form. In canonical form P' is expressed relative to P , and P'' is expressed relative to P' . However, in order to produce a reconstruction these cameras must be specified in a consistent frame. To achieve this the common frame between the two pairs can be used to make P'' relative to P rather than P' . This can be achieved by combining the transformations of P' with P'' .

One downside to this approach is that the error associated with computing the tensors or fundamental matrices, and the associated camera matrices extracted from them, is accumulated through each triplet. This means the more images there are—and the further a given image is from the start of the sequence—the more error there is in the camera matrix.

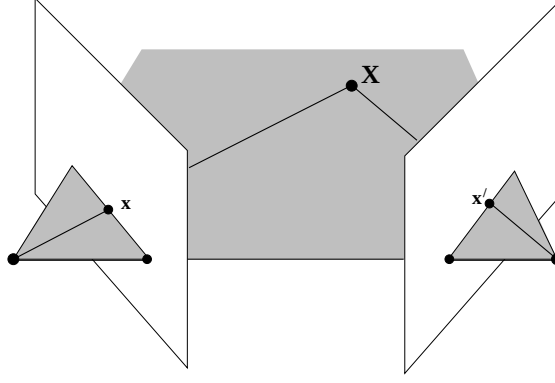


Figure 3.21: Triangulation. x and x' are two corresponding points. X lies at the intersection of the two rays cast through x and x' which all lie on a plane (Image courtesy of Richard Hartley and Andrew Zisserman – Australian National University and University of Oxford. Image reproduced from [HZ04])

3.2.8 3D Reconstruction

Having obtained the camera matrices for the images, 3D reconstructions for all the 2D multi-view matches can be computed. In the simplest case of two views, the 3D location of any corresponding point can be determined as the intersection of two rays back-projected from two corresponding image points as shown in Figure 3.21. This process is known as triangulation.

As pointed out by Hartley and Sturm [HS95], triangulation is only assured when the corresponding points $x \leftrightarrow x'$ exactly satisfy the relation $x'^T F x = 0$. Otherwise there is no guarantee the rays cast lie on the same plane, thus the rays may never meet, making a reconstruction impossible. For this reason the points are first corrected, producing a pair $\hat{x} \leftrightarrow \hat{x}'$ which exactly satisfies the epipolar constraint. Figure 3.22 shows this process. For example, when computing $\hat{x}$, it is chosen as the intersection of the line perpendicular to the epipolar line l which passes through x ; in essence $\hat{x}$ is the point on l which is closest to x .

The 3D location for each multi-view match can then be computed yielding a 3D reconstruction. The type of reconstruction produced is dependent on the amount of information known about the camera's calibration and placement. When uncalibrated cameras are used, a projective reconstruction is the best that can be achieved. The term “projective reconstruction” describes the ambiguity between a reconstruction produced purely from feature correspondences detected in images of a scene, and the ‘true’ reconstruction which generated the measured correspondences. These reconstructions differ by an unknown arbitrary projective transformation such that the true camera matrices $\bar{P}$ and $\bar{P}'$ are related to the projective cameras P and

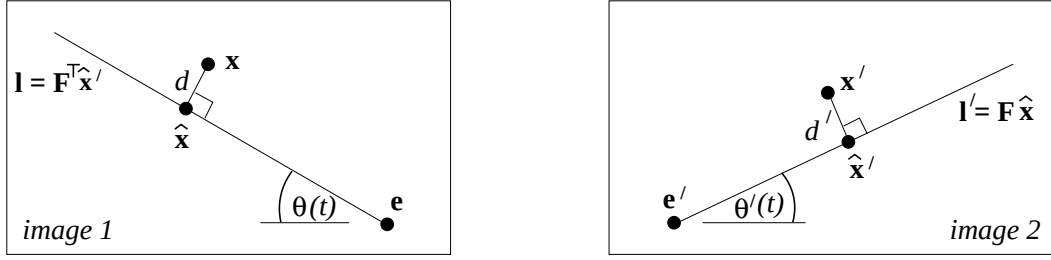


Figure 3.22: Obtaining corrected points $\hat{x}, \hat{x}'$ involves finding points which lie precisely on the respective epipolar lines, l or l' . The closest point is the intersection of the line through x or x' that is perpendicular to the epipolar line (Image courtesy of Richard Hartley and Andrew Zisserman – Australian National University and University of Oxford. Image reproduced from [HZ04])

P' by:

$$\bar{P} = PH^{-1}, \quad \bar{P}' = P'H^{-1} \quad (3.29)$$

where H is a 4×4 matrix representing a projective transformation. The 3D points for the true and projective reconstructions, $\bar{X}_i$ and X_i , are related by $\bar{X} = HX$.

This principle is illustrated in Figure 3.23(b). The reconstructions in the figure differ by a projective transformation. However, the measured feature correspondences remain the same. Because the feature correspondences are the same under any projective transformation and these correspondences are the only information available, the ambiguity cannot be resolved without further information or constraints. The reconstruction shown in the bottom of the figure is different due to longer focal lengths in the cameras. Focal length is unknown when using uncalibrated cameras, therefore the true reconstruction could be either of these options, hence the ambiguity. This ambiguity means projective reconstructions can appear highly deformed with respect to the actual scene. Figure 3.24 shows three types of reconstruction generated from the images in Figures 3.24(a) and 3.24(b). Figures 3.24(c) and 3.24(d) show a projective reconstruction of this building. Notice the high degree of deformation. This model would be harder to recognise if not for the lines joining the points that have been added to aid in visualising the results.

The projective ambiguity has serious consequences when augmenting the scene. During live pose estimation (covered in more detail in upcoming Section 3.2.11) the live camera pose is established with respect to the model. This model is in turn specified relative to the coordinate frame of the first image, which by implication means the live camera matrix is determined with respect to the first camera. By

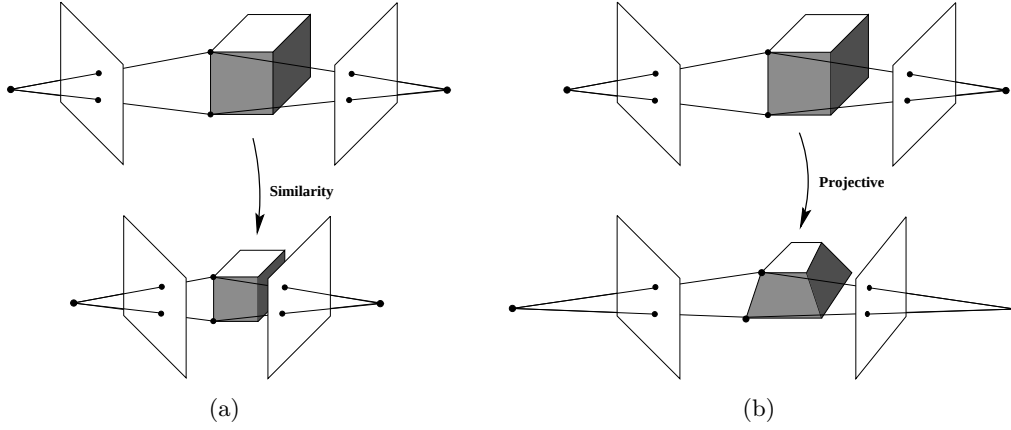
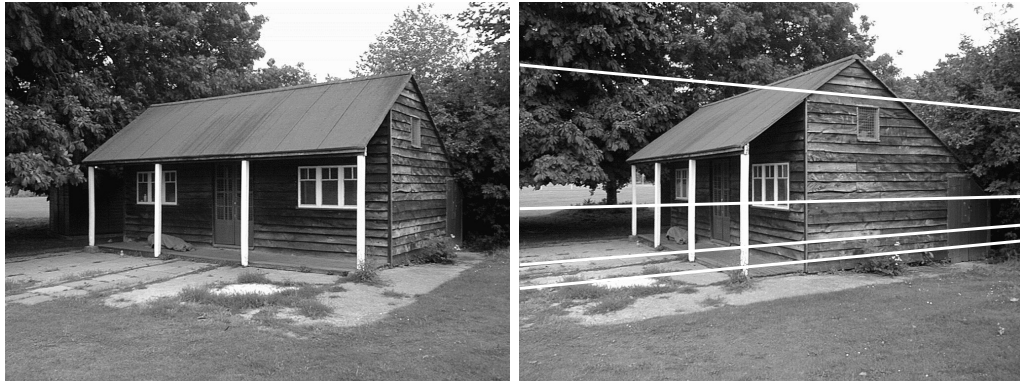


Figure 3.23: 3.23(a) shows two reconstructions that differ by a similarity transform. 3.23(b) shows two reconstructions that differ by a projective transformation. (Images courtesy of Richard Hartley and Andrew Zisserman – Australian National University and University of Oxford. Images reproduced from [HZ04])

specifying the virtual objects's location and orientation in the first frame, the position in the reconstruction is also established. The virtual object's position in the live frame can be determined by projecting its location from the model through the computed live camera and into the live image. However, correctly orienting this object is a more difficult problem.

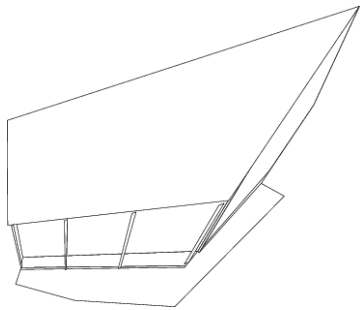
In order to orientate the object correctly, the transformation between the first camera and the live camera is required. This transformation is encapsulated in the camera matrix P , which as shown in Equation (3.21) is $P = K[R \mid t]$. In order to apply the transformation $[R \mid t]$ to the augmentation, the effect of K , which represents the intrinsic camera parameters, must be mitigated.

Obtaining the internal camera parameters is a process called camera calibration. Removing the effect of the camera calibration removes the projective ambiguity from the construction model and produces a Euclidean (often called metric) reconstruction. A Euclidean reconstruction differs from the true reconstruction by a similarity transform, meaning it differs by an unknown rotation, scale or translation which cannot be determined without *a priori* knowledge. For example when performing reconstruction on the Arenberg castle images, the longitude and latitude of the building cannot be determined. Similarly, the absolute orientation of the building cannot be established, *i.e.* whether the building is north facing or not. Ambiguity in scale is present as it is unknown whether the images capture an actual building or a miniature model of the castle. Two reconstructions differing by a similarity transformation are shown in Figure 3.23(a). These reconstructions differ in scale.

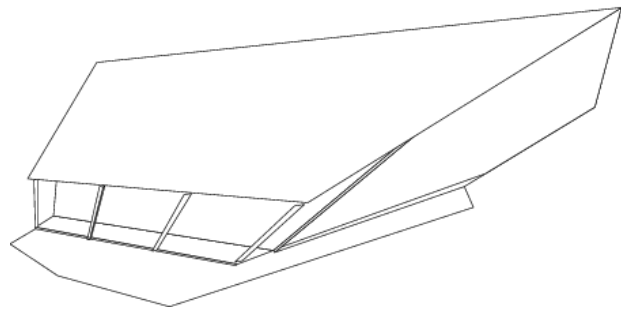


(a)

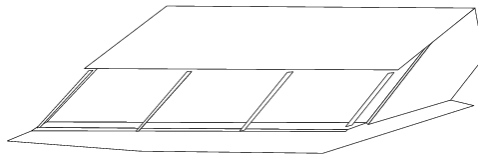
(b)



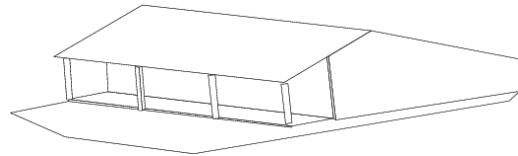
(c)



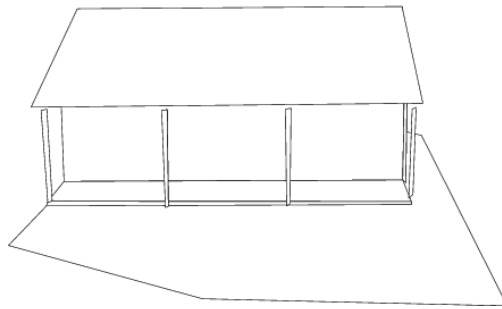
(d)



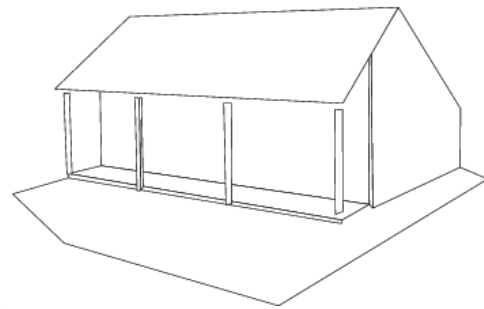
(e)



(f)



(g)



(h)

Figure 3.24: Examples of projective, affine, and Euclidean reconstructions (Image courtesy of Richard Hartley and Andrew Zisserman – Australian National University and University of Oxford. Images reproduced from [HZ04])

Figures 3.24(g) and 3.24(h) show a Euclidean reconstruction of the house. Euclidean reconstructions provide an accurate reconstruction and the most visually pleasing results.

Several approaches exist to perform camera calibration. The first involves directly computing the calibration from the camera itself using a calibration rig and is described next.

Camera Calibration using a Calibration Rig

If the camera which took the images is available it can be calibrated by using an object with known geometry, often called a calibration rig. Typically this is a specialised object specifically designed for calibration. They are often covered with high contrast patterns such as checkers, making it easy to detect and localise points on their surface.

Some approaches make use of a known 3D structure such as that shown in Figure 3.25. The advantage of this approach is that only one image of the calibration object is required. The downside is that construction of this object must be precise as the relative 3D positions of all the squares on its surface must be known *a priori*.



Figure 3.25: Calibration object with known geometry and a checker pattern on all sides (Images courtesy of Stefano Soatto – UCLA. Reproduced from [MSKS03])

Another approach makes use of a planar checkerboard pattern. This is simpler in construction as all that is required is a printed checkerboard on a flat surface. It does, however, require a number of images (at least 25) to be taken of this pattern from a number of views in order to successfully calibrate the camera. Images of such a checkerboard planar object are shown in Figure 3.26.

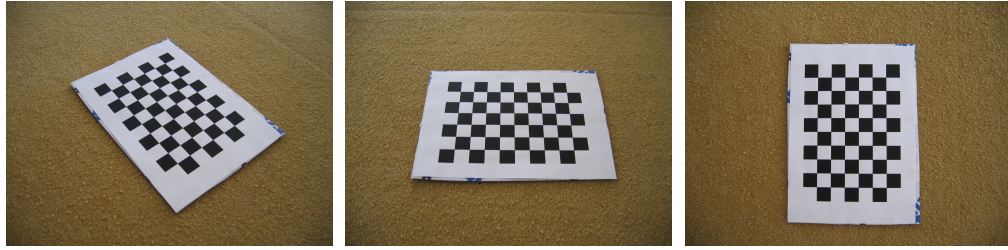


Figure 3.26: Calibration grid for assisting in calibrating an unknown camera

A disadvantage of this manual calibration approach is that it must be done for every camera and requires quite a bit of time. In addition, calibration must be redone if the internal parameters of the camera change, for example if the focal length of the camera changes due to a zoom operation. Also, if the camera that took the images is not available, calibration in this way is impossible. For this reason several other ways of achieving a Euclidean reconstruction have been devised.

Other Calibration Methods

Partial knowledge about the camera in use or about the scene can be used to determine the camera calibration when multiple images of the scene are available. For example, as Ma *et al.* [MSKS03] point out, knowing that the same camera was used for all captured images places additional constraints on the calibration matrix K , as the skew, principle point, aspect ratio are all the same. Knowing or making assumptions about other calibration factors can also help. For example: assuming rectangular pixels where the skew parameter is zero, or that the optical centre is the middle of the image. Such assumptions can form an approximate model of K , which when used during reconstruction might produce a model sufficient for visualisation; these are called quasi-Euclidean reconstructions.

Hartley and Zisserman [HZ04] note that if two cameras with the same calibration differ by a translational motion, an affine reconstruction is possible. An affine reconstruction differs from the true reconstruction by an affine transformation. The advantage of affine reconstructions over projective ones is that lines which are parallel in the scene remain parallel in the reconstruction. Hartley and Zisserman also note that if the only unknown in the calibration matrix is the focal length, a Euclidean reconstruction is possible.

Another approach to achieving a Euclidean reconstruction is to perform a stratified reconstruction where a projective reconstruction is successively upgraded from projective to affine to Euclidean. Figure 3.24 shows three different types of

reconstructions built using a stratified method from the images in Figures 3.24(a) and 3.24(b).

Figures 3.24(e) and 3.24(f) show an affine reconstruction, where the projective reconstruction has been upgraded by identifying the plane at infinity. The plane at infinity can be established by extending lines identified as parallel in the image out, until they intersect. This point of intersection is also a point on the plane at infinity. Examples of these lines are shown in 3.24(b). Notice edges in the scene that are parallel are now parallel in the model, however perpendicular lines in the scene are not perpendicular in the model; this is the defining characteristic of an affine reconstruction.

The affine reconstruction is upgraded to Euclidean by identifying the absolute conic at infinity. This improves on the affine reconstruction as now perpendicular lines in the scene are also perpendicular in the model.

Projective to Euclidean using Camera Calibration

For the sake of simplicity, the 3D system assumes a calibration matrix computed *a priori*. With the calibration matrix K , upgrading the projective reconstruction to a Euclidean one is a simple task.

In the case of uncalibrated cameras, 2D points are a function of the camera matrix and the points' 3D coordinates such that $x = PX$, where $P = K[R \mid t]$. When K is known normalised 2D points can be obtained by applying the inverse of K to x , such that $\hat{x} = K^{-1}x$.

A normalised camera matrix can also be obtained by applying the inverse of K to the camera matrix such that $K^{-1}P = [R \mid t]$. This removes the internal camera parameters from the equation, meaning that the image of point X is now only a function of the 3D point's location and the rotation and translation parameters of the camera $\hat{x} = [R \mid t]X$. Once P is calibrated it only contains information on the extrinsic parameters of the camera, *i.e.* the camera's pose.

Applying triangulation (discussed earlier in Section 3.2.8) to the normalised 2D points and the normalised cameras yields a euclidean reconstruction. When performing triangulation on uncalibrated cameras, two rays only meet if the 2D correspondences exactly satisfy the relation $x^T F x = 0$. When operating in a calibrated framework, it is important to note that the equivalent of the fundamental matrix is the essential matrix. The essential matrix can be considered a specialisation

of the fundamental matrix where normalised image coordinates are used. The essential matrix E operates in the same way as F and encapsulates the epipolar geometry for the scene. It effectively encapsulates only the extrinsic parameters of the camera, whereas the fundamental matrix encapsulates both the intrinsic and extrinsic parameters. The defining equation for the essential matrix is:

$$\hat{x}'^T E \hat{x} = 0 \quad (3.30)$$

Two normalised points $\hat{x} \leftrightarrow \hat{x}'$ must exactly satisfy the relation $\hat{x}'^T E \hat{x}$ for triangulation to be successful. The fundamental matrix can be converted to an essential matrix using:

$$E = K'^T F K \quad (3.31)$$

where K and K' are the calibration matrices for the first and second views respectively. In all cases during this reconstruction it is assumed the same camera is used for all images, therefore $K' = K$. Conversion of F to E removes the intrinsic parameters from F .

3.2.9 Optimisation

Having obtained an initial Euclidean reconstruction, the final step is to optimise the model. Bundle adjustment (BA) [TMHF99] seeks a model of the scene structure and camera poses that minimises reprojection error. Reprojection error is the image distance between a measured 2D point x_i^j with index i seen in image j , and the corresponding scene point X_i projected through camera matrix P^j . BA ensures that all the scene points viewed in every image conform exactly to the relation $x_i^j = P^j X_i$.

2D points extracted from the images and the reconstructed cameras and 3D points all have noise, and therefore do not satisfy this constraint exactly. Thus, an estimate of the reconstruction parameters is sought that minimises the reprojection error. Essentially, each point in the optimised model must exactly satisfy the relation $x_i^j = \hat{P}^j \hat{X}_i$, where $\hat{X}_i$ is the estimated 3D point i and $\hat{P}^j$ is the estimated camera matrix for image j . BA seeks to minimise this reprojection error globally for all points i as seen in all images j . The equation for this minimisation is shown in Equation (3.32).

$$\min_{\hat{P}^j, \hat{X}_i} \sum_{ij} d(\hat{P}^j \hat{X}_j, x_i^j)^2 \quad (3.32)$$

Figure 3.27 shows four views of the final bundle adjusted Euclidean reconstruction for the castle sequence. This model was constructed using nine images from the castle sequence. The blue pyramid shaped objects depict the camera locations for each image used to construct the model. The first image is shown from an isometric view, the second from in front of the building, the third from the side and the fourth from directly above. The images are shown as point clouds made up of the reconstructed 2D features.

3.2.10 Virtual Object Insertion

The final step in preparing the system involves interactively specifying the location and orientation of the virtual object with respect to the computed reconstruction. The process here is identical to that employed by Skrypnik and Lowe [SL04].

This process utilises two images used in the reconstruction stage that have an available fundamental matrix F . In the first image the position at which the object will appear is specified by designating a point in the image. Using this 2D point the epipolar line Fx is automatically rendered in the second image on which the designated point must lie. This is shown in Figure 3.28. The image on the left shows the desired position and the image on the right shows the epipolar line in the second image which corresponds to this point. By designating the position on the epipolar line in the second image the points' 3D position can be established. After this the orientation of the object can be specified. An oriented virtual object (in this case a simple co-ordinate frame) is shown in Figure 3.29.

The views of the final reconstruction shown in Figure 3.27 correctly show the 3D position of this virtual object inside the model, and can be seen as a red dot. This position was established by back-projecting the point through the known camera matrices.

With respect to virtual object insertion, the 3D system's method is far superior to the approach employed by the 2D system described in Section 3.1. There, the virtual object had to be specified in every reference frame used. In contrast, the 3D system only requires the virtual object to be specified once inside the reconstruction. Furthermore, it proved infeasible to provide full 6DOF operation for the 2D system thus the 3D system's inherent ability in this regard makes it superior. The accuracy

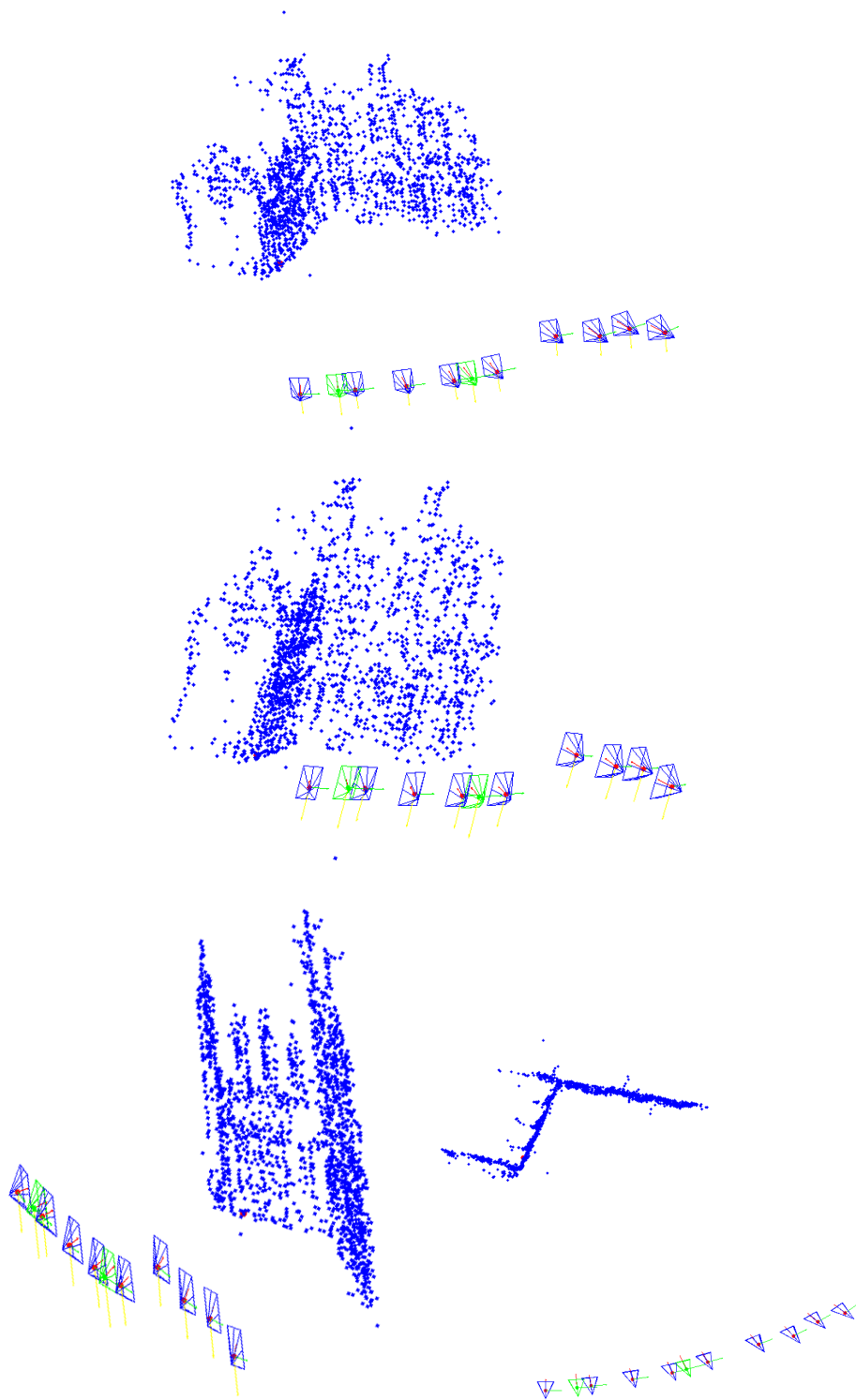


Figure 3.27: Four views of the final reconstruction. The point clouds represent the reconstruction of the scene points. The pyramid shaped objects represent the cameras. The blue cameras were used to construct the model. The green cameras represent two live views



Figure 3.28: The left image shows the selected location for the virtual object. The right image shows the corresponding epipolar line on which the object must lie



Figure 3.29: The assigned orientation of the co-ordinate frame in the first and second images

of the Fourier-based algorithm as well as a range of local feature based algorithms is established in Chapter 5.

3.2.11 Live Pose Estimation

Having established the pose of the virtual object with respect to the scene, live tracking is possible. During this phase the live camera pose is determined with respect to the reconstruction. By determining the live camera's pose the virtual object can be correctly rendered in the live view.

The initial stages of this process are listed in Table 3.1 and are the same as for 3D reconstruction. To begin with, feature extraction is performed on the live image, as described in Section 3.2.2. Once features have been extracted putative 2D-3D matches are calculated by determining SIFT matches between the descriptors for the extracted 2D points, and the descriptors for the 3D points that make up the reconstruction.

Having obtained these 2D-3D matches, the pose of the live camera is computed



Figure 3.30: Augmented images produced by the 3D system

by performing an operation similar to BA. The 2D-3D matches may contain many false matches, therefore RANSAC is used to compute the live camera pose in the presence of outliers. For each RANSAC sample (which is a small subset of the 2D-3D matches) the LM algorithm iteratively searches for a camera capable of generating the measured 2D-3D correspondences. As the algorithm iterates LM converges on a camera model that minimises the reprojection error between the 2D features their corresponding 3D points. Once RANSAC has determined the camera model with the most support, LM is reapplied to 2D-3D matches classified as inliers, yielding a camera model that minimises the reprojection error over all matches. The live camera pose can then be applied to the pose of the virtual object and rendered inside the live view. Figure 3.30 shows two novel images augmented with a coordinate frame. The green pyramids in Figure 3.27 show the position of these cameras.

The minimisation used during live operation is significantly faster than the BA used for model optimisation as the only unknown parameter is the current camera. The 3D coordinates for the model as well as the other camera parameters are kept constant. This makes the live operation stage significantly faster than bundle adjusting an entire reconstruction model.

3.2.12 Problems

The approach employed by the 3D system has some disadvantages. To work, it requires a Euclidean reconstruction of the scene that is only possible with calibrated cameras. This means either the camera is restricted to have constant settings *i.e.* not permitting zoom operations, or the camera must be re-calibrated using a calibration rig every time a change occurs. The other option is to perform a stratified reconstruction by making assumptions about the scene structure. This often requires user intervention to specify parallel lines in the image *etc.*

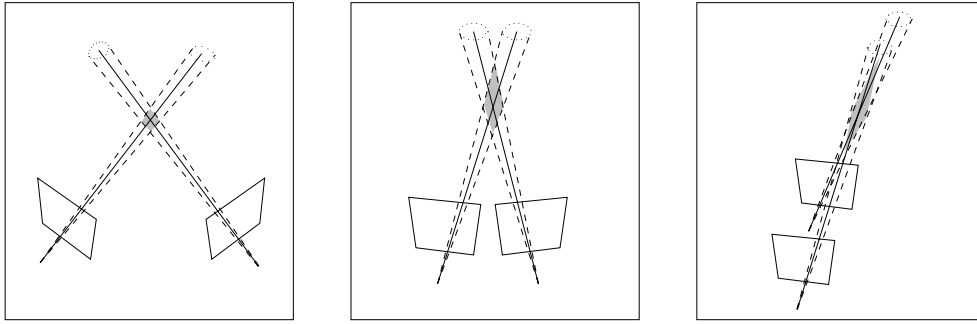


Figure 3.31: The region of uncertainty when triangulating points. (Image courtesy of Richard Hartley and Andrew Zisserman – Australian National University and University of Oxford. Images reproduced from [HZ04])

Another problem with this approach is caused by the compounding error associated with long image sequences. As the camera poses are computed relative to the first frame and the sequence is merged by combining smaller subsequences of three frames, the camera pose of a later image is the combination of all computed camera poses before it, in addition to its own. This means cameras further away from the initial frame suffer from cumulative error.

Low-level computations often impose specific constraints to ensure successful operation. For example, computation of the fundamental matrix is not possible either when all the world points lie on a plane, or when the camera centres are coincident, *i.e.* when there is no translation difference between the cameras. This means in a completely autonomous system, these situations must never occur or they must be detected and dealt with.

In addition, the choice of images used to reconstruct the scene is important. Images separated by a sufficient baseline are required in order to ensure accurate point localisation when reconstructing points. As the images come closer together, rays cast through image correspondences become more parallel and their point of intersection becomes increasingly ambiguous. This is illustrated in Figure 3.31 where the image on the left provides the best localisation accuracy, while the point of intersection in the other images is less certain. Notice views with decreasing separation obtain larger regions of uncertainty meaning reconstructed points from these views may incur poor localisation accuracy. Images separated by too large a distance run the risk of detecting significantly different features, or obtaining features with descriptors too dissimilar to be matched.

3.2.13 Summary

In summary, the 3D system can be split up into two phases: 3D reconstruction and live operation. The general flow of reconstruction starts with feature extraction and matching. The SIFT algorithm is employed to both detect and describe features. These features can then be matched by calculating the distance between the feature descriptors. From these putative matches two-view geometry constraints can be applied, namely the epipolar geometry constraint. This refines the initial set of matches and eliminates a lot of outliers.

Three-view geometry extends the benefits of the two-view case and provides more restrictive constraints, further eliminating remaining outliers. Multi-view matches can then be formed of feature correspondences appearing over the entire sequence of images. At this point a projective reconstruction is possible. However, in order to be able to augment a scene, a Euclidean reconstruction is necessary.

In order to obtain a Euclidean reconstruction the internal camera calibration is required, to eliminate the projective ambiguity. In this system it is assumed the same camera is used for all images and that the calibration matrix is known *a priori*. These assumptions are unrealistic in practice as generally camera calibration is not known, however it is satisfactory for the purpose of the reference implementation and to demonstrate the principles pertinent to achieving a 3D AR system using natural features.

Having obtained a 3D reconstruction, virtual objects can easily be positioned inside the model. Live operation is then possible. A live camera feed is augmented by matching local features to the points that make up the model and computing the live camera's pose. A virtual object can then be correctly rendered from the live camera's point-of-view.

The general design of the approach is based on the work of Skrypnik and Lowe [SL04], with the 3D reconstruction process more closely following that prescribed by Hartley and Zisserman [HZ04]. Several differences exist between the approach of Skrypnik and Lowe [SL04] and that used by the 3D system. The first is the difference in constructing the 3D model. Skrypnik and Lowe first obtained the multi-view matches, then a small selection of the best points (those with the most correspondences) were used to determine all the model parameters simultaneously using the LM optimisation algorithm. This, as described by the authors, is achieved by "minimisation of the reprojection errors over all camera parameters and world

point coordinates, given image projections of the world points” [SL04] and is formulated as Equation (3.33). Where a_{ij} is comprised of three entities: the 3D coordinates of a given world point X_j , the camera pose parameters for the image i , and the global camera calibration parameters. w_j expresses a weighting for X_j which is adjusted based on the reprojection error associated with this point. Π is the projection function.

$$\min_{a_{ij}} \sum_i \sum_j \| w_j (\Pi(a_{ij}) - x_{ij}) \|^2 \quad (3.33)$$

The 3D system presented in this chapter made use of a more traditional approach, as outlined by Hartley and Zisserman [HZ04]. There are, however, some minor steps described by Hartley and Zisserman that are not described or implemented, as the improvement in the accuracy of the model does not justify the effort required in implementation. For example, after computing each fundamental matrix it is recommended that a non-linear estimation of F is made, using the LM algorithm to minimise the reprojection error on all the correspondences classified as inliers by RANSAC when computing the initial estimate. After this, guided matching takes place and these two steps are repeated until the number of correspondences is stable. In the 3D system, computation of F involves performing a ‘least squares’ fit on the inliers determined by RANSAC and then performing guided matching, however this process was not iterated until stable as suggested.

3.3 System Comparison

In this chapter the two reference AR implementations, built for this project have been presented: the 2D system and the 3D system. The 2D system is based on a 2D Fourier-based technique that is global in nature, meaning it uses the image as a whole to determine registration. This method’s use of phase data was appealing due to its robustness to illumination change. Unfortunately, extending this approach from use in a purely 2D setting, where users can only view an augmented scene from one location, to a 3D system that provides freedom of movement, proved impractical. The amount of preparation required to set up the system was too great. In effect the algorithm’s inherent 2D nature was impossible to break. This means the algorithm is only suitable for use in AR systems that are intended to operate from fixed positions.

In contrast, the 3D system is natively capable of operating with 6DOF. This

approach makes use of local features to automatically construct a 3D model of the scene that will be augmented. Despite being complicated to compute, the reconstruction provides a simple mechanism to position the virtual content in the scene as well as an uncomplicated process for augmenting a live camera feed. This process is simple enough to be performed in real-time as demonstrated by Skrypnik and Lowe's [SL04] implementation, which achieved a frame rate of 4 fps when using a 3D model comprised of 5000 3D points and video images captured at a resolution of 640×480 . Skrypnik and Lowe's system was built in 2004 thus with the increases in computational power since then performance would likely be even higher now.

The crucial issue when applying any local feature based approach to a computer vision based AR system—including the other systems discussed in Chapter 2—is that they rely on a sufficient number of correctly matching features to work. This means that too many outliers or too few inliers obtained in the initial feature extraction and matching stage will prevent later stages from working. Thus an important factor is whether the feature extraction algorithms are sufficiently robust to cope with large variations in illumination or weather conditions.

In order to ensure a sufficient level of performance for this stage of the process an analysis into the effects of variable weather and illumination is required in order to ensure a level of confidence in the system that AR requires. To begin this analysis test data designed specifically to evaluate such conditions is needed. The following chapter details two data sources useful for analysis.

Chapter 4

Test Data

“If I had eight hours to chop down a tree, I’d spend six sharpening my axe”

Abraham Lincoln (1809–1865)

This chapter examines the test data used for analysing the performance of feature extraction and matching algorithms. Many different data sets exist for testing computer vision algorithms; few are tailored specifically towards weather and illumination conditions. The possibilities are further limited by the need for ground truth data to accompany the images in order to test performance and accuracy.

Many image databases are discussed in this chapter with two explored in detail. The first is the Columbia Weather and Illumination Database (WILD). The second is the Slovenian video Database (SloDB) which was captured during the course of this project. Both datasets provide still images captured over a range of weather and illumination conditions. SloDB also provides motion video enabling tests on data with controlled movement and environmental changes. Details on what data these systems provide, how they were captured, and how they can be used are discussed. The lessons learned in implementing the SloDB system are also presented.

4.1 Related Databases

There are numerous image repositories intended for testing computer vision algorithms. The type and range of images in each set are determined by the type of algorithm the data is intended to test: from facial recognition to optical flow and feature tracking. In fact facial recognition is one field that has acknowledged the

significant effect of illumination on vision algorithms. Adini *et al.* noted the degree of variability in facial appearance due to changes in illumination are the same as those induced by changes in viewpoint or facial expression [AMU97]. Therefore, changes in illumination have been included in image databases intended for facial recognition testing.

One example dataset is provided by the Carnegie Mellon University (CMU). This database has a substantial collection of images capturing people under different poses and illumination. Sim *et al.* [SBB03] describe the collection as consisting of over 40,000 images capturing 68 people “across 13 different poses, under 43 different illumination conditions, and with 4 different expressions.” The database is called the CMU Pose, Illumination and Expression (PIE) database. The PIE database was captured by a setup called the “3D Room” built by Kanade *et al.* at CMU [KSV98]. The setup consists of a large array of cameras, 13 of which were used to simultaneously capture the 13 poses of each participant. Controlled illumination changes are implemented by 21 Minolta flashes programmed to go off together with image acquisition. By using different combinations of cameras and flashes, a variety of poses under a range of illumination conditions is obtained. Ambient light sources are also available and add a further dimension to the spectrum of images produced by the flashes by utilising room lights both on and off. PIE was developed to assist in assessing the performance and robustness of facial recognition systems under changes in head pose, expression and illumination.

The Context Aware Vision using Image-based Active Recognition project is a video database containing footage of people in various scenarios. Scenarios such as people walking, meeting with each other, window shopping, entering and exiting buildings, fighting, and leaving packages in a public place [FSVC07]. There are two applications for the database, the first of which is testing video surveillance systems designed to analyse video streams for behaviour which may be of interest, for example people fighting or committing crimes. The other potential application is for marketers who might be interested in analysing people’s behavioural patterns in shopping malls *etc.* in order to provide information which could potentially boost sales. The database is useful for testing computer vision algorithms designed for these applications, as they provide a variety of common circumstances, yet under controlled conditions.

There are several other databases containing images of the outdoors. The

Computational Vision Group at the California Institute of Technology (CALTECH) has a selection of images of Mt. Wilson [Per] taken at different times of day, from roughly the same position on a roof top. The images have no ground truth data and each image is not precisely aligned to those of a subsequent day, making direct comparisons of images of the same scene under different conditions difficult. The collection is useful as it contains various views of a mountain range, and covers a sufficient period of time to encompass some seasonal variations. Figure 4.1 shows some images from this collection. Figures 4.1(a)–4.1(c) show variations in illumination from roughly the same season. Figures 4.1(d) and 4.1(e) show variations in season—notice the leaves on the tree on the right of the image in 4.1(e), and the same tree without leaves in 4.1(d).

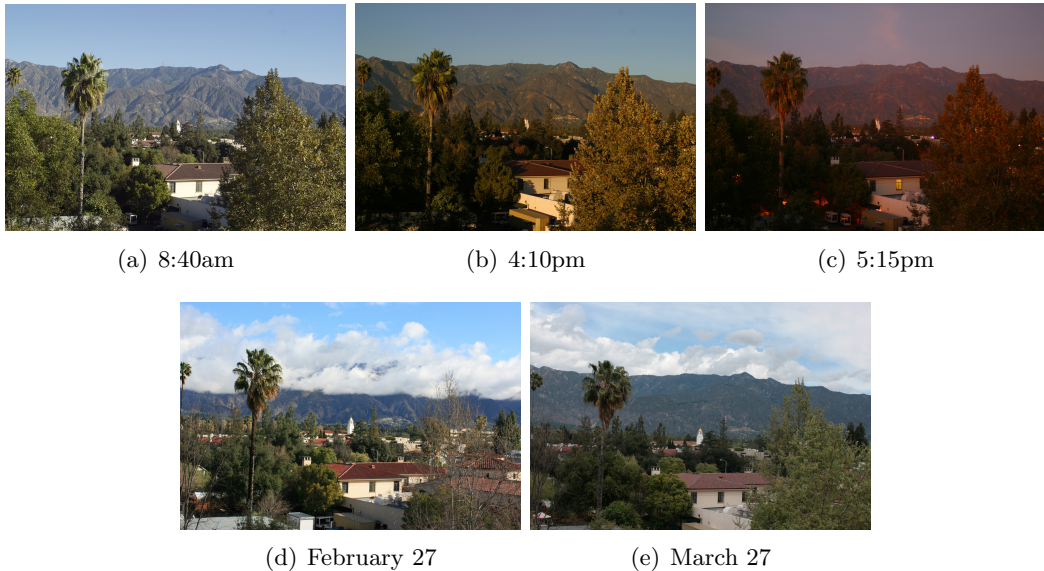


Figure 4.1: Images of Mt. Wilson, California, USA. (Images courtesy of Pietro Perona – CALTECH)

The Natural Stimuli Database captured by Hateren and Schaaf [HS98] is a collection of over 4000 images of various outdoor scenes and objects. The intended application was for testing computational models of simple cells which make up the human visual system: techniques such as independent component analysis. The set allows testing of the response of these simple cells to stimuli generated by this assortment of images. Multiple images from a given scene type were captured, typically 100–200 consecutive images. These images allow the possibility for image matching applications, however most images contain no overlap, therefore would not be appropriate. Of the images that do overlap, there is no extrinsic camera calibration data, therefore their use in the context of evaluating the performance of

feature extractors is limited. In addition, the illumination during capture was not the focus of this collection, therefore there is an insufficient range of illumination change to warrant testing.

In their comparison of local feature descriptors Mikolajczyk and Schmid examined the performance of a large range of descriptor techniques [MS05]. To do so, a set of images were utilised which demonstrate viewpoint changes when viewing a planar scene, scale and rotation changes, image blur, JPEG compression, and illumination. The same dataset was also used by Mikolajczyk *et al.* in a comparison of affine region detectors [MTS⁺05]. There are six images under each transformation category with ground truth homographies relating all images to the first image of the set. The illumination change set is produced by a somewhat artificial means. The aperture on the camera is changed to let less light through; this simulates the drop in illumination often present with changes due to time of day. This approach only produces uniform changes in illumination, as opposed to the more realistic non-uniform changes.

4.2 WILD

A useful database for evaluating natural feature tracking algorithms is WILD [NWN02] built by Narasimhan *et al.* from Columbia University. This database is comprised of images taken from a fixed perspective. Images were captured every hour for the period of approximately one year. This database improves on the others by capturing the exact same scene under virtually every illumination and weather condition possible. In order to be a valid comparison when evaluating the impact of illumination or weather effects, all other variables must remain fixed or must be accounted for. For example, a change in the viewpoint between two images being compared will make it difficult to establish whether a change in performance is due the weather effect being tested or the viewpoint change. To avoid bias in any test results, factors such as viewpoint position and orientation must remain constant, with the only variability caused by environmental changes.

The scene captured in WILD is a city scape of New York and is shown in Figure 4.2. Buildings in the scene range from about 20 metres to 5 kilometres away from the camera. This scene depth allows for pronounced effects from haze and fog to be visible, and therefore tested. In contrast, a scene that is very close to the camera witnesses little or no haze or fog, as the degree of scattering through



Figure 4.2: The image on the left was captured under clear weather. The image on the right was taken under misty conditions with light rain. The zoomed in portion of this image shows an example of surface appearance change due to weather effects

these mediums is a function of the distance light has to travel through the medium to the camera.

Weather conditions captured during this time include: clear and overcast weather, haze, fog, mist, and rain. A large range of illumination changes were also captured. These were induced by changes in the time of day, from inter-reflections or from point light sources within the scene. The duration of operation means variations in season were also captured. Such seasonal variations often manifest as changes in illumination, due to differences in the sun’s position during seasons and changes in atmospheric conditions. Narasimhan *et al.* describe their motivation for the collection of this database as: “Ultimately, vision algorithms are expected to work robustly in outdoor environments. This necessitates a principled collection and study of images of an outdoor scene under all illumination and weather conditions” [NWN02].

4.2.1 Capture Setup

The camera used was a KODAK Professional DCS 314 together with a 24–70mm zoom lens. The camera was mounted on a Pan Tilt Unit (PTU), and housed in a box coated with black paint on the inside to prevent reflections. The glass panel on the front was made of anti-reflection glass and the entire unit was mounted on a panel attached to a window. The apparatus is shown in Figure 4.3.

The images were captured at a resolution of 1520×1008 pixels with 10 bits *per* pixel *per* colour. In order to capture both large and subtle changes in illumination, High Dynamic Range (HDR) images were computed for each image by combining



Figure 4.3: Apparatus used to capture WILD images (Images courtesy of Srinivasa Narasimhan – Carnegie Mellon University)

images taken using an exposure bracketing technique. Exposure bracketing involves taking a series of images both under and over exposed relative to the auto exposure settings determined by the camera as being optimal. The exposure controls the amount of light received by the sensor and can be adjusted by either altering the aperture or the shutter speed of the camera. The aperture acts like the iris of an eye, and controls the amount of light let in. The shutter speed controls the duration of exposure; shorter durations allow less light onto the sensor, while longer durations allow more. In the case of WILD the camera aperture remains constant and the shutter speed is varied to produce different exposures. Longer exposure times result in more accurate and detailed readings in dark regions of an image, but allow too much light through for light regions resulting in over-saturated and over-exposed images. In contrast, low exposure times result in the most detail in light areas with little data in dark regions and produce underexposed images. HDR images exploit this complementary nature, and by merging the images produce an image with more detail over the entire scene.

The first image is captured using an auto exposure settings determined by the camera. Two under and overexposed images are then captured relative to the measured auto exposure settings. The combination of these under and over exposed images produce HDR images with an effective dynamic range of 12 bits *per* pixel *per* colour channel; an increase over the 10 bit dynamic range available from a single exposure image. The intent of these images, as described by Narasimhan *et al.*, is to “capture both subtle and large changes in illumination and weather” [NWN02]. Photographer Sean McHugh [McH] explains the benefits of higher dynamic range are that “your final photo may still improve from a side benefit: decreased shadow

noise. Ever noticed how digital images always have more noise in the shadows than in brighter tones? This is because the image’s signal to noise ratio is higher where the image has collected more of a light signal. You can take advantage of this by combining a properly exposed image with one which has been overexposed.”

4.2.2 Ground Truth Data

The WILD database contains ground truth data covering multiple aspects of the system. For each image captured, weather information is recorded from the National Weather Service. As described by Narasimhan *et al.*, this data “includes information about sky condition (sunny, cloudy), weather condition (clear, fog, haze, rain), visibility, temperature, pressure, humidity and wind” [NWN02].

In addition, the camera was calibrated using a planar checkerboard pattern and a calibration toolbox implemented by Bouguet [Bou]. Ground truth data on the structure of the scene was obtained using information supplied by the United States Geological Survey. This, in conjunction with specialised software, allowed computation of distances between any two scene points yielding an accuracy of 1 metre. Narasimhan *et al.* note that depth information is not available for every pixel of the image. However, since the structure of the scene consists mainly of planar buildings, an approximate model could be constructed. This, combined with a knowledge of the time and date each image was captured, together with the known longitude and latitude of the system, allows accurate computation of the relative position of the sun and moon to the apparatus.

4.2.3 Limitations

Despite the static nature of the apparatus, minor image misalignments still exist due to various factors. For instance the authors report misalignments due to the weight of the apparatus causing it to slowly bend over time [NWN02]. In addition, the potential for misalignments was introduced by equipment malfunctions, which necessitated the apparatus being removed, repaired, and subsequently reinstalled. Even examining images from the same day shows minor differences in alignment, potentially caused by temperature fluctuations inside the camera box.

Another potential limitation is the scene itself. The WILD scene is perfect for testing purely urban environments, however this scene contains very few natural elements, making it an unrealistic test bed for algorithms intended to work in



Figure 4.4: The image on the left shows a daylight image with snow falling. The 6pm image on the right shows snow falling as well as an accumulation of snow on the tops of buildings

suburban settings. The selection of images with snow in them is limited. Figure 4.4 shows an image with snow falling, however few daylight images exist which show significant amounts of cumulative snow; *i.e.* the white ground associated with snow fall as shown in the right hand image of Figure 4.4.

In addition, although WILD covers all four seasons, the urban scene type means variations due to seasonal changes are mostly limited to illumination differences with only isolated portions of the scene changing in appearance due to snow *etc.* A more extensive study into seasonal effects would require a less urban setting, or at least being able to see the ground where snow is more visible. The most notable changes in appearance between seasons occur in nature, where trees undergo significant changes in appearance due to leaves falling off and then later growing back.

Inherently these still images cannot be used to test any algorithms which rely on motion, for example optical flow techniques. Indeed, the WILD scene contains no moving objects, *e.g.* no moving people or cars. This can be seen as both an advantage and a disadvantage. Movement in the scene could confuse an algorithm thus introducing bias; conversely, most typical applications of computer vision will be in scenes containing motion, therefore ideally, controlled movement would be beneficial to include as part of the database.

4.2.4 WILD Selection

The WILD database is comprised of a large collection of HDR images formed from 5 images taken under different exposures. These HDR images, however, present several problems when used for evaluating the performance of feature extraction and matching techniques. The first is that these images are not representative of typical cameras used for computer vision, meaning the results might not be a fair

representation of expected performance. Most cameras currently used for computer vision use an 8 bit Charge Coupled Device (CCD) sensor, meaning they capture a range of 256 possible intensity values *per pixel per channel*; a substantial difference from the 4096 values possible with a 12 bit image. In addition, the implementations of most of the algorithms presented in Chapter 5, Section 5.3, only support 8 bit images. Furthermore, this restriction is also imposed by current Graphics Processing Units (GPUs), which utilise 8 bit images. This is an important consideration, given the trend of offloading vision algorithms to the GPU. Using GPUs provides substantial performance benefits due to their highly parallel nature.

The second problem is that these HDR images provide more detail in both dark and light regions than is typically possible in Standard Dynamic Range (SDR) images. This advantage becomes increasingly prominent when capturing scenes containing a large range of intensity values produced by both dark and light regions; WILD is an example of such a scene. In clear weather, direct sunlight produces very bright regions, while the structure of the scene produces dark regions due to shadows. These effects combined in a single frame produce a large range of intensity values the camera must capture. When using an 8 bit sensor, the camera’s dynamic range is often insufficient to retain the subtle details in the dark and light regions. Furthermore, even using a camera with a sufficient dynamic range, it is impossible to capture the entire range with one exposure, therefore multiple exposures are required as *per* WILD. The increase in detail in both light and dark regions afforded by HDR images provides an unfair advantage compared with images typically used in such a matching scenario, thereby introducing bias into the results.

In addition, previous projects analysing the performance of computer vision systems—such as those by Mikolajczyk *et al.* [MTS⁺05]—do not utilise HDR images, therefore utilising HDR images would preclude the possibility of making comparisons to the results from existing literature. For these reasons any images used from WILD must be converted to 8 bit SDR images and should approximate what would have been obtained if a standard camera utilising a single exposure had been used.

HDR to SDR

The HDR WILD images are stored in a file format using 16 bits *per pixel per channel*. This means all the images used from this database must undergo a transformation from 16 to 8 bits. When converting these 16 bit images down to 8 bits, a tradeoff

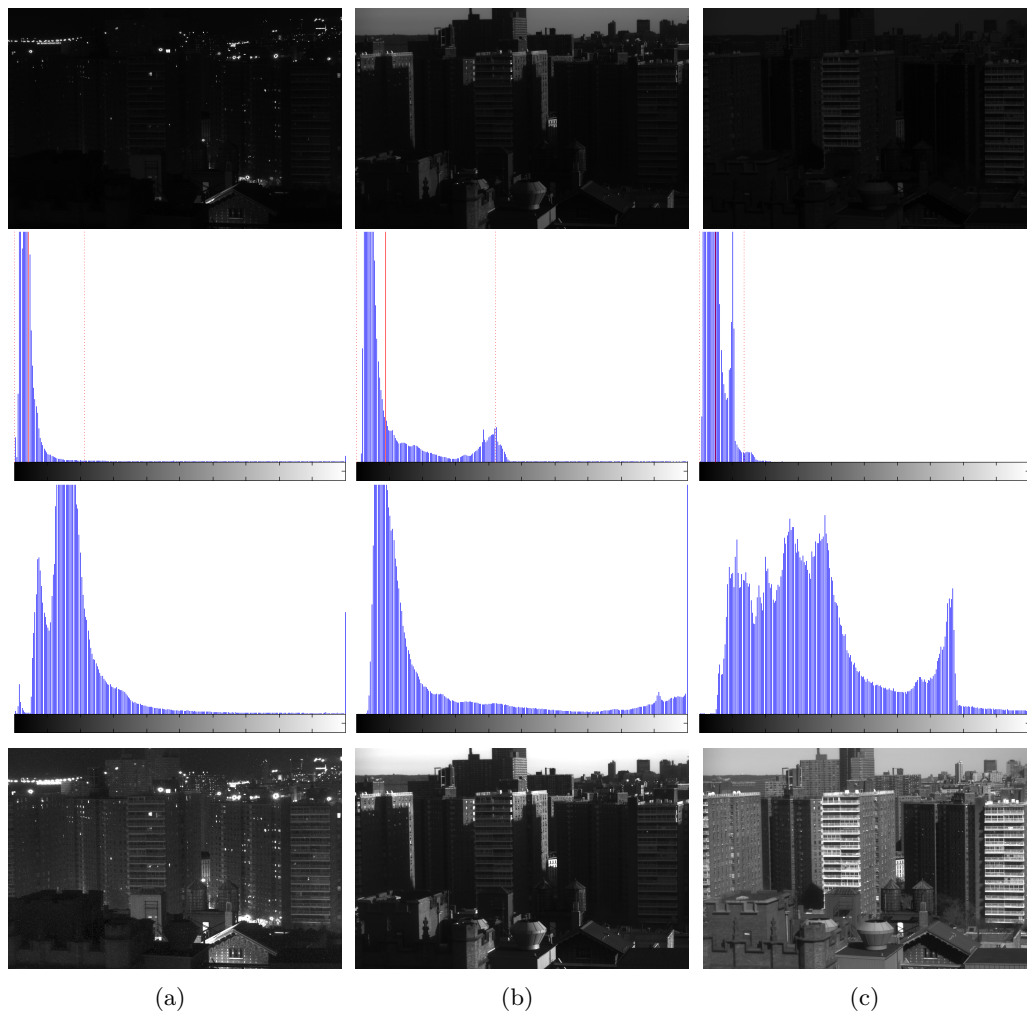


Figure 4.5: Histogram truncation method for correcting WILD images. The top row shows the original images with their histograms in the second row. The third row contains the histograms for the corrected images shown in the last row

must be made between how much of the intensity spectrum is retained, and the amount of detail inside this range that is retained. If the entire spectrum of values from the 16 bit images is maintained the resulting 8 bit image loses differentiation between pixels with similar intensity values, as these values are now grouped into the same intensity value. This loss means subtle changes in intensity are no longer represented. This typically affects dark regions the most, as they are the regions containing the smallest differences. If the conversion ensures subtle changes are preserved, there is insufficient scope for a large range of intensity values, resulting in either low or high intensity data being truncated, producing either very dark images or images containing over-saturated regions. Therefore, a balance between the range of intensity values represented and the resolution of these intensity values must be established.

To establish a balance between these two factors an approach was devised that can be seen as an approximation to the process an 8 bit camera would undergo. The procedure for this is to truncate the histogram for each image, where the upper and lower bounds for this truncation are determined as three standard deviations on either side of the mean value¹. This approximates an 8 bit camera that chooses an auto exposure setting based on the average light intensity in the scene and then captures an 8 bit range around this. To the author’s knowledge this approach for converting a HDR image to SDR is novel.

Three unmodified images are shown on the top row of Figure 4.5. The histograms for these images are shown on the second row. The solid red line shows the location of the computed mean. The dotted red lines show the locations of the lower and upper bounds of the truncation operation and represents the $\mu \pm (3 * \sigma)$ where μ is the mean and σ is the standard deviation. As the images are dark, the computed means tend to be low, meaning the lower bounds are always set to 0. A byproduct of this is that the resulting images tend to retain more detail in darker regions than in lighter regions. Any intensity data higher than the upper bound is truncated to the maximum value available, producing a spike in intensity on the right hand side of the resulting histograms shown on the third row. This approach can be thought of as the equivalent of a camera employing bottom-weighted averaging, where a camera calculates the average light intensity based on a region just below the centre of an image, as opposed to a centre-weight approach which uses a region around the centre of the image. The bottom portion of images are generally darker than the upper area, thus bottom-weighted averaging favours retainment of detail in darker regions over lighter ones.

The three images shown in Figure 4.5 are captured at *5am*, *7am* and *11am* respectively. The histograms for the unadjusted images on the second row show that only a fraction of the entire dynamic range is utilised by the data. The mean and $\mu + (3 * \sigma)$ lines shown demonstrate that the upper limit correctly adapts to the distribution of the unadjusted data. Notice the *11am* image; most of the intensity data falls in a narrow band on the left side of the histogram. In this case using the same limits as those placed on the *7am* image shown in the centre column would result in an 8 bit image that was unrealistically dark, and would not utilise the dynamic range available. The computed standard deviation correctly establishes

¹The effect of choosing three standard deviations is explored in Appendix A

that the distribution of intensity data in this image is small, thus it truncates only a small portion containing the bulk of the image data. The centre column for the *7am* image shows the intensity data is spread over a larger range of the histogram, and the distribution produces a larger standard deviation and consequently a larger range is truncated. The peak values on the right of the histograms on the third row represent all intensity values from the original histogram that are higher than the upper bound.

Final Images

The WILD selection generated using this technique is shown in Figure 4.6. The reference image shown here is the *2pm* image from April 4th. The remaining images are chosen from February 18th from *5am* to *6pm*. These days were selected as the visibility was high and conditions were clear.

4.2.5 Summary

In short, the WILD database provides a good test bed for many computer vision algorithms including the ones investigated in the next chapter of this thesis. The stability of the apparatus provides accurate results eliminating any effect a viewpoint change might have on a computer vision system and permits a focused analysis on the influences of illumination and weather.

Despite this, there is still additional scope for other databases. As computer vision research advances, the algorithms can operate in an increasing range of scenes. Therefore, the range of test data should also increase to ensure more difficult, yet realistic, circumstances are accounted for, thus progressing towards the goal of algorithms which work under virtually any circumstance.

Furthermore, a logical extension to WILD is to provide the ability to test both a viewpoint change and an illumination change simultaneously. A further test on illumination would thus be a test combining both illumination and positional differences, likely compounding the effect of either change in isolation. For example, examining performance when both an illumination and a viewpoint change exists between any two images. This leads us to our next dataset, SloDB, which was developed during the course of this project.



Figure 4.6: The selection of WILD images taken in clear conditions used for testing. (a) is the reference image used. (b) onwards are images from the 18th of February at the time indicated.



Figure 4.7: Panoramic View from Window of Computer Vision Lab, University of Ljubljana, Slovenia

4.3 SloDB

SloDB captures a scene in Ljubljana, Slovenia over a period of three weeks. This database differs from WILD in that rather than providing only still images, SloDB captures 25 fps video. This adds a further dimension to that of WILD that permits testing of either solely environmental changes without viewpoint differences, as *per* WILD, or a combination of viewpoint and environment changes. It does this by providing ground truth data for every frame of video, storing the position and orientation each frame was captured at. This ground truth data distinguishes SloDB from other datasets—such as the dataset from CALTECH—that contain somewhat arbitrary, non-repeatable viewpoint changes with no ground truth data available. It also provides a larger, more methodical coverage of weather and illumination variations within a scene than most other databases, and is similar in this respect to WILD.

Another significant difference between WILD and SloDB is the structure of the scene itself. The SloDB scene is a view from the Computer Vision Lab at the University of Ljubljana, Slovenia. A panorama of the scene is shown in Figure 4.7. The scene was captured during winter and contains various types of buildings, trees, cars, people, mountains, and hills. Trains are sometimes visible through a gap in between buildings. Most of the environment is fixed; however there is a moderate amount of movement due to traffic on the road and car park which are visible, as well as people coming and going. In contrast, WILD contains many tall buildings with no natural scenery and does not contain any movement.

Weather conditions recorded in this scene range from clear weather as shown in Figure 4.7, through to dense fog. The database captures virtually every time of day in order to provide a range of videos with varying illumination.

During each video capture the exact same motion sequence is executed. As the footage is from the same location and follows the same path, comparisons can be made between captures taken at different times or under differing environmental conditions. Because the video is captured at a smooth frame rate, testing of applications such as human or car tracking, motion detection, or optic flow algorithms is also feasible.

Each capture run includes a variety of different types of motion. The system goes through seven distinct types of motion. The first slowly moves the position of the camera to the left while the camera pans to the right, the system then changes direction and moves to the right while panning left. The second stage involves a faster panning left and right motion, while the position remains constant. Next, the camera tilts up and down, also remaining in the same position. The fourth stage involves the camera performing a diamond shape motion where the camera moves up and to the left, then up and to the right, then down and further to the right, and finally down and to the left bringing it back to its start position. Stage five involves moving to the left while tilting down, then moving to the right while tilting up. Stage six consists of a backwards movement while tilting up, then moving forward while tilting down. Lastly a series of motions is performed which keeps the orientation constant and only varies the position. The camera moves left and right and performs a motion that simultaneously moves forward and to the right. Additionally, some of these motions are performed at different speeds, increasing the usefulness of the database for testing applications where speed of movement may also influence the results.

Davison *et al.* [DCK04] observed that wider FOVs can help to stabilise camera pose estimation, therefore in SloDB two different FOVs were captured. This allows further testing on the effect FOV has on algorithm performance, particularly in the presence of environmental changes. 97 videos were captured; 75 using the larger FOV while the remaining 22 were captured using a smaller FOV. Two images captured at these different FOV values are shown in Figure 4.8.

4.3.1 The Capture System

The capture system was constructed by the author during the course of this project specifically to capture this database. The capture apparatus is shown in Figure 4.9. Positioning and orienting the camera was handled by an Arrick Robotics XY

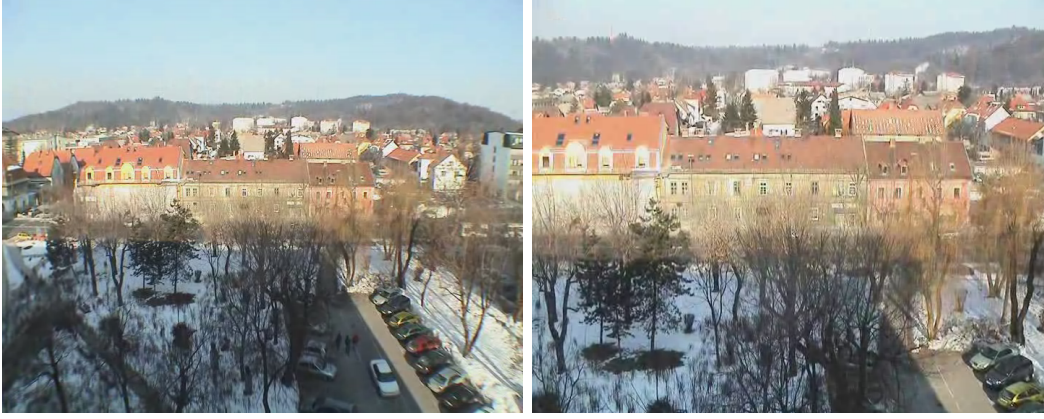


Figure 4.8: The image on the left shows an image captured using the wide angle lens. The image on the right shows an image from the same position and orientation captured without the wide angle lens resulting in a narrower FOV.

positioning table with MD-2 Dual Stepper Motor system, and a Directed Perception PTU respectively. A similar apparatus was used by Gomez *et al.* to assess the accuracy of a given camera model used for AR in the presence of typical calibration errors [GSB05]. Their apparatus also consists of an XY positioning table with a camera mounted on a PTU. The system moves along a predefined path, capturing images as it goes. The known camera pose at each image is known from the positioning table and PTU, thus this can be compared with the pose computed from 2D-3D matches on a known calibrated scene being captured by the camera.

The apparatus used for SloDB is similar in construction, however, a Sony DCR-TRV70 DV HandyCam is used for capturing, while an Intersense IC2 provides orientation information. The IC2 was used during capture to provide additional orientation data. However, upon analysis of the results, it was found to be too unstable to be useful, even when elevated above the camera to reduce the influence of magnetic distortions.

The XY positioning table is controlled using an API provided by Arrick robotics. The API was modified so that the current position of the motors is stored *per* step. The video is captured at a resolution of 720×576 (PAL) with a frame rate of 25 fps using a Sony VCL-0637 wide angle lens that provides a $0.6\times$ magnification. Footage was also recorded without the wide angle lens, yielding a smaller FOV. The FOV for each lens is 53° for the normal lens and 89° for the wide angle lens.

For each frame the following information is stored: frame number, x and y position measured from the positioning table, yaw, pitch and roll from the IC2, time of frame capture, the last known pan and tilt position of the PTU, and the times

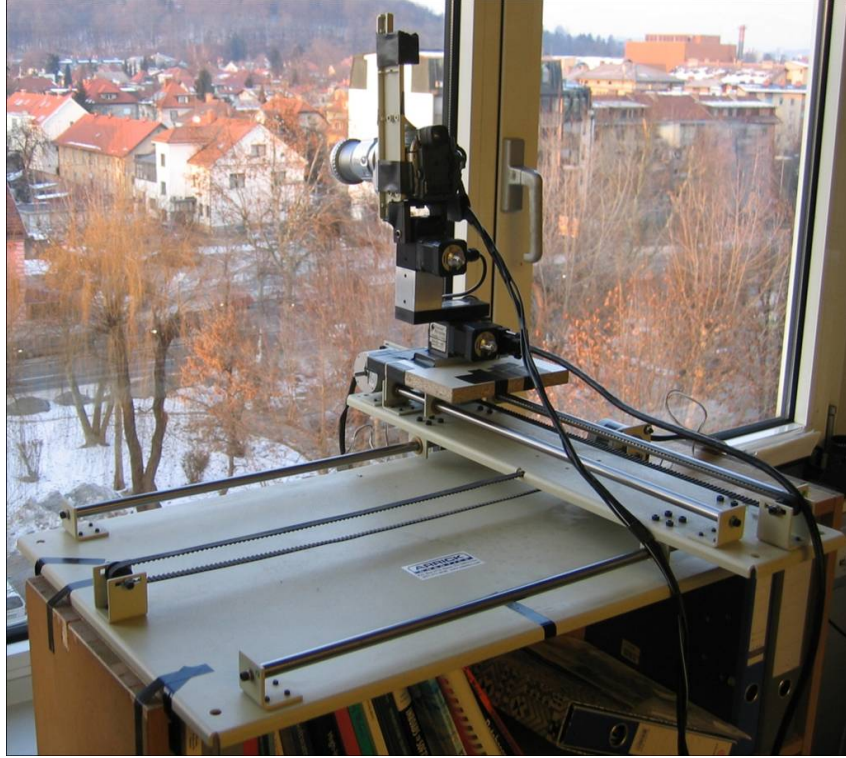


Figure 4.9: SloDB capture system in situ

at which those readings were taken. The stage of the movement is also stored; this breaks the motion sequence into stages based on the type of motion allowing easier retrieval of aligned frames. In addition, at the beginning of each pan-tilt motion the time and frame number at which the movement started is recorded. Using this information together with known parameters such as the speed, acceleration, and position of the PTU, which are precisely controlled, the orientation can be accurately predicted. This provides a secondary method of determining orientation in addition to the measurements recorded directly from the PTU.

Two methods of establishing the orientation are used because the first method has a limited accuracy. The PTU was controlled using a serial interface. Attempting to retrieve the current position of the PTU *per* frame was not feasible, as doing so caused the system to move erratically. For this reason the PTU position is queried every 30ms, meaning for any given frame the recorded pan and tilt position could be off by at most 30ms worth of movement. Each time the PTU was queried the results were stored along with the time of the query. This way, for every frame recorded, the last known PTU position and the time at which that result was taken is known. Using the time at which the current frame was captured, the orientation results can be interpolated between frames during post processing. This provides

accurate orientation positions for each frame. During this step, the secondary source of orientation information on when a pan tilt motion started is also incorporated to further increase the accuracy of the orientation data.

4.3.2 Limitations

There are several limitations of the SloDB system and the data captured. In contrast to WILD, no ground truth data on the weather conditions at the time of capture are available. It is envisaged that each video could be tagged at a later stage if desired.

Furthermore, due to the relatively short duration of the capture system being in place (three weeks), significant seasonal changes were not captured. An interesting result would have been videos capturing the gradual change from winter to spring and the subsequent melting of the snow on the ground and to examine any impact on performance this might have. There was also no rain, and no snow fall captured during this time, meaning tests under these conditions are not possible.

The most significant limitation of this dataset is the reflections present in the images, notably the reflections from parts of the metallic camera. These reflections can be seen as the ring shaped objects on several of images in Figure 4.10. Performing feature extraction on these images revealed that no spurious features were detected due to the reflection, therefore their impact on performance is negligible.

4.3.3 SloDB Selection

The selection of SloDB images used for testing is shown in Figure 4.10. These images were extracted from the database using the ground truth information to determine matching frames from videos captured at different times. Each image is manually registered to the reference frame to ensure precise alignment.

4.3.4 Lessons Learned in Setting up SloDB

In setting up this system a number of important lessons were learned. These lessons are presented here to assist future research implementing a similar system.

Ensure the Apparatus is Light and Low

Everything mounted on the platform should be as light and as low as possible. All these things contribute to reducing sway and jitter in the video.

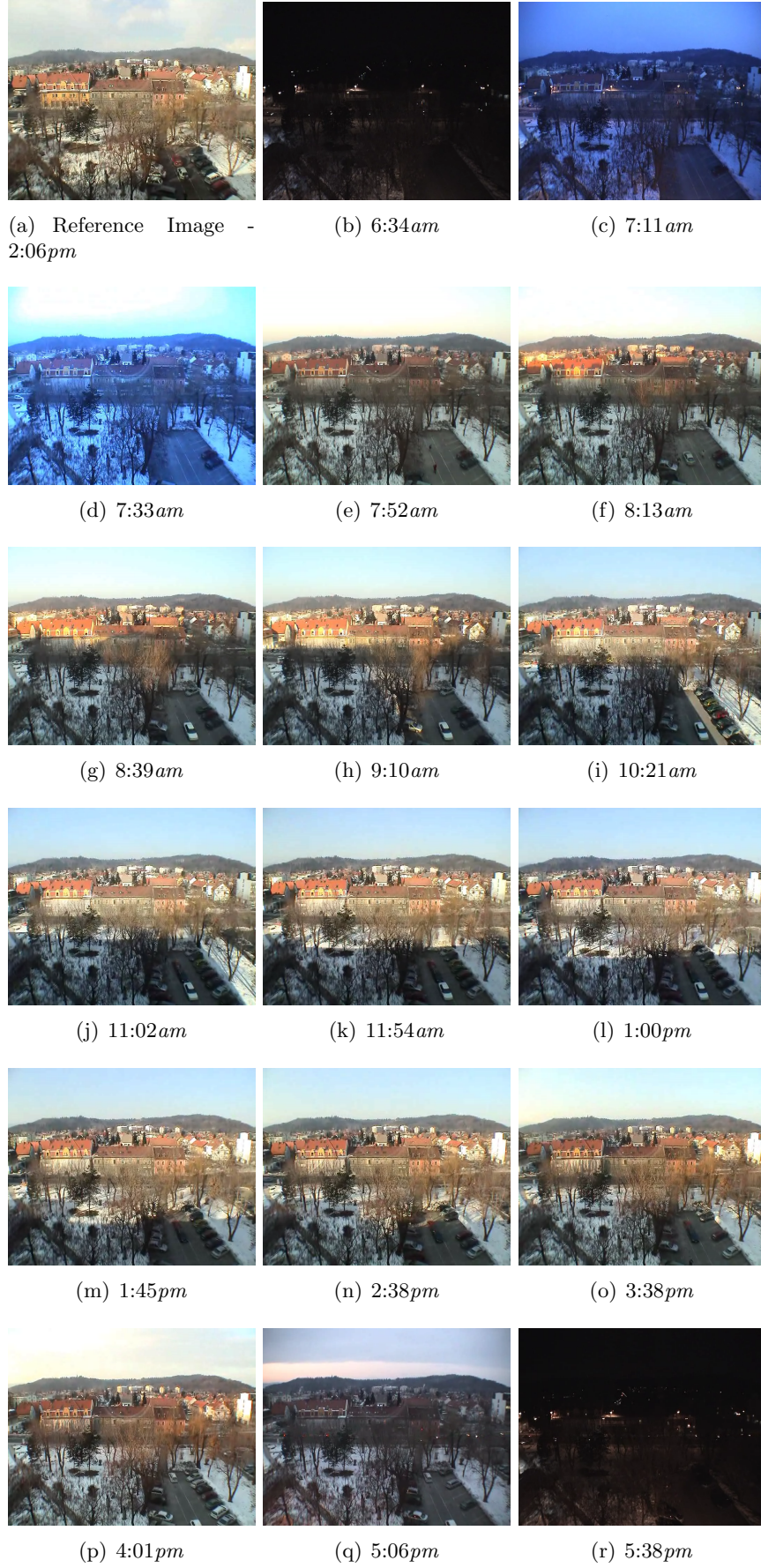


Figure 4.10: A selection of images from SloDB registered to each other

If the unit is quite high, as was the case in the SloDB capture system, the apparatus has the possibility of swaying, particularly after the XY table changes directions. The PTU used in this project had a large metal block which displaced the top motor from the bottom so that the top motor was not directly above the bottom one (it was displaced to the left as shown in Figure 4.9) and it was unnecessarily elevated a few centimetres above. If the top motor was mounted directly above and in line with the bottom camera, sway would have been reduced substantially. The IC2 could also have been removed; it added to height but ultimately did not contribute to the results, as its performance was inadequate.

The weight of the components mounted on the platform should be as light as possible. An early implementation of the SloDB capture system used a Point Grey Flea camera that weighed 45 grams; unfortunately due to other problems relating to the limited range of resolutions this camera provided, it could not be used and a Sony camera that weighed 740 grams was used instead.

Use a Camera with built-in PTU

Another enhancement would be a pan-tilt capable camera, where the camera has a built-in PTU. This would make the whole system a lot smaller and lower. The only requirement is that the unit can report its current position *per* frame so it can be recorded. If this cannot be obtained, the best that can be achieved is either from interpolated values between subsequent readings, or *via* a model of PTU motion obtained through knowledge of the speed, acceleration, start time of the motion, and the beginning and end position.

Selection of XY Table

A more modern positioning table might be better suited. The table used for SloDB required the host PC processor to be available almost exclusively to table requests in order to produce smooth movements. The host PC typically has several processor intensive jobs such as capturing and storing video as well as coordinating all the devices together. Therefore, reducing the burden on this device will ensure smoother movements and a higher level of confidence. A larger range of X displacement would also be beneficial. Y displacement is not so important as this can be approximated using camera zoom.

Minimise Cable Clutter

Due to the number of devices used in this setup, there were a large number of cables. Two for the camera for power and Firewire, two for the PTU, one for the IC2, and two for the XY table. Many of these cables were bundled together as shown in Figure 4.9. However, the number of cables meant an operator had to be present each time the system ran to prevent the cables interfering in the system's operation. A reduction in the number of cables will make it easier to implement an autonomous system. For example eliminating the IC2 and using a camera with built in PTU, would reduce the number of cables from seven to four where, most importantly, the cables on the XY platform that move the most are curtailed. A useful addition would also be something to hold the cables while the system runs to prevent them getting caught or tangled while still allowing some movement.

Ensure System Operation is Completely Autonomous

For the system to be useful for long periods of time it must be completely autonomous. Otherwise the amount of supervision will be prohibitive and might prevent capture during the night or at times when an operator is not available. Also, autonomous capture will allow a more frequent and methodical coverage of each day. As mentioned above, the number of cables in the SloDB capture system prevented autonomous operation and required user supervision to ensure the system did not become entangled.

Mounting Location Suggestions

The system tends to be fairly noisy, and given that it often runs for a few minutes each time (five minutes in the case of SloDB) and many times each day it is best to mount it somewhere it will not distract people. In addition, a location should be chosen where it will never be touched or moved. Choosing the distance from the scene is also important. If it is too far, the camera will have to move substantial distances on the XY table to produce noticeable image differences. In SloDB the mounting distance was too far, as movements along the X axis did not produce enough measurable differences in the image. The only time they did is when the camera was looking down at the parking lot, which was a lot closer to the camera than the buildings across the street and beyond.

Preventing Reflections

Find a location which can be as dark as possible so as to limit the reflections on the window. WILD uses its own mounting box with anti-reflective glass on the front. A similar weather proof box could be constructed for this apparatus; however, this is significantly more difficult due to the size of this system with the XY table.

Masking out all areas around the apparatus with black non-reflective paper or special paint which absorbs all light will eliminate the bulk of the reflections. Also ensure that any reflective objects on the apparatus itself are masked out. In the case of SloDB the metallic looking wide angle lens should have been masked out to prevent the reflection that is common in the videos.

Use a Hardware Encoding Camera

Although still images can be kept in raw form to eliminate any potential issues to do with compression artifacts, it is not so easy when using motion video as the amount of data being recorded makes capture and storage of the video difficult. Capturing raw video at these frame rates requires an extremely fast Hard Disk Drive (HDD), or another such medium. Under-performing HDDs will result in dropped frames and motion which is jerky and erratic. This project initially took advantage of a Point Grey Flea camera. This camera captured 30 fps at a resolution of 1024×768 meaning 67.5MB/second of video data needed to be written to the HDD. Although this was possible given the technology at the time the work was conducted, capturing at this rate caused control of the XY table and PTU to be adversely affected, producing erratic movements.

The only way to produce raw captured video would be to have a separate computer solely for video capture. However, this increases the complexity of the system, as synchronisation between computers must be considered. In addition, the problem of storing so much data becomes an issue. For this reason some form of compression is necessary. Using a camera which encoded directly into H.264 (or another advanced compression method) reduces requirements on HDD writing speeds and storage space, but may introduce artifacts due to lossy compression that may affect subsequent image processing algorithms. Alternatively, a dedicated hardware encoder can compress the raw video in real-time before writing it to disk. This would permit higher quality streams to be captured while not requiring the CPU to perform real-time compression on high resolution streams. The Sony DV

camera used in SloDB utilised DV compression. Although not the most efficient form of compression, this approach reduced the amount of data, enabling the computer to write the video to file without incurring dropped frames or needing further compression.

Scene Choice Suggestions

Optimally a scene will consist of objects both near and far. This provides significant depth in the scene, permitting effects such as fog or haze to be noticeable. Near objects produce significant differences in position inside the images providing more scope for object or feature trackers. Furthermore, significant object motion enables more precise 3D reconstruction formulation.

Ideally, the scene will also contain several different types of objects such as man-made objects like different types and sizes of buildings, natural features such as trees, mountains, and hills. Additionally, the presence of both static and dynamic portions of the scene will increase the applicability of the collection to several different research fields.

Use More Frequent Captures at Sunrise and Sunset

Illumination in the scene drops off quickly as the sun sets, therefore in order to capture a more gradual range of illumination change during this time, captures should be more frequent. In WILD for example, images are taken every hour. Often the difference from 5pm to 6pm is significant. At 5pm there is still a significant amount of light, while at 6pm it can be pitch black. During this transitional period it would be beneficial to capture images more frequently, perhaps every 15 minutes. The same principal also applies during sunrise.

Duration of Capture

The longer the period of time the apparatus is setup, the more useful the dataset. Performing capture over a year or more provides increased certainty of the coverage of all possible conditions that might arise in the scene. In addition, it ensures coverage over all seasons allowing for seasonal differences to be tested.

4.4 Arizona Test Set

The Arizona test set captures a scene overlooking the University of Arizona campus and the Santa Catalina Mountains beyond. The still images that make up the test and reference sets used for testing in this thesis are frames extracted from time-lapse videos produced by the Computer Science department from the University of Arizona [Tow]. Although no time information is available for when each frame from the time-lapse video was captured, each of the images in both the test and reference sets are in temporal order thus relationships between when the image was captured and the performance at various times can still be established.

It should be noted that these images are not evenly spaced as more images were taken from periods towards the start and end of the day when the shadows are changing the most. In reality, there would be a large period during the middle of the day where the shadows do not change so quickly because the sun is directly overhead, however, as this thesis is evaluating feature detection and matching performance under non-uniform illumination changes, periods containing the most variable illumination are of most interest.

Figure 4.11 and 4.12 show the images that constitute the test set. Each image from the test set is designated with an index from 1–22 as shown in the figure. The time-lapse video was captured on March 12th 2006.

Figure 4.13 shows the reference set that will be used during testing. The reference images are designated from reference 1–15 and were captured on November 30th 2006.

4.5 Summary

Many databases exist designed specifically for testing computer vision algorithms. Of these, few are tailored for the evaluation of algorithms under the range of conditions possible in the outdoors, especially given a requirement of precise alignment, which is only possible through a fixed camera or ground truth data on the camera’s position and orientation.

WILD is one such database which provides complete coverage of the variability in environmental conditions of a city scape in New York. This system is fixed in one position, and provides images with accurate alignment for evaluating the effect of weather and illumination on computer vision performance.



Figure 4.11: A selection of images from the Arizona time-lapse sequence registered to each other (images 1–15)

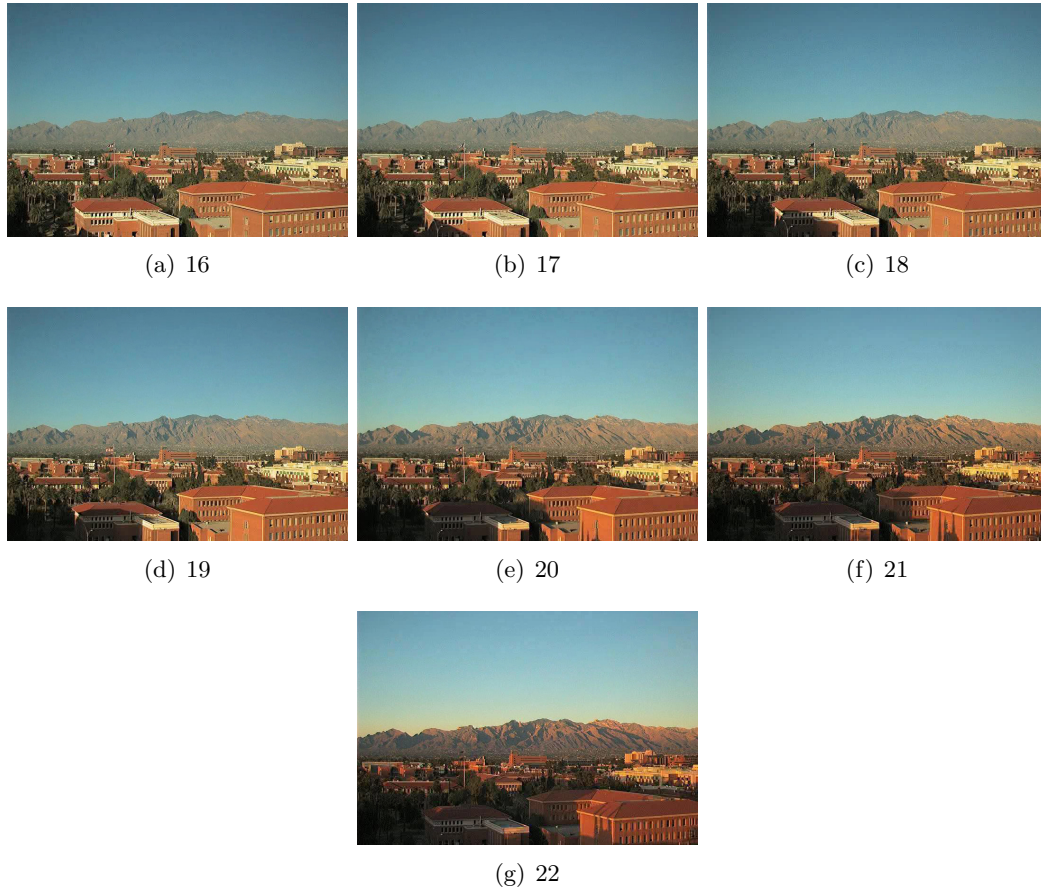


Figure 4.12: A selection of images from the Arizona time-lapse sequence registered to each other (images 16–22)

SloDB adds to WILD in that it provides a different scene type that can be tested. The scene captured is a mixture of urban and countryside in Slovenia. Natural elements include distant mountains and hills permitting horizon silhouette algorithms such as that employed by Behringer [Beh98].

In addition, SloDB extends on WILD and tackles a substantially more difficult problem by capturing motion video of a scene while providing ground truth data *per* frame. Ground truth data allows tests not only under environmental changes, but also those effected by both environmental and viewpoint change. Camera motion also increases the usefulness of this database, as it can be used to test illumination effects on algorithms which use or require motion such as optic flow or video surveillance. The scene captured also contains moving objects such as people, cars, trucks, and buses, opening up fields such as human or object detection or tracking.

The amount of movement in the scene is sufficient for it to be detected, but should not adversely affect algorithm performance. The redundancy of the dataset



Figure 4.13: A selection of images from a Arizona time-lapse sequence used as reference images

is such that videos containing a range of movement can be selected.

In the context of AR, motion video in addition to still images is important from a testing standpoint. Still images are necessary for analysing the performance of the vision algorithm and ensuring a sufficient level of accuracy. However motion video is required to assess the consistency of the estimates from the vision system. Small inconsistencies in pose estimation manifest themselves as objects jittering over the subsequent frames, something only visible with video data.

There is significant scope for additional image databases in the future. Other scene types with a larger range of positional change and a larger range of seasonal variations will always be advantageous for evaluating the performance of vision systems. Only by ensuring that test data contains sufficient coverage of all conditions an algorithm will be expected to work in, can a high degree of confidence be obtained in its ability to operate successfully.

Chapter 5

Analysis

“No question is so difficult to answer as that to which the answer is obvious”

George Bernard Shaw (1856–1950)

5.1 Introduction

This chapter presents an analysis of the performance of the feature extraction and matching stages of a computer vision based AR system under a range of illumination conditions. In Chapter 3, two reference AR systems implemented over the course of this project were introduced: the 2D system and the 3D system. The underlying algorithms these systems used were the global Fourier-based image registration algorithm for the 2D system, and a local feature based approach called SIFT for the 3D system. To provide a more comprehensive survey into how illumination affects computer vision algorithms, an additional five feature detection algorithms will be tested in this chapter. It should also be noted that any of these new algorithms could have been used in place of SIFT in the 3D system of Chapter 3, as they provide the same functionality.

Three sources of data are used for testing the algorithms. The first two are the WILD database and the SloDB database, presented in Chapter 4. The third is the set of light change images used by Mikolajczyk *et al.* in their comparison of region detectors [MTS⁺05]. The inclusion of this dataset provides a useful comparison with previous work, of which the test framework is based. The usefulness of this dataset is limited, however, as the illumination changes are generated by artificial

means. This results in images effectively differing by uniform changes in illumination. This chapter instead focuses on non-uniform changes, therefore the inclusion of Mikolajczyk’s light database is simply for reference purposes.

The 3D system presented in Chapter 3 operates by matching the features extracted from the current, ‘live’ image, to a 3D model made up of features extracted from a set of images of a scene. Instead of matching the current image to a 3D model constructed from a set of images, this chapter evaluates the performance when matching one current image to a single reference frame. Typically, tests in this chapter involve matching a single reference frame to a series of images which are variations of the same view, but contain a spectrum of photometric effects. These results are directly applicable to the 3D system, and can be considered a scaled down test required to provide clear, meaningful analysis.

Three distinct but related stages are being tested here: feature detection, description, and matching. Table 5.1 shows these stages in the context of the overall operation of the 3D system. Feature descriptors describe regions determined at the detector stage. The matcher then uses these descriptors to find a region similar enough to be considered a match. SIFT—as described in Chapter 3, Section 3.2.2—detects features by identifying local extrema points in a pyramid of DOG images constructed from an image that is successively scaled.

3D Reconstruction	Live System
1) Image Capture or Existing Images	1) Image Capture
2) <i>Feature Detection</i>	2) <i>Feature Detection</i>
3) <i>Feature Description</i>	3) <i>Feature Description</i>
4) <i>Feature Matching</i>	4) <i>Feature Matching</i>
5) Two-View Geometry	5) Pose Recovery
6) Three-View Geometry	
7) Camera Pose Recovery	
8) 3D Reconstruction	
9) Bundle Adjustment	

Table 5.1: The stages highlighted correspond to those tested in this chapter

The most simple descriptor is a vector of image pixel intensity values in a region around the detected point. Cross-correlation can then be used to determine how similar two descriptors are, allowing correspondences to be computed. A more robust approach uses a histogram formed from the distribution of the pixel intensities in a region around a detected point; this makes the relative location of the intensities less important and can thus cope with small point localisation errors. SIFT extends

this by utilising histograms built using the orientation and magnitudes of pixel differences. This makes the descriptor more robust to photometric and geometric differences than simpler approaches using pixel intensity data, which are adversely affected by variations in intensity between images.

5.2 Background

The analysis in this chapter builds on the work of Mikolajczyk *et al.* [MTS⁺05] and Mikolajczyk and Schmid [MS05], in their evaluations of both feature detection and feature descriptor algorithms respectively. In their performance evaluation of region detectors, Mikolajczyk *et al.* compare six different detection algorithms [MTS⁺05]. The algorithms evaluated included an affine version of both a Harris (HARAFF) and a Hessian (HESAFF) point detector, the Maximally Stable Extremal Regions (MSER) detector, edge (EBR) and intensity (IBR) based region detectors, and a ‘salient region’ detector. The developers of each of the algorithms were also authors of the paper. Five of these algorithms are utilised in the analysis presented in this chapter and a detailed discussion of these algorithms is presented in the next section. A test framework for the evaluation of the performance of these algorithms was also presented and is used in this thesis; a description of this framework is given in Section 5.4. Test results from Mikolajczyk *et al.* showed that no algorithm performed the best in every situation, but that the best overall results were achieved by MSER, followed by HESAFF. MSER was found to perform well on images containing distinctive region boundaries, as did IBR. It was noted that HARAFF and HESAFF detect more regions than the other algorithms, making them more suitable for matching in the presence of occlusions.

Mikolajczyk and Schmid [MS05] conducted an evaluation of the performance of 10 local descriptor algorithms: “SIFT, Gradient Location and Orientation Histogram (GLOH), shape context, PCA SIFT, spin images, steerable filters, differential invariants, complex filters, moment invariants, and cross-correlation of sampled pixel values.” This paper built on previous work by the authors on this topic [MS03]. For histogram based descriptors such as SIFT, GLOH, PCA-SIFT, shape content, and spin images, similarity between descriptors was determined using the Euclidean distance as the distance measure. The Mahalanobis distance is used for steerable filters, differential invariants, moment invariants, and complex filters.

Furthermore, Mikolajczyk and Schmid compare the performance of three dif-

ferent matching strategies: threshold-based, nearest neighbour and distance ratio matching. As described by Mikolajczyk and Schmid [MS05], in the threshold-based approach “two regions are matched if the distance between their descriptors is below a threshold.” This means there could be several matches for any one feature. In nearest neighbour matching, a matching region is sought whose descriptor yields the smallest Euclidean distance, and where the distance between them is less than a threshold. Distance ratio matching is utilised by SIFT and is described in Chapter 3, Section 3.2.3. The distance ratio matching approach ensures the distance between the closest and second closest nearest neighbour matches are sufficiently distinct to be confident the region is the correct match.

For consistency, both papers make use of the same data set, which contains images with rotational differences, rotations with scale changes, viewpoint changes of a planar scene, image blurring, JPEG compression, and light change. For most of the test data two different scene types were used, one containing structured elements with distinctive boundaries, the other containing repeated textures. This allowed testing of the influence of both different scenes types and of image transformations.

Mikolajczyk and Schmid’s results showed that the GLOH descriptor algorithm performs well for structured scenes and SIFT performs well for textured scenes. This indicates SIFT has good discriminating power. When evaluating the performance of the descriptor algorithms on the set of light change images, GLOH performed the best, followed by SIFT. Many other approaches such as cross correlation, steerable filters, and differential invariants did not fare so well. The authors concluded that the robustness and distinctive nature of SIFT-based descriptors produced the best results; specifically the GLOH and SIFT algorithms. They also noted that in the case of SIFT the distance ratio technique for matching produces better results than the other approaches. Three types of error that influence the resulting descriptor for a region were identified: “the region error, the localisation error, and the error of the estimated orientation angle.” Based on these results the tests in this chapter make use of the SIFT descriptor and the distance ratio matching approach.

Mikolajczyk and Schmid [MS05] identify that changes in image illumination are typically approximated by an affine transformation such as: $aI(x) + b$. This is essentially a linear mapping using a scaling factor a applied to the intensity values of image I with an offset of b . Many approaches, including SIFT, normalise the descriptor to eliminate the effects of illumination changes. Normalising the

descriptor vector to unit length, effectively eliminates contrast differences a , while the brightness b , is eliminated through the use of pixel differences. The approach makes some attempt at accounting for non-uniform changes in illumination by thresholding the values to ensure highly saturated regions do not overly influence the descriptor.

A limitation in approximating illumination change as an affine transformation is that non-uniform changes in illumination are not represented. The test data used in both the detector [MTS⁺05] and descriptor evaluations [MS05] focus more on testing geometric transformations in order to assess the algorithm’s robustness to affine transformations. The illumination test that is present is produced artificially, thus may not be indicative of real world performance. The analysis in this thesis seeks to build on the successful test framework provided by these authors, and provides results and analysis of a larger and more methodical collection of real world images exhibiting variable illumination.

5.3 Algorithms Evaluated

Recently, a substantial body of work has been presented on detecting image regions that are ‘covariant’ to a given class of transformation [MCMP02, MS04, MTS⁺05, TG99, TG00, KZB04], meaning the detected regions vary according to the transformation present without *a priori* knowledge. Typically, in the area of wide baseline matching the regions are designed to be invariant to affine deformations. This ensures extracted regions from differing viewpoints cover the same physical region of the viewed object, meaning the descriptors are more similar, thus improving matching and increasing the robustness of the approach to viewpoint changes. Matas *et al.* identify that standard regions that only model translation and rotation are insufficient to cope with the local image deformations introduced by significant viewpoint changes. They state “correspondence cannot be therefore established by comparing regions of a fixed (Euclidean) shape like rectangles or circles since their shape is not preserved under affine transformation” [MCMP02]. Once regions that are covariant to affine deformations are established, the regions can be normalised to a consistent shape and size. The descriptors of these regions are then effectively invariant to affine deformations.

This process is shown in Figure 5.1. Figures 5.1(a) and 5.1(b) show two images separated by a viewpoint change. The regions shown are not affine covariant, like

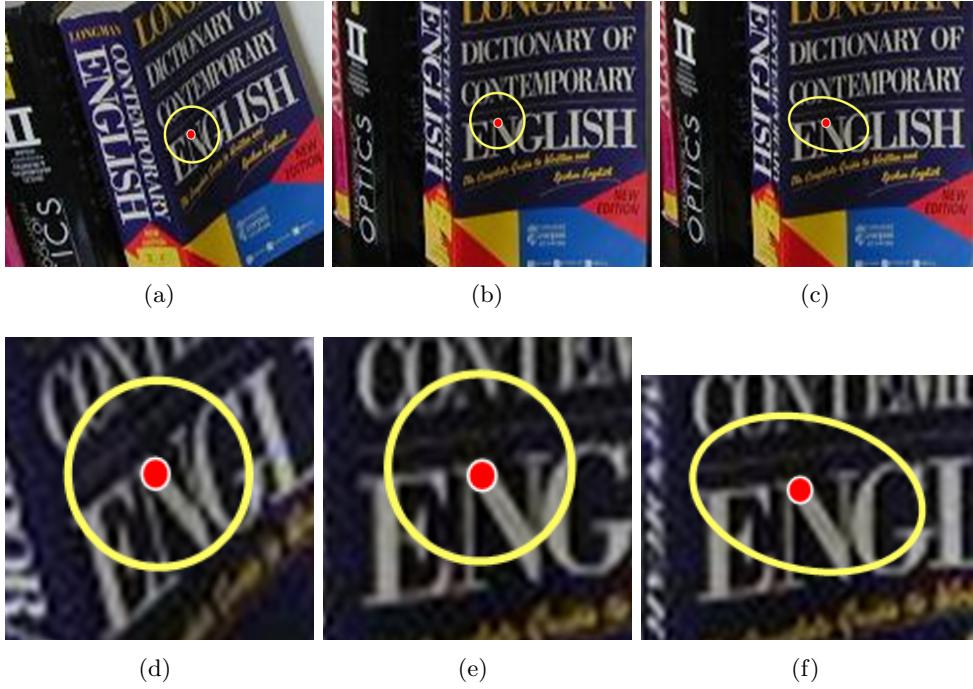


Figure 5.1: Affine invariant regions. (Images used courtesy of Jiri Matas – Czech Technical University, Prague. Images reproduced from [MTS⁺05])

those produced by SIFT. Figures 5.1(d) and 5.1(e) show magnified images of these regions. Notice how the region covered in the second image is now quite different to that of the first due to the viewpoint change, even though the feature location is the same. This means the descriptors for these regions will be dissimilar, decreasing the likelihood of obtaining a match. Figures 5.1(c) and 5.1(f) show an affine covariant version of the region in the second image. In these two images the physical region covered is similar to that of the first image.

The only drawback of affine covariant approaches is that either the camera motion between the images being matched must be purely rotational, or the extracted regions must lie on a plane in order for the deformation to be accurately described by the affine model. However as Mikolajczyk and Schmid point out [MS04]:

“Although smooth surfaces are almost never planar in the large, they are always planar in the small that is, sufficiently small surface patches can always be thought of as being comprised of coplanar points. Of course this does not hold if the point is localised on a depth boundary.”

There are many planar surfaces in both the WILD and SloDB scenes. Therefore these datasets are well suited to testing affine covariant algorithms. In addition, as the tests conducted in this chapter use images that contain no viewpoint changes,

non-affine approaches are equally suited.

In the following section the six feature detectors evaluated in this chapter are discussed. These algorithms include: SIFT, MSER, HARRAFF, HESAFF, EBR and IBR. Mikolajczyk *et al.* used an algorithm called “Salient Regions” [KZB04] instead of SIFT. The Salient Regions algorithm is left out of this analysis as Mikolajczyk *et al.* found that the algorithm obtained low scores during their evaluation. SIFT was used in this study as it is representative of the performance of a non-affine region detector. SIFT is better suited to cases where local image deformations cannot be approximated by planar patches. Lowe [Low04] notes that while SIFT is

“not fully affine invariant, a different approach is used in which the local descriptor allows relative feature positions to shift significantly with only small changes in the descriptor. This approach not only allows the descriptors to be reliably matched across a considerable range of affine distortion, but it also makes the features more robust against changes in 3D viewpoint for non-planar surfaces.”

It should be noted that for three of the four algorithms the output type is an ellipse. This is not the case for EBR, MSER and SIFT, which use parallelograms, arbitrary regions, and circles respectively. In the case of SIFT a circle can easily be represented in ellipse form, as a circle is a specialisation of an ellipse. An ellipse is fitted to the shapes generated by EBR and MSER. This approximation loses some information, as identified by Mikolajczyk *et al.* [MTS⁺05] who note that performing this ellipse fitting should not be used in practical applications as “orientation information is lost.”

Finally, the performance of the Fourier-based image registration algorithm used by the 2D system of Chapter 3 is also evaluated in this chapter. This global image technique provides a comparison to the six local feature approaches discussed thus far. The six local feature detectors will now be discussed in detail.

5.3.1 SIFT

The Scale Invariant Feature Transform algorithm—called SIFT—is described in detail in Chapter 3, Section 3.2.2. Figure 5.2 shows an example of the type of features produced by SIFT.

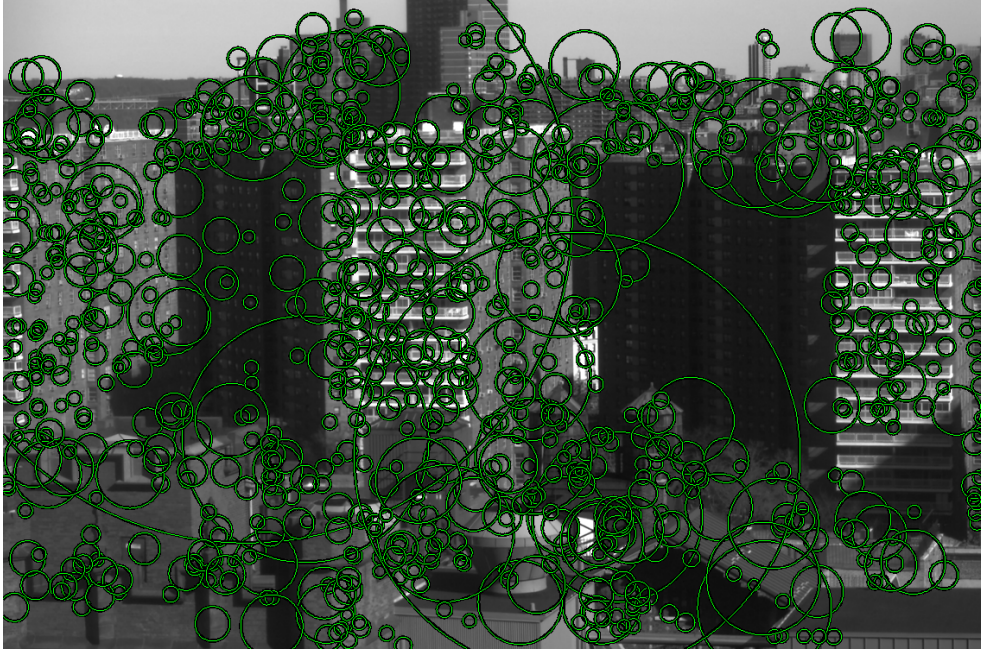


Figure 5.2: Detected regions produced by SIFT. Only 1000 out of 4923 features are shown here

5.3.2 MSER

MSER is described by Matas *et al.* [MCMP02]. MSER is a feature extractor that detects a stable subset of all ‘extremal regions’. Extremal regions (either maximal or minimal) are detected by applying a binarisation operation with a varying threshold, and detect local intensity extrema. For example, when applying a threshold, t , beginning from zero and gradually increasing, the entire image is white, indicating all pixels in the image are greater than t . As t is increased, black spots corresponding to intensity minima will appear and grow as t is increased further. Subsequently, more minimal regions appear and adjacent regions merge until finally when t is at its maximum the image is completely black. The set of all connected components for every value of t is the set of maximal regions. These extremal regions have the property that all pixels in the region have either higher or lower intensity—for maximal or minimal regions respectively—than all the pixels at their boundary. The maximally stable subset of these regions are those regions that are stable over a range of thresholds. Figure 5.3 shows an example of the type of features produced by MSER.

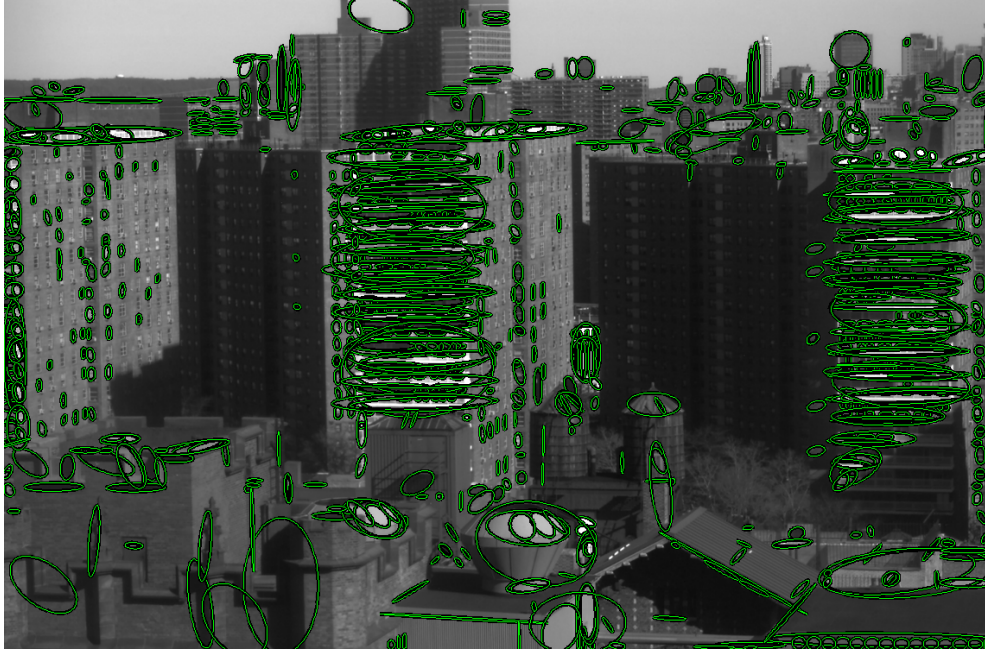


Figure 5.3: Detected regions produced by MSER

5.3.3 HARAFF

The HARAFF algorithm described by Mikolajczyk and Schmid [MS04], builds upon the “Harris-Laplace” algorithm described in an earlier project by Mikolajczyk and Schmid [MS01]. Harris-Laplace is used to determine the location, rotation and scale of the regions and can cope with scale changes up to a factor of four. Potential points are located using the Harris detector. These points are selected if the location also yields a maximum for the Laplacian of Gaussian (LoG) operating over neighbouring scales constructed in a similar manner to the scale approach employed by SIFT. The LoG operator identifies regions of rapid intensity changes by analysing the second derivative of the image. LoG operates in a similar manner to the DOG operator employed by SIFT. In fact, the DOG operator provides an approximation to LoG and benefits by providing a significantly lower computational complexity. The authors describe it as detecting “corner like structures.” HARAFF then extends on this and determines the shape of the region around the point, permitting an affine invariant description of the region to be constructed. To determine this shape, the second moment matrix of the intensity gradient is computed. This describes the gradient distribution in a region and is used to determine an elliptical region capable of deforming along with affine transformations. Figure 5.4 shows an example of the type of features produced by HARAFF.

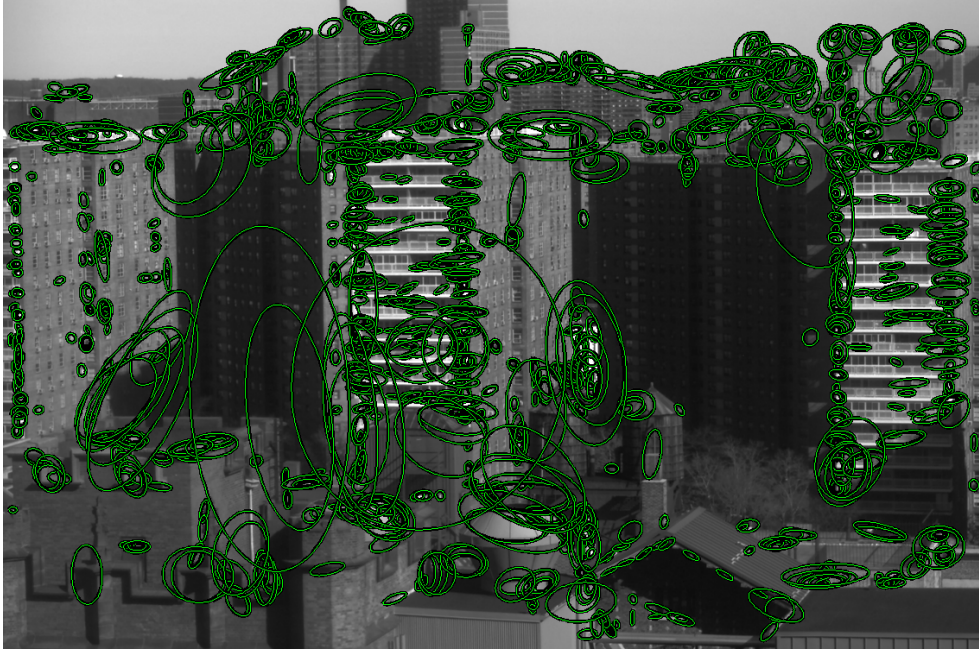


Figure 5.4: Detected regions produced by HARAFF

5.3.4 HESAFF

HESAFF [MTS⁺05] is an affine extension of the Hessian-Laplace algorithm. Hessian-Laplace detects features that are invariant to rotation and scale. Points are localised at the maxima of the Hessian determinant, while the scale—like the Harris-Laplace algorithm used by HARAFF—is determined using the local maxima of the Laplacian-of-Gaussian. HESAFF, like SIFT, mainly detects blob-like structures in the image. Affine region approximation follows the same method as that described above for HARAFF. Figure 5.5 shows an example of the type of features produced by HESAFF.

5.3.5 EBR

EBR was developed by Tuytelaars and Gool [TG99] and is an affine invariant algorithm that utilises point and curve features. First, a Harris corner point is determined p , then a nearby Canny edge is extracted. To permit scale change, these features are extracted at multiple scales. Two points, p_1 and p_2 are then selected along the edges moving away from the Harris point in two directions. The points are selected based on the rate at which they move away from p at speed l such that the two points are $p_1(l)$ and $p_2(l)$. The authors note that the area between the edge formed between (p, p_1) and the Canny edge, and area of the edge between (p, p_1) and the Canny edge, are invariant under affine transformations. The vectors formed

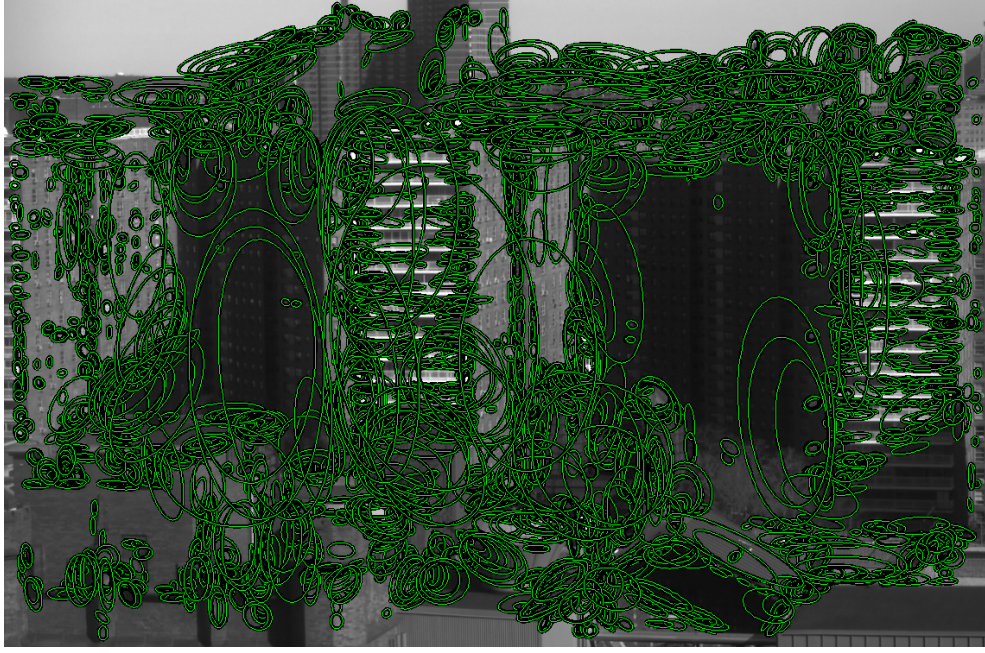


Figure 5.5: Detected regions produced by HESAFF

between $(p, p_1(l))$ and $(p, p_2(l))$ can be used to form a family of parallelograms as a function of l . Any parallelogram from this family whose region is an extrema for certain photometric contrasts is selected for use. Figure 5.6 shows an example of the type of features produced by EBR.

5.3.6 IBR

Subsequently Tuytelaars and Gool presented another feature detector called IBR, which does not require corners and edges to be present in the scene like EBR does, and makes use only of intensity [TG00]. The algorithm works by first identifying intensity extremum in the image. Rays are then emanated from one of these identified points, and the intensity data along each ray is examined. When the intensity function along a ray encounters an extremum due to a substantial intensity increase or decrease, the ray is stopped. The point where the ray stops is invariant under affine geometric or photometric transformations. Each ray from the detected point undergoes this function until all rays intersect with an extremum, generally considered the boundary of a region. The end points of all these rays are then linked together forming an enclosed region. An ellipse is then fitted to this region. Figure 5.7 shows an example of the type of features produced by IBR.

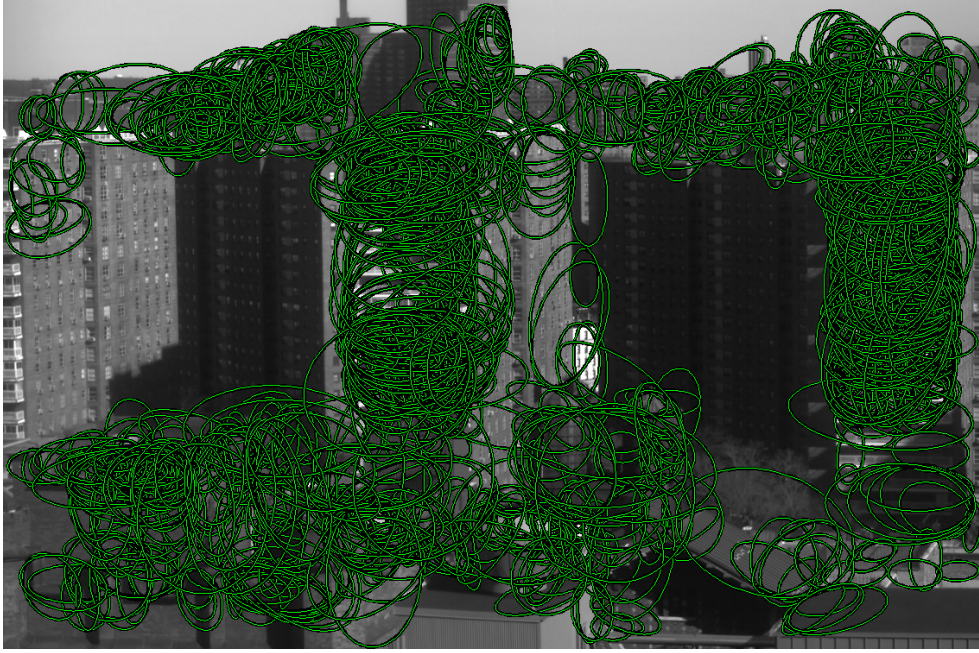


Figure 5.6: Detected regions produced by EBR

5.3.7 Algorithm Summary

In summary, seven algorithms will be tested in this chapter. The Fourier-based algorithm was presented as part of the 2D system in Chapter 3. This global registration algorithm operates in the Fourier domain and is reported to be robust to illumination changes between images. The 3D system of Chapter 3 made use of a local feature detector called SIFT. This algorithm is invariant to scale and rotation. The remaining five local feature detectors—MSER, HARAFF, HESAFF, EBR and IBR—were evaluated by Mikolajczyk *et al.* [MTS⁺05] and are developed by a range of authors. Like SIFT, these algorithms are invariant to scale and rotation. In addition, these algorithms are affine covariant, in that the regions they produce are covariant to any affine transformation in the scene.

5.4 Testing Methodology

Three stages of the 3D system are evaluated using the testing methodology presented in this chapter: feature detection, description and matching. Three distinct tests are used. The first is based on the framework presented by Mikolajczyk *et al.* [MTS⁺05] in their comparison of affine covariant region detectors. The second test suite is based on the work of Mikolajczyk and Schmid in their comparison of descriptor algorithms [MS04]. The last is a simple example of the use of the output of these

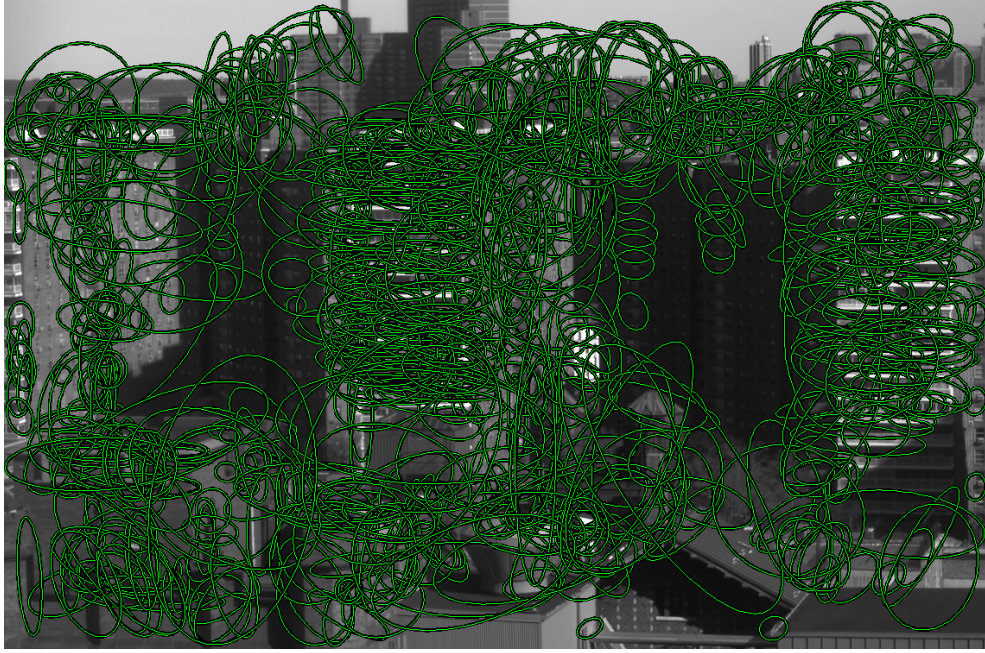


Figure 5.7: Detected regions produced by IBR

algorithms to test the impact of the performance in a real AR application. A detailed description of each of these tests follows.

5.4.1 Repeatability

The repeatability metric was originally conceived by Schmid *et al.* [SMB98] and provides a strong measure of detector performance irrespective of the abilities of any other stage in the feature tracking process. This approach was also used by Mikolajczyk *et al.* [MTS⁺05] in their comparison of affine covariant region detectors.

This metric is based on the realisation that matching performance is dependent on the detector identifying the same real world 3D points in images of the same scene under varying geometric and photometric conditions. Therefore the repeatability determines the detector's ability to extract the same 2D features from images of the same scene. A re-detected feature is called a correspondence, and if no correspondences exist between two images, there are no common features, thus there can be no correct matches. The number of correspondences therefore constitutes the set of all potential matches.

In order to determine whether two points are correspondences, the geometric relationship between the images must be known *a priori*. Using a known homography between the images, a region extracted from one image can be transferred into the other. A region located in the same position in the other image is considered a

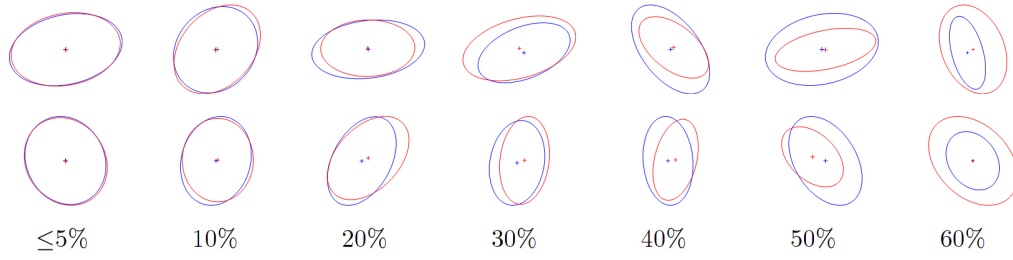


Figure 5.8: Depiction of overlap error thresholds (Image courtesy of Jiri Matas – Czech Technical University, Prague. Image reproduced from [MTS⁺05])

correspondence if its overlap error is less than some threshold e . The overlap error is defined as the area not in common between two potentially corresponding regions. If this area is too large, it indicates the regions cover areas insufficiently similar to classify the features as correspondences. Mikolajczyk *et al.* [MTS⁺05] state that this metric can be determined by computing the ratio between the intersection of the regions and their union as follows:

$$1 - \frac{R_1 \cap H^T R_2 H}{R_1 \cup H^T R_2 H} < e \quad (5.1)$$

where R_1 and R_2 are the potential corresponding regions between a pair of images related by a homography H .

Figure 5.8 demonstrates the indicated overlap errors for a series of regions. Following Mikolajczyk *et al.*, an overlap error of 40% is used throughout the tests. Any regions for which the error is larger than this are assumed to be non-corresponding. Errors larger than this usually produce descriptors representing regions sufficiently different as to preclude correct matching anyway.

The repeatability can then be computed as the number of correspondences that have an overlap error less than e , over the lower of the total number of features extracted from the two images. Mikolajczyk *et al.* [MTS⁺05] express this relationship as:

$$\text{repeatability} = \frac{\# \text{correspondences}}{\min(\# \text{features in 1st image}, \# \text{features in 2nd image})} \quad (5.2)$$

Low repeatability scores indicate a poorly performing region detector generating few re-detected regions. Any subsequent matching stage no matter how robust, will produce few correct matches, or worse, a large number of incorrect matches.

Theoretically, if two different images were taken from exactly the same position

and orientation, a completely robust region detector would obtain 100% repeatability in that it would re-detect the exact same points in each instance. In practice this is unlikely as there are always minor differences between images, even from the same position. These differences occur due to sampling variations between images from a camera, or noise introduced by minor illumination changes or other photometric variations.

Note also that the repeatability score needs to be assessed with respect to the absolute number of correspondences and the total number of features detected. If the number of features detected in one of the images being matched is low, a high repeatability can still be obtained since it is computed as the number of successfully re-detected features over the total number of features. Thus a high repeatability score and a low number of correspondences is worthless as no further geometric entity can be computed.

5.4.2 Recall and 1-Precision

The recall and 1-precision metrics have been used recently by both Ke and Sukthankar [KS04] and Mikolajczyk and Schmid [MS05] to evaluate the performance of region descriptors and subsequent matching performance. The aim of these metrics is to capture the fact that during matching the number of correct matches should be maximised, while minimising the number of false matches. Often these statistics are used to produce recall *vs.* 1-precision graphs where the lines in the graph are formed by varying a threshold parameter. In Mikolajczyk and Schmid's work the threshold used was the maximum distance between matching descriptors. In the experiments of this chapter the performance of the descriptor is not the focus, but rather performance over a range of illumination conditions. Therefore, both recall and 1-precision are separately plotted against time.

Recall is the number of correct matches as a percentage of the total number of possible matches, which is effectively the number of correspondences. Mikolajczyk and Schmid [MS05] identify that it is computed using:

$$recall = \frac{\#correct\ matches}{\#correspondences} \quad (5.3)$$

where matches are determined using the distance ratio technique and the correct matches are those verified as correspondences. This metric is useful as it gives a clearer idea of the descriptor performance given maximum theoretical performance

of the detector. It does this by taking into account the best possible performance that could be obtained, *i.e.* the number of correspondences. For example, assume there are two images with 1000 and 1100 features detected respectively. Fifty of these features are deemed as correspondences, and forty are established to be correct matches. The recall in this case would be $40/50 = 80\%$.

Knowing how many correct matches there are is not useful by itself, the number of false matches is also needed. For instance, take the example above where 40 correct matches were computed out of 50 correspondences. Recall by itself does not take into account the number of false matches—an important consideration in final performance. Imagine in that example there were 500 false matches, a number which might make computing any further geometric entities—such as homographies or epipolar geometry—difficult. Robust statistical techniques such as RANSAC can cope with a large number of outliers, and only begin to fail when groups of outliers form valid models of the entity sought or when a group wholly comprised of inliers is not selected. Both of these prospects become increasingly likely as the ratio of correct to false matches drops. The lower the ratio of correct to false matches, the less confidence there is in obtaining a correct final result.

In the field of information retrieval the precision metric is computed as the number of correct matches over the total number of matches. In these matching experiments however, a metric that describes the number of false matches rather than the number of correct ones is needed. Therefore, 1-precision is used instead of precision. 1-precision is computed as the number of false matches relative to the total number of matches:

$$1 - \text{precision} = \frac{\# \text{false matches}}{\# \text{correct matches} + \# \text{false matches}} \quad (5.4)$$

where $\# \text{correct matches} + \# \text{false matches}$ is effectively the total number of matches, correct or false. Using the same example as used for the recall, 1-precision is computed as $500/(500 + 40) = 92.5\%$, meaning 92.5% of all matches for this image pair are false matches.

1-precision graphs are also useful as the area above the graph indicates the percentage of correct matches out of all matches. Therefore it also depicts the ratio of correct to false matches. For example a 1-precision score of 75% indicates only 25% of all matches were correct, meaning there are three false matches for every correct one.

In their comparison of affine region detectors Mikolajczyk *et al.* utilised a metric called the ‘matching score’ to evaluate matching performance. This metric is computed as the percentage of correct matches out of the minimum of the total number of features extracted from the reference or live frame. A matching score, however, can give a false impression of performance as it does not take detector performance into consideration. In the example above for instance, where 50 correct matches were found from two images with 1000 and 1100 total features respectively, the matching score would be calculated as $40/1000 = 4\%$, indicating poor matching performance. However, given there were only 50 correspondences to begin with—meaning 50 possible correct matches—40 correct matches is a good result. For this reason recall and 1–precision were selected as the metrics to represent matching performance. To provide a suitable comparison with the work of Mikolajczyk *et al.* matching score results are given in Appendix B.

5.4.3 Practical Tests

The final test involves a more realistic scenario where the data from the feature extractors is used to automatically compute a homography between the images using nothing more than the features extracted. As the selection of WILD images are taken from the same position, the geometric relationship between the images is described by a homography, which provides a point-to-point relationship. The process for computing these homographies is similar to that for computing epipolar geometry, as described in Chapter 3, Section 3.2.4. The features are extracted and then matched using the distance ratio method. RANSAC is used to determine a homography model describing the geometric relationship between the images. The model with the most support is selected. A minimum of four points is needed to compute a homography. The computed homography is verified by comparing the estimate with the ground truth homography available for every image. The number of ‘inliers’ is plotted for each image where a value of zero indicates homography computation failed. Inliers are matches consistent with the computed homography model.

Due to the random nature of RANSAC, a computed homography can differ between successive runs. This is particularly true when there are large numbers of outliers in the putative match set, providing the potential for RANSAC to select a model that is incorrect. RANSAC is used to generate 10 homographies, providing evidence of the reliability of the feature sets. These reliability scores are also plotted.

In a hypothetical AR system, the computer vision system may employ this approach to establish the geometric relationship between a reference view—for which a virtual object’s pose has been calibrated—and live images from a camera. Homography computation would then be performed for each frame. Unreliable estimates would produce frames where the virtual object would be rendered in a manner inconsistent with the motion of the camera, *i.e.* a registration error would be present. With 70% reliability only 70% of the frames would be rendered correctly. This would produce an object that appears to jitter due to the numerical instabilities in the homography estimation. To be practically useful a system needs close to 100% reliability on this metric.

5.4.4 Test Framework Summary

Table 5.2 summarises the relationship between the stages under evaluation, and the algorithms and metrics used to analyse performance at each stage. The performance of six feature detection algorithms is evaluated. The algorithms are: SIFT, MSER, HARRAFF, HESAFF, EBR and IBR. The metrics used to analyse detector performance are: the repeatability score, the absolute number of correspondences and the total number of features detected.

Reconstruction Stages	Algorithms Evaluated	Metrics Used
1) Image Capture		
2) Feature Detection	SIFT, MSER, HARRAFF, HESAFF, EBR, IBR	Repeatability, correspondences, total #features
3) Feature Description	SIFT, PCA-SIFT, SC, Spin Images, GLOH	Recall, #correct matches, 1–precision, #false matches
4) Feature Matching		#Homography inliers, homography reliability
5) Two-View Geometry		
6) Three-View Geometry		
7) Camera Pose Recovery		
8) 3D Reconstruction		
9) Bundle Adjustment		

Table 5.2: The left column shows the stages of the 3D system. Stages 2, 3 and 4 are highlighted as their performance is evaluated in this chapter. The centre column lists the corresponding algorithms tested for each of the corresponding stages. The right hand column shows the applicable test metrics for the stage

The five feature description algorithms tested are: SIFT, PCA-SIFT, SC, Spin Images, and GLOH. The metrics used to evaluate descriptor performance are: the recall score, the absolute number of correct matches, the 1–precision score and

the absolute number of false matches. Note also that these metrics evaluate the description and matching stage as their respective performance is linked.

The final metrics are produced by computing a homography from the feature matching results and evaluating the reliability of the computations. These metrics are dependent on all three stages and demonstrate the applicability of the feature extraction and matching results.

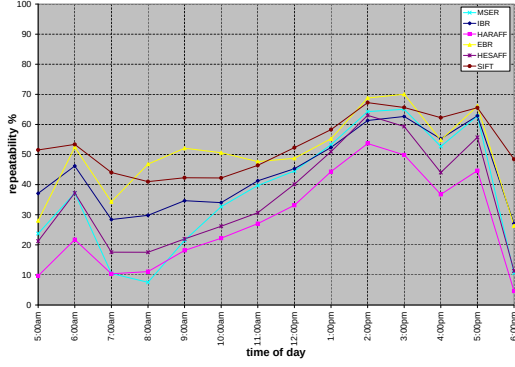
5.5 Test Results

The first experiment involves analysing the feature detector's performance on the selection of clear weather WILD images which vary by the time of day they were captured. These images were captured on February 18th, while the reference image used for these tests was captured in clear conditions at 2pm on April 4th. These images are shown in Chapter 4, Figure 4.6 where Figure 4.6(a) is the April reference image.

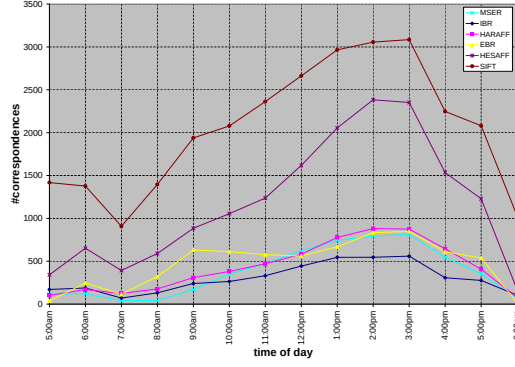
Figure 5.9 shows the detector results where Figure 5.9(a) shows the repeatability. It can be seen that all the algorithms exhibit similar repeatability patterns. The performance starts low at 5am then all the algorithms increase to a small peak at 6am. Performance then drops off at 7pm, and then gradually increases to the maximum repeatability score at 2-3pm. The decrease in performance from 6-7am is likely the result of the introduction of strong shadows in the latter image. At 6am there are few shadows visible. This means features are detected more evenly over the whole image, yielding more correspondences to the reference image despite detecting fewer features overall. The bad performance at 5am and 6pm is due to the lack of sunlight, meaning the only features in the image are generated by artificial light sources. The performance at 5am is slightly better than at 6pm, as there is slightly more global illumination at 5am. The reddish tinge on the 5am image suggests the sun is rising and is providing a small amount of illumination across the scene, whereas at 6pm the sun is completely set.

The number of correspondences shown in Figure 5.9(b) shows that from 6-7am the number of correspondences drops. HARAFF seems less affected in this graph. However, the comparatively small decrease from 170 to 125 correspondences comes despite an increase in the total number of features from 782 to 1204, which means the repeatability still decreases.

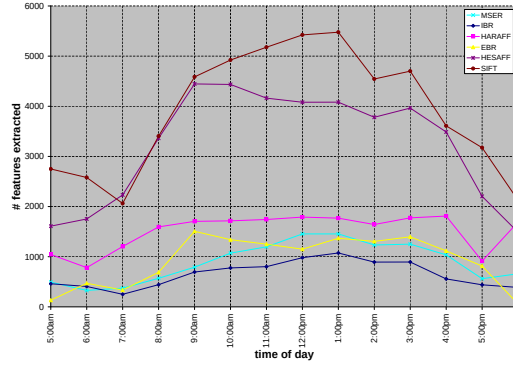
The rankings of these algorithms in terms of their repeatability scores generally



(a) Repeatability



(b) Number of Correspondences



(c) Number of Features Detected

Figure 5.9: Detector performance for WILD selection

remains consistent throughout the test range. The ranking of SIFT and EBR changes at several points in the sequence, with EBR performing better between 8am and 11am, and between 2–3pm. The other algorithms fall roughly into the order of IBR, HESAFF, MSER, then HARAFF. Figure 5.9(b) shows the number of corresponding regions between the reference image and the WILD selection. It can be seen that SIFT produces the most correspondences, followed by HESAFF, while the remaining algorithms achieve similar numbers. Although SIFT and HESAFF produce a lot more correspondences, their repeatability scores do not benefit, indicating an increase in non-corresponding regions proportional to the correspondences. These results are consistent with Figure 5.9(c) which shows the total number of features extracted for each of the images. The graph shows that SIFT extracts the most features, followed by HESAFF then the remaining algorithms.

Figure 5.10 provides an analysis of the matching performance for these images when using the SIFT descriptor and distance ratio matching technique. Figure 5.10(a) shows the recall for the algorithms, indicating the percentage of correctly computed matches out of the actual number of re-detected features between the

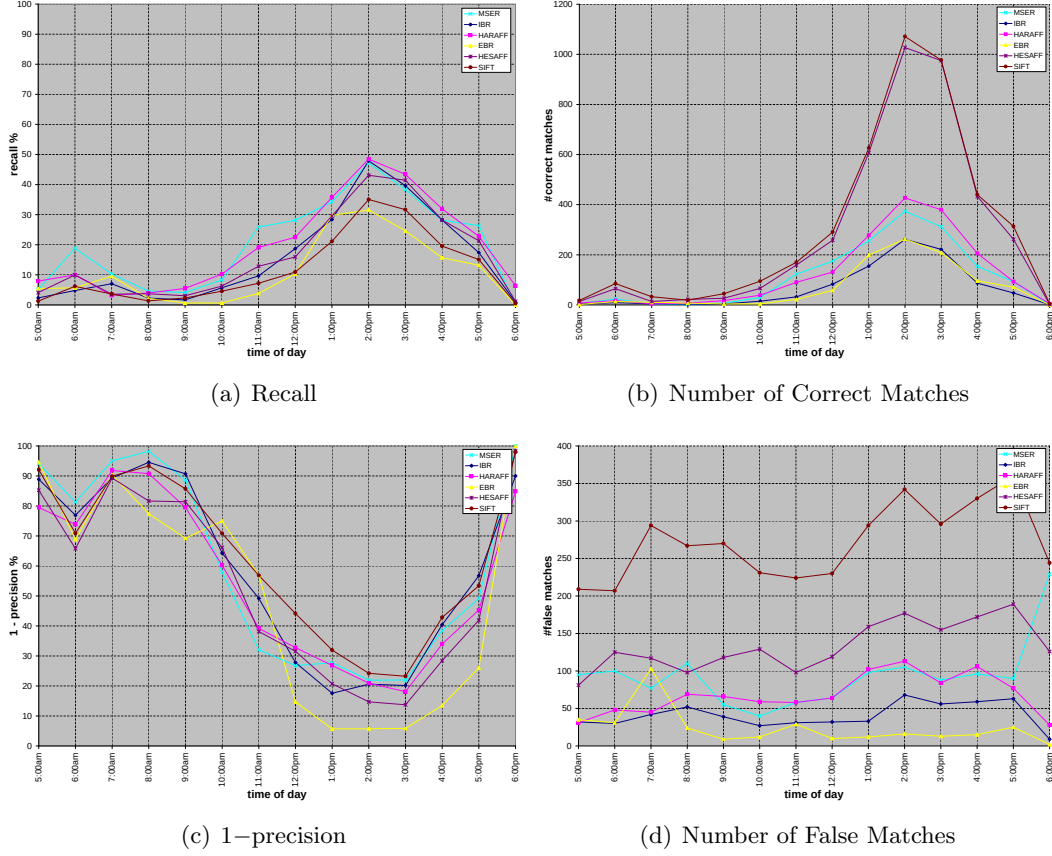


Figure 5.10: Matching results for WILD selection

images. MSER performs well, indicating that in some cases a higher percentage of re-detected MSER regions are likely to be correctly matched. EBR performs the worst, indicating that matching SIFT descriptors for EBR features can be the most unreliable. What is interesting here is the low recall rates of images captured in early times of the day. This means that only a low percentage of the features that are re-detected in these images are being correctly matched, indicating inadequacies in the matching process when matching images from dissimilar illuminations. These poor matching results cannot be explained by any deficiency in the 7 or 8am feature sets, as matching MSER features between these views produces good results with a recall score of 60% and a 1-precision score of 9%. The absolute number of correct matches of Figure 5.10(b) shows that SIFT and HESAFF obtain the highest number, followed by MSER. This means MSER's top performance in the recall scores is the result of a higher proportion of correspondences being correctly matched compared with the other algorithms.

At 2pm HARAFF achieves a 50% recall rate, meaning half of the features that are re-detected are correctly matched. At 7am this is significantly lower and HARAFF

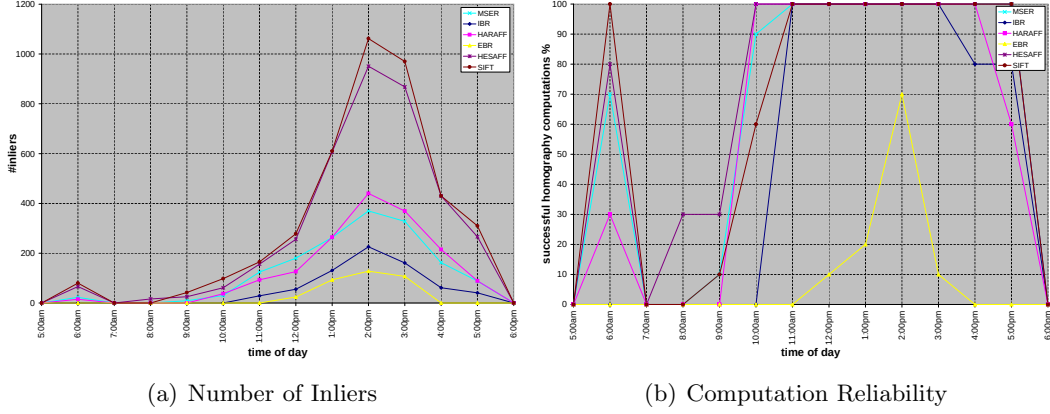


Figure 5.11: Homography computation results for WILD selection. A value of zero inliers in (a) indicates the system failed to compute a homography for this time

only achieves a rate of 3%, where only four correspondences are deemed correct out of 125. All the algorithms follow a similar trend here, indicating a deficiency in the descriptor and matching process under large illumination differences. Given that all algorithms exhibit the same trends, it suggests it is not what regions are being chosen—as these detectors all use different image properties to locate features and produce different regions—but rather how they are being described and what effect the illumination discrepancy is having.

Figure 5.10(c) shows the 1-precision scores. All the algorithms produce a large percentage of false matches until approximately 9am after which the performance quickly improves. EBR often performs best producing fewer false matches, indicating that although it extracts fewer features than some of the other algorithms, the features it does use are robust ones. There are many tradeoffs shown in these graphs. In some instances an algorithm may produce fewer false matches, but also fewer correct ones. MSER, HARAFF and HESAFF seem to provide the best balance of performance over both the metrics. Note that at 6am the algorithms obtain a local peak in performance compared with the other early morning images. The precision of all algorithms increases here despite similar or even increased numbers of false matches between 6 and 7am as shown in Figure 5.10(d) in the case of MSER and HARAFF. This means the precision improvements are a result of the increase in the number of correct matches. The poor precision at 7 and 8am is a result of a high number of false matches and a low number of correct matches.

To establish what these results mean with respect to the final performance of an AR system, or any application relying on successfully tracking image features—for example a video surveillance system—Figure 5.11 shows the results of computing

a homography using each of the algorithms. Figure 5.11(a) shows the number of inliers. If the homography computation failed to determine a correct homography in all 10 attempts, the number of inliers is set to zero indicating failure. Figure 5.11(b) shows the success rate for these computations. 100% indicates all 10 runs successfully resulted in a correct homography model. Less than a 100% success rate in computing the homography would manifest itself in a real-time AR system as the virtual object intermittently moving erratically around the scene. Examining the graphs it can be seen that all the algorithms fail to establish a correct homography for the *5am* image. At *6am* HARAFF, MSER, HESAFF, and SIFT succeed with varying levels of success: 30, 70, and 80% for the first three algorithms respectively, while only SIFT managed to achieve 100% success in computing the correct homography. All the algorithms fail at 7 and 8*am* while only HESAFF manages to successfully compute a homography at 8*am*, however, only 30% of the time. At 9*am* HESAFF, MSER, and SIFT are capable of computing the homography with limited reliability. At 10*am* the reliability and the number of inliers increases, with HESAFF and HARAFF managing to achieve a 100% success rate. At 11*am* all the algorithms except EBR compute the homographies with 100% success until 3*pm*. The number of inliers for each of the algorithms also increases and reaches its maximum at 2*pm*. SIFT, HESAFF and MSER maintain 100% reliability until 5*pm*. All the algorithms fail at 6*pm*.

EBR obtains the lowest number of inliers consistently for each image, and fails to achieve 100% success for any image, with its highest reliability achieved at 2*pm* with 70%. This is surprising given that EBR obtained the highest precision of any algorithm from 12–5*pm*. Analysing the homographies generated from the matches, it seems the localisation accuracy of the points generated by EBR is insufficient to produce a homography close enough to the ground truth data to be considered an accurate result. This means that corresponding regions produced from EBR overlap enough to be considered matches in the matching tests of Figure 5.10, but when used for computing a homography, yield a model less accurate than those produced by the other algorithms. Relaxing the constraints on what is considered a correct homography model increases the reliability at times when sufficient correct matches exist, for example from 11*am* to 5*pm*. However the test results shown in Figure 5.11 reflect the accuracy expected from all approaches, and the level required in order to be useful in a real world application. Although not ideal for producing accurate

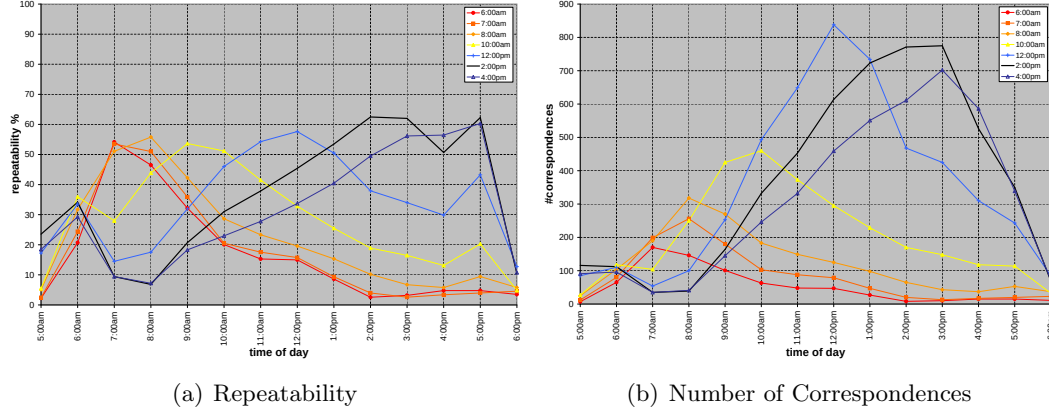


Figure 5.12: Detector performance for MSER using different reference frames

homographies, the precision of EBR could be utilised under near ideal imaging conditions to provide a rough estimate of geometry that is later refined by one of the more robust algorithms shown in the tests. In addition, the poor localisation of EBR could be the result of the ellipse fitted to this region. Mikolajczyk *et al.* noted that ellipse fitting should not be used in practical applications as “orientation information is lost” [MTS⁺05].

Choice of Reference Frame

The choice of the *2pm* reference frame used in the previous test significantly impacts the detection and matching performance for images from disparate times containing differing shadow configurations. The following experiment demonstrates that the choice of reference frame can produce highly variable performance. This test makes use of the MSER algorithm and utilises a number of reference frames from different times. The black bold line shows the default performance using the *2pm* reference frame as *per* the previous test. Only a selection of the available reference images are used in order to simplify the graphs; however, these results are representative of those seen in the excluded reference images.

Figure 5.12 shows the detector performance when using reference frames from different times. The repeatability performance shown in Figure 5.12(b) shows a clear correlation between the choice of reference frame and the performance for a given live image. Reference images from times closer to the live images produce the highest performance. For example the *12pm* reference image produces the highest performance when matched to the *12pm* live image. The same applies to the *2pm*, *10am*, *8am*, and *7am* reference images, which all produce performance peaks for live

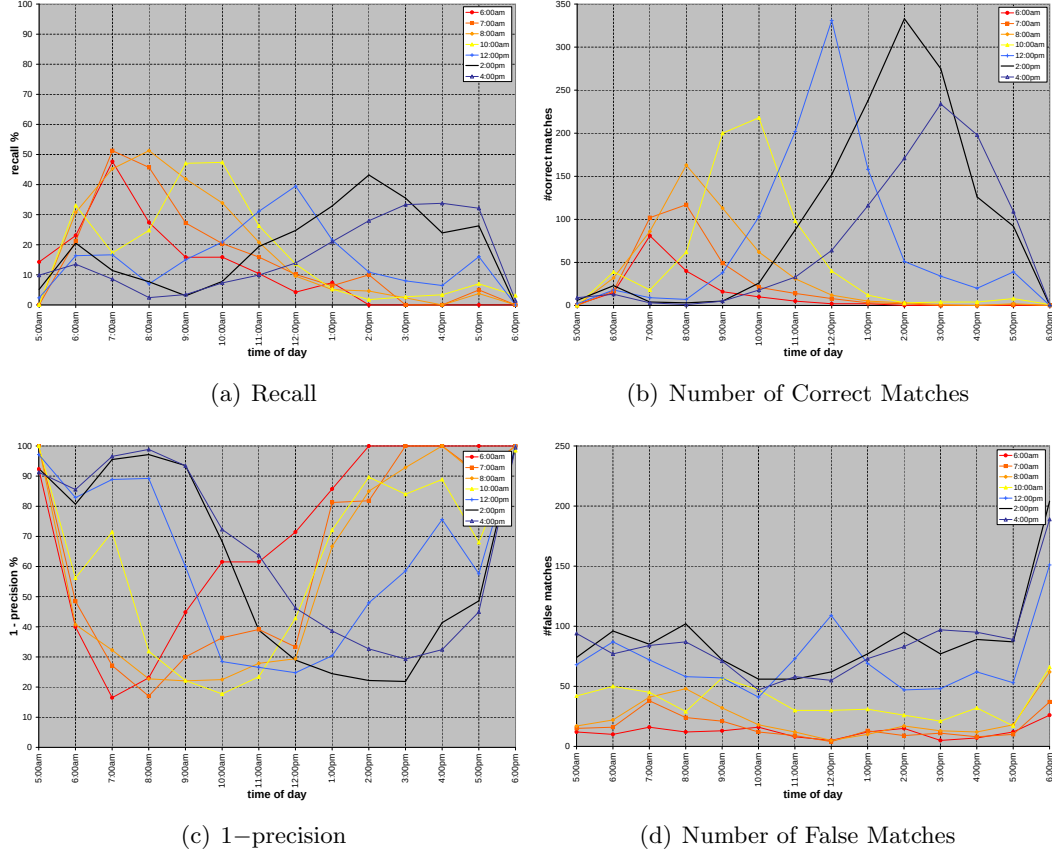


Figure 5.13: Matching results for WILD selection

images at corresponding times. The number of correspondences found is shown in Figure 5.12(b), and exhibits the same trend as seen in the repeatability data. The number of correspondences produced using early reference frames with early live images, are lower than those from during the middle of the day with corresponding reference frames. This is because fewer features are detected in both early reference and live images.

Figure 5.13 shows the matching performance using reference frames from different times. Even using a 12 pm reference frame 1-precision results still drop off by 60% between 10 am and 8 am . The same effect as seen in the detector performance is seen in this situation. Peak performance is obtained in both recall and 1-precision when the reference and live images being matched are from the same time. This produces reference frames with performance characteristics opposite to each other. For example using a 2 pm reference frame produces the best performance at times between 11 am and 4 pm , while the worst performance is obtained between 7 am and 10 am . The opposite is true of the 7 am reference frame which produces optimal performance at approximately 7 am while the worst is between 12 pm and 6 pm .

Summary

To summarise, the performance of these feature extractors over a range of illumination conditions can be variable, even in clear weather and with no other image differences other than the illumination changes. The choice of reference frame has a significant impact on performance, where live images from times disparate from the reference image produce poor matching performance. The impact is so large that matching performance is insufficient to compute further geometric entities. If the images contained viewpoint changes or some other difference the performance would be even worse.

When utilising a *2pm* reference image the performance is best from approximately *10am–5pm*, however the performance outside these times is insufficient to provide a reliable solution. Inspecting the images, it seems there is certainly sufficient illumination from *7am* onwards that the algorithms should work. It is hypothesised that the lack of performance is not because these images are taken at dissimilar times to the reference frame, but that they contain dissimilar shadow configurations. This hypothesis fits the data: the increase in performance at *6am* can be explained by the lack of shadows in this image compared with the other morning images from *7am* onwards, even though the level of contrast in these images increases as the morning progresses. At *7am*, although the sun is up providing higher intensity regions than at *6am*, a substantial portion of the image is in shadow. The small parts that are not in shadow are in the reference image, resulting in poor performance at this point. Another factor supporting this hypothesis is that the performance is substantially better at *2pm* than at *12pm*, even though the *12pm* image appears better in that it does not contain many regions in shadow. The improvement comes from the fact that at *2pm* the shadow configuration is the most similar to that of the reference image. This means at *12pm* there are a lot of additional features detected that are not in the reference image, resulting in a lower repeatability score. The matching performance also decreases as there are additional features in the set that are not in the reference set, increasing the probability that an incorrect match will be computed.

For the remaining experiments the *2pm* reference image will be utilised unless noted otherwise. The choice of this frame is somewhat arbitrary, however, as shown in the previous test, any reference image chosen will not be optimal for all live images. Therefore the analysis focuses on improving performance between dissimilar shadow configurations, an effect common to all choices of reference frame.

5.5.1 The Shadow Effect

Shadows are caused by an object occluding a light source. In an outdoor setting shadows are produced by occlusions of the sun. Prati *et al.* [PMTTC03] identify several fundamental properties of shadows which are shown in Figure 5.14. First, an object’s shadow can be separated into two parts: the self-shadow and cast shadow. The self-shadow is the area of the occluding object not facing the light source. In the 2pm image from the WILD selection, this would be the right hand side of the three facing buildings.

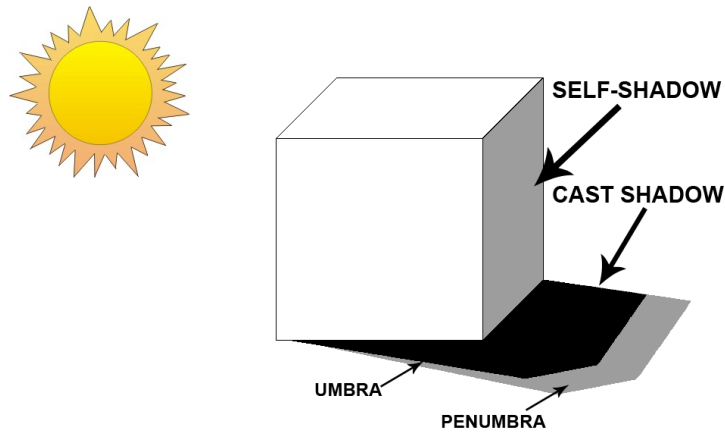


Figure 5.14: The properties of a shadow

Cast shadows are the projection of the occluding object from the light source and is identified in the mockup of Figure 5.14. These are often visible in the images from WILD. Figure 5.15 shows an example of both a self and cast shadow on the 2pm and 8am images respectively.

Cast shadows can be further broken down into two constituent parts: the umbra and penumbra. These are depicted in Figure 5.14. The umbra is the darker portion of the shadow and corresponds to the region for which light from the source is completely blocked. In contrast, the light is only partially blocked in the penumbra, therefore this region appears lighter. The rightmost image of Figure 5.15 shows the cast-shadow broken down into the umbra and penumbra regions.

Prati *et al.* examine methods for detecting moving shadows that are cast by moving objects [PMTTC03]. These shadows cause problems when segmenting and extracting moving objects, as shadows can cause “object merging, object shape distortion and even object losses (due to the shadow cast over another object)” [PMTTC03]. In addition, because the object’s shadow moves, background points in



Figure 5.15: WILD shadow examples. The leftmost image shows a self-shadow in the 2pm image. The centre image shows a cast shadow in the 8am image. The rightmost image shows a magnified version of the highlighted area in the centre image. Here the umbra and penumbra are identified.

shadow can be misclassified as being in the foreground.

Despite the fact shadows produce undesirable effects in some computer vision applications—such as feature tracking—it is worth noting that other computer vision algorithms make use of, and even rely on shadows. Examples include shape from shadow algorithms that infer the geometric properties of an object based on the shadow it casts [DD98]. Other algorithms exploit shadows in order to determine the orientation of object surfaces [SK92]. Shadows can also be used to determine the location of the light source that produces the shadow.

Despite this, shadows clearly provide an adverse impact on feature tracking applications. Four ways in which shadows adversely impact the feature set extracted from an image are identified. The first has been seen in the previous test, where shadows essentially cause there to be no overlapping regions in the image and produces a result similar to that when the images are taken from different positions. The magnitude of this effect is dependent on the intensity of the shadow, where darker shadow regions result in fewer features being detected. This is dependent on several factors such as the location of the light source—in this case the sun—as well as the scene structure and particularly the tonal range. The tonal range of an image refers to the region in the image’s histogram where most of the brightness values occur. If the tonal range of the scene is high, it means a larger range of intensity values is required to represent the scene content, leaving less data to represent any subtle changes that may exist in shadow areas, producing a further pronounced effect with fewer features detected.

In addition, shadows can cause the introduction of anomalous features into the feature set. For example, features can be introduced which are a direct cause of the

shadows themselves. Alternatively, shadows can deform features that might already exist. This deformation may be significant enough that the descriptors now cover a region so different that they no longer match. Alternatively, even if a corresponding feature is detected in a shadow region, its descriptor may be impacted to the point it is no longer matched.

The shadow effect is clear in Figure 5.16 which shows the MSER features detected for all the images. Boundaries to shadow regions mark the edge of a significant change in the number of features extracted. Virtually no features are detected in these regions. The *2pm* reference image shown in Figure 5.16(a) shows features extracted predominantly on the front faces of the buildings, but virtually no features detected on the right sides of the buildings. The *8am* image shown in Figure 5.16(e) shows substantially fewer features on the front of the middle building and virtually none on the buildings on the left and right of this building.

To confirm the hypothesis that the shadows are affecting the performance, a selection of images without shadows can be tested. The only time shadows are not present during daylight is when there are overcast conditions. Such a selection of images is now presented.

5.5.2 Cloudy Selection

A set of images taken in overcast conditions was extracted from the WILD database. The overcast conditions simulate a more diffuse light source. This selection of images is shown in Figure 5.17. Note that on this day there was no *5am* image available therefore this selection starts at *6am*.

Figure 5.18 shows the detection performance obtained between the cloudy selection and the *2pm* reference image. The repeatability results of Figure 5.18(a) show a substantial increase for all algorithms over the performance seen when operating on the images from the clear weather selection. For example at *7am* the performance increase is between 20–60%. The rankings of the algorithms remain similar with the exception of MSER which now performs best in many of the images. Figure 5.18(b) shows a consistent number of correspondences up until *6pm*. The large peak at *2pm* seen under clear weather is less pronounced here and is likely the product of reduced image contrast and a reduction in visibility produced by the cloudy conditions. For example at *2pm* the number of correspondences from SIFT drops from 3056 to 2322.

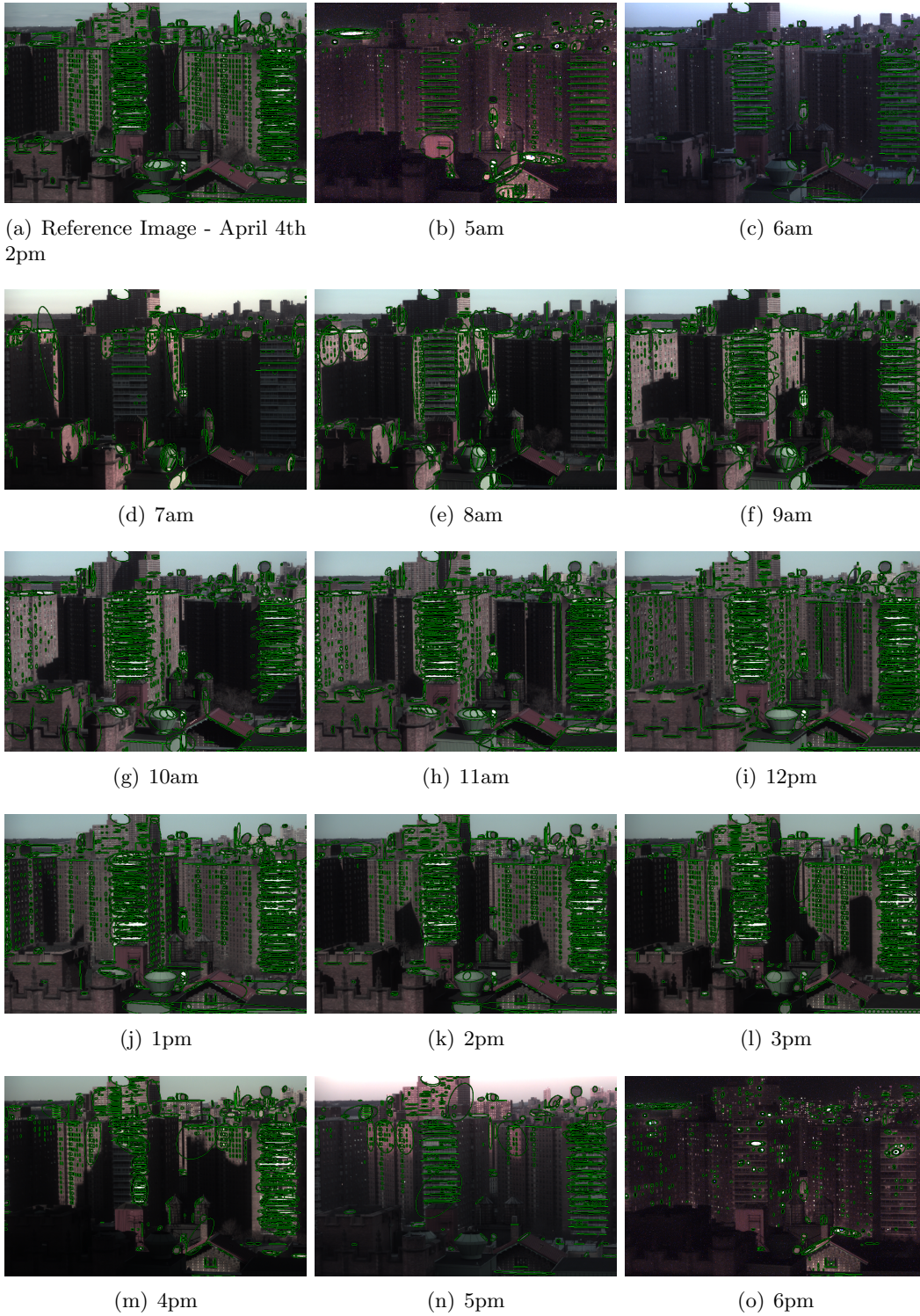
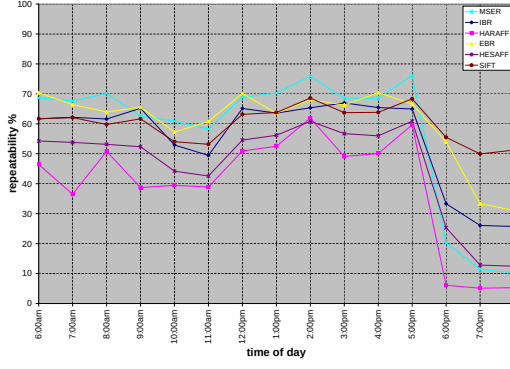


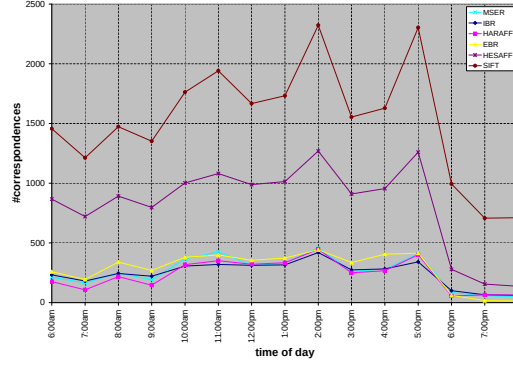
Figure 5.16: MSER features extracted from a clear weather WILD selection



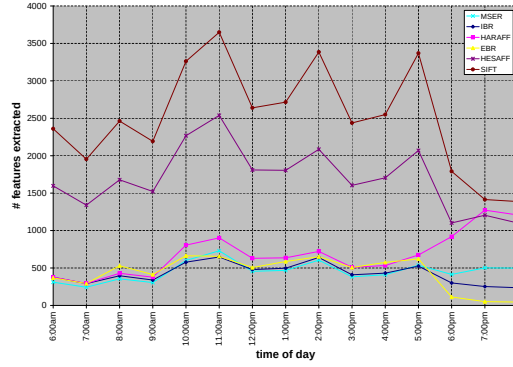
Figure 5.17: A selection of WILD images taken from a cloudy day to avoid shadow effects



(a) Repeatability



(b) Number of Correspondences



(c) Number of Features Detected

Figure 5.18: Detector performance for selection of cloudy WILD images

Examining the matching results in Figure 5.19 it can be seen that from 6am to 12pm the recall is greatly increased over the results from the clear weather selection. Another important fact is the consistency of this performance. The recall rates, however, aren't all that high, indicating that a large number of corresponding regions still aren't being matched. It is hypothesised that this is a consequence of the uniformity of the scene. The structure of the scene is such that there are many copies of similar features detected over the image. This leads to similar feature descriptors and a higher probability that the correct match is not the closest in descriptor space.

The 1-precision results in Figure 5.19(c) also show that from 6am to 12pm the percentage of false matches is substantially lower than the clear weather selection. Many algorithms such as MSER achieve an average of approximately 50% over these times, meaning that out of all the matches detected the ratio of correct to false matches is approximately 1:1. Given the only moderate success in recall scores this is an important consideration, as it means despite the matcher not living up to the feature set's full potential, it is robust enough to reliably provide a high enough

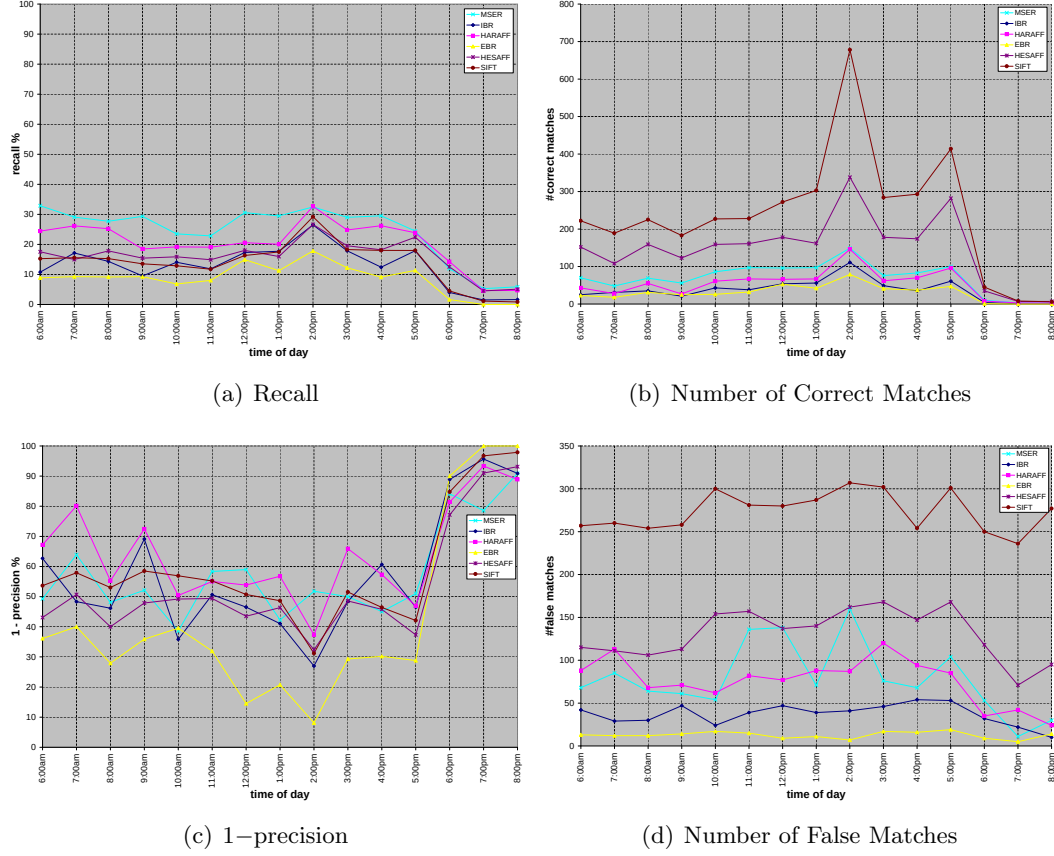


Figure 5.19: Matching Results for cloudy WILD Selection

number of correct matches to be useful, while not saturating the match set with false matches. After 12pm the 1-precision results are better for the clear weather selection, generating fewer false matches.

Another interesting result here is the high precision of the EBR algorithm over the other techniques. This was also witnessed in the clear weather experiment where from 12-5pm EBR obtained significantly lower percentages of false matches: approximately 10-30%. The algorithm produces the lowest number of correct matches, despite obtaining a similar number of correspondences to many of the other algorithms, thus achieving the lowest recall rate. However, of the matches it finds, a high percentage are correct compared with the other algorithms. This could be a result of this algorithm combining two algorithms, both a point and edge detector. When operating on good quality images this combination produces very few false matches.

Figure 5.20 shows the homography computation results. All the algorithms except EBR successfully compute homographies for all images until 6pm. The reliability of the computations shows that SIFT, HESAFF, and MSER consistently

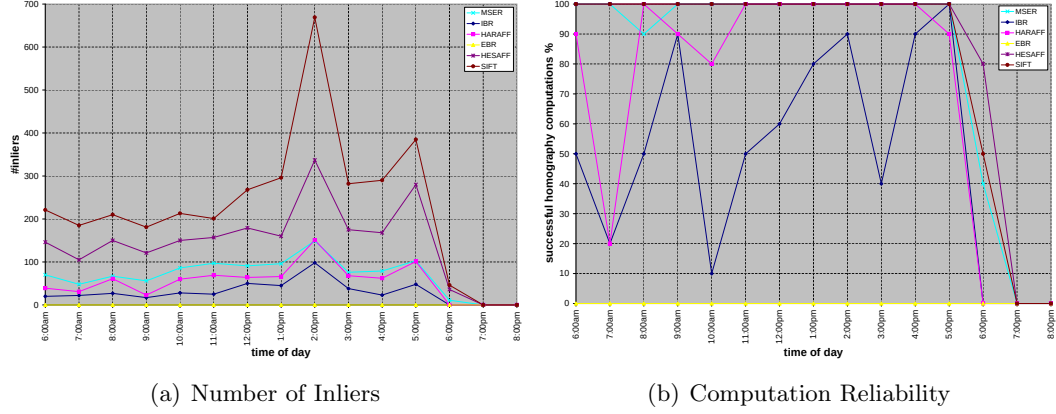


Figure 5.20: Homography computation results for cloudy WILD selection. A value of 0 inliers in 5.11(a) indicates the system failed to compute a homography for this time

compute the homographies successfully and even manage to compute the homographies at 6pm; a feat only possible due to the increased levels of illumination in this image compared with the 6pm image from the clear weather set. HESAFF performs best at 6pm, obtaining an 80% success rate. All algorithms from 7pm onwards fail to compute homographies as the images contain only man-made light sources and performance is similar to that seen at 5am and 6pm in the clear weather set.

As seen in the clear weather selection, EBR fails to provide a homography model accurate enough to be considered useful. The results are even worse here due to the decreased image quality caused by the reduced visibility and lowered contrast, which impacts edge based approaches the most.

5.5.3 Seasonal Changes

Time of day is not the only factor that brings about non-uniform changes in illumination such as shadows. Seasonal differences also introduce the same effects as those seen at various times during one day. This is not surprising, given the sun's position changes from season to season just as it does during the course of a day, thus also introducing differences in shadow location.

To test this, a selection of images was taken from the WILD database from the same time of day and under clear conditions, but differing in the date on which they were taken. This selection is shown in Figure 5.21 and covers a period of six months with each image captured at 3pm. Selecting images under the same clear conditions means the images are not regularly spaced. The dates for each image are indicated in the figure.

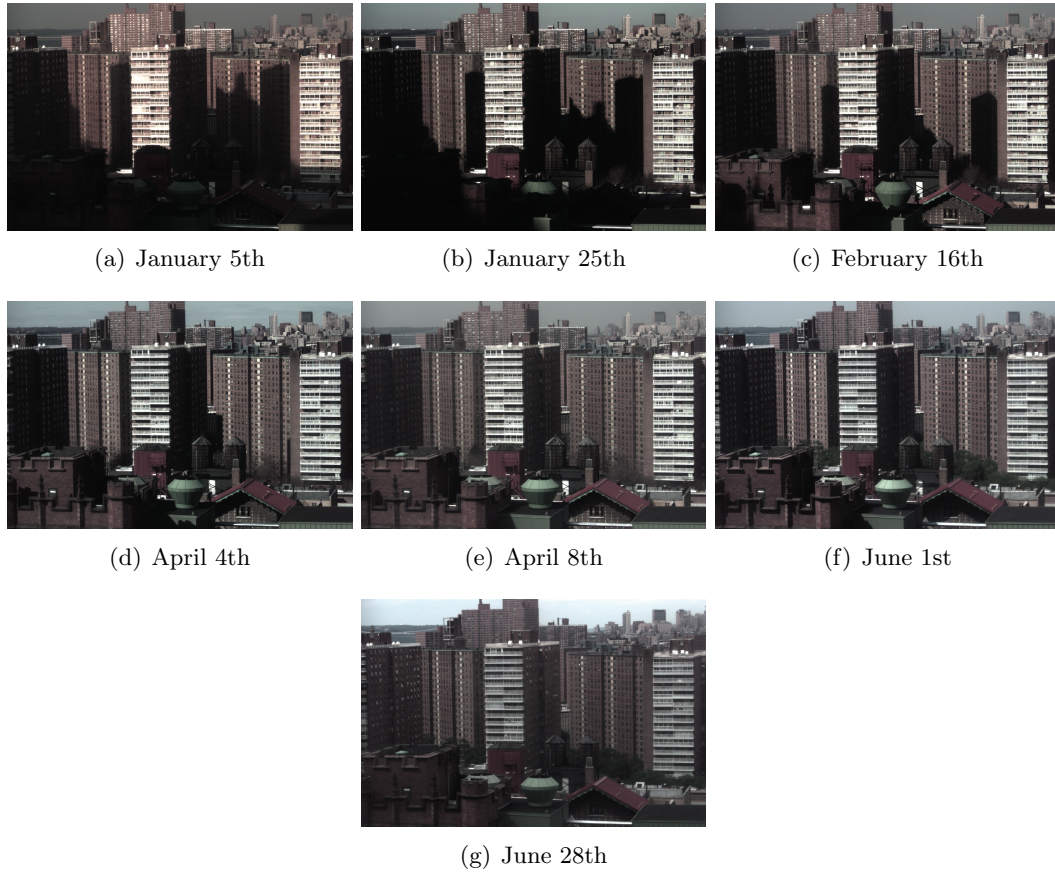


Figure 5.21: Set of seasonal change images from the 5th of January to the 28th of June

In order to test the effect that seasonal differences have, the detection and matching results are computed for every image combination from the seasonal set. This provides data covering a range of seasonal differences from small changes of only a few days to the maximum between images 5.21(a) and 5.21(g) covering almost six months. Figure 5.22 shows the image combinations used for this process. Each image is compared with every other subsequent image, although only the first four images are shown in the figure. The results are then averaged over all combinations that lie within the same time range.

Figure 5.23 shows the detector performance results. From Figure 5.23(a) it can be seen that the repeatability of all algorithms decreases as the seasonal difference increases, and in many cases almost halves the performance. EBR is the exception to this rule and exhibits several outliers to this trend, where the difference in performance at 5–6 months actually improves over that from 4–5 months. Analysing the number of correspondences in Figure 5.23(b) shows a matching downward trend. It can be seen in this graph that EBR decreases between 4–5 and 5–6 months,

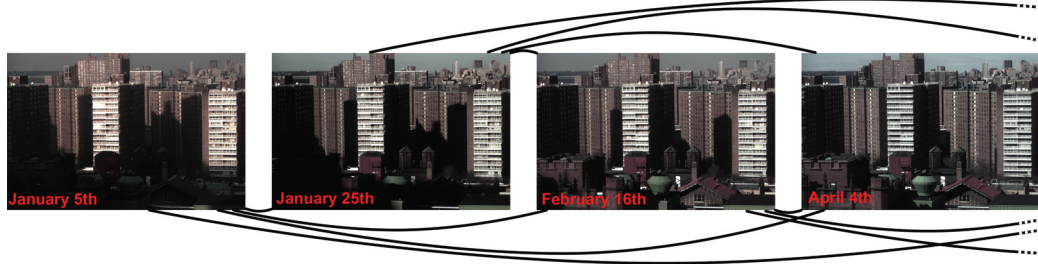


Figure 5.22: Image combinations for computing seasonal performance. Repeatability and matching results are computed for every combination of a given image and every image after it

therefore the increases in repeatability seen in Figure 5.23(a) are likely the result of a disproportionate decrease in the total number of features extracted.

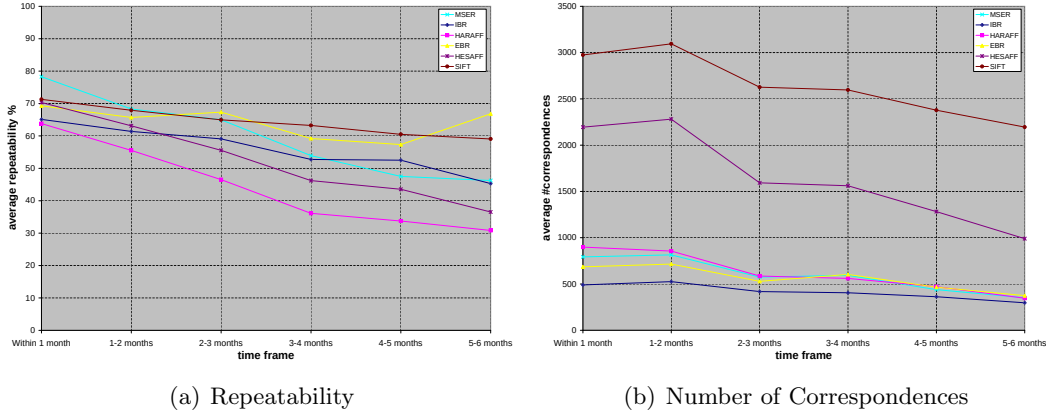


Figure 5.23: Detector performance changes for seasonal WILD selection

The matching results are shown in Figure 5.24. The recall decreases dramatically over the 6 month period, and all algorithms exhibit the same downward trend. This is the result of a disproportionate decrease in the number of correct matches compared with the drop in the number of corresponding regions. The 1-precision results shown in Figure 5.24(c) show that as the seasonal difference increases, the precision decreases. Despite this, the number of false matches shown in Figure 5.24(d) actually decrease over this period, indicating the resulting loss of precision is a product of the substantial drop in the number of correct matches and the number of putative matches found by the distance ratio matching technique.

The impact seasonal changes have on both detection and matching performance is significant. The worst performance is obtained between images taken six months apart, however, the level of performance at this time is still substantially better than the worst performance seen between times from the same day. This indicates shadows do impact performance between images from different seasons, but not

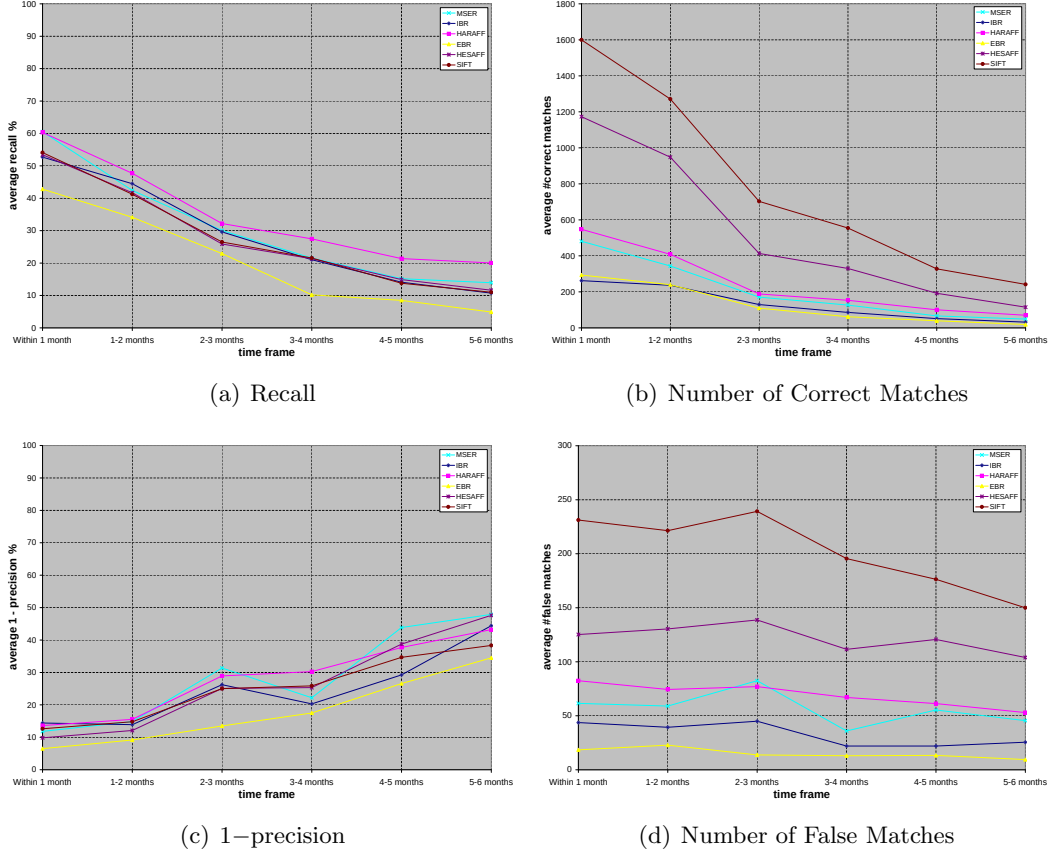


Figure 5.24: Matching results for seasonal WILD selection

as severely as images from different times on the same day. Inspecting the set of seasonal images in Figure 5.21 it can be seen that the difference in shadow configuration is not as prominent as in the clear weather set. For example, all images in this set receive sunlight on the front faces of the buildings, meaning there is always a region in common where features are detected. This, however, is not the case in the clear weather set that is captured on a single day.

5.5.4 Fourier Analysis

The Fourier-based method presented in Chapter 3, Section 3.1, is a global image approach and cannot be evaluated using the same test framework as the other local feature algorithms. Therefore, tests are conducted using the same data set but performance is based on how closely the recovered parameters match the ground truth data. Since the Fourier-based algorithm recovers translation differences between two images, as well as the rotational and scale difference, these parameters are compared with the ground truth values where the translation and rotational differences between the images is ≈ 0 , and the scale is approximately 1. Also reported

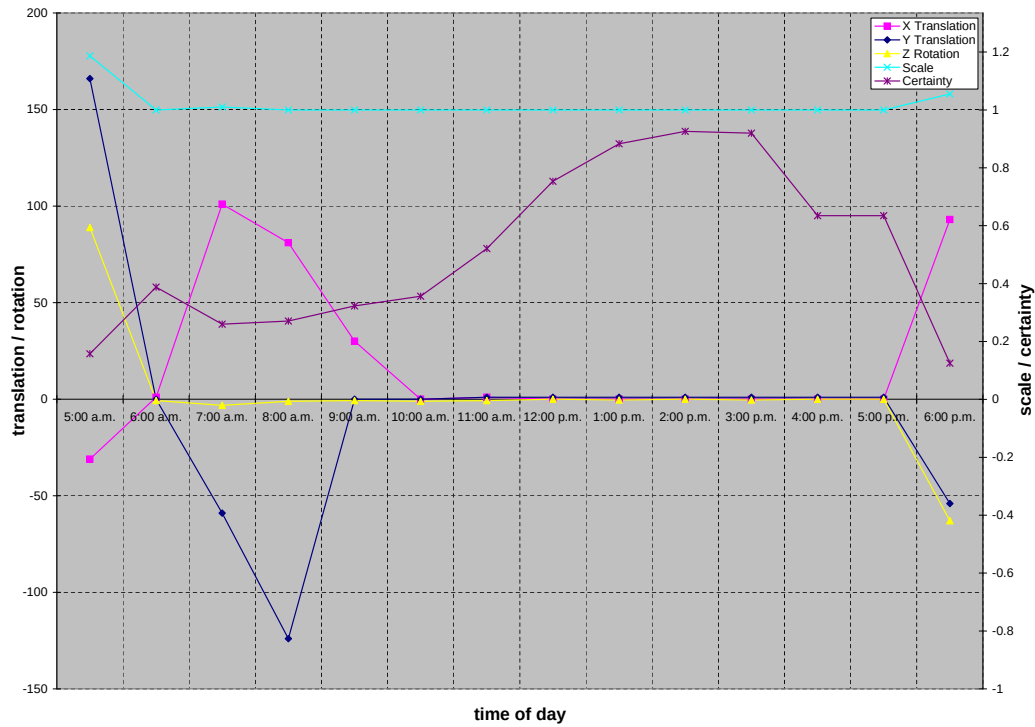


Figure 5.25: Results of Fourier-based algorithm on clear weather WILD selection

is a certainty value. The certainty value indicates the magnitude of the peak value and represents the confidence the algorithm has in these estimates; ideally this value approaches 1.

The results from the Fourier-based method are shown in Figure 5.25 and convey a pattern similar to those produced by the local feature based algorithms. The graph shows that at 5am all estimated parameters are incorrect and the algorithm indicates its uncertainty with a correlation peak value of 16%. At 6am the parameters are estimated correctly and confidence increases, but then dips between 7–10am before steadily rising, peaking at 93% at 2pm. At 6pm the algorithm fails again.

Interestingly, the scale and rotation parameters quickly converge to near correct values from 5am to 6am and remain similar to the actual values until 9am, after which the estimated values are correct. The small errors during this time indicate the peak location used when determining the scale and rotation was close to the correct peak location. In contrast, the translation parameters are highly dissimilar to the true values until 10am. The level of error in the translation estimates is likely the result of the small amount of error in the scale and rotation estimates. Recalling from the algorithm description, scale and rotation are estimated first by performing phase correlation on log-polar versions of the Fourier magnitude spectra of the images. The images are rotated and scaled to reverse these estimated differences,

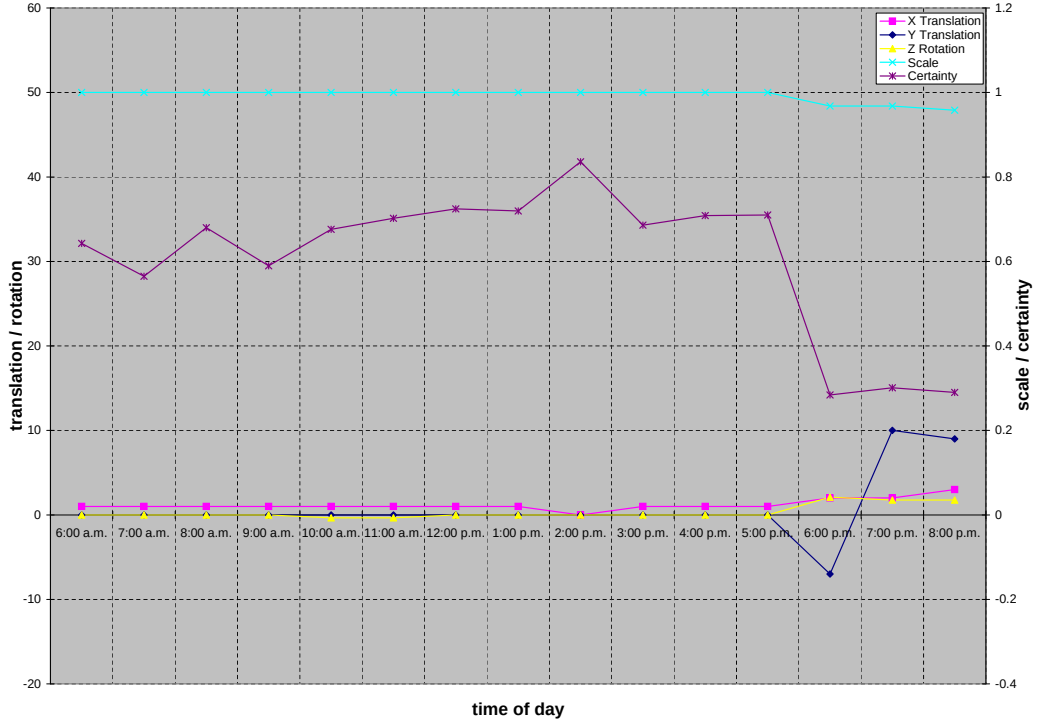


Figure 5.26: Results of Fourier-based algorithm on cloudy WILD images

producing images that differ by an unknown translation. Incorrect estimates of the scale and rotation can cause problems in translation estimation, as phase correlation is susceptible to even small differences in scale and rotation.

The failure of this algorithm operating on the WILD selection is also due to the non-uniform changes in illumination present in this set. Although the Fourier-based approach makes use of properties that are robust to illumination changes, its robustness, like the local feature approaches, only accounts for uniform changes in illumination, not the non-uniform changes present in this image set. Foroosh *et al.* [FZB02] report that phase correlation, which the Fourier-based algorithm uses, is

“robust to those types of noise that are correlated to the image function, *i.e.*, uniform variations of illumination, offsets in average intensity, and fixed gain errors due to calibration.”

Performing the same test on the selection of cloudy WILD images produces a similar effect to that seen when using the local feature based algorithms. These results are shown in Figure 5.26, and confirm that the shadows in the clear weather WILD selection are also preventing the Fourier-based algorithm from working successfully. The graph shows correct results from 6am–5pm, after which the

algorithm fails and precipitates a substantial drop in the certainty of the algorithm from 71% to 28%. The X translation of 1 pixel detected throughout this sequence is a result of the limited accuracy of the manual alignment method. The images are aligned to within 1 pixel; however, the Fourier-based approach does not provide sub-pixel accuracy, therefore these values are estimated at 1 pixel which is deemed correct.

5.5.5 Light Dataset

The shadow effect can also be seen in the illumination dataset employed by Mikolajczyk and Schmid [MS05] and Mikolajczyk *et al.* [MTS⁺05]. This dataset is shown in Figure 5.27. This figure shows the MSER features extracted from each of the images. The features shown in green are those that are correctly matched; those in red are not. It can be seen that the density of features detected in the shadow region of the image—where the cars are—is lower than in the sunlight region. This effect becomes more pronounced as the illumination in the scene decreases.

The influence of illumination change on performance is more subtle in these images, due to the artificial way in which they were created. The images were formed by adjusting the aperture of the camera, which allows less light onto the sensor. This produces a uniform change in illumination across the image despite the fact there is a shadow present in the foreground of the image. The fairly consistent performance seen in their results—in terms of the repeatability and matching score—are the result of a proportional decrease in the total number of features extracted across the entire image, caused by the uniform illumination change. In real situations when shadows are present, uniform changes in illumination are uncommon. Typically the illumination for many regions of the image remain fixed while only those in shadow change. In addition, changing illumination is often accompanied by a change in the location of the shadow, something not modelled in this dataset. This factor would certainly produce a more pronounced performance decrease as the shadow locations in the reference and current images become more dissimilar.

5.5.6 SloDB Selection

The experiments were repeated on a selection SloDB images. This image selection is shown in Chapter 4, Figure 4.10. The images were captured during clear weather over the course of two days. The reference image used in this test was taken at



(a) Original Image



(b) Reference Image



(c) 2



(d) 3



(e) 4



(f) 5



(g) 6

Figure 5.27: Features extracted from the Light change selection of images from [MTS⁺05]. (a) shows the original image. The set of features shown in (b) is the reference feature set. In the remaining images regions in green are correct matches determined by the repeatability tests. The regions in red were classified as incorrect matches, or did not exist in the reference feature set

2:06pm under clear conditions.

Analysing the locations of the features detected as shown in Figure 5.29, it can be seen that the shadows in this image do affect the number of features detected. For example from 8:13am to 1:45pm the shadows can be seen progressively moving down from the front building, across the road, and proceeding towards the location of the camera. The number of features is clearly larger up to the edge of the shadow, after which the number of features is diminished. This effect is not as pronounced as seen in the WILD selection, as the shadows are not as intense, most likely due to the scene structure and the position of the sun.

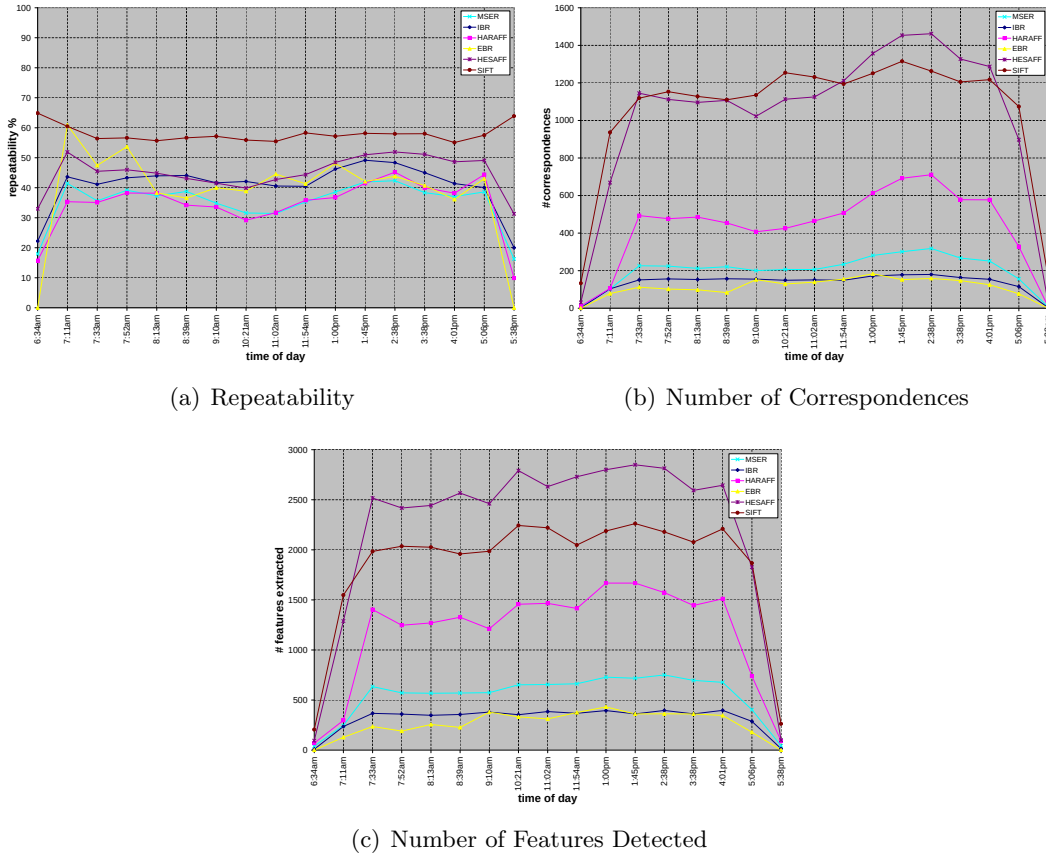


Figure 5.28: Detector performance results for SloDB selection

The detector results are also impacted less by the location of the shadow than in WILD, as is shown in Figure 5.28. All algorithms perform at a consistent level throughout the sequence. The only exceptions are at 6:34am and 5:38pm when it is dark. It can be seen that SIFT consistently outperforms the other algorithms in terms of repeatability. In practice SIFT is better suited to this scene as there are not many large planar surfaces in the image. This means that the surface patches for the regions contain depth discontinuities and fail to take advantage of the affine

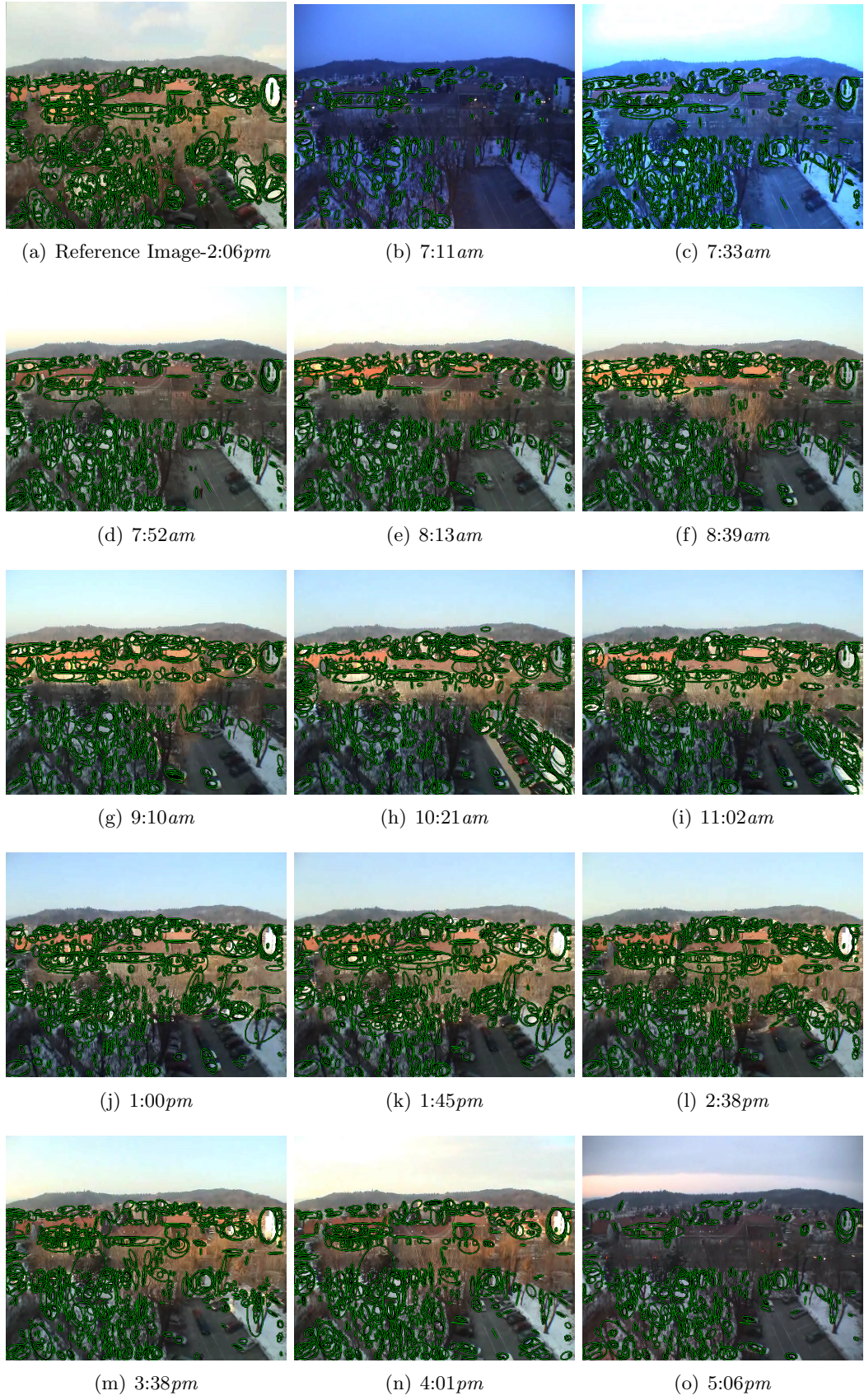


Figure 5.29: MSER features extracted from the SloDB selection. Note that for viewing clarity only 14 of the 17 test images are shown in this figure. See Figure 4.10 for the original images

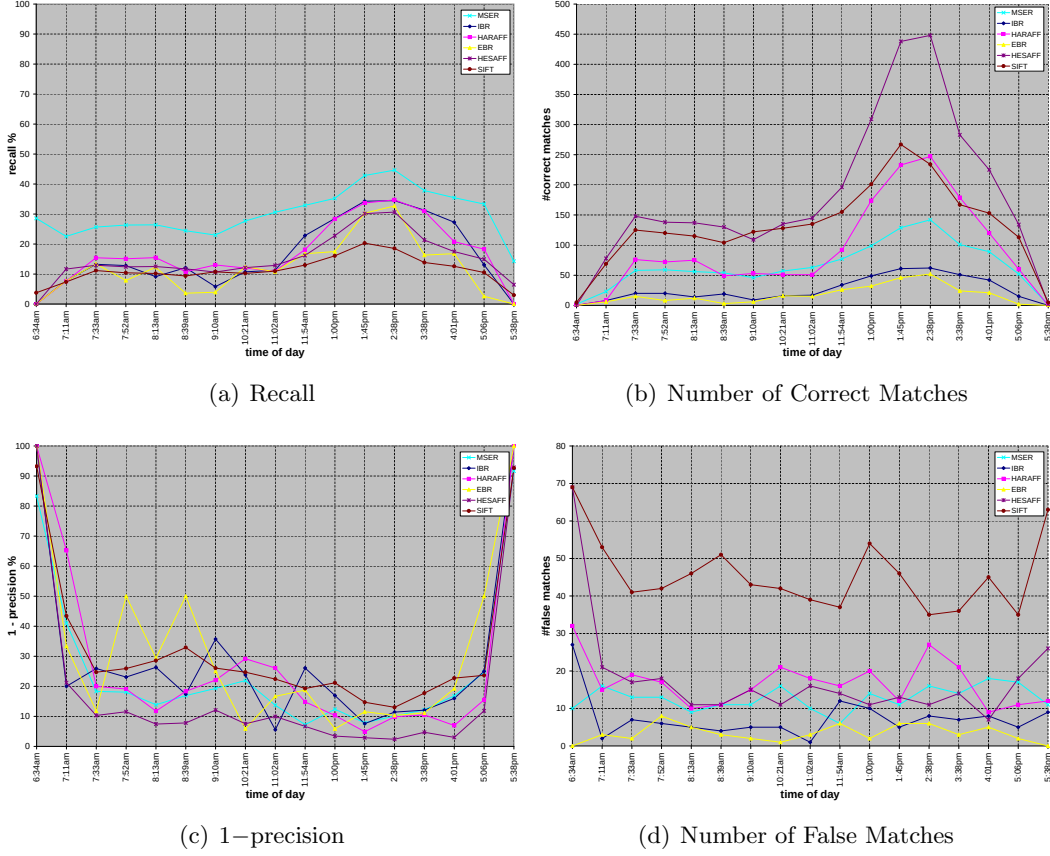


Figure 5.30: Matching results for SloDB selection

invariance of the other approaches. If the camera was closer to the buildings in the background these other algorithms might be better suited.

Examining the number of correspondences in Figure 5.28(b) it can be seen that all algorithms increase quickly from low numbers at 6:34am through to moderate levels at 7:11am. After this, in the case of SIFT and HESAFF, a substantial number of correspondences are obtained at 7:33am as the amount of light in the scene increases. From this point on, all algorithms increase slowly to reach the maximum levels at approximately 2pm, which corresponds to the time of the chosen reference frame. These trends are similar to those seen for the total number of features detected by the algorithms, as shown in Figure 5.28(c).

The matching results are shown in Figure 5.30. The recall results show that all algorithms correctly match only a small percentage of the total number of potential matches (which are the correspondences). Recall scores reach their maximum at 2pm where MSER correctly matches 45% of all correspondences. MSER consistently performs best of all the algorithms. At the earliest time of 6:34am, it even achieves a recall rate of 28%, far greater than any other algorithm. To put this achievement

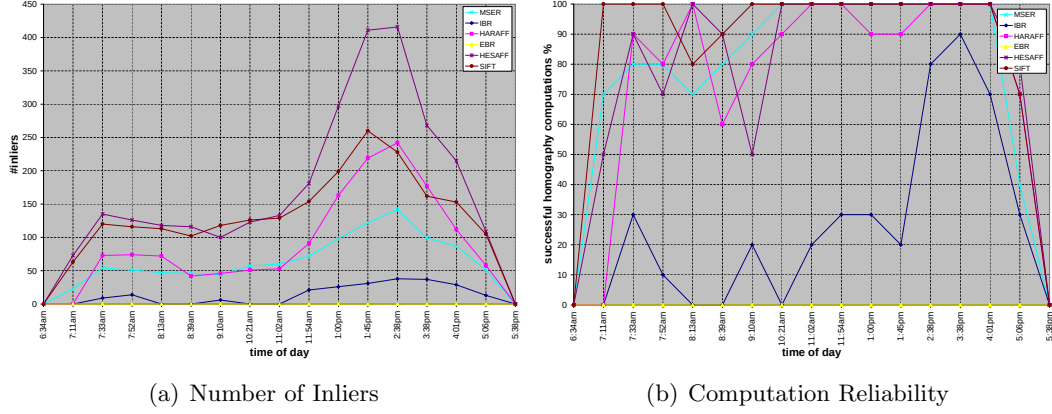


Figure 5.31: Homography computation results for SloDB selection

in perspective however, only two matches were deemed correct out of a total number of seven correspondences. In terms of precision, HESAFF achieves the best results. At 2:38pm it achieves a false positive rate of 2.3% as only 11 of the 459 matches are mismatches while 448 are correct. The 1-precision scores show that all algorithms obtain poor precision at 6:34am which quickly improves at 7:11am and further still at 7:33am. From this time till 4:10pm high precision results are achieved. After this, precision drops off quickly and mirrors the results seen in the early images, with poor scores nearing 100%.

Figure 5.31 shows the homography computation results on the SloDB selection. HESAFF performs best for these images, consistently obtaining the highest number of inliers and reaching a maximum at 2:38pm. Despite obtaining respectable numbers of inliers from 7:33am onwards, HESAFF's performance is unreliable and only produces consistently high performance between 10:21am and 4:01pm. At 8:13am HESAFF obtains 100% reliability, but then drops to 50% at 9:10am before increasing again. The other algorithms show similar trends and only obtain consistent performance after 10:21am. This time roughly coincides with the beginning of the sharp increase in the number of inliers and the improvement in precision leading to the respective performance peaks at 2:38pm. SIFT obtains the best performance and obtains consistent performance except at 8:13am and 8:39am where performance drops to 80%.

5.6 Descriptor Analysis

In this section five different descriptor techniques are evaluated in the context of images with non-uniform changes in illumination, in particular the WILD selection

taken under clear conditions used in the previous section. The descriptor algorithms used here are a subset of those evaluated by Mikolajczyk and Schmid [MS05]. They include SIFT, PCA-SIFT, Shape Context (SC), Spin Images, and GLOH. These are all similar to the SIFT approach. PCA-SIFT uses a large vector of image gradients whose dimensionality is reduced through the use of PCA. GLOH extends SIFT by computing the SIFT descriptor for a grid in log-polar co-ordinates. SC is also similar to SIFT, except it makes use of edges. ‘Spin Images’ use a histogram of intensity values rather than gradients. For each of these descriptors the distance ratio matching technique is again employed.

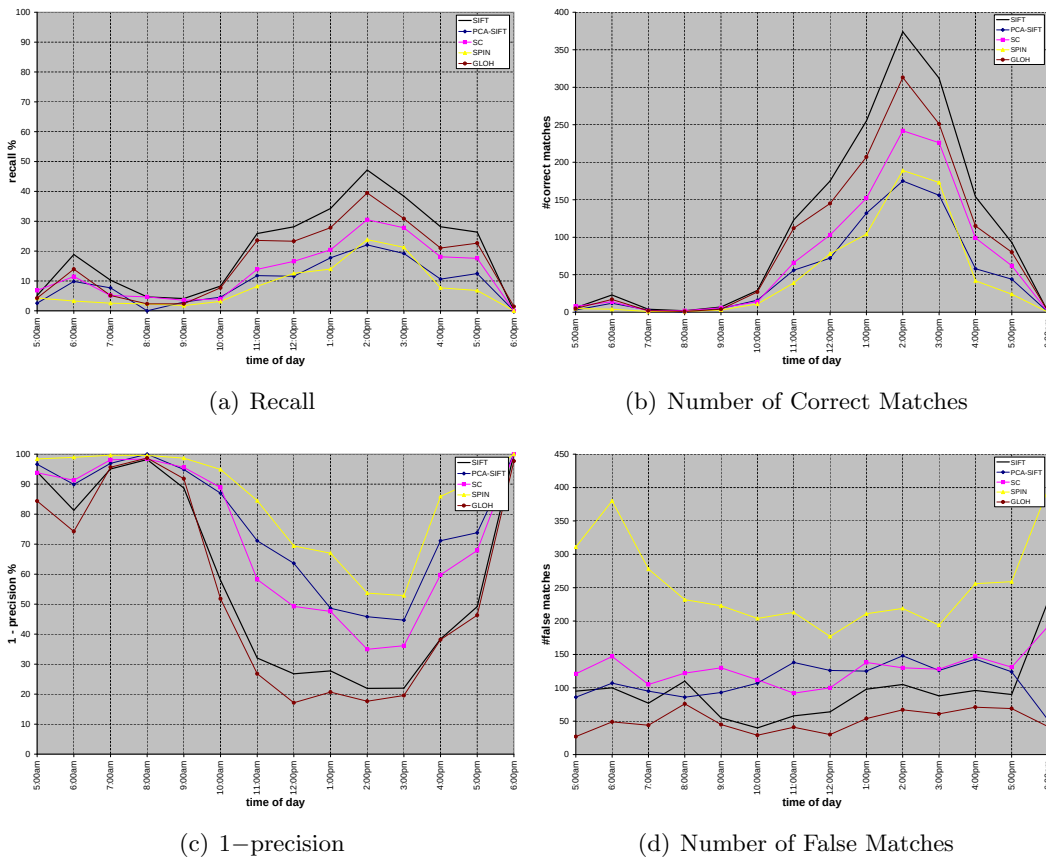


Figure 5.32: Matching results for Different Descriptors on WILD selection

Figure 5.32 shows the matching performance for each of the descriptors. The line in bold is the performance for the SIFT descriptor and represents the default results seen in previous tests. It can be seen that in terms of recall, SIFT descriptors perform the best out of all the descriptor algorithms tested. The number of correct matches shown in Figure 5.32(b) is consistently higher for SIFT. GLOH performs next best, followed by SC, while SPIN and PCA-SIFT exchange positions throughout the sequence. The 1-precision results, however, show that GLOH outperforms SIFT

for virtually every image, and produces a substantially lower number of false matches as shown in Figure 5.32(d). This means that out of all the putative matches found, GLOH obtains a better ratio of correct to false matches. SPIN obtains the worst precision results and produces substantially higher numbers of false matches.

The homography computation results shown in Figure 5.33 reflect the matching results seen in Figure 5.32. Here, however, SIFT is bested by GLOH in terms of reliability at 6am and 10am. At 6am GLOH achieves a 1-precision score 7% better than SIFT. This provides an improvement in reliability of 30% allowing GLOH to achieve 100% reliability at this time. Another significant result is that GLOH manages to achieve 30% reliability at 5am, a time where SIFT descriptors fail completely. This is the result of the improvement in precision provided by GLOH which obtains a significantly lower number of false matches, while finding a similar number of correct matches.

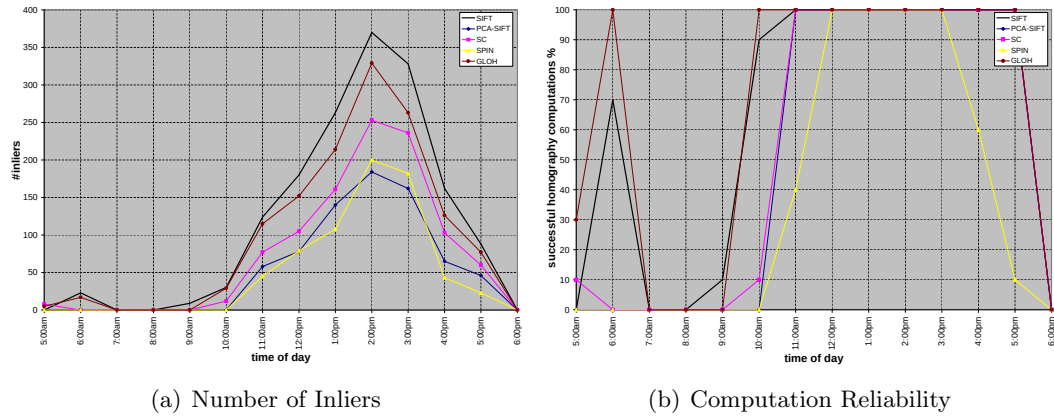


Figure 5.33: Homography computation using a variety of descriptors

Regardless of descriptor choice, the same effect that was seen in the initial tests on the WILD selection is also present here. The shadows greatly impact performance for the early images and none of the new descriptor types alleviate this problem. This suggests that none of the descriptor approaches are robust enough to cope with the differences introduced by non-uniform changes in illumination.

The other factor affecting matching performance is the distance ratio being used. The distance ratio is used in the SIFT descriptor matching process and is described in detail in Chapter 3, Section 3.2.3. The choice of this parameter is investigated next.

5.6.1 Choice of Distance Ratio

Figure 5.34 shows the results of adjusting the distance ratio threshold during matching for MSER features. The bold black line shows the matching performance seen in previous tests using a default value of 1.5. Figure 5.34(a) shows the recall results. It can be seen that reducing the distance ratio threshold increases the recall rate significantly. For most images, the more relaxed the thresholds, the greater the number of correct matches found. As shown in Figure 5.34(b), the number of correct matches is generally proportional to the decrease in the ratio threshold. Most notable is the performance for early images such as 7, 8 and 9 *am*, is higher for more relaxed ratios. When the ratio is 1, the nearest neighbour is effectively chosen regardless of how close the second nearest neighbour is. Performance at these times increases by a factor of two, four and five respectively. The ratio reduction helps the situation considerably, and in the case of MSER increases the recall rate from 10% to 20%.

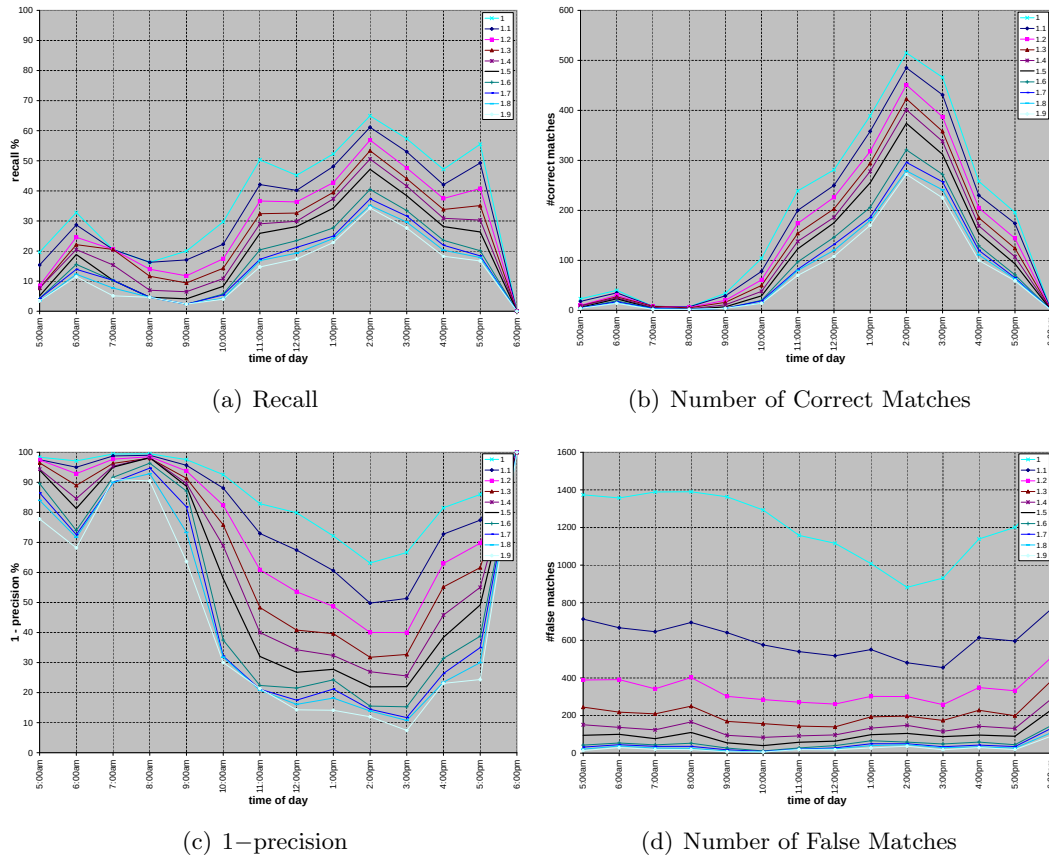


Figure 5.34: Matching results using modified distance ratio thresholds for MSER. The bold black line is the performance using the default value of 1.5. This value was used for all previous tests

Two factors stand out here. Even if the distance ratio is 1, meaning effectively the nearest neighbour is chosen regardless of the distance to the second closest feature, 100% recall is still not obtained. This indicates that a large percentage of the corresponding features are not matched because they are not the closest match, not because the confidence in the match is too low. This could be the result of either a lack of similarity between descriptors from matching regions, or that other descriptors are more similar indicating a problem with the distinctiveness of descriptors in the set. The second factor is that reducing ratio thresholds from the default values produces more matches. This indicates that descriptors that are the closest are not being selected as the distinctiveness from the rest of the feature set is insufficient to be confident these are matches. If the sole problem was similarity between matching descriptors, the number of correct matches would not increase regardless of the distance ratio, *i.e.* the nearest neighbour would still be wrong, so decreasing the threshold to the second nearest match would not help. This is not the case, however, meaning both problems are encountered. By analysing the recall graph when the ratio is set to 1, it can be seen that at 8am, for instance, 84% of the corresponding features are not matched due to the corresponding descriptor not being the closest match. 12% of the corresponding features at this time are not matched as they are not distinctive enough from the other features in the set. The remaining 4% are both similar and distinctive enough to be classed as matches. This is significantly different at 2pm, where the performance is at its maximum. 35% of the corresponding features are not matched due to similarity issues, while 19% are not matched due to a lack of distinctiveness.

Decreasing the distance ratio has a tradeoff in that it increases the number of false matches. In this situation the opposite case from the recall results is seen. The more restrictive the threshold the better the precision performance, where the value 1.9 achieves the best results. The more relaxed threshold adversely affects the normal times of day most, where for instance at 2pm the percentage of false matches increases from 22% when using default values, to 63% when using a ratio of 1. This only produces a gain of 18% in the number of correct matches. This indicates that relaxing the ratio produces a disproportionate increase in the number of false matches over the number of correct matches. This is clearly shown in the absolute number of false matches of Figure 5.34(d), where the number of false matches almost doubles between ratio values of 1.4, 1.3, 1.2, 1.1 and 1. The benefits come for earlier

images. For instance, if a modest ratio of 1.3 is used, the change in the percentage difference in the number of false matches at times such as 7, 8 and 9 *am* is only a few percent, yet the number of correct matches approximately doubles. These results indicate that in such instances, either reduced thresholds are required or perhaps another matching strategy may be better suited, for example the nearest neighbour approach that uses a threshold on the maximum distance.

5.7 Summary

The results in this chapter have demonstrated the significant effect non-uniform changes in illumination can have on feature extraction and matching. In this case the non-uniform changes in illumination were brought about by shadows rather than specular reflections, *etc.* Specifically, problems occur when there are differences between the shadow configurations between a reference image and a live image. If the shadow configurations correspond, for example from similar times of day in the same season, performance is good. As the configurations become increasingly dissimilar, performance drops accordingly. This effect was seen over a range of algorithms, utilising different image properties to locate features, including the Fourier-based approach discussed in Chapter 3, demonstrating that none of the algorithms presented are immune to its effects. Both the detection and matching processes were affected. Detector performance suffered and produced substantially fewer correspondences than temporally adjacent times. Matching performance was also impacted and manifested in low recall rates and reduced precision, indicating that a low percentage of the correspondences that did exist were correctly matched. Other descriptor algorithms provided little benefit over the popular SIFT algorithm.

The results were verified using several datasets. The first utilised a selection of images taken in overcast conditions. This set was devoid of shadows due to the more diffuse nature of the light source, producing more even lighting across the scene. Performance in this instance was far more consistent, and varied little until the images were completely dark. Practical performance was also greatly increased in early images which suffered the most in the selection of WILD images taken under clear conditions. Computation of the homography was successful for virtually every algorithm and was also substantially more reliable.

The SloDB selection was also impacted by the presence of shadows, but performance suffered to a lesser extent than the WILD images. Feature extraction was

visibly affected by the location of the shadows but did not suffer as much due to the low intensity of the shadows brought about by the scene structure and the sun's position. The WILD database, however, produced large shadows due to the large, tall buildings in the scene, which induce large shadows projected onto the rest of the scene. The illumination test set used by Mikolajczyk and Schmid [MS05] and Mikolajczyk *et al.* [MTS⁺05] also showed this effect to some extent. However, the artificial means by which the test images were created meant performance was not impacted as significantly.

The shadow effect was also demonstrated as being present during changes in season and resulted in the same impact as that from different times of day. All these tests demonstrate that performance can be highly variable and unpredictable in the presence of shadows, and is dependent on the time of day, season, and the structure of the scene itself; an undesirable outcome for a system attempting to achieve robust performance in all circumstances.

Several suggestions can be made at this point to reduce the impact of shadows on feature matching. The first is to ensure the reference data is free of shadows. This ensures features are detected over the full extent of the scene, and makes performance only dependent on the quality of the live images. This may not be possible either because an existing set of images is used as the reference data—in which case it is too late to fix—or because there may be no time during the day when the scene is completely devoid of shadows. Alternatively, images captured under cloudy conditions could be used, much like photographers do to maximise the quality of their images. Unfortunately this often brings about a reduction in visibility which may degrade performance elsewhere.

The best solution will involve dealing with the shadows rather than attempting to avoid them. This will likely come from advances on several fronts. Improving the reference data so that it is not so heavily influenced by shadow effects will greatly increase reliability. In the case of AR, reference data is typically processed offline, therefore there is greater flexibility in the amount of processing time allocated to improving the reference set. Improvements in the current images is also required. Increasing the number of features extracted in shadow areas could potentially improve performance. Furthermore, improving the matching by fine tuning algorithm parameters may be of benefit. The following chapter looks at these methods of improving the performance in detail.

Chapter 6

Dealing with Shadows

“To raise new questions, new possibilities, to regard old questions from a new angle, requires creative imagination and marks real advances in science”

Albert Einstein (1879–1955)

Most computer vision image or feature matching systems assume unchanging or uniformly changing illumination. This assumption is violated in many practical situations in which AR might be applied. The previous chapter demonstrated several scenarios where shadows impacted performance, and in the case of the WILD selection completely prevented successful operation at several times of day. This variable and unpredictable performance is an undesirable characteristic in a computer vision system. At a minimum it decreases confidence in the system, and in the worst case will render the system useless, given that immersive AR systems require near perfect registration for useful real-time operation. Therefore, having shown conditions under which shadows can impact performance and the degree to which they do so, the next question is: can these effects be lessened or mitigated? This chapter seeks to answer this question.

Achieving performance improvements in the presence of shadows may be addressed from two perspectives. Both would likely be required in order to maximise robustness. The first is to attempt to minimise shadow impact on the live images. As was shown in the previous chapter, performance is at its worst when the shadow overlap between the reference image and the live image is at its lowest. Non-overlapping shadow regions decrease image overlap and therefore decrease the image

area where correspondences are possible. This reduction in correspondences in turn reduces the number of matches, breaking later geometric entity computation stages. Secondly, the previous chapter also showed problems in matching images under disparate illumination conditions. If the problem was solely the result of poor detector performance, a high percentage of the correspondences that did exist still should have matched. This was not the case however. The results showed low recall scores between the *2pm* reference frame and early live images such as *8am*, *i.e.* re-detected features were not being matched.

There is room for improvements in both the detection and matching stages. In the detection stage, improvements can come if features can be extracted inside the shadow areas of the live image, which increases the image overlap. Ensuring features are detected in these regions may not be enough as these new features must still be matched to corresponding features in the reference set. It is, however, the first step required. The second aspect of improving performance is to increase the number of features detected in the shadow regions of the reference data set. This increases the degree of image overlap, thereby increasing the chances of successful matching. The first section of this chapter deals with improving detection in live images. The second section looks at an algorithm to improve the reference feature set.

6.1 Improving Detection in Live Images

Increasing the regions of feature overlap between the reference feature set and the live image feature set should result in a higher number of correspondences and the possibility of an increased number of matches. One method for increasing the number of features detected in these shadow regions is to use relaxed thresholds for the detection algorithm, allowing it to use features not considered under default parameters. Another option is to apply an image enhancement algorithm to the image in order to enhance the data in shadow regions, producing more features and the chance of additional matches.

To test these hypotheses, knowledge of shadow locations in the image is useful as it allows different operations to be performed in shadow regions. In the case of image enhancement operations, it allows features to be selected from enhanced shadows, while in the rest of the image the standard feature set is used.

A number of research groups have addressed the issue of shadow identification. Salvador *et al.* present an approach for identifying shadows and classifying them

as either self or cast shadows [SCE01]. Potential shadow regions are identified by performing edge detection and then locating regions that are darker than their surroundings. Edge detection is then performed on an illumination invariant colour model of the image. Hue and saturation are examples of colour spaces that are invariant to shadows and shading, however, Salvador *et al.* found that the $c_1c_2c_3$ model introduced by Gevers and Smeulders [GS97] produced the best results. The $c_1c_2c_3$ model is formed from a combination of the RGB colour channels, such that

$$c_1 = \arctan \left(\frac{R}{\max\{G, B\}} \right), \quad (6.1)$$

$$c_2 = \arctan \left(\frac{G}{\max\{R, B\}} \right), \quad (6.2)$$

$$c_3 = \arctan \left(\frac{B}{\max\{R, G\}} \right). \quad (6.3)$$

This model only works under the assumption of white light and matte, dull surfaces that do not produce reflections. The edges from this illumination invariant $c_1c_2c_3$ model are then used to classify the detected shadow as falling inside the contour of the object—in which case it is a self-shadow—or falling outside where it is classified as a cast shadow. The usefulness of the approach is limited to simple scenes where shadows are cast on flat, non-textured surfaces that are uniformly coloured. Also, the technique only supports one light source, which must be strong enough to produce clear shadows.

Levine and Bhattacharyya also presented an automated approach to shadow detection and removal [LB05]. Potential shadow boundaries are located in the image and then shadow regions are obtained from this boundary information. Shadow boundaries are located by segmenting the image and then examining pixel boundaries between segmented regions. Pixel values produced by a camera are modelled as a combination of the illumination from a light source, the surface reflectance, and the camera sensitivity function. A ratio between two pixels can be computed using:

$$ratio_{KSN} = \frac{p_{KS}}{p_{KN}} = \frac{c_{ij}E_L(\lambda_K) + A}{E_L(\lambda_K) + A} \quad 0 \leq c_{ij} < 1 \quad K = R, G, B. \quad (6.4)$$

E_L is the illumination provided by the light source while A is the ambient illumination. c_{ij} represents the illumination intensity change between the pixels and is influenced “by the geometry of the scene, such as the angle of incidence of the

illuminant and the viewing angle.” p_{KS} and p_{KN} are boundary pixels, one in shadow, the other not. As p_{KS} is in shadow its intensity must be lower than the pixel out of shadow, therefore c_{ij} is between zero and one. The formula states that two boundary pixels with the same surface reflectance and illuminated by the same light source, E_L and ambient light, A , will give the same ratio in all three colour channels. Therefore, by extension, any pixel that violates this constraint is either not in shadow, covers different surfaces, or is illuminated by a different light source. The approach also supports pixel boundaries lying in both the umbra and penumbra areas of a shadow.

The shadows can even be removed by assigning shadow regions the same colour of non-shadow neighbours, assuming they have the same material. The usefulness of this approach is again limited to shadows that lie on common surface materials. Therefore, when shadows are cast over a range of objects and surfaces—like those in SloDB—the algorithm may be unable to locate them. The examples presented by Levine and Bhattacharyya indicate the approach is better suited to locating shadows on faces *etc.*

These projects show that automatic shadow detection is possible, thus this chapter addresses the question: would it help improve feature matching? Shadows in each of the images are manually identified as an implementation of one of the presented automatic shadow detection algorithms was not available and is not needed to successfully answer this question. Figure 6.1 shows an example of the shadow regions identified in the *2pm* image. These regions can then be used to mask out either all features inside the region to obtain a feature set not in shadow, or alternatively, mask out all features outside the shadow region. Note that for the sake of simplicity only shadows in the centre band of the image are identified. Shadows on the smaller buildings in the foreground are not masked out. The vast majority of features are extracted in this centre region, thus it is the most important.

6.1.1 Relaxed Thresholds

One way to increase the overlap between early images and the *2pm* reference frame is to relax the thresholds used by the algorithm in order to increase the number of features detected, especially in the shadow regions of early images. Modifying the thresholds brings the possibility of improvements in performance. In previous tests, default parameters have been used for each of the algorithms. Generally, these parameters have been determined experimentally by the algorithm developers, and



Figure 6.1: The shadow region for the $2pm$ image is outlined in red

are intended to give good performance over the largest range of conditions. This, however, does not mean the parameters provide ‘optimal’ performance in every situation. It is possible that in more specialised circumstances such as those present in the WILD selection, algorithm performance could be improved using modified values.

To test this hypothesis, one feature detection algorithm was tested under a range of parameter modifications. Only one algorithm was used, as using all six, together with all their possible parameter choices would have been impractical. Although the way each algorithm works—and hence the parameters available for modification—are different, all algorithms provide parameters that could improve their performance. The algorithm chosen in this example is MSER, as it often provided the best matching performance in many tests in the previous chapter. Furthermore, MSER provides several interesting parameters that could improve performance in regions of shadow.

There are three parameters in MSER that could be useful for improving the algorithm’s performance. The first is called the “minimum size of output region,” and will be referred to symbolically as ms . Regions smaller than the minimum size are not selected for use. Lowering this parameter from its default value of 30 pixels permits smaller features to be used. Larger features tend to be more stable than smaller ones, therefore lowering this parameter can yield an increased number of

non-corresponding regions and false matches.

The second parameter is called the “minimum margin,” and will be referred to symbolically as *mm*. Recalling Chapter 5, Section 5.3.2, MSER operates by searching for regions that are stable under binarisation using changing thresholds. This parameter specifies the minimum number of threshold levels that a region must be stable over to be considered a valid feature. Lowering this parameter allows features whose size is stable over only a limited number of thresholds to be used. This value has a default value of 10. It is anticipated that this parameter will be especially relevant, as features inside shadow regions are likely to be less stable than those outside shadows. This means the default settings may be too stringent to detect any features at all.

The final parameter is called the “use relative margins” toggle, and is referred to symbolically as *rel*. This compensates for the non-linearity of the intensity function by adaptively applying different minimum margin settings over the image. For example, less restrictive *mm* values are selected for lower intensity regions such as shadows, allowing more features to be detected, despite being stable over a lower number of threshold levels. It is expected that this parameter will prove the most useful as it provides more features in shadow regions, while maintaining similar feature numbers in illuminated regions, which already produce good performance with default parameter settings.

Seven different parameter combinations were tested. The values for each parameter were selected in order to cover a range of values around the default settings for the parameters. These combinations are: *mm*=6 and *ms*=10, *mm*=6 and *ms*=25, *mm*=8 and *ms*=30, *mm*=6 and *ms*=30, and *mm*=10 and *ms*=30 (the default parameter values). The last two combinations use default *mm* and *ms* parameters with the *rel* parameter enabled on the live images, and where the final configuration enables *rel* on the reference frame as well. The bold line in the graphs represents the performance from previous tests using default algorithm parameters (*mm*=10 and *ms*=30). The red line shows the performance with the *rel* parameter enabled on both the live images and the reference image, while the green line uses standard setting on the reference frame, with the *rel* parameter enabled on the live image.

Figure 6.2 shows the detector performance. Examining the repeatability scores in Figure 6.2(a) when *mm*=6 and *ms*=10, it can be seen that up until 11am,

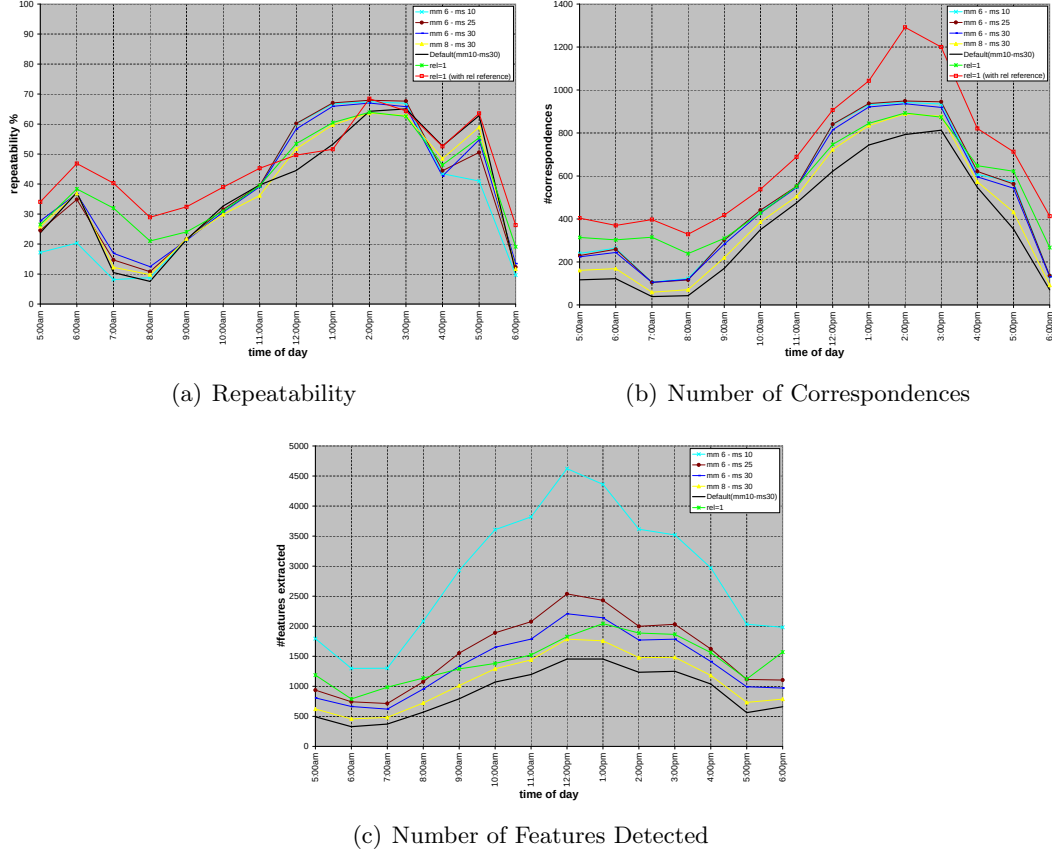


Figure 6.2: Detection results for MSER using relaxed thresholds. The bold line shows the results using default parameters where $mm=10$ and $ms=30$.

reducing both the mm and ms parameters has a detrimental effect on performance. Figure 6.2(b), however, shows an increase in the absolute number of correspondences. The drop in repeatability score therefore, is a result of a disproportionate increase in the total number of features extracted compared to the increase in the number of correspondences. This performance drop is not shown in the repeatability scores between 12–3pm however, as the performance is better than default. Higher repeatability scores are only produced because the divisor is set to the number of features extracted in the reference image (which is 1397), since that is the lowest of the totals. A better indication of this is 12pm, where the number of correspondences increases by 34% while the total number of features at this time increased by 218%, demonstrating the disproportionate increase in the total number of features is still in effect. If the repeatability scores were computed using the total number of features detected in the live image as the divisor, the results would be significantly worse.

The graphs also show that virtually identical performance is obtained regardless of the ms parameter. For example, examining Figure 6.2(b) it can be seen that

when $mm=6$ the number of correspondences produced is virtually identical when $ms=10$, 25, or 30. The repeatability scores show some differences. However, these differences only occur up until 8am and after 4pm. The differences exist because the repeatability is computed using a different divisor at these times. When $ms=25$ or 30, the repeatability score is computed with the divisor set to the total number of features found in the live frames. Between 8am and 4pm the total number of features in the reference frame—which equals 1397—is used for all values of ms . These results indicate that lowering the ms parameter in only one of the images being matched provides no benefit in terms of additional corresponding features, despite a large increase in the total number of features detected. As might have been expected, the small features detected using lower ms values are not present in the reference set, making these additional features worthless.

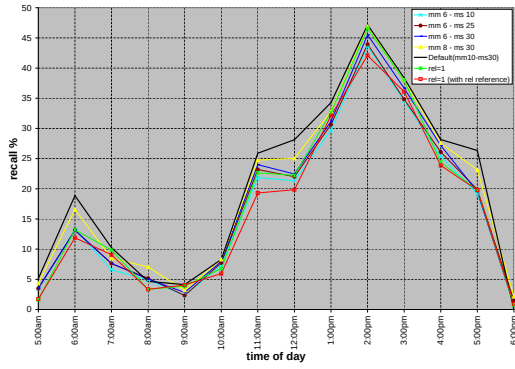
Comparing the performance between different mm settings (with ms at default levels) shows that lowering the value of mm produces more correspondences and often higher repeatability scores. At 6am the increase in the number of correspondences is proportional to the increase in the total number of features. Therefore, repeatability scores are similar to those produced by default parameters. At 7 and 8am reduced mm values produce a slight improvement in repeatability scores. At 9, 10, and 11am the divisor used when computing the repeatability scores is different for the default settings and the other combinations. The results generated from default parameters are computed using the total number of features in the live image as the divisor. In contrast, combinations using relaxed parameters utilise the total number of features in the reference frame as the divisor. This difference means the low number of correspondences generated using default parameters is divided by a small divisor. The relaxed combinations, however, have higher numbers of correspondences but use a higher divisor, cancelling out any performance advantage and producing near identical repeatability scores over these times. From 12pm to 4pm the divisor used by all mm combinations is the total number of features detected in the reference image. This produces increases in the repeatability scores for relaxed combinations. After 4pm performance for relaxed combinations is worse than at default levels due to a disproportionate decrease in the number of correspondences.

The previous results show performance when relaxed detection thresholds are applied across the entire image. Now performance inside shadow regions is examined in isolation, specifically in images from early times of day that exhibited some

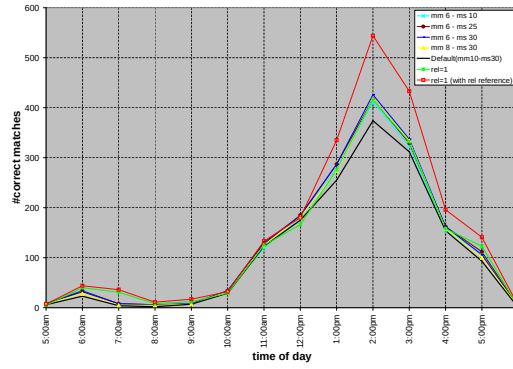
improvement in performance. By applying the shadow mask to the *7am* feature set, it was found that in the shadow region the number of features detected increased from 22 to 97 when the *mm* parameter was set to 6; a 341% increase over default settings. The illuminated portion of the image—which covers a larger proportion of the image—only jumped from 350 to 524; an increase of only 50%. The new correspondences added due to the relaxed parameters also follow a similar trend, where virtually all new additions appear in the shadow region. These results were also verified for the *8am* image, where the number of features detected in the shadow region increased by 374%. This indicates that relaxing the thresholds has the desired effect and does allow more features to be extracted, particularly in shadow regions. This increases the probability of obtaining matches in later stages. These results also show that shadow detection and masking may not be essential, as applying relaxed constraints over the entire image—as opposed to applying relaxed constraints only in shadow regions—still yields performance improvements.

The results with the *rel* parameter enabled show a marked improvement for the early images. At 7 and *8am* the number of correspondences jumps from 39 to 315 and 43 to 239 respectively. The large performance hit from 6–*7am* is also eliminated, with only a drop for 7–*8am* present. From 10–11*am* the results closely follow those for where *mm*=6, indicating that the algorithm selected the same *mm* parameter at this time. From 12–3*pm* however, the results are similar to when *mm*=8. The last test showed that decreasing *mm* can help performance, particularly in the shadow regions. The *rel* parameter applies different *mm* parameters adaptively across the image so it provides a similar effect. This parameter is beneficial in that it provides reduced *mm* thresholds in shadow regions while maintaining high thresholds in high intensity regions of the image. This means more repeatable features are extracted in shadow regions, while a consistent number of high stability features in the high intensity regions are maintained. The dark portions of the early images likely benefit from an *mm* value less than six, as performance at these times is even higher than combinations where *mm*=6. Use of the *rel* parameter is also beneficial in that it produces a lower number of features than many of the other parameter choices. When the feature set for the reference frame is also extracted using the *rel* toggle, a further increase in the number of correspondences can be obtained, yielding the highest number of correspondences out of any approach. The number of features extracted from the reference frame jumps from 1397 to 2022.

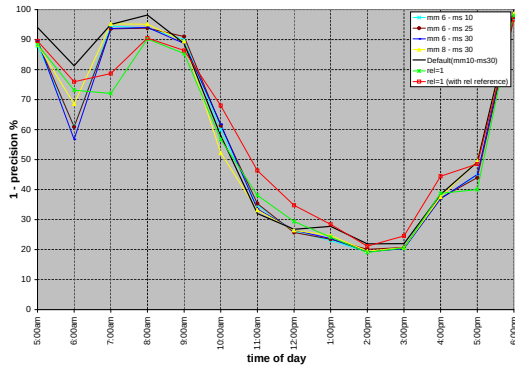
The performance of the matching stage is now analysed. The recall scores of Figure 6.3(a) show that default parameter values often provide the best performance. Note that in this case the scale of the recall graph is from 0–50%. The reduction in performance for all the other parameter settings means that the increase in the number of correct matches shown in Figure 6.3(b), was not proportional to the increase in the number of correspondences. Looking at the absolute number of correct matches in Figure 6.3(b), however, shows that from 6–9am the modified parameters did provide a small benefit. The *rel* toggle provides the largest increase over this time period. At 7am the number of correct matches jumps from 4 to 31, while at 8am it goes from 2 to 8. There is also some benefit at 2pm where performance peaks for all values.



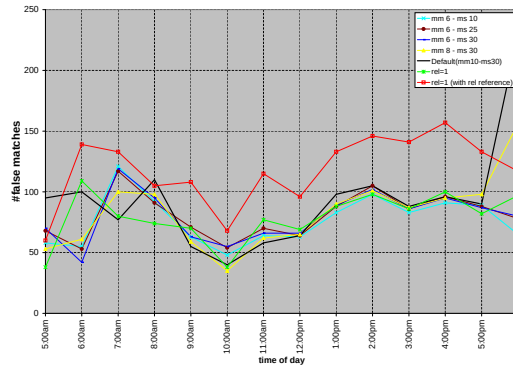
(a) Recall



(b) Number of Correct Matches



(c) 1-precision



(d) Number of False Matches

Figure 6.3: Matching results for MSER using relaxed thresholds. The bold line shows the results using default parameters where $mm=10$ and $ms=30$.

The precision of MSER using modified parameters also shows some improvement. As can be seen in the 1-precision results shown in Figure 6.3(c), the early images achieve higher precision when using more relaxed parameters. The best results at 6am are obtained when the mm parameter is at its lowest. However, at 7 and

8am the precision when *rel* is enabled is significantly better. Using the standard reference frame the 1–precision score at 7am improves from 95% when using default parameters, to 72% when *rel* is enabled. The near identical number of false matches at this time means the improvement in performance is because of the eightfold increase in the number of correct matches. At 8am the performance decreases significantly to 90%, yet still remains better than the default level at 98%. At 9am performance begins to converge to default levels, and remains at similar levels for the remainder of the images. The absolute number of false matches shown in Figure 6.3(d), shows most settings producing similar levels to default; the exceptions to this being when *mm*=6 and *ms*=30, which consistently produces higher numbers of false matches from 7am onwards, as is also the case when the *rel* parameter is enabled on the reference image. The fact that the 1–precision score remains at almost default levels indicates the rate of change in the correct and false matches is proportional.

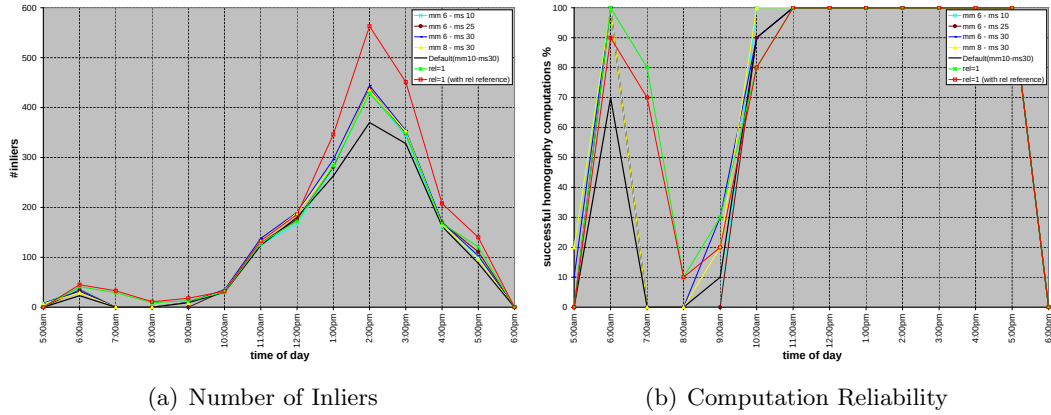


Figure 6.4: Homography computation results for MSER using relaxed thresholds

Figure 6.4 shows the effect these relaxed parameters have on a realistic application scenario. As can be seen, the *rel* parameter in particular can provide a significant increase in performance for images containing dissimilar shadow configurations from the reference image. In particular, when the *rel* parameter is enabled solely on the live images, between 6–9am, reliability increases by 30–80%. The 7am image increases by 80% which is particularly significant given that homography computation completely failed for this image when using default parameters.

Relaxed Thresholds Summary

Summarising the performance changes brought about by relaxed threshold arguments, it can be seen that improvements are possible for some times during the day. This is not surprising as default algorithm parameters provide no guarantee of yielding the best performance in every situation, but are determined experimentally by the algorithm designers to provide the best results over a range of typical situations. The images with the most to gain are those whose shadow regions do not correspond to the the reference frame. In these experiments a *2pm* reference frame was used. Therefore, images taken at 7, 8, and 9*am* contained the most dissimilar shadow configurations and consequently the worst performance.

Three options exist when using relaxed parameters in the feature detection stage of a complete system. The first option is to apply the relaxed parameter settings over the entire image. The results in this chapter followed this approach and provided improvements in detection performance. Tests indicated that relaxed detection thresholds benefited shadow regions more than regions under direct illumination. This suggests that matching performance should also improve. Results showed otherwise, with little or no gain in matching performance due to the significant size increase of the live feature set. Using relaxed parameters over the entire image resulted in increased numbers of features in both the shadow and illuminated portions of the image. Performance in the illuminated regions is already good. Therefore, applying relaxed thresholds on illuminated regions increases the number of unstable features. This is detrimental to performance as the additional features increase computational complexity and increase the probability of obtaining false matches. Mikolajczyk and Schmid [MS04] identify the downside of larger feature sets, stating “the unnecessarily high number of points increases the probability of mismatches and the complexity of the matching algorithms”.

The second option is to make two detection passes on the image; the first using default parameters, the second using relaxed threshold settings. With knowledge of the shadow locations in the image these two feature sets can be combined. The final set would be comprised of the features detected in the illuminated portion of the image using default parameters, and features found in the shadow portion of the image when using relaxed parameters. One disadvantage of this approach is that two feature detection passes would be required for each frame, which doubles the computational cost of the feature detection stage. Another downside is that only

one parameter setting can be used over the entire shadow region. This means an image containing shadows of varying intensity would still have the same threshold setting applied for all shadow regions, resulting in less than optimal performance.

Lastly, the third option is to use an adaptive threshold selection approach, like that provided by enabling the *rel* parameter. Enabling the *rel* parameter provides a good demonstration of the effectiveness of an adaptive technique, applying different thresholds to various regions of a single image. Adaptively selecting threshold parameters avoids the disadvantages of the second option discussed earlier. The approach is embedded in the MSER algorithm and threshold selection occurs during feature detection. Therefore, multiple passes of the algorithm are not necessary. The algorithm also copes with shadows of varying intensity in a single image, as thresholds are selected based on the intensity of local regions in the image. This means there could be many threshold settings applied to a single image.

6.1.2 Image Enhancement

The second option for improving performance in the live images is to apply an image enhancement operation before feature detection. This may increase the number of features detected in the shadow regions of the image; however it may have a detrimental effect on regions which already have adequate image quality. For this reason many operations should not be performed over the whole image, but rather the area in shadow only. To this end, shadow masks as used to combine features found in the shadow region of the enhanced live images, with the features found outside the shadow region of the original images.

The danger with these operations is that due to lower signal to noise ratios in shadow regions, any image enhancement will also amplify the noise. Three approaches are discussed in this section: two simple approaches using a contrast and gamma adjustment, then a more advanced algorithm called homomorphic filtering.

Contrast Adjustment

A contrast adjustment was applied to each of the images in the WILD selection to increase the contrast by $2.5\times$. This value was determined experimentally and produced the largest improvements out of the values tested. The features extracted from these images were then masked so that only the features in the shadow region were used. This set was then combined with all the features outside the masked

area from the unadjusted version of the image. Because the *5am* and *6pm* images were taken in the dark, and therefore contain no shadows, they were omitted from this test.

Figure 6.5 shows the difference in detector performance from the unadjusted results. The difference in repeatability performance of Figure 6.5(a) shows that up until *12pm* virtually all algorithms provide increased performance. At *7am* MSER increases by 21%, and 14% at *8am*. The performance of all algorithms in all these graphs is identical at *12pm* because in this image there are no shadows, therefore the entire live feature set is comprised of features from the unadjusted set producing the same results. After *1pm* performance drops to levels below the unadjusted results. This is a result of substantial increase in the total number of features detected as shown in Figure 6.5(c) with only a minor increase in the number of correspondences. The worst performance comes at *2pm* when the shadow overlap with the reference image is almost 100%, meaning that any new features found in the shadow region of the live image are not in the reference image. Therefore, despite the number of correspondences remaining similar, as a percentage of the total features extracted, performance is worse.

Analysing the matching performance of Figure 6.6 it can be seen that the benefits to the matching process are not as great as those seen in detector performance. At *7am* despite obtaining an additional 150 new correspondences, MSER only yields another 13 correct matches. To put this into perspective, however, at this time in the original data only four matches were correct; using the enhanced version 17 are now found, demonstrating a large percentage increase. The disproportionate increase in the total number of correspondences means at this time recall is 1% less than in the unadjusted case. The number of correct matches shown in Figure 6.6(b) illustrates that all algorithms except EBR produce more matches in the early images. During mid-morning the difference is negligible, and when performance is best in the unadjusted case, the performance from the enhanced images is less.

Interestingly, the precision of the algorithms improves for the enhanced images, at *7am* in particular. At this time MSER improves by 20%, dropping from 77 false matches to only 51. This performance difference quickly converges to levels similar to default for the remaining images later in the day.

Figure 6.7 shows the difference in performance when attempting to compute a homography using the contrast enhanced shadow regions. Mixed results are obtained

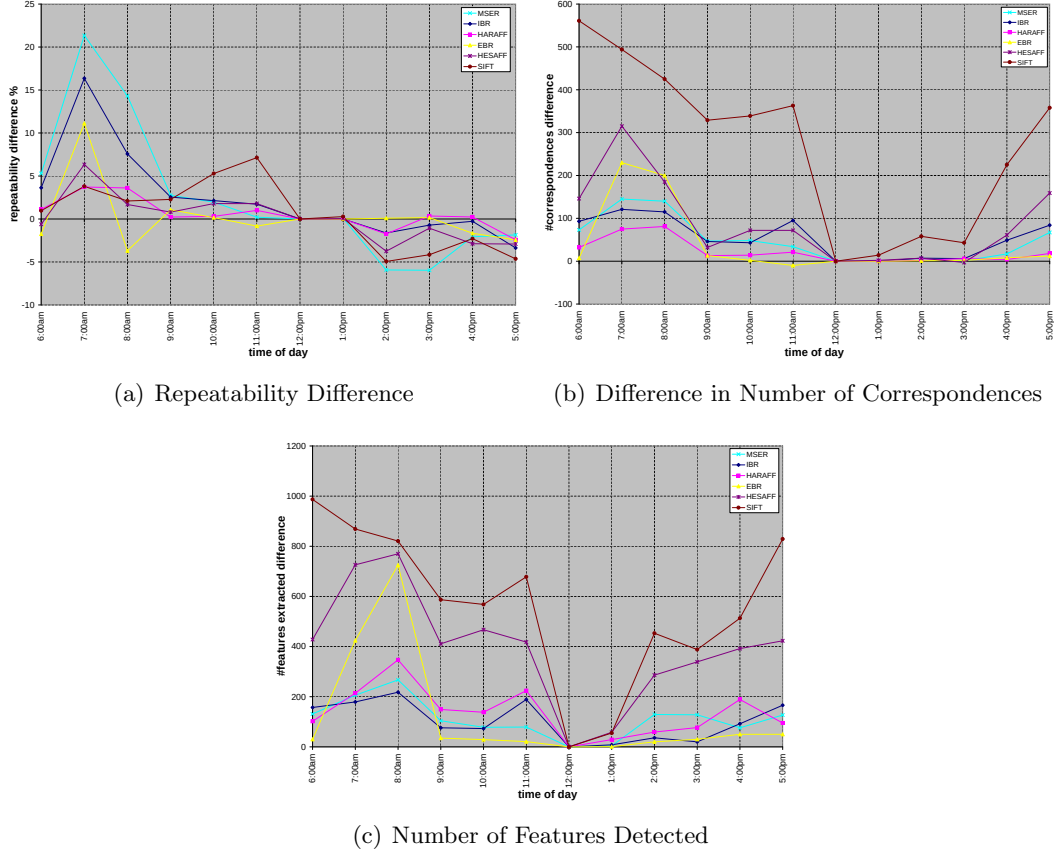


Figure 6.5: Differences in detection performance for contrast adjusted shadow region of clear weather WILD selection

here. For the MSER algorithm, the contrast enhancement provides improvements in some images, and drops in other images. At 7am the enhancement enables the homography to be computed, where it had failed in the original test. The reliability at this time, however, is only 10%. At 9am the computation fails, where in the original data it achieved 10% reliability. This failure is the result of the small increase in the number of false matches, indicating the original data was on the brink of complete failure. SIFT also fails at 9am with the enhanced images. However, unlike MSER it achieves substantially better performance at 7am. This success is the result of a large increase in the number of correct matches—with 36 additional correct matches—together with a drop in the number of false matches. The impact on HESAFF mirrors that seen in SIFT. At 7am performance increases significantly, considering that the original data failed to compute a homography at all. Like SIFT the reliability at this time is 80%. A typical trend seen in these graphs is that the contrast enhancement provides a detrimental effect on performance at 9am. The remaining images show improvements in the number of inliers found, but little change is exhibited in the reliability of the approaches. This is to be expected, as

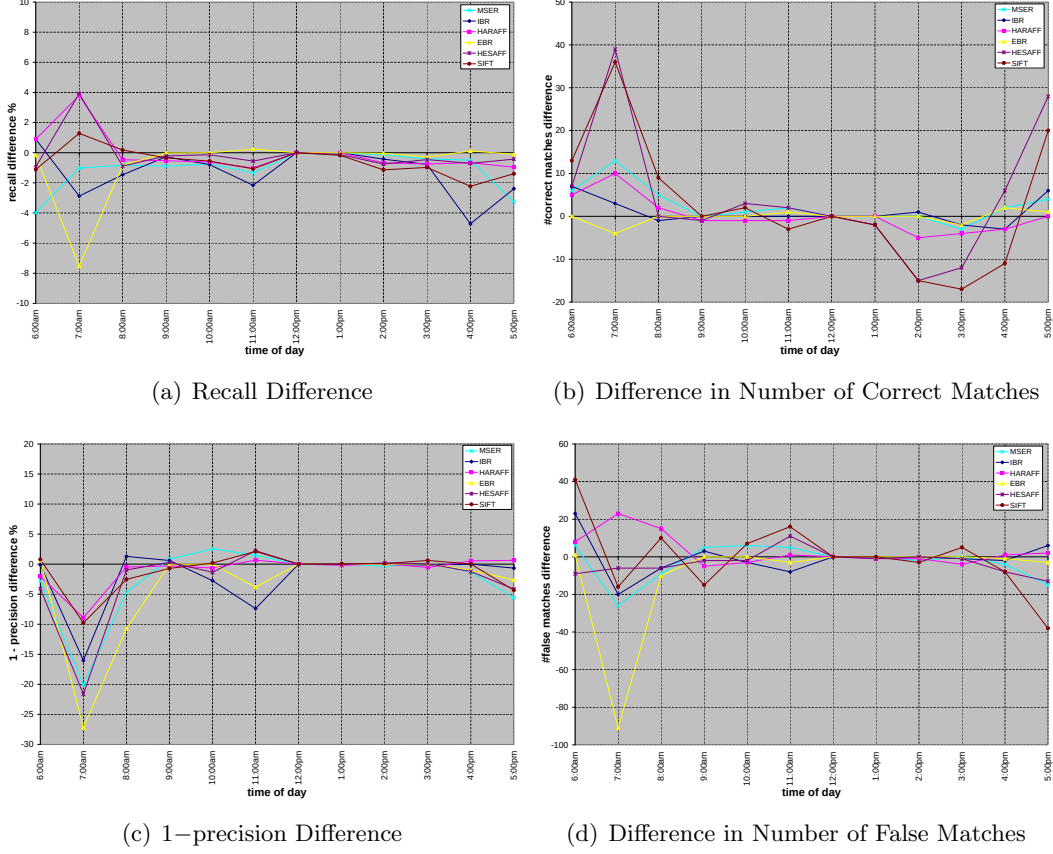


Figure 6.6: Difference in matching performance for contrast adjusted shadow region of clear weather WILD selection

the original data obtained 100% reliability for these images already.

Gamma Adjustment

Gamma adjustment provides a non-linear mapping of input to output pixel intensities called the gamma correction function:

$$\hat{I} = I^{1/\gamma} \quad (6.5)$$

where $\hat{I}$ is the gamma adjusted image, I is the input image and γ is the gamma amount to adjust. Gamma values between 0–1 one enhance contrast in bright regions, while values greater than 1 enhance dark regions. The following tests utilise a gamma value of 1.7 to enhance contrast in the dark shadow regions. This value was determined experimentally and produced the largest improvements out of the values tested.

Figure 6.8 shows the difference in detector performance between the gamma adjusted images and the unadjusted set. It can be seen in Figure 6.8(a) that the

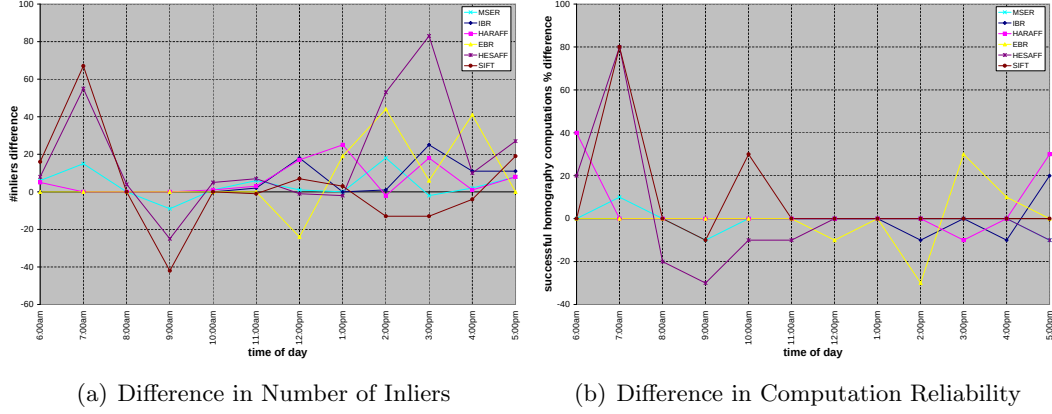
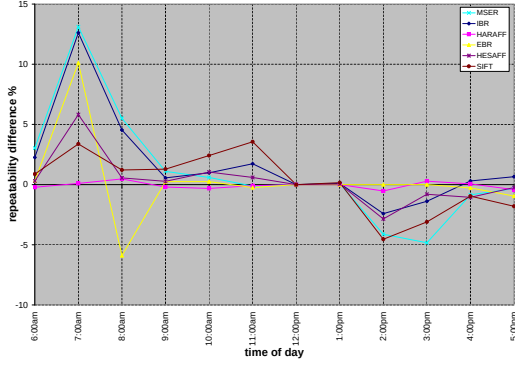


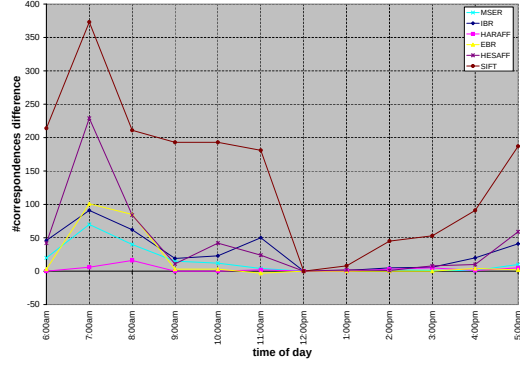
Figure 6.7: Difference in homography computation performance on contrast adjusted shadow region

repeatability of virtually all algorithms is improved up until 12pm. 7am again benefits the most from this enhancement, with MSER and IBR gaining the most. The number of correspondences in Figure 6.8(b) shows MSER gaining approximately 70 new correspondences while only gaining 90 new features in total, meaning a high percentage of the new correspondences were already in the reference set. SIFT’s total feature count, on the other hand, increases by approximately 650 while only gaining 370 new correspondences. This lower percentage results in only a minor increase in repeatability performance. Like the contrast adjustment results, from 1pm onwards the number of new correspondences is minimal, and results in worse repeatability over this time.

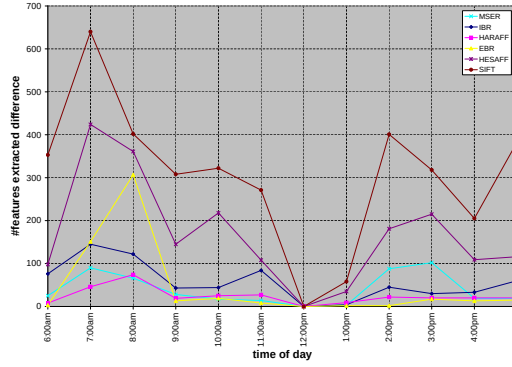
Figure 6.9 shows the differences in matching performance. The recall results of Figure 6.9(a) show little benefit for any of the algorithms. HESAFF obtains the highest increase at 7am with a gain of 6%. The similar performance is the result of a proportional increase between the number of correct matches and the number of correspondences. Figure 6.9(b) shows the actual number of new correct matches. HESAFF increases from 14 to 59 at 7am—a difference of 45. Other algorithms such as MSER do not receive such large gains, and MSER itself only obtains an additional seven matches. Given that this algorithm only obtained four matches in the unadjusted case however, this 175% increase doesn’t seem so bad. The precision of the HESAFF algorithm also benefits greatly from the gamma adjustment. At 7 and 8am the number of false matches drops by approximately 20, and at 7am the 1-precision score drops by 27%. The other algorithms at this time also generate better 1-precision scores. Even MSER—whose false matches actually increase by five—gain improved precision, indicating that the precision enhancement is a result



(a) Repeatability Difference



(b) Difference in Number of Correspondences



(c) Number of Features Detected

Figure 6.8: Differences in detection performance for gamma adjusted shadow region of clear weather WILD selection

of a higher percentage of correct matches.

The increase in the number of correct matches found by MSER at 7am enables successful homography computation, albeit at only 10% reliability, whereas the original data failed. Despite the large increase in the number of correct matches at 7am and a substantial drop in the number of false matches, HESAFF only achieves 10% reliability. SIFT gains the most at this time achieving 50% reliability where the original data failed. The enhancement is generally detrimental to the reliability of the homography computation for images after 10am and is likely the result of a drop in the number of correct matches for most algorithms over this time. At 5pm, HARAFF and IBR achieve higher levels of reliability with HARAFF even obtaining 100% reliability.

Homomorphic Filtering

This operation as described by Kovesi [Kov] “sharpens features and flattens lighting variations in an image. It usually is very effective on images which have large variations in lighting”. Figure 6.11 shows the 7am image before and after homomorphic

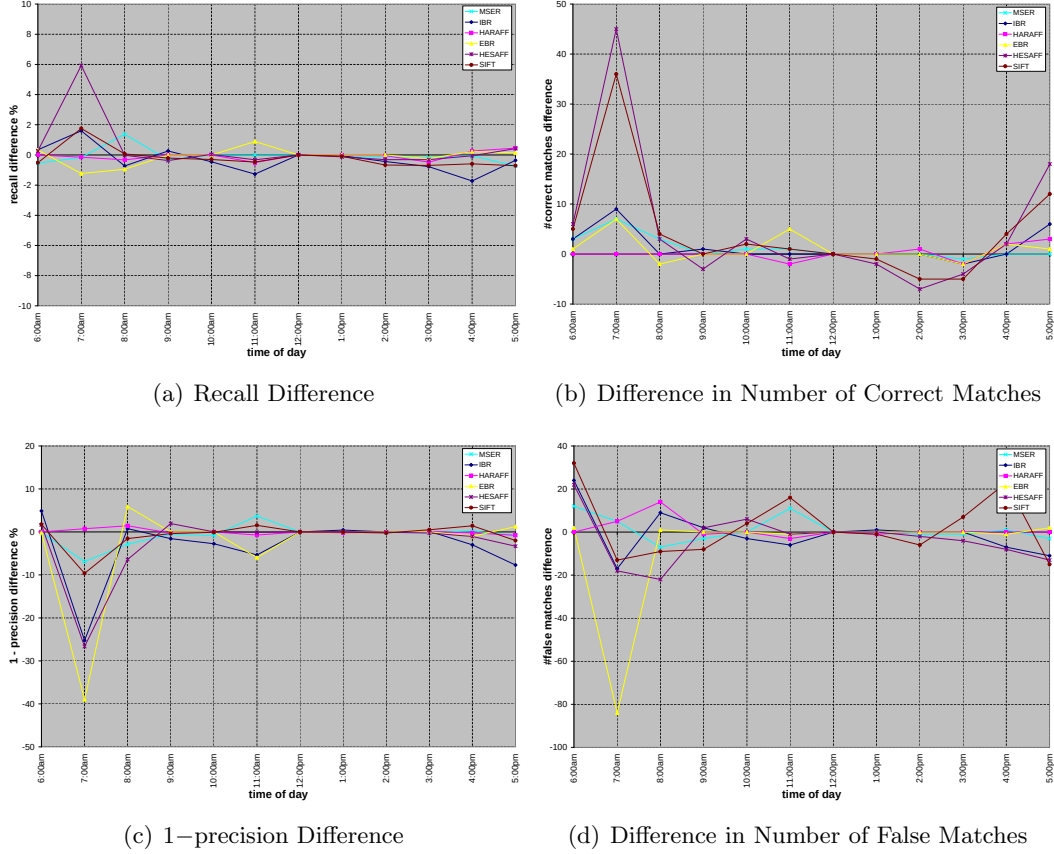


Figure 6.9: Difference in matching performance for gamma adjusted shadow region of clear weather WILD selection

filtering. Notice how the shadows are less prominent and contrast within shadow regions has been enhanced.

This algorithm works by separating an image f into its two constitutional components: the illumination and reflectance. The illumination is the amount of light cast onto the scene by the light source, whereas the reflectance is the amount of light reflected off surfaces in the scene. Image f can thus be represented by the combination of these two components $f(x, y) = i(x, y)r(x, y)$ where i is the illumination and r is the reflectance. In order to improve the image and flatten lighting variations caused by shadows, the illumination component must be suppressed while preserving the light reflected from the objects in the scene; maintaining this reflectance is also the objective in colour constancy research. The illumination frequency components could be suppressed and the reflectance components preserved by applying a frequency domain filter—commonly termed a transfer function. To do this however, i and r must be separated; this is not easy given they are combined multiplicatively.

By taking the natural logarithm of the image such that $f_{\ln} = \ln f(x, y)$, the

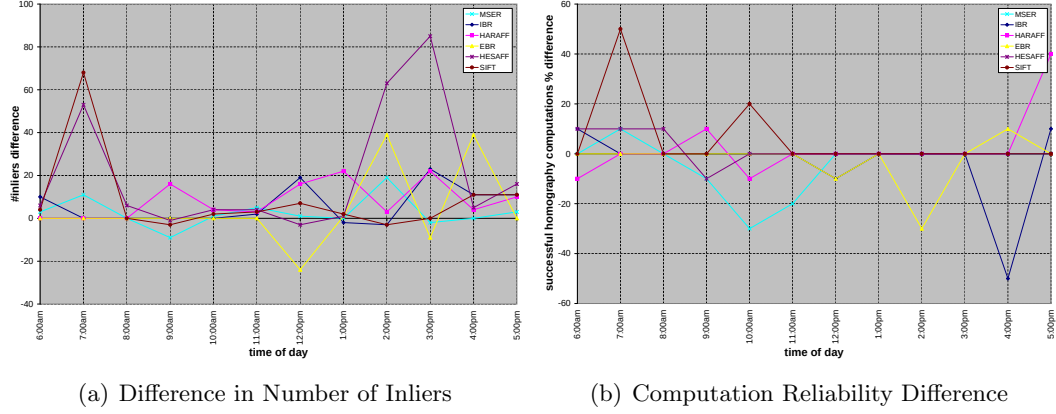


Figure 6.10: Difference in homography computation performance with the gamma adjusted shadow region



Figure 6.11: The image on the left shows the original unadjusted 7am image, while the image on the right shows the image after Homomorphic Filtering

components can be separated, producing $f_{\ln} = \ln i(x, y) + \ln r(x, y)$. The Fourier transform of this then becomes $Z(\xi, \eta) = I(\xi, \eta) + R(\xi, \eta)$, where $I(\xi, \eta)$ is the Fourier transform of $\ln i(x, y)$ and $R(\xi, \eta)$ is the transform of $\ln r(x, y)$.

The transfer function T can now be applied to both I and R such that $G(\xi, \eta) = T(\xi, \eta)I(\xi, \eta) + T(\xi, \eta)R(\xi, \eta)$. An inverse Fourier transform brings G back into the spatial domain producing image g . Recalling that the original image had a natural logarithm applied to it, this process must also be corrected by applying the inverse of the $\ln$, which is $\exp$, yielding the final enhanced image $\hat{f}$. To summarise the process follows:

$$f(x, y) \rightarrow \ln \rightarrow \text{FFT} \rightarrow \text{applying } T \rightarrow \text{FFT}^{-1} \rightarrow \exp \rightarrow \hat{f}(x, y).$$

The detection results of Figure 6.12 show that many algorithms incur small benefits from the homomorphic enhancement during the period from 6am–12pm. Repeatability scores are higher due to a consistent increase in the number of correspondences despite less significant increases in the total number of features compared with the contrast and gamma adjusted images. For MSER at 7am an

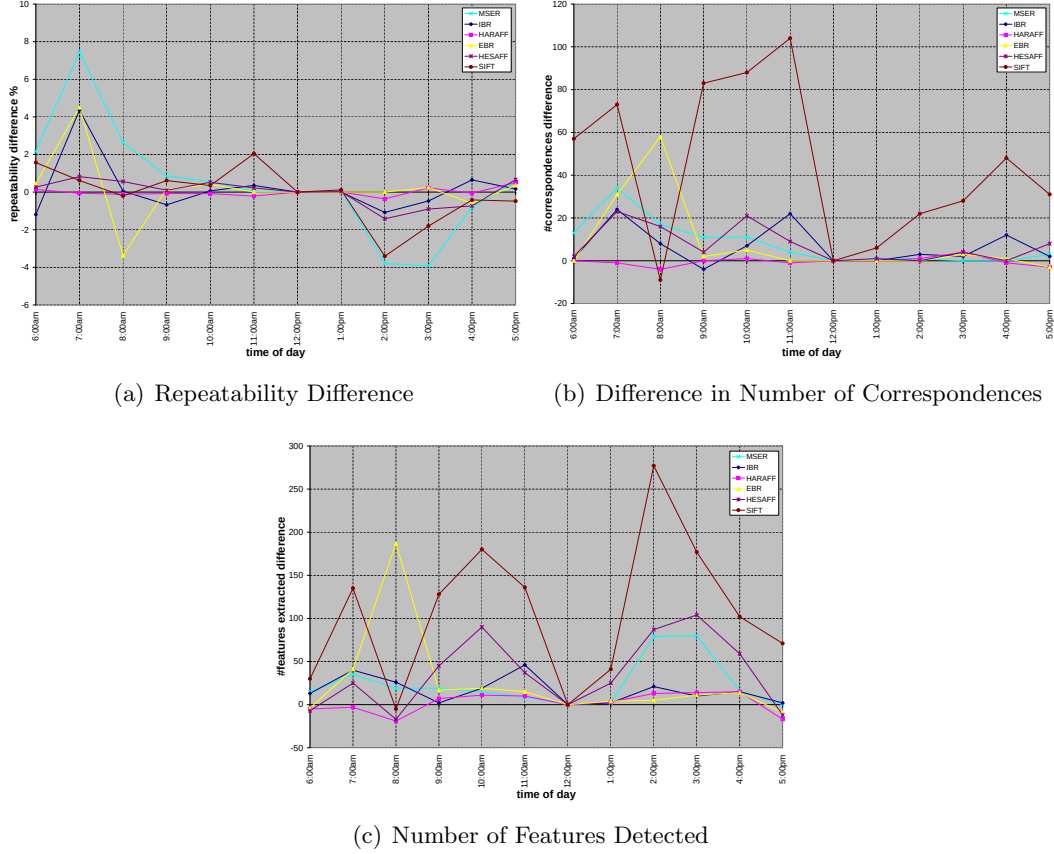


Figure 6.12: Differences in detection performance for homomorphic filtering on shadow region of clear weather WILD selection

additional 34 correspondences are found, increasing from 39 to 73. Repeatability performance from 1pm onwards tends to be worse than the original performance, due to a disproportionate increase in the total number of features detected as compared with the number of new correspondences.

The matching results of Figure 6.13 show mixed results compared with the original data set. Only minor differences are present in the recall rates, resulting from relatively few additional correct matches. MSER at 7am only obtains an additional 5 correct matches, the lowest result of the 3 image enhancement operations. The precision results however, paint a different picture. The number of false matches at this time for MSER drops by 42, the largest of any approach. The improvement in 1-precision score of 16% is also high and is bested only by the contrast adjustment which improves by 20%. This is because the number of correct matches in the homomorphic case does not increase as much as in the contrast adjusted images. From 12pm onwards, the difference in recall performance is negligible, and generally less than 1%.

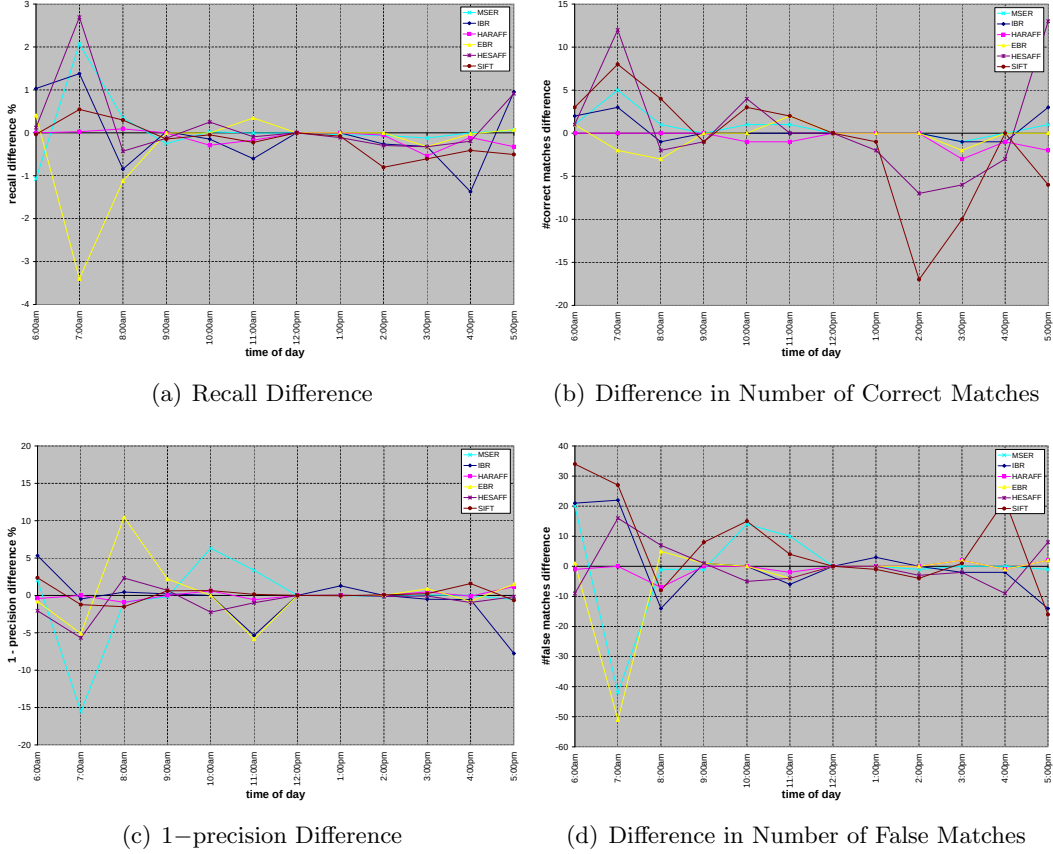


Figure 6.13: Difference in matching performance for homomorphic filtering on shadow region of clear weather WILD selection

Homography computation performance is also mixed. MSER obtains significantly worse performance at 6am, where the computation now fails despite it obtaining 60% reliability in the original data. This is a result of the large increase in false matches at this time. Conversely, the large decrease in the number of false matches at 7am produces successful computations and a reliability of 20% where the original data failed. The only other noteworthy result is the increase in reliability at 10am for SIFT that increases from 60–100%. The decrease in reliability at 9, 10 and 11am for MSER is likely the result in an increased number of false matches at this time.

Enhancement Summary

Overall, HARAFF produces the smallest performance difference for the gamma and homomorphic adjusted images meaning its detection process is relatively immune to these changes and derives no benefit. All the algorithms only produced significant variations in the presence of contrast adjustments. Adjusting images that already

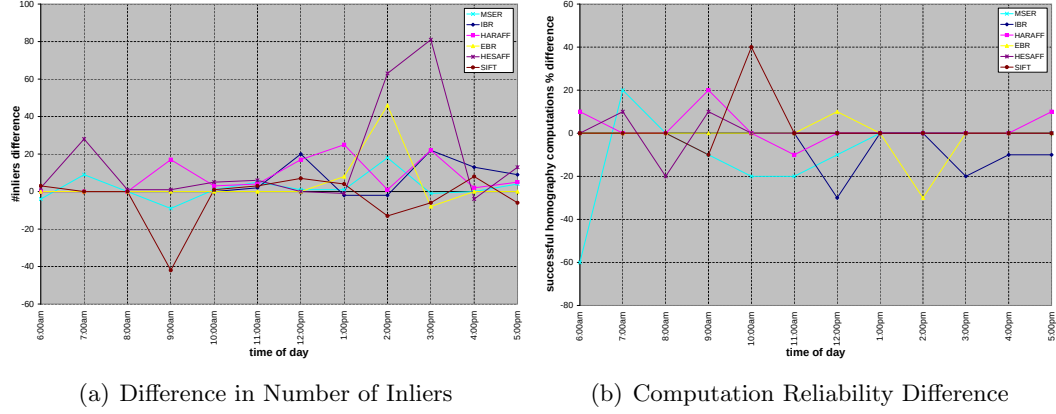


Figure 6.14: Difference in homography computation performance with the homomorphic adjusted shadow region

produce good results often results in degraded performance. However, the level of performance degradation is generally small; the existing results are robust enough to cope with minor degradations and most importantly do not suffer in terms of reliability in homography computation. The performance of the early unadjusted images is so bad that even small improvements brought about by these enhancements can yield tangible results.

Performing the homography computation on the enhanced images often yields improvements in both the number of inliers when homography computation is successful and the success rate of the algorithms. For example SIFT and HESAFF both benefit greatly from the contrast adjustment at 7am and manage to successfully compute the homography 80% of the time, compared with failing every time on the unadjusted images. Reliability at 8 and 9am however is still low; only HESAFF manages to achieve 40% reliability when operating on the gamma adjusted set at 8am and 20% at 9am. SIFT and HARAFF manage to achieve 10% reliability at 9am on the gamma images. Interestingly HESAFF also manages to achieve a 40% success rate at 9am on the homomorphic images. The general trend across all algorithms and each of the enhancement operations is that performance at 7am can be enhanced while performance at 8 and 9am is generally worse. MSER in particular does not produce many tangible benefits from any enhancement operation.

6.2 The Pillage Algorithm

This section introduces a new algorithm called Pyramid of Illumination Images (PILLAGE). The PILLAGE algorithm seeks to reduce the impact of shadow effects in

the reference data. As it has been shown, shadow effects in the reference data can cause problems if the location of shadows in the live images do not correspond. This reduces the area of overlap where feature correspondences can occur, and greatly restricts matching performance. Section 6.1 of this chapter looked at two approaches to increasing the number of features extracted in the shadow areas of the live images; this section now looks at doing the same in the reference images.

The approach discussed in Chapter 3 for constructing multi-view matches took spatially adjacent images and computed two-view and three-view matches between the images by utilising the epipolar and trifocal geometry respectively. The PILLAGE algorithm extends this concept to form multi-view matches over sequences of temporally adjacent images. By combining feature sets from a range of images taken under different illumination conditions, a complete feature set covering all regions of the image can be formed. If the images cover all possible shadow configurations, with a sufficient level of overlap between adjacent times, a reference feature set that provides minimal shadow influence can be formed. The set achieves minimal shadow impact but not necessarily a feature set completely devoid of shadow regions. This is because some scenes may always have some regions in shadow regardless of the time of day.

The PILLAGE algorithm makes the following assumptions:

- Feature overlap between spatially adjacent images is sufficient to compute the geometric relationship between the images
- Temporally adjacent feature sets are built from images taken in the same season

The first assumption dictates that the images must contain sufficient overlap to establish a reliable set of matches capable of being used to compute the geometric relationship between the images. The second assumption specifies that temporally adjacent reference images should be from the same season, otherwise there is no guarantee the shadow regions will overlap.

Figure 6.15 shows a mockup of the intention of PILLAGE. Here, feature sets from two aligned images taken under differing illumination are combined. The top image has shadows on the front faces of the buildings, thus does not extract features there. It does, however, detect features on the sides of the buildings. In contrast, the middle image detects features on the fronts of the buildings but not on the sides, therefore these two feature sets are complementary, and their combination produces

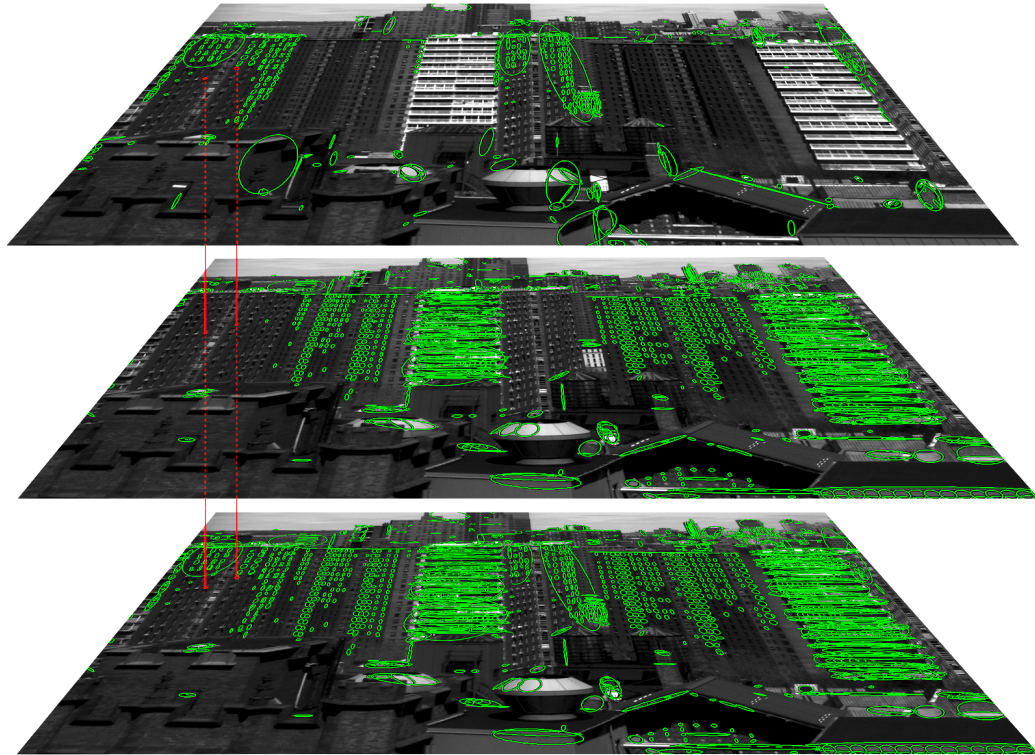


Figure 6.15: Combining two feature sets from images with different shadow regions. The red lines show two features detected in the top image but not the middle image. The bottom image shows the combined feature set

a feature set covering most surfaces of the scene, as shown in the bottom image. The red lines here show the transference of two points in the first image to the final combined set.

The advantage of the PILLAGE algorithm is that it does not require knowledge of the locations of the shadows in the images. The approach is also completely automatic, much like the process for building standard multi-view matches.

6.2.1 How Pillage Works

The simplified example of PILLAGE shown in Figure 6.15 demonstrates the combination of two feature sets from precisely aligned images. In general, image alignment is either completely unknown or only roughly known in the case of a hybrid system, therefore, establishing the geometric relationship between the feature sets being merged is part of the problem. The geometric relationships must be established in order to determine which features in the sets represent the same real world points. This process is the same as used for computing multi-view matches, however in the case of a temporally adjacent sequence of images, it is assumed that not all features exist throughout the entire sequence. If they did there would be no need to use the

algorithm.

Algorithm 6.1 provides an overview of the main stages in PILLAGE. The term ‘track’ is used in the algorithm description to describe multiple views of the same real world feature. The track list structure called TL, contains all the tracks currently being built. The width of this structure is based on the number of temporal sets being used. The temporal image set S is the input to the algorithm and covers the spectrum of illumination changes for a given scene.

The number of image sets required, n , should be sufficient to cover the range of possible shadow configurations for the scene while maintaining a level of similarity between any two adjacent sets in terms of their shadow configuration. This ensures the success of geometric relationship calculations. The temporal order of these sets should also be known.

The number of images in a given set S_j should be enough to cover the range of possible viewing locations in the scene. For example if using this technique to build a 3D model, as described in Chapter 3.2, the same requirements apply, *i.e.* images must be captured from a range of positions to build up a 3D model which encompasses all potential viewing locations. The PILLAGE algorithm differs from the algorithm described in Chapter 3.2, however, as it adds a further dimension to the system: that of time and differing illumination characteristics. For simplicity, the following experiment uses only one image in each temporally adjacent set. This enables a PILLAGE frame to be constructed using images from WILD that only provide one viewing location. Performance can then be compared to previous tests which only used one reference frame. The selection of temporal reference frames from WILD covers images taken at successive times on 4th April 2002. These images were taken during clear weather and exhibit the same shadow effects seen in the clear weather WILD selection. As these images are precisely aligned, their geometric relationship is described by a homography. In any system where the images are not taken from the same viewing position, a trifocal tensor should be used instead. The important ability this algorithm requires is being able to perform point transfer, where a point-to-point operation is performed to precisely map the location of a feature in one image to a point in the second. The use of this ability will be discussed further shortly.

The algorithm begins by extracting features from all the images. For each temporally adjacent image pair S_j and S_{j+1} putative matches are established using

Objective Combine n temporally adjacent feature sets from images S . The image set should cover the scene both geometrically and photometrically, and capture the range of illumination conditions present in the scene.

Algorithm

- (i) **Feature Extraction:** Load or extract feature points for each image used
- (ii) **Multi-view Matches:** For each temporally adjacent feature set:
 - (a) Compute putative matches
 - (b) Compute the geometric relationship between the two sets and the inliers
 - (c) Extend existing tracks with any inliers that share a common first element index
 - (d) For any matches without a common index, utilise the geometric relationships computed previously to transfer the current point to each of the previous images. If a matching feature exists in any image, the current match may represent the same feature, thus could be added to this track
 - (e) If there is ambiguity in which track is the best choice to extend—*i.e.* there are multiple tracks this feature could belong to, some of which may be the same track—choose the best one as follows:
 - If a number of tracks have the same number of hits, select the track with corresponding features in TL closest to the current image
 - If there are a different number of hits, ensure that the track with the most hits differs from the second highest by a sufficient amount to be confident this is the best track to choose.
 - If any ambiguity exists in any of the previous methods, *i.e.* two tracks scored equally, select the track with the closest descriptors by computing the average Euclidean distance between the current match and all features on this track
 - (f) Any matches that have not been added to an existing track are considered new features and are added to TL accordingly
- (iii) **Descriptor Choice:** Determine the best descriptor to represent each track. Chosen as the descriptor producing the smallest distance to all other descriptors in this track
- (iv) **Remove Duplicates:** Minimise in the final feature set by ensuring that any feature from a given image is only used once

Algorithm 6.1: PILLAGE Algorithm

the distance ratio matching technique. A homography is then computed from these putative matches using RANSAC, yielding a model H_j and the inliers consistent with it. All features from the first pair are directly inserted into the TL structure as, in effect, they are all new features. All subsequent image pair matches however, are compared with existing feature tracks in TL to establish whether any of the matches already exist in TL and therefore extend previous match tracks. Establishing whether any given feature x in S_j already exists in TL is an important process in this algorithm. Failing to recognise that a feature already exists means it will be added as a new feature, introducing duplicates into our final set. Duplicates should be avoided for several reasons: first they inflate the reference set which increases computational complexity; second, and more serious, is that duplicate feature tracks have almost identical descriptors because they represent the same physical region of the scene. When using the distance ratio matching technique the likelihood of determining the closest and second closest matches as these duplicate descriptors is high. As the descriptors are so similar, the distance between them will be insufficient to pass the distance ratio check and the match will be rejected due to insufficient confidence that the closest match is in fact the best one.

Two different approaches are used to determine which features from any new pair already exist in TL, and hence the tracks these new matches should extend. The first step is the simplest and follows that outlined for the multi-view match approach in Chapter 3, Section 3.2. In short, the common image between two adjacent match pairs is used to establish which matches refer to the same real world point. For example, having stored the matches for images 1–2 in TL, the matches between images 2–3 are then compared with TL based on the common feature set of image 2. Any matches that share the same feature index in 2 represent the same point, and can be extended to cover images 1–2–3.

Problems arise when a feature track is lost through the sequence of images due to changes in shadow location or other photometric causes. In essence it means feature tracks can be broken, and subsequent reoccurrences of this real world feature will result in a new feature track being constructed. This produces duplicate tracks. Recall the example from above, where a given feature x has been detected across images 1–2–3. Feature matching between images 3–4 has been performed, however feature x was not found in image 4. The next iteration processes images 4–5 and again finds feature x , however because the existing feature track for x only goes

to image 3, no common feature set exists between the current match pair and the correct track in TL, therefore this feature would be added again, creating a duplicate. This is an undesirable outcome and provides the motivation for the second more complicated phase for establishing these multi-view matches.

As geometric relationships have been established between every image pair up until the current computed pair, the relationships can be used to establish the location of the current feature x in every preceding image. This necessitates a geometric entity capable of providing a point-to-point operation between images, thus epipolar geometry would be insufficient. Homographies provide such a relationship and the location of x in any image can be computed by multiplying together all homographies preceding this image then multiplying by x . For example to find the location of x in image 2 given its location $\hat{x}$ in image 4. The result is computed using $x = (H_3 * H_2) * \hat{x}$ where H_3 is the homography computed between images 3–4.

Given the location of x in image 2, the matches stored in TL for this image can be examined to determine if there is an existing feature which corresponds to $\hat{x}$. To speed the process of locating matches in previous images, an index image is constructed with the same dimensions as the original image. At any pixel where there is a match, the unique index of the feature is stored. Every image preceding the current one is examined to establish any tracks that the current feature may belong to. This results in a set of potential tracks, one of which must be selected for the inclusion of the match.

If only one potential track exists—which may receive several hits if there were matching features in multiple images along a given track—the choice is clear. If this is not the case, selection follows several options depending on the distribution of the potential tracks. If multiple tracks exist with the same number of hits, the track with the most matches in TL from previous images closest to the current image is selected. Temporally close matches are preferred over those from more disparate images, as the accumulation of transfer error is higher for increasing temporal distances. If the tracks have matches in identical images, the track yielding the lowest total Euclidean distance between all previous descriptors for images on the track is selected. When computing the combined distance, images closer to the current image are given more weight than those further away. Again this accounts for the difference in accuracy between temporally close images and those further away.

If multiple tracks with a different number of hits exist, the track with the highest

number of hits is selected if it is clear this is the best choice. If the difference between the track with the highest number of hits and the second highest is small, it indicates a potential ambiguity in track choice, and the Euclidean distance approach is again used to resolve the situation. The Euclidean distance calculation is used as a last resort, as it incurs the highest computational cost.

The remaining matches between the pair that are not extensions to existing tracks are added to the end of TL as new features. This entire process then repeats for the next temporally adjacent image pair, until all the images have been processed.

Choosing the best descriptor

Once the multi-view matches have been established over the temporal images, the descriptor representing each track must be selected. This is an important choice, as this descriptor will be used for subsequent matching operations between the newly constructed PILLAGE reference frame and the live data.

For each track, the descriptor most similar to all the other feature descriptors of the same track is classified as the best descriptor and is used in the final combined dataset. The most similar track is used as it's descriptor is the most likely to be matched to a descriptor representing the same physical region in a live image. Figure 6.16 shows the resulting feature set constructed using the PILLAGE algorithm on MSER data.

6.2.2 Performance

To demonstrate the performance of the PILLAGE algorithm, tests on four data sets are conducted. The first two are the the clear and cloudy WILD selections, utilising the reference data constructed for each feature type. The third is the Arizona data set with detection and matching results for a range of reference images as well as a PILLAGE reference frame. Lastly, detection and matching performance is evaluated for the SloDB selection.

6.2.3 Clear Weather WILD Image Performance

Figure 6.17 shows the detector performance for the clear weather WILD selection. Included in these graphs is the performance of MSER when the *rel* parameter is enabled on the live images while utilising the PILLAGE reference set. This case represents the optimal performance for the MSER algorithm based on improvements

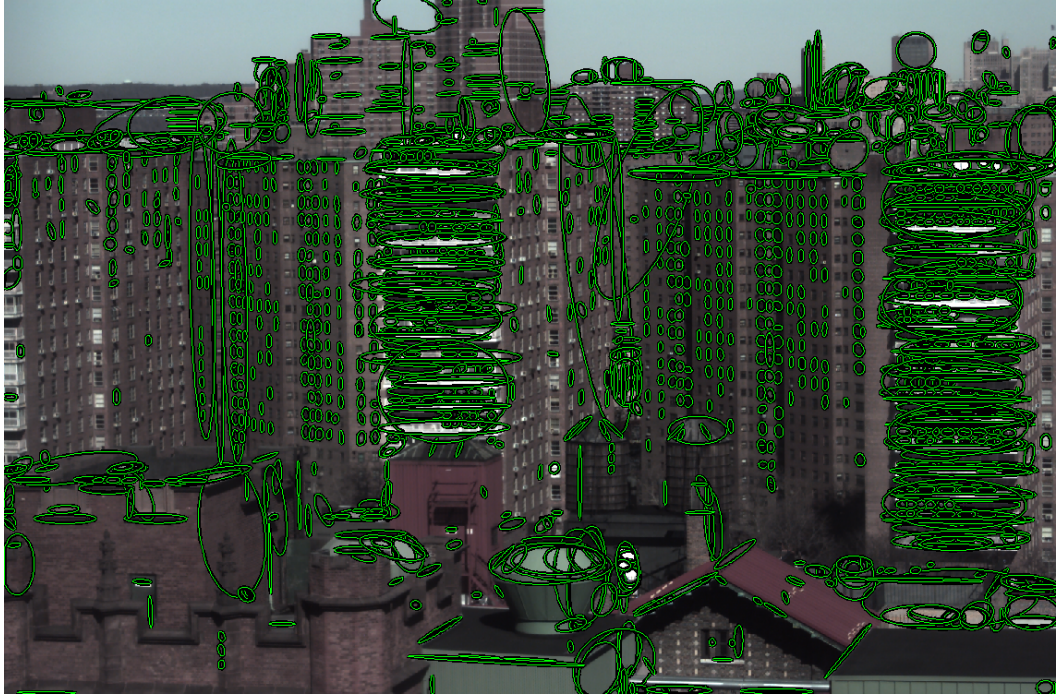
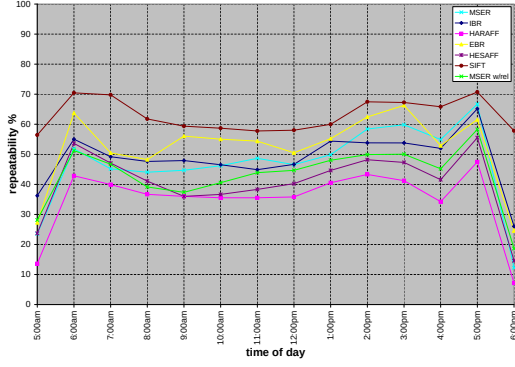


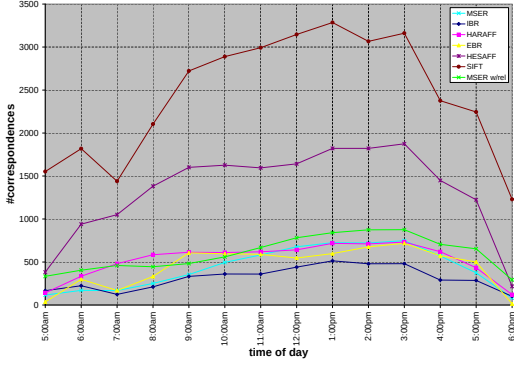
Figure 6.16: The combined reference frame created by the PILLAGE algorithm. 1751 features are in this set

to both the live and reference data. The detector performance shown in Figure 6.17 shows a marked improvement in detector performance when using the reference frame constructed using the PILLAGE algorithm. Most notably, the large drop in repeatability performance seen in the original tests of Figure 6.17(c) at 7 and 8 *am* are no longer present. Performance over all early images with the exception of 6 *am*, has been improved and, most importantly, is consistent over all images. This is a result of a large increase in the number of correspondences for the early images. The number is not as high as in the middle of the day where 2 *pm* still produces the highest number of correspondences, however the rate is now proportional to the rate of increase in the total number of features detected in Figure 6.17(c). Note that the total number of features detected here is the same as in the original test, as the live images have not been modified, only the reference data.

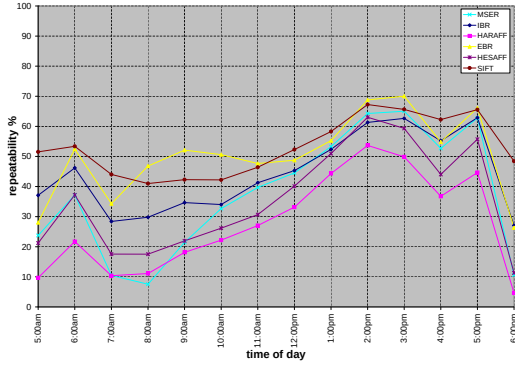
The matching performance of Figure 6.18 shows substantial improvements in both the recall and 1-precision results for the early images. The number of correct and false matches shows more consistent performance across all images when compared with the original results using one standard reference frame. Specifically, in terms of the number of correct matches shown in Figure 6.18(b), at 7 *am* MSER increases from 4 to 71. In addition, the 1-precision score drops from 95% to 61%. As was the case in the original tests, EBR provides the best 1-precision results in



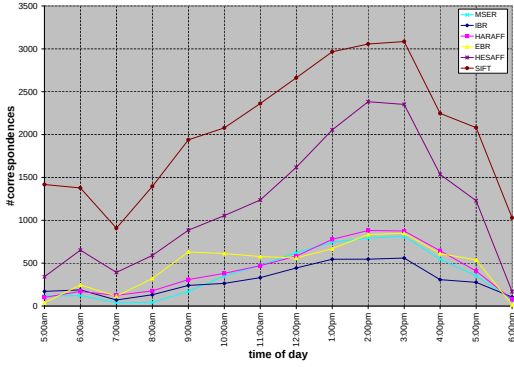
(a) Repeatability - PILLAGE



(b) Number of Correspondences - PILLAGE



(c) Repeatability

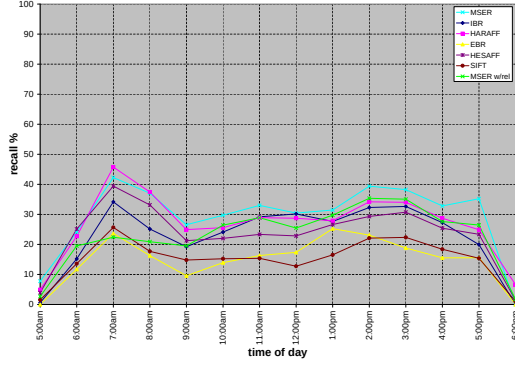


(d) Number of Correspondences

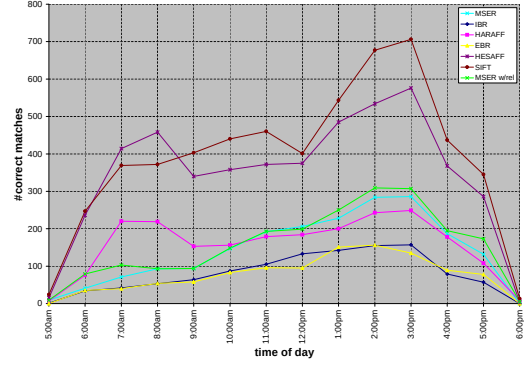
Figure 6.17: Detector performance using PILLAGE reference data and a clear weather WILD selection. Figures 6.17(c) and 6.17(d) are reproduced from Figure 5.9 to assist in comparison. The number of features detected in the live images has not changed thus this graph is not included.

many instances. This is due to the lowest number of false matches for many images. The ratio of correct to false matches also shows EBR as obtaining a significantly better ratio than the other algorithms, and at 1pm gets 25 correct matches for every false match, in this case 151 correct matches and only 6 false matches. When *rel* enabled on MSER, lower recall rates are obtained due to a disproportionate increase in the number of correspondences compared with how many new correct matches are obtained. However, as shown in Figure 6.18(b) at 6 and 7am the increase in the number of correct matches is significant, and most importantly, improves the correct to false ratio. This is demonstrated in the 1-precision results where enabling the *rel* parameter produces a respectable improvement in precision, especially in the early images.

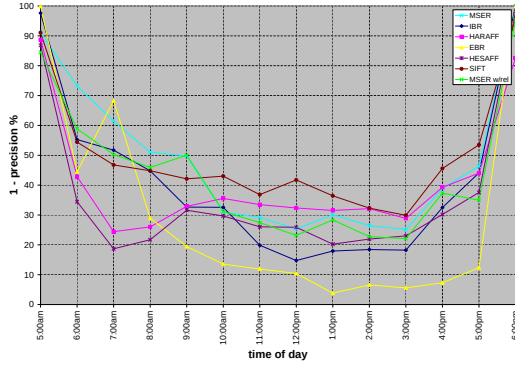
Computation of the homographies between the reference image and each of the live images reflects the increase in performance seen in the previous tests utilising PILLAGE. All algorithms except EBR succeed in computing the homographies



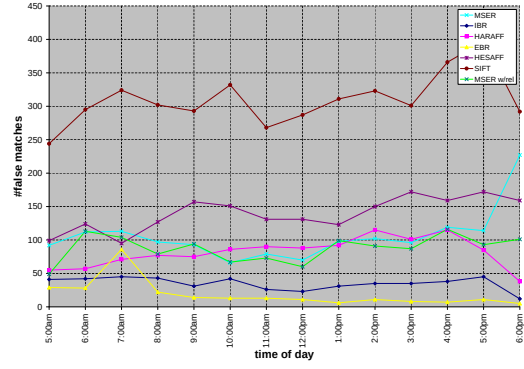
(a) Recall - PILLAGE



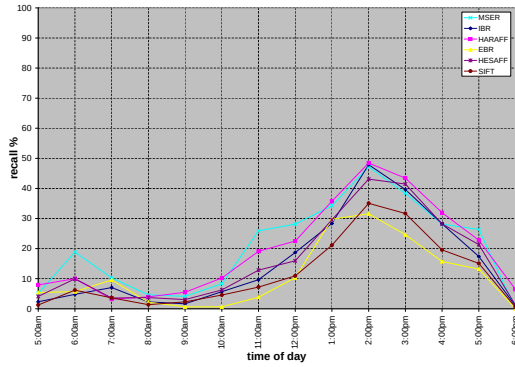
(b) Number of Correct Matches - PILLAGE



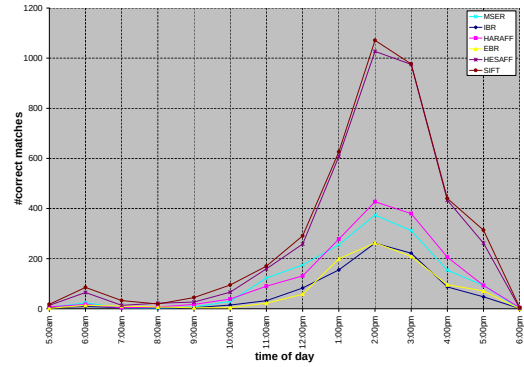
(c) 1-precision - PILLAGE



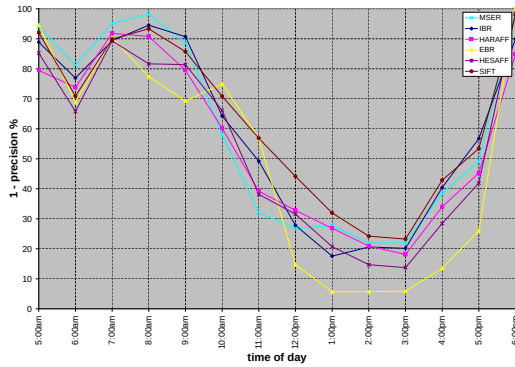
(d) Number of False Matches - PILLAGE



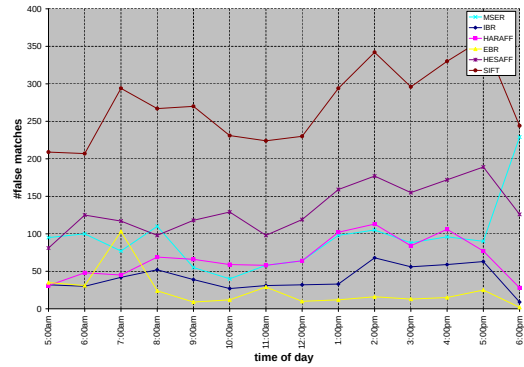
(e) Recall



(f) Number of Correct Matches

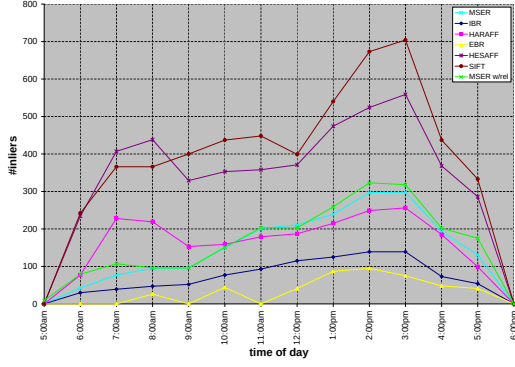


(g) 1-precision

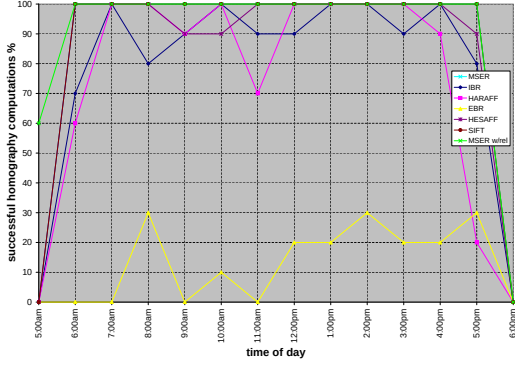


(h) Number of False Matches

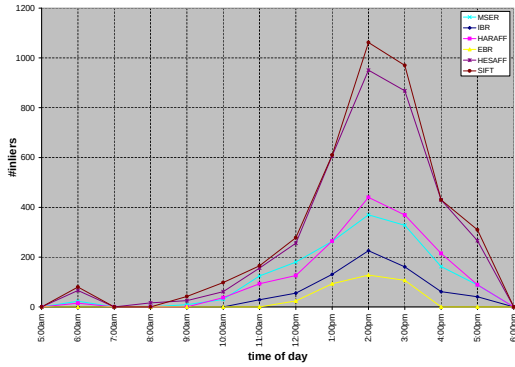
Figure 6.18: Matching results using PILLAGE reference data with a clear weather WILD selection. Figures 6.18(e) to 6.18(h) are reproduced from Figure 5.10 to assist in comparison



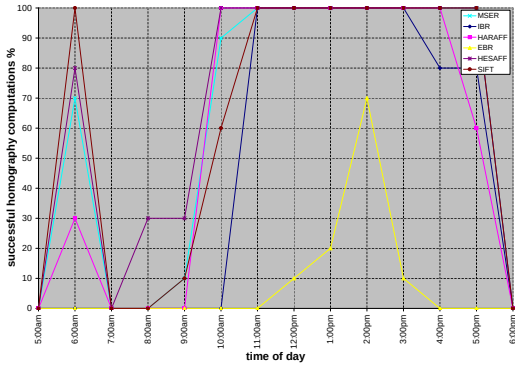
(a) Number of Inliers - PILLAGE



(b) Computation Reliability - PILLAGE



(c) Number of Inliers



(d) Computation Reliability

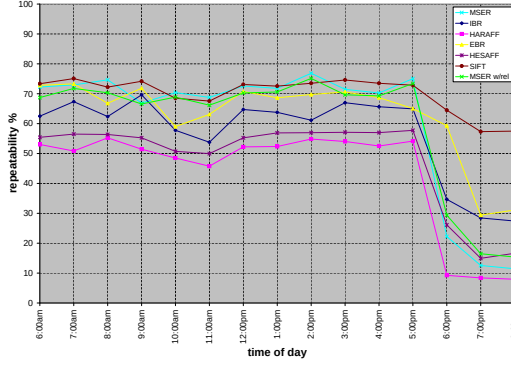
Figure 6.19: Homography computation using PILLAGE reference data and a clear weather WILD selection. Figures 6.19(c) and 6.19(d) are reproduced from Figure 5.11 to assist in comparison

between 7am and 5pm with a high degree of reliability. MSER, for instance, obtains 100% reliability for these times and only 5am and 6pm fail.

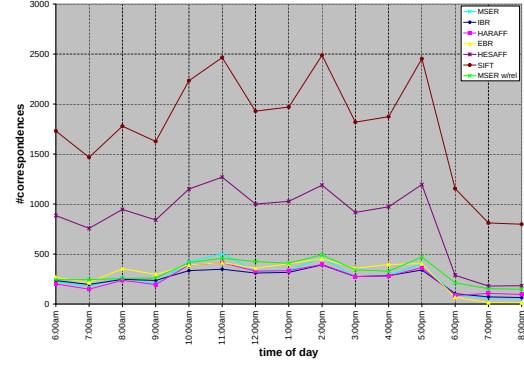
As can be seen in Figure 6.19(a) enabling the *rel* parameter produces an increased number of inliers over most of the sequence, compared with the default MSER performance. This provides the same reliability as the default parameters with the exception of the 5am image that now succeeds in computing a homography 60% of the time. This is particularly significant as no other algorithm manages to compute a homography for this image.

6.2.4 Cloudy WILD Image Performance

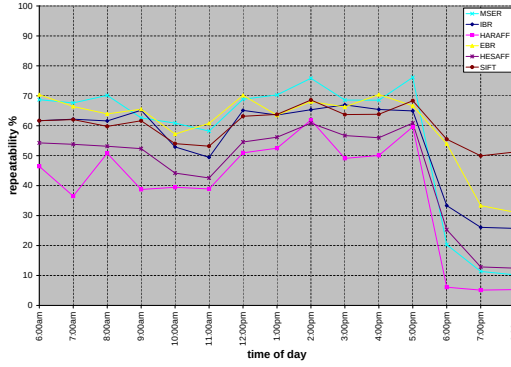
The detector performance shown in Figure 6.20 demonstrates that PILLAGE reference frames can even slightly improve the repeatability performance under cloudy conditions. The number of correspondences shown in Figure 6.20(b) are similar to those obtained in the previous test using the cloudy selection, when the standard 2pm reference frame was used.



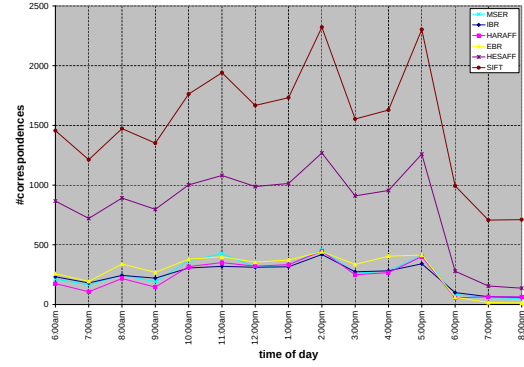
(a) Repeatability - PILLAGE



(b) Number of Correspondences - PILLAGE



(c) Repeatability

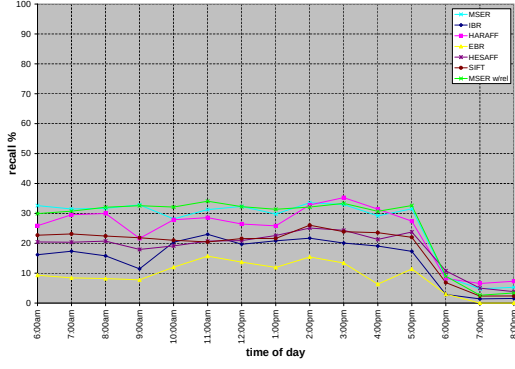


(d) Number of Correspondences

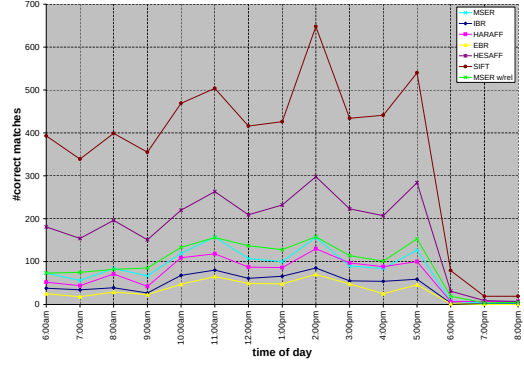
Figure 6.20: Detector performance using PILLAGE reference data and cloudy WILD selection. Figures 6.20(c) and 6.20(d) are reproduced from Figure 5.18 to assist in comparison

The use of PILLAGE also produces a slight increase in recall performance. Utilising MSER with *rel* enabled produces less variable results than with default settings. As shown in Figure 6.21(b) there is a slight increase in the number of correct matches for all algorithms, with SIFT's performance almost doubling in the early images. MSER with *rel* once again produces less variable results in this graph. This is also seen in the 1-precision scores where the results are better when *rel* is enabled and produces less variability between live images. All algorithms here obtain improved precision except between 6–8pm. Interestingly, even at 6pm MSER with *rel* achieves a 1-precision score 13% better than MSER with default parameters.

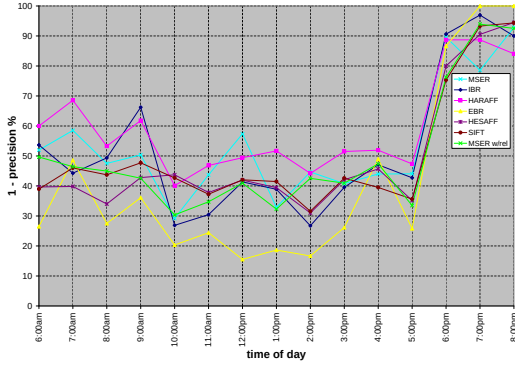
The impact on homography computation is shown in Figure 6.22 and demonstrates that all algorithms obtain an increase in the number of inliers when utilising PILLAGE. In addition, improvements in the reliability of the results show more consistent performance, particularly for algorithms less reliable when using a single reference frame, as shown between Figures 6.22(b) and 6.22(d). The improvement in precision afforded by MSER with *rel* at 6pm even allows computation with 100%



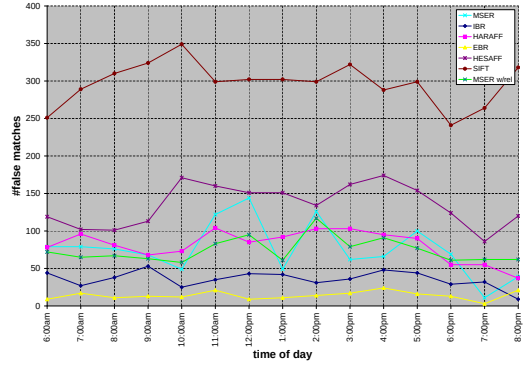
(a) Recall - PILLAGE



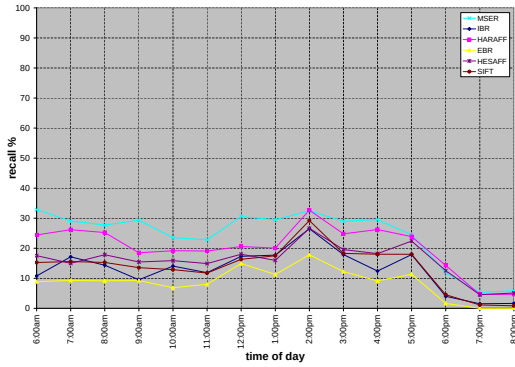
(b) Number of Correct Matches - PILLAGE



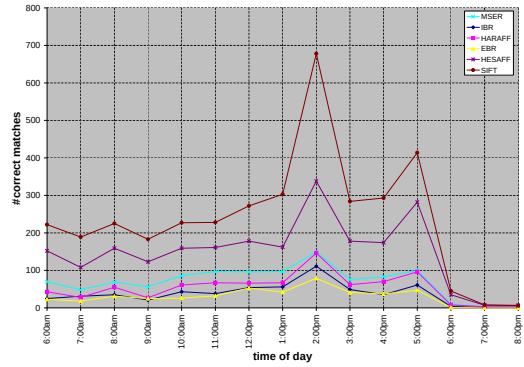
(c) 1-precision - PILLAGE



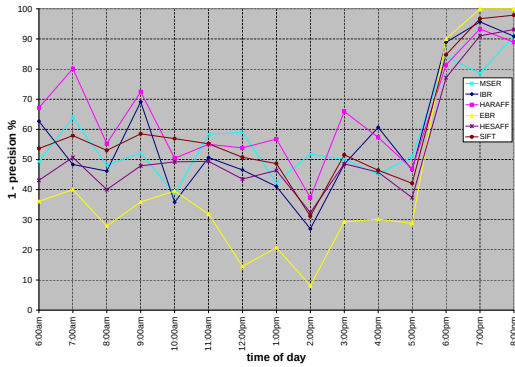
(d) Number of False Matches - PILLAGE



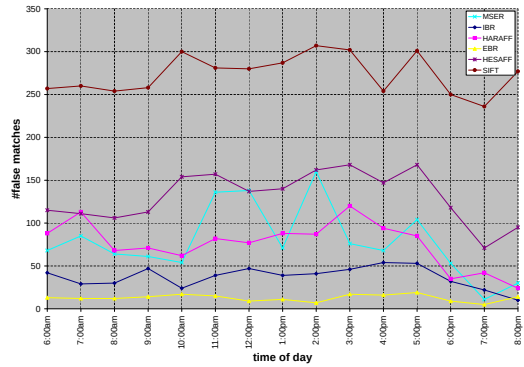
(e) Recall



(f) Number of Correct Matches



(g) 1-precision



(h) Number of False Matches

Figure 6.21: Matching results using PILLAGE reference data and cloudy WILD selection. Figures 6.21(e) to 6.21(h) are reproduced from Figure 5.19 to assist in comparison

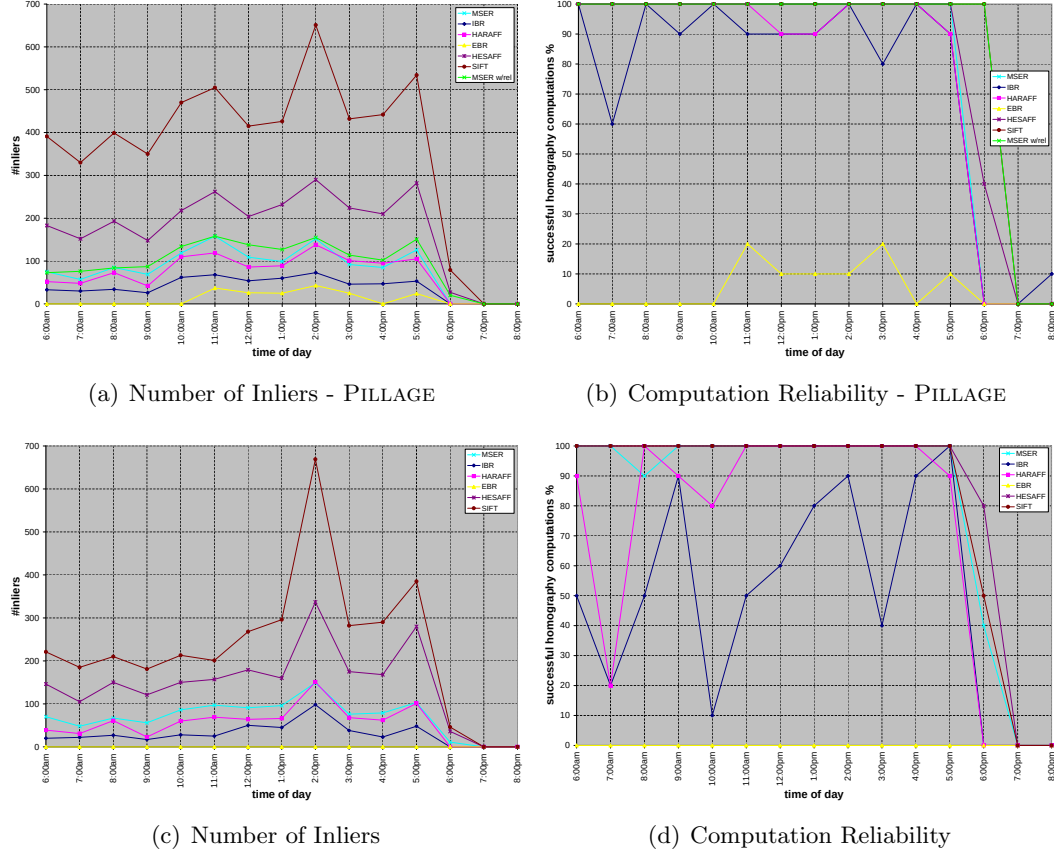


Figure 6.22: Homography computation using PILLAGE reference data and cloudy WILD selection. Figures 6.22(c) and 6.22(d) are reproduced from Figure 5.20 to assist in comparison

reliability, something no other algorithm can achieve even when utilising PILLAGE.

6.2.5 Arizona Test Set Performance

Figures 6.23 to 6.25 show the feature detection and matching performance for the Arizona test set described in Chapter 4, Section 4.4. Note that like the WILD clear weather selection, the Arizona images contain strong effects thus it is likely they will exhibit similar feature detection and matching performance to the WILD results in Chapter 5. The X axis in these graphs represents the index of the image from the test set (from 1–22) shown in Figure 4.11. Each line in the graphs represents the use of a different reference frame (the reference set is shown in Figure 4.13) that is matched to each of the 22 images from the test set, thus this test is analogous to the “Choice of Reference Frame” experiments of Chapter 5, Section 5.5. The black line shows the results when utilising a reference frame constructed from 25 reference frames (15 of which are shown in Figure 4.13) using PILLAGE. The PILLAGE frame takes 59 seconds to build on a 2.4GHz PC. Figure 6.23(c) shows the number of

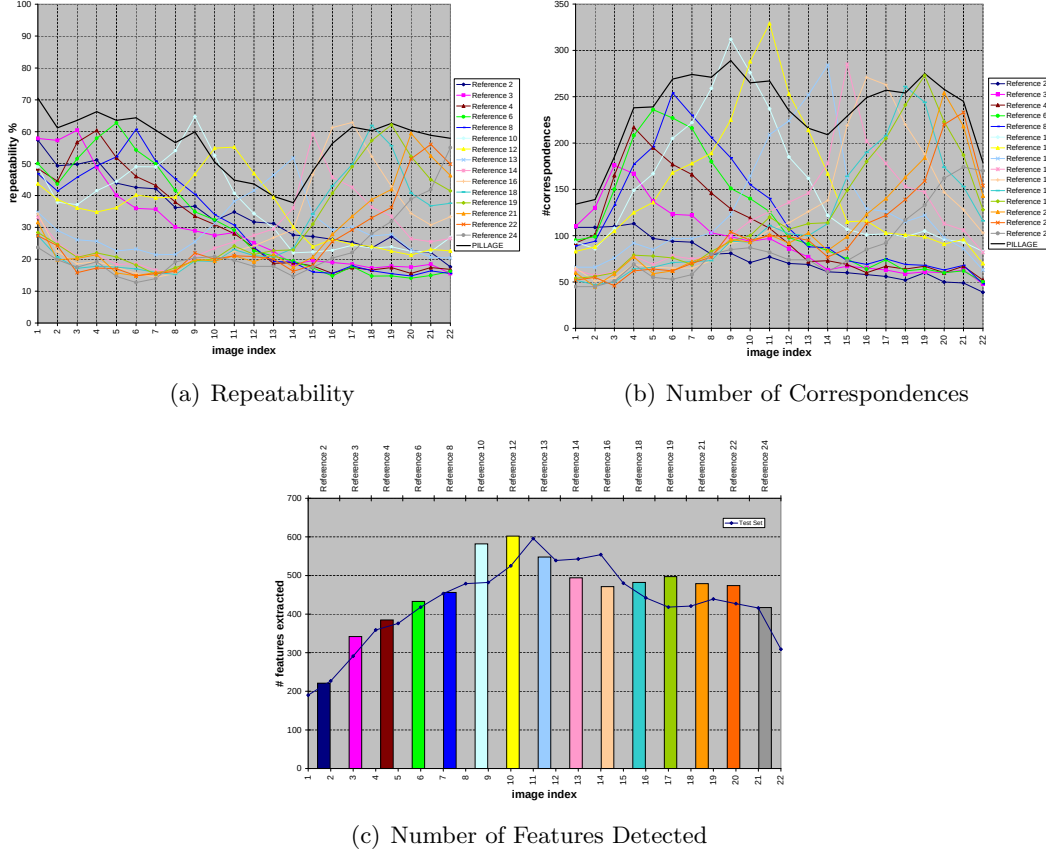


Figure 6.23: Detector performance using Arizona test set

features extracted in both the reference frames (shown as the bar chart) and the test set (represented by the line). The feature detection algorithm used for all of these tests is MSER.

Figure 6.23 shows the detector performance on the Arizona test set taking a selection of 15 individual reference frames (from the total set of 25) and the feature set constructed using PILLAGE. These results paint a very similar picture to that seen in Chapter 5 when using various WILD reference frames. As can be seen in the repeatability results (Figure 6.23(a)), early reference frames produce the best performance with early images from the test set. For example, the bright pink line representing reference frame 3 demonstrates its best performance for the second and third test images. In contrast, later reference frames exhibit their best performance when operating on test images taken at similar times. For example, the two orange lines representing reference frames 21 and 22 produce the best performance for test images 20 and 21 respectively. In fact, as in the case of WILD, successive reference frames produce successive performance peaks as the image index increases. It is interesting to note that for test images 13 and 14 reference image

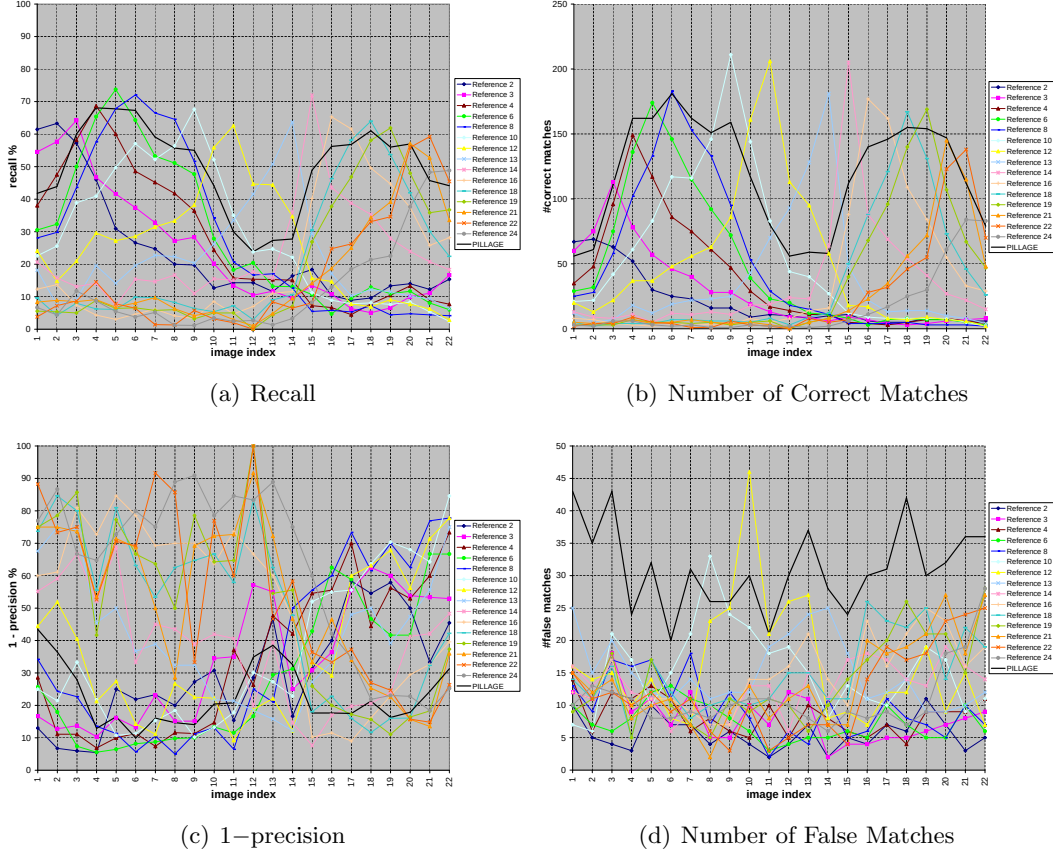


Figure 6.24: Matching results using Arizona test set

13 performs the best, and yet its peak repeatability scores are noticeably lower (by approximately 10%) than the peaks produced by other reference images at their best times. Examining Figure 6.23(b) shows a high number of correspondences for reference image 13 at these times, meaning that the dip in repeatability score is a result of the high number of features extracted for this frame, as shown in Figure 6.23(c).

In contrast to the highly variable results obtained when using individual reference frames, the repeatability scores when using a reference frame constructed by PILLAGE shows more consistent performance for both early and late images, often obtaining scores higher than any individual reference image. There is a dip in PILLAGE performance for test images 11–15 that is shown in both the repeatability and the total number of correspondences. It should be noted, however, that PILLAGE performance in this dip is still significantly higher than most of the individual reference frames with the exception of those captured at times roughly similar to these particular test images, *i.e.* reference frames 12, 13, and 14.

Figure 6.24 shows the matching performance. Similar trends to those seen in the

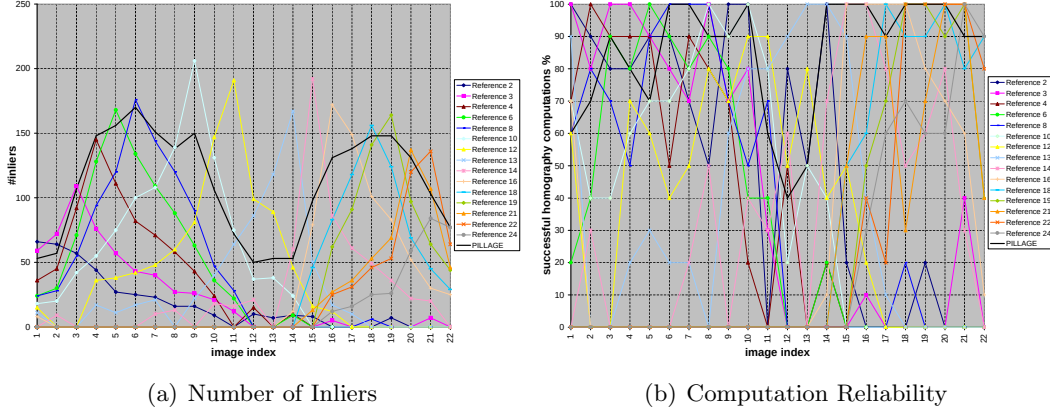


Figure 6.25: Homography computation on Arizona test set

detection results can be seen in the recall results and the number of correct matches as shown in Figures 6.24(a) and 6.24(b). Specifically, individual reference frames produce optimal matching performance for test images captured at similar times. In contrast, when using a reference frame captured at a significantly different time from the test image performance is very low. The PILLAGE reference frame once again produces the most consistent performance of all the reference frames.

The same dip in PILLAGE performance as witnessed in the detection results is seen in the recall scores and in the number of correct matches. Figure 6.24(c) also shows a significant spike in the 1-precision score for test images 11-15, indicating a loss in precision for these images. Like the detection results, PILLAGE performance is still better than most of the individual reference frames at these times but is noticeably less than the highest performing individual reference frames.

Figure 6.25 shows the results of computing a homography on the Arizona data set. As can be seen, PILLAGE performance is high for most of the test images and is significantly more robust than any single reference frame. The dip in performance is visible for images 11-14 with reliability at these times dropping down to 40-60%.

The performance dip seen in these results was not present in the WILD results from Chapter 5. In order to establish the cause of this performance dip further analysis will now be presented. The dip in detection performance when compared to that of the individual reference frames that produced peak performance during these times indicates that not all of the features from the individual reference frames are in the PILLAGE frame, otherwise there would be a similar number of correspondences produced and consequently similar repeatability. Recall from Section 6.2.1, PILLAGE reference frames are constructed by performing feature extraction and matching

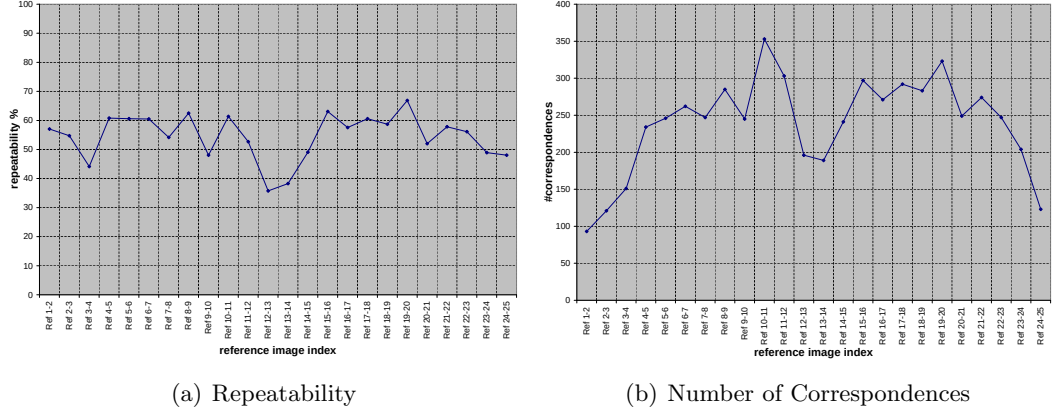
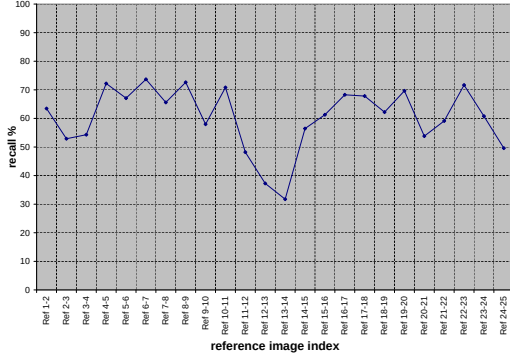


Figure 6.26: Detector performance between Arizona reference images

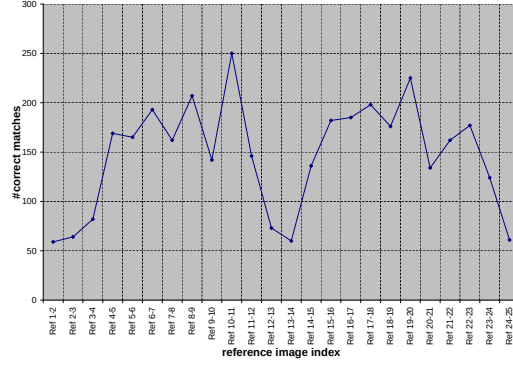
between temporally adjacent reference frames. For some features to be available in the individual reference frames but not in the PILLAGE frame, matching must have failed for these features during construction, otherwise PILLAGE detector performance would be in line with that of the individual reference frames. There are three possible causes for matching to fail on these features: either they did not exist in one of the frames being matched; the corresponding features were not the closest matches; or lastly, there were other close matches that diminished the matcher's confidence, thus causing it to reject the match. The dip in matching performance seen in Figure 6.24 indicates that in addition to some features being present in the individual reference frame but not in the PILLAGE frame, the features that are left are not the most robust ones when matching to test images 11–15. If the remaining features were equally suited to those that were not included, the recall results would be proportional to the best obtained by reference frame 13 for test image 14, for instance.

Figures 6.26 to 6.28 shows the detection and matching performance between the reference images used to construct the PILLAGE reference frame. These results represent the performance of the PILLAGE reference frame construction process. Low detection and matching performance between any of these reference frames will mean the final PILLAGE reference frame may be compromised.

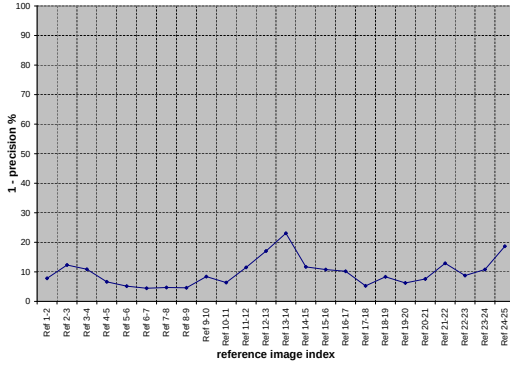
Figure 6.26 shows the detection performance when computing the repeatability scores between reference frames. As can be seen from the repeatability scores in Figure 6.26(a) there is a large dip in performance between reference images 12-13, 13-14. This indicates there is a large number of features between reference images 12 and 13 that are not in common (*i.e.* not correspondences).



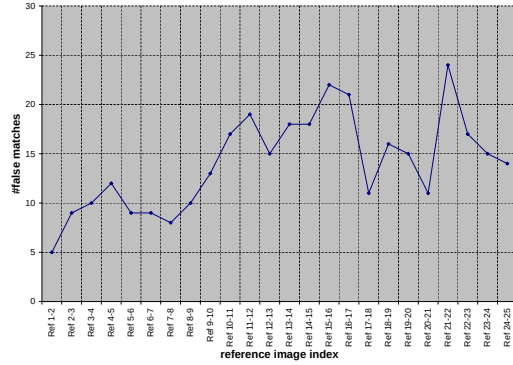
(a) Recall



(b) Number of Correct Matches



(c) 1-precision



(d) Number of False Matches

Figure 6.27: Matching performance between Arizona reference images

The matching results show a similar dip between reference images 12, 13 and 14 in terms of the recall and the number of correct matches. Furthermore, the 1-precision scores show a spike at these times indicating a loss in precision that is the result of a disproportionate decrease in the number of correct matches when compared to the number of false matches.

Lastly Figure 6.28 shows the homography results when computed between adjacent reference images. The same dip in performance between images 12, 13 and 14 is seen here in both the number of inliers and in the reliability. In fact, the reliability is 100% for most reference images until images 12, 13 and 14 at which point reliability drops to 50 and 60% respectively before returning to 100% for virtually all the remaining images.

These detection and matching results are consistent with frames that have dissimilar shadow configurations. Examining reference images 13 and 14 it is clear this is the likely problem. In reference image 13 the left faces of the buildings are completely in shadow whereas in reference image 14 the left faces are completely illuminated. This again demonstrates the point that it is very important to ensure

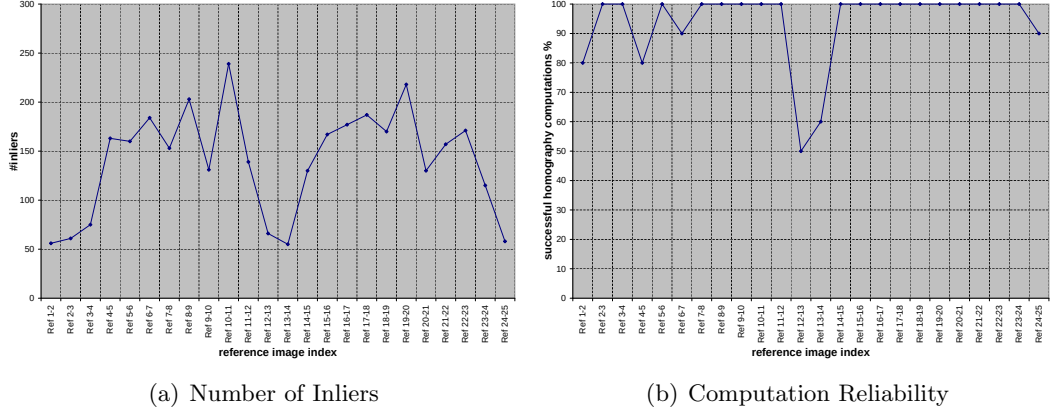


Figure 6.28: Homography computation between Arizona reference images

shadow configurations between temporally adjacent reference images are similar. This may require inconsistent spacing in terms of the frequency at which the temporally adjacent reference data is collected.

6.2.6 SloDB Test Set Performance

The following figures illustrate the performance of the SloDB data set introduced in Chapter 4, Section 4.3. The performance of this data set was evaluated for one reference frame under six different detection algorithms in Chapter 5, Section 5.5.6. Here the performance is evaluated when using individual reference frames taken from different times together with a PILLAGE reference frame that is a combination of the individual reference frames.

Figure 6.29 shows the detector performance. These graphs show similar trends to the other data sets, however, as noted in the analysis of these images in Chapter 5 the shadows in the SloDB scene are not as intense. As a consequence of this the performance trends are not as pronounced as in WILD or the Arizona test set.

Figure 6.29(a) shows that peak performance for a given image from the test set (along the X axis) is often obtained by a reference frame taken at a similar time. For instance the 9:37am reference frame obtains high performance for the early reference images but drops off from 10:21am to 3:38pm. Conversely, the 11:45am reference frame obtains lower performance for the early test images and reaches peak performance at 11:54am. It is interesting to note that early reference images produce high performance for both early and late test images as do late reference images. This indicates that the early and late test images—specifically the 7:11am and 5:06pm test images—have similar shadow configurations thus reference images

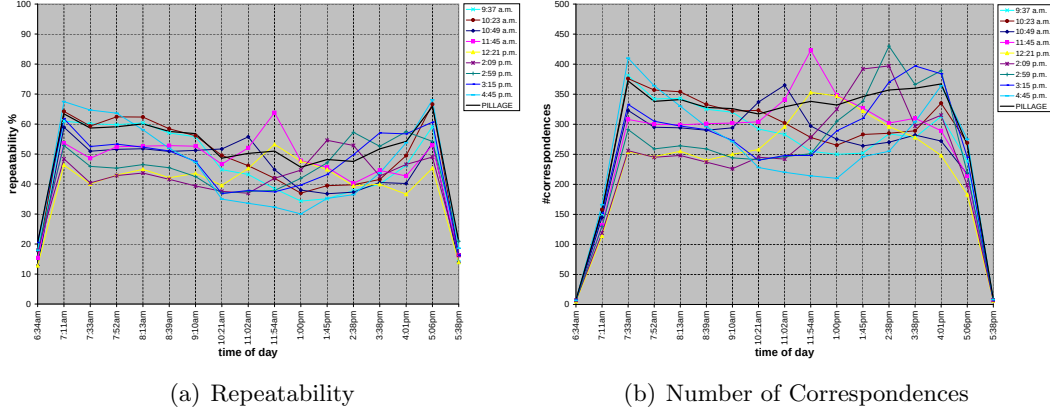


Figure 6.29: Detector performance using PILLAGE reference data and the SloDB image selection

that perform well on either image will perform well on both. The PILLAGE reference frame demonstrates consistent performance in these graphs.

Virtually identical trends to those seen in the detector performance are exhibited in the matching performance graphs of Figure 6.30. Peak individual reference frame performance is correlated to test images with similar shadow configurations that are generally captured at similar times. This produces variable performance for the individual reference frames where they achieve a performance peak for images with the most similar shadow configurations but drop off for other images. In contrast, PILLAGE once again produces consistently high performance.

Despite some variability in the detection and matching performance, all the individual reference frames and the PILLAGE reference frame produce reliable homography computation results at virtually 100% for all test images except the images taken when it was dark (at 6:34am and 5:38pm). As previously identified, the impact of the shadows on performance is not as significant as other scenes as the shadows are not as intense. As a result of this, matching performance is sufficient for all reference frames to produce good homography performance. PILLAGE produces the most consistent performance in terms of the number of inliers.

6.3 Summary

This chapter has detailed several experiments towards improving the performance between images containing differences in illumination—specifically differences in shadow configurations. Utilising the parameters provided by the algorithms, in this case MSER, provides the ability to improve the performance of the algorithm.

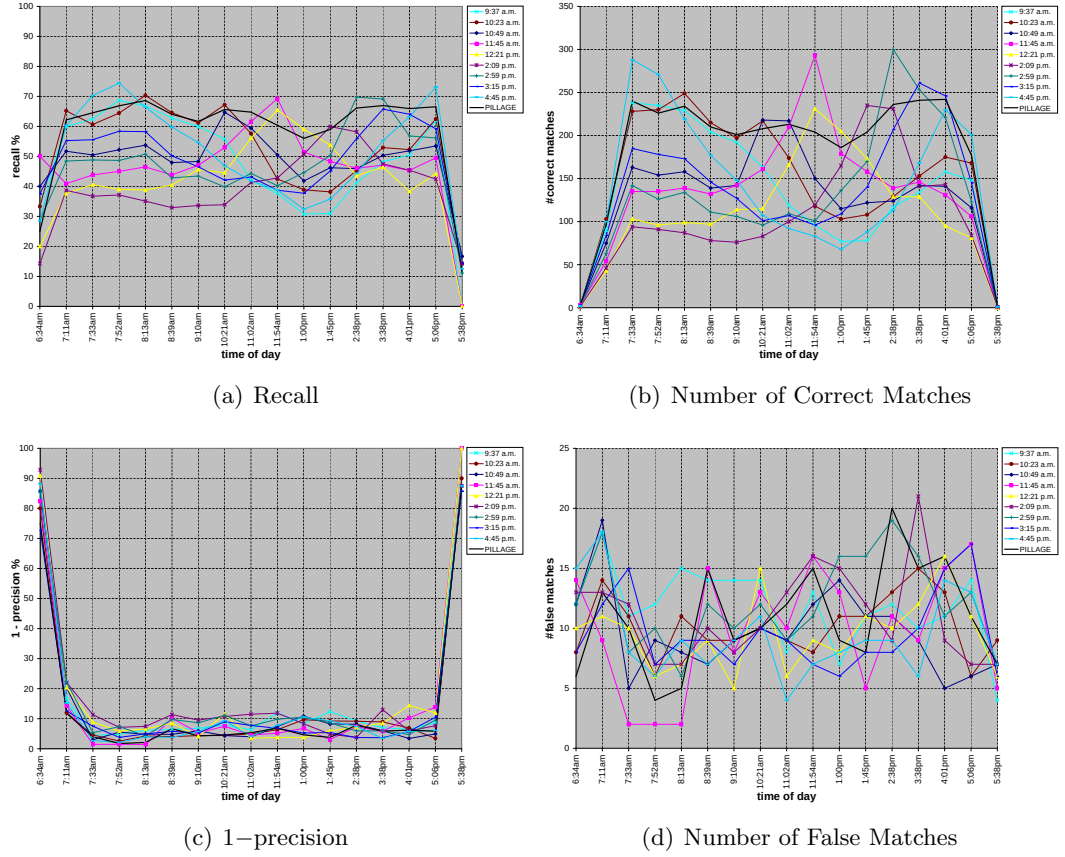


Figure 6.30: Matching results using PILLAGE reference data with the SloDB image selection

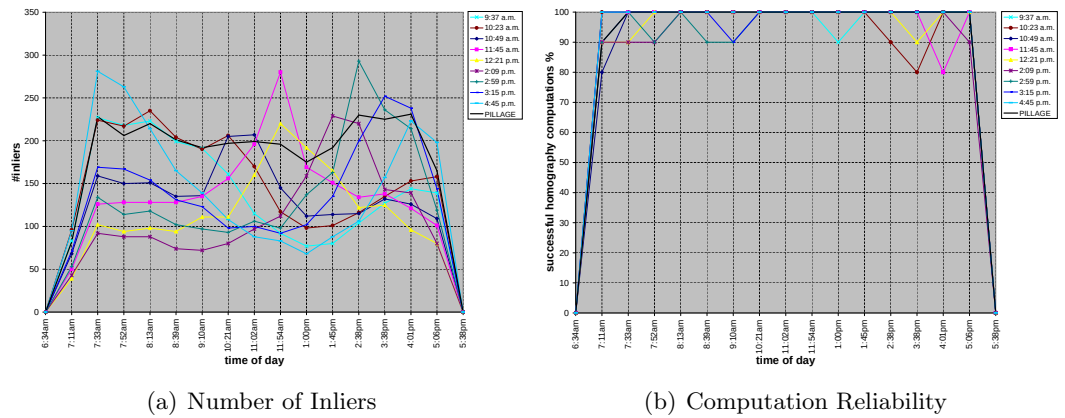


Figure 6.31: Homography computation using PILLAGE reference data and the SloDB image selection.

Naively relaxing these constraints produced more features in shadow regions when operating on the WILD images; however, the best results were obtained using an adaptive method embedded in the MSER algorithm. This instead utilised different parameter values for different parts of the scene and produced the best performance, especially in the early images where performance in the original data was worst.

Applying image enhancement operations to the regions in shadow resulted in improved detector performance in the early images. This often did not translate into significant improvements in matching performance however, and only the *7am* image consistently yielded tangible benefits.

The PILLAGE algorithm reduced the impact of shadows in the reference feature set by combining multiple temporally adjacent image sets. This approach yielded significant increases in performance for the WILD and Arizona image databases when compared to performance obtained when naively using one reference frame. More importantly it increased the consistency of the results and for WILD produced reliable results for most algorithms at all times except the dark images at *5am* and *6pm*. Furthermore, future improvements in feature extractors and matchers will not only improve performance when matching images to the PILLAGE frame, but will also improve the PILLAGE frame itself by ensuring that during construction the maximum number of robust features are maintained between each of the reference frames that comprise the resulting feature set. The disadvantage of the PILLAGE algorithm is that it requires images of the same scene taken at many times during one day. The images must contain sufficient overlap both geometrically and photometrically to be able to compute the geometric relationships between the images, and thus avoid the problem this technique seeks to solve.

When combining temporally adjacent sets of reference images that cover a range of viewpoints—for instance applying PILLAGE to multiple reference image sets like the Leuven Castle sequence used for the 3D reconstruction in Chapter 3—the problem is almost identical to that of computing multi-view matches solely over geometrically adjacent images. As has been shown in the results differences caused by shadow configurations behave similarly to viewpoint changes in that matching performance drops off as the shadow configurations become more disparate. This means the process needed is effectively the same except another dimension has been added; that of time. If ground truth data is available, or the developer has some prior knowledge about which images are approximately from the same

viewpoint but taken at different times, PILLAGE can be applied to these temporally adjacent images. Once combined these image sets—that now only differ spatially—can be combined using typical multi-view match computation approaches like that described in Chapter 3, Section 3.2.6. The important consideration when performing this process is that matching can only be performed between images that are similar enough such that they contain sufficient overlap. This means matching should be performed between spatially and temporally similar images. If applied to images that differ in both their shadow configuration and their viewpoint the differences may prevent the images being successfully matched.

None of the approaches here help in matching the dark images at *5am* and *6pm*. Such cases do not suffer from shadow effects, thus were not considered here. Dealing with such images will require further research.

Naturally the best approach to dealing with shadow effects is to avoid them. However in any computer vision system intended for long term use, such an approach is impossible as shadows will be encountered eventually. Ensuring reference data is free of significant shadow effects can provide robust performance. The PILLAGE algorithm developed during the course of this project proved this, yielding significant improvements in performance and reliability. In contrast, dealing with the shadows in the live images is more difficult. The experiments did show however, that adjusting the parameters of the algorithm can be sufficient to provide significant increases in performance when the illumination conditions are different. In particular the “use relative margins” toggle of MSER provided a clear indication of the performance advantage possible under early images. In addition, it demonstrated the advantage of employing adaptive threshold selection during feature extraction, ensuring that the correct parameter values are employed for specific regions of an image. This process is best implemented inside the feature detector, as reproducing the effect outside it would require several passes utilising different parameter values, then integrating the resulting feature sets; a highly inefficient solution.

Chapter 7

Conclusions

“Knowledge is a process of piling up facts; wisdom lies in their simplification.”

Martin Fischer (1856–1950)

The field of AR is still in its infancy, but is improving rapidly due to advances both directly and in related fields such as computer graphics and computer vision. In recent years, techniques currently in use from the broader realm of computer vision have seen considerable improvement in their robustness to several geometric and photometric transformations. However, in order to improve the algorithms further, it is important to have a complete and concise understanding of their inadequacies and limitations. This thesis demonstrates one such inadequacy: the inability to cope with non-uniform changes in illumination. Recent algorithms have achieved invariance to uniform changes in illumination, so mitigating the effects of non-uniform changes is a logical next step. A major contribution of this thesis is a careful formulation and analysis of the impact of the most common non-uniform illumination effects: shadows.

In order to provide such an analysis a range of images with differences that are due solely to illumination changes is required. Three datasets were utilised. The first—called WILD [NWN02]—provides images covering the complete spectrum of weather and illumination conditions present in a fixed scene in New York. The second dataset is another contribution of this thesis. The dataset is called SloDB and extends the type of attributes encapsulated in WILD, by capturing video from a repeated camera motion path. This contrasts to the fixed position that WILD uses.

Still frames can be extracted from these videos using ground truth information that was captured *per* frame. This information enables easy extraction of images taken from the same position, but under differing illumination and weather conditions. Alternatively, full motion video is available. This provides a useful test set for other fields in computer vision that require video, such as video surveillance. The third dataset utilised is called the Arizona test set and is comprised of frames extracted from several time-lapse videos from the University of Arizona.

The impact of non-uniform illumination was evaluated on seven different algorithms. Six of these algorithms were variations of local feature based approaches, while the seventh was a global Fourier-based method. The use of both the local feature based algorithms and the global Fourier-based method were presented in the context of two reference AR systems built during the course of this project. The system using the Fourier-based algorithm registered live camera images to a set of reference frames of the scene. This registration information allowed correct placement of a virtual object into the live view of the scene. The Fourier method can provide accurate registration results. However, as it operates on image differences, the resulting AR system provides only 4DOF. Essentially, users can only move around within a small distance from predefined viewing locations. Extending this system to support 6DOF—where the user can move around freely—proved impractical. In contrast, the system using local features provides a native 6DOF solution and was called the “3D system.” Through pre-processing, a 3D model is constructed from a series of images by extracting features, establishing geometric relationships between the images using features, and then using the relationships to construct the model. Live operation of the AR system involves matching live features from a single image, to the features that comprise the 3D model of the scene. Shadows impact this process at the feature detection and matching stages where putative matches between the live image and the 3D model are determined. If the 3D model is formed from images containing a dissimilar shadow configuration to that in the live image, performance will suffer, and the AR system may even fail.

The simplest way to test algorithms designed for use in such a situation is to compare one live image with one reference image, as explained in Chapter 5. A single reference image was compared to a set of images captured over the course of a single day. Experiments were performed to evaluate the detection, description and matching stages of the reference 3D system. These stages represent the essential

aspects of the larger AR process, and are affected the most by variable illumination. It was found that no feature detector was immune to the effect of shadows, and that performance was dependent on the level of similarity between the shadows in the reference image and the live image. Several different descriptor algorithms were also tested to ensure the inadequate performance could not be remedied through the use of a different descriptor technique. The results showed that the SIFT descriptor being used provided excellent performance compared with the other approaches, but that all methods were affected by shadows. Only the GLOH algorithm produced slightly better results, and only for a couple of images.

At first it is not apparent why shadows adversely affect computer vision performance, given that the human visual system—which computer vision systems are often modelled after—can cope with such changes easily. Chapter 5 showed that shadows impact both the detection and matching stages. Feature detection was found to be affected due to the loss of image data in shadow regions which precipitated a substantial reduction in the number of features detected in such areas. When the position of shadows in a scene did not match the reference data, the area of overlap where features were extracted was often severely reduced, resulting in fewer correspondences, essentially a reduction in the number of potential matches. This effect is analogous to that produced by viewpoint changes in that it reduces the amount of image overlap between the test and reference images. Testing with reference images taken from different times during the day confirmed this and showed that performance was highly correlated between the time at which a given live image was captured and the time that the reference image was captured. More specifically, this relationship is based on the configuration of the shadows and tests showed that individual reference frames produced good performance for images with similar shadow configurations but poor performance elsewhere. This means that no one reference frame produced robust performance for all live images.

In addition, the matching results showed that even with the correspondences that were found, matching images with dissimilar shadow configurations resulted in lower recall results than with images containing similar shadow configurations. An investigation into the cause of these matching deficiencies revealed that the bulk of the corresponding regions failed to match, as corresponding descriptors were no longer the best match. A smaller, but still significant number of correspondences failed due to a lack of distinctiveness from the other features in the set. The

uniformity of the WILD scene was likely the cause of this.

Utilising a selection of cloudy images from WILD, it was shown that images without shadows can provide consistent performance over most times during the day. This suggests that either the reference or live data must not contain shadows to allow successful operation. Otherwise the results are highly dependent on the relative shadow positions between the reference and live images. Imposing such a condition however, is not always practical given that most scenes contain shadows at some times of day. Therefore, Chapter 6 explored techniques to improve feature detection and matching in the presence of shadows, and is also a contribution of this thesis. Two approaches were discussed in the chapter: improving the live, or the reference data.

Methods evaluated to improve the live images included using modified thresholds for the detector algorithm and applying image enhancement operations. Both techniques were intended to increase the number of features detected in shadow regions, thereby increasing the feature overlap to the reference image. Using less restrictive thresholds increased the number of correspondences but seldom provided additional correct matches. Of the parameters available for modification, one that enabled the technique to adaptively select threshold parameters for different regions of an image based on their intensity provided the best performance. This was the *rel* parameter for MSER. With this parameter enabled, there were small improvements in the number of correspondences, correct matches and in precision. When operating on images with dissimilar shadow configurations these changes provided greater reliability in practical uses of the features. This demonstrated the advantage of adaptive approaches to threshold selection based on the region of the image.

Image enhancement, however, provided mixed results. For images captured early in the day, for which performance was poor, enhancements from contrast or gamma adjustment to homomorphic filtering provided a small increase in both recall and precision. For images during the middle of the day the changes produced poorer results. In addition, it was found that the performance of some algorithms improved while others declined. This suggests that such an approach would need to be finely tuned for specific circumstances, and that the choice of enhancement should be dependent on the extractor algorithm used.

Another approach to improving performance involves minimising the effect of shadows on the reference data. The major contribution of this thesis is the

PILLAGE algorithm, which takes advantage of reference data capturing the same scene under different illumination conditions. The PILLAGE algorithm exploits the robust performance between temporally adjacent sets to combine a series of reference sets, thereby forming matches that are both spatially and temporally stable over all available reference data. When merging reference data containing shadows, given sufficient coverage over all shadow configurations, a reference set that minimises the shadow’s impact can be formed.

Performance tests utilising reference data constructed using the PILLAGE algorithm showed significant improvements in the performance, and eliminated the effect of dissimilar shadow configurations. Consistent detector and matching performance was obtained, yielding reliable practical performance for virtually all images. This was verified on four data sets: the clear and cloudy selections of images from WILD, a set of images called the Arizona test set, and the SloDB data set. Results showed that the PILLAGE algorithm produced significant performance improvements in scenes with strong shadow effects, and importantly, does not adversely affect performance when there are no shadows. Utilising PILLAGE reference data together with MSER’s adaptive threshold selection applied to the live images yielded better performance than MSER using default values. This combination even provided good results on WILD images taken at 5am, a time where all other algorithms failed.

7.1 Design Guidelines

Based on the results obtained several suggestions can be made about how to deal with shadow effects. These suggestions are separated into two sections based on whether you are a developer who will be *using* feature extraction and matching algorithms, or a researcher *creating new* feature extraction and matching algorithms.

7.1.1 Developers using Existing Feature Extraction and Matching Algorithms

The best way to mitigate the effects of shadows is to avoid them to begin with. If there is a time of day where the sun is in a position such that there are no shadows in your scene, capture reference data at that time. Even if live data has shadows, removing the shadow effects from the reference data is enough to significantly improve performance. If no such time exists, capture reference data in overcast conditions. This avoids shadows, or at least softens them. Capturing in

overcast conditions has long been a technique photographers have used to improve image quality.

If shadows cannot be avoided because you want the maximum visibility that comes with clear weather or the scene you are capturing always has shadows regardless of the time of day try to capture scenes that do not contain large tonal ranges, or capture them from a position that avoids obtaining very bright and very dark (i.e. in shadow) regions in one image. Scenes with broad tonal ranges force the camera to try and capture as much of the range as possible while sacrificing subtle details in shadow regions. This makes shadows ‘stronger’, more intense and reduces the chances of detecting features in such regions. Alternatively, adjusting the camera exposure to let more light in will increase detail in areas of shadow but may wash out bright areas. If maximum robustness is required, capture reference data at various times during one day. Try to capture every possible shadow configuration and ensure there is sufficient overlap between temporally adjacent reference sets that they will match with high performance. These reference sets can then be merged using the PILLAGE algorithm effectively minimising the effect of the shadows.

If shadows cannot be avoided because you have existing reference data for the scene in question then your options are more limited. Experimenting with the feature detector’s parameters may yield improved performance if you are suffering from shadow effects, particularly if the algorithm provides adaptive options to automatically adjust the parameters for different parts of the image it is operating on. Applying enhancement operations in regions of shadow may also provide a small increase in performance but should be used sparingly as they often have a detrimental effect on performance in regions not in shadow. Although these techniques may only provide small increases in performance, this can yield tangible improvements in reliability in practical applications.

7.1.2 Researchers Creating New Feature Extraction and Matching Algorithms

When developing new feature extraction and matching algorithms it is important to think about how the algorithm will work not just under uniform illumination changes but non-uniform changes as well. Test the algorithm on images containing strong shadow effects such as those described in Chapter 4. Investigate the effect various algorithm parameters have on feature detection in regions of shadow. One parameter

setting may work best in most cases but does it work best in every situation? If not perhaps consider an adaptive approach to parameter selection. Jiri Matas' MSER algorithm is a good demonstration of the effectiveness of applying different algorithm parameters to different regions of the image based on their intensity.

7.2 Future Improvements

Future improvements are possible in several areas. As previously mentioned, adaptive feature detectors have been shown to provide advantages over those that use fixed threshold values over an entire image. This is particularly the case in images with shadows, where the default parameters can be too restrictive for features to be detected. This process is best performed during feature detection rather than integrating feature sets from multiple passes of the algorithm using different settings, as that increases computational complexity.

Another direction future work could take is combining multiple feature detection algorithms. This allows the weaknesses of any one algorithm to be mitigated by another approach that makes use of different image properties. As the analysis in Chapter 5 showed, all the algorithms presented suffered from the same weakness in the presence of shadows, thus combining algorithms would not help in such a circumstance. With improvements in the robustness of these techniques, however, perhaps employing the aforementioned adaptive improvements, differences between the algorithms may begin to emerge, particularly over different scene types.

In their discussion of EBR and IBR, Tuytelaars and Gool [TG00] state

“the idea is not to replace the above parallelogram type regions, but rather to complement them with other types. This should result in an opportunistic system that exploits a wide diversity of invariant regions depending on what is on offer. This should increase robustness and the number of correspondences found.”

It is postulated here that combining multiple algorithms with more restrictive detection and matching thresholds, will achieve better results by ensuring only the best features from each algorithm are used. Utilising only the best features provides a decrease in false matches at a rate greater than the decrease in correct matches, thereby improving the ratio of correct to false matches. Using restrictive thresholds with a single algorithm may reduce the number of correct matches so much that

no further computations are possible on the data. When using multiple algorithms, however, this may not be the case as the final feature set is comprised of the results generated by all the detectors. This could produce a number of features sufficient to compute geometric entities.

Using multiple feature extraction algorithms simultaneously greatly increases the computational complexity. By leveraging increases in computational power afforded by multi-core processors and powerful GPUs, however, significant speed increases can be obtained. Research has already begun in this area. Yang and Pollefeys [YP05] examined the use of a real-time correlation-based stereo algorithm implemented on a GPU, while Lalonde *et al.* utilise a GPU implementation of SIFT available in the OpenVIDIA library. The GPU implementation of SIFT obtains a speed-up of a factor of 10 over CPU implementations. GPUs are ideally suited for vision tasks due to the parallel nature of both the algorithms and the processors. Therefore, GPUs provide a clear path towards real-time computation of increasingly complex and robust algorithms. MSER, HARRAFF and SIFT have also been implemented on Field Programmable Gate Arrays (FPGAs) by Kristensen and MacLean [KM07], Cabani and MacLean [CM06], and Se *et al.* [SBJ05] respectively. FPGA implementations yield a customised circuit better suited to specialised applications such as the planetary rover presented by Se *et al.* FPGA implementations can also obtain significant performance increases; for example the FPGA implementation of MSER is reported to obtain an order of magnitude increase in performance over a software version.

Feature description methods are another area where improvements should be possible. GLOH has already been shown to obtain improvements in performance over SIFT in some instances. In addition, more robust statistical methods could allow the system to cope with a higher percentage of outliers than are possible with RANSAC. For example Subbarao *et al.* [SMG05] make use of a Projection based M-Estimator [CM03] in place of RANSAC. This technique, from a methodical standpoint, is more useful than RANSAC, as a user-defined threshold parameter is not required. Improper selection of the threshold parameter can affect the accuracy of the results provided by RANSAC.

Finally, and more speculatively, improvements could be made with knowledge of the locations of shadows in the images. With this knowledge, detection algorithms could remove anomalistic features generated because of shadows. Doing so would

eliminate a number of features guaranteed not to be present in another image containing a different shadow configuration. This process would increase computational complexity during detection, but would decrease complexity during matching and reduce the probability of obtaining false matches.

7.3 Closing Remarks

The goal of this thesis was to increase the usefulness of computer vision algorithms by extending the range of conditions under which they could successfully operate. Many applications of computer vision strive towards operation under any condition. To achieve this, the performance under all conditions must be evaluated to ensure robustness in every situation.

This thesis shows that shadows produce a significant impact on computer vision algorithms, and can even prevent systems working at all. Although not every scene or time of day will produce shadows severe enough to prevent successful operation, they are the most likely non-uniform illumination change a system will encounter, given that they occur over the course of each day. Therefore, a good understanding of their effects, and ways to mitigate these effects are desirable. This thesis has demonstrated when and how shadows impact performance, ways their impact can be reduced or eliminated, and approaches to further improve performance in the future. Therefore this thesis has achieved its goal of extending the range of conditions under which computer vision algorithms can be used, and takes another step towards the goal of robust computer vision systems.

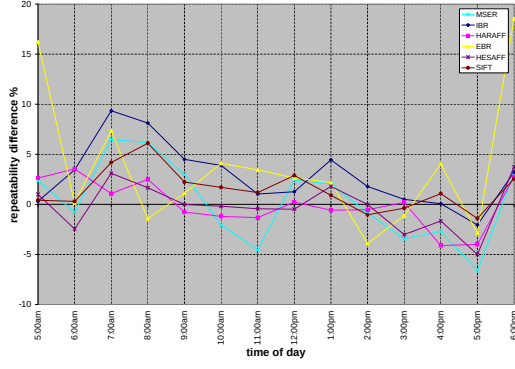
Appendix A

WILD Transformation Choice

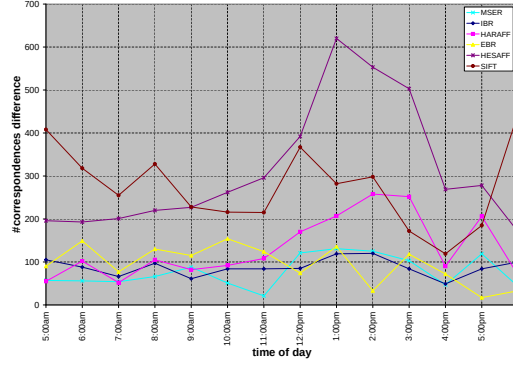
Chapter 4 presented two databases designed for testing computer vision algorithms under a range of weather and illumination conditions. One of these databases, WILD, is comprised of HDR images. Before testing, these images must first be transformed in order to be representative of images typically tested on computer vision algorithms. The choice of transformation presented involved a similar process to that used by a camera performing an auto-exposure capture. In essence, the mean intensity value is computed, μ , then the histogram is truncated to three standard deviations σ on either side of the mean, *i.e.* $\mu \pm 3 * \sigma$. Please note that it is purely coincidental that $\mu \pm 3 * \sigma$ is commonly used when discussing normal distributions as a way to select 99.7% of all values. The distribution of intensity values in most images in this dataset are not normally distributed. In fact, most images are considered low key, in that dark tones dominate the image, producing histograms skewed to the left.

It is conjectured that this transformation process would actually be beneficial over that of an auto-exposure camera, as the camera can obtain an incorrect reading of average intensity, potentially obtaining an under or over exposed image. As the database is comprised of HDR images, which capture a significant portion of the scene's intensity range, the computed mean intensity value is closer to the actual value than that computed from a light metre.

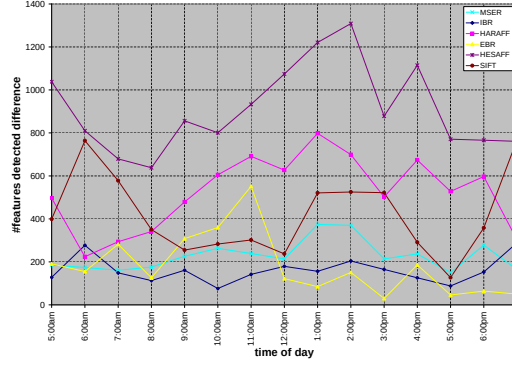
In order to verify that the choice of three standard deviations is a sensible one, this appendix explores the performance when using images transformed using two and four standard deviations. These tests make use of the *2pm* reference frame. Section A.1 explores the performance difference between two and three standard deviations, while Section A.2 compares the difference between three and four.



(a) Repeatability Difference



(b) Difference in Number of Correspondences



(c) Difference in Number of Features Detected

Figure A.1: Differences in detection performance between WILD images constructed using 3σ and 2σ

A.1 Performance of 3std vs 2std

Using two standard deviations from the mean, rather than three, provides an advantage in that the resolution of the intensity values of the original image are maintained in the region either side of the mean. This means it should perform better in dark regions where subtle differences might be lost when using three σ . In addition, performance in images with low tonal range should be improved, as image data for such images tends to be in a smaller range to begin with.

The downside of an increased resolution inside the histogram region covered by the truncation operation is a reduction in the range of possible intensity values that can be represented. This produces many regions that are over-saturated in images containing a large tonal range, for example at 12pm.

Figure A.1 shows the difference in detection performance using a transformation with σ set to two and three. A positive performance difference indicates higher performance when $\sigma = 2$.

The repeatability scores show some mixed results when comparing 2σ with 3σ .

Many of the algorithms obtained small increases from 7–9*am* but seem to exhibit a downward trend as the day goes on. From 2*pm* performance becomes increasingly worse than 3σ . This indicates that the early images can benefit from the increase in the resolution of the intensity values, despite a reduction in the overall range. This is not surprising given that these images obtain low performance due to a difference in the shadow configurations between the live and reference frame. With the resolution increase inside shadow regions, comes an increase in the number of features detected as shown in Figure A.1(c). The increase in correspondences shown in Figure A.1(b), however, is not as large as the increase in the total number of features, meaning a large number of new features are not detected in the reference image. Furthermore, as the early images do not contain highly illuminated regions—such as the white faces of the buildings—they do not suffer from the over-saturation problem produced by clipping the upper intensity portions of the histogram. This means these images obtain all the benefits of higher resolution, without paying the price for a reduction in overall range.

Later images, however, do suffer. The front faces of the three parallel buildings are the whitest region of the scene, and with the sun shining directly on them, these regions have the highest intensity in the image. Therefore these regions suffer the most from a range reduction. The subtle changes in intensity in these bright regions are lost, resulting in over-saturation. The truncated process is depicted in Figure A.2. The top row shows the original histograms for the 9*am* and 2*pm* images with the mean line and standard deviation lines rendered as the solid red, and dotted red lines respectively. The histograms on the second line show the truncated histograms, while the last row shows the resulting image. Notice in the final 9*am* image the front faces of the buildings are not over-saturated as they are not in direct sunlight. The detail in the rails on top of the middle building, however, have been completely washed out.

The 2*pm* image has sun directly on the front faces of the buildings, for example the middle building. As a consequence a lot of the detail on this building is washed out. These clipped intensity values are shown as the high intensity peak on the right hand side of the truncated histograms in Figure A.2.

Figure A.3 shows the difference in matching performance between images transformed using two or three standard deviations. Recall results are generally worse for all algorithms when using 2σ compared with 3σ , consistently producing lower scores.

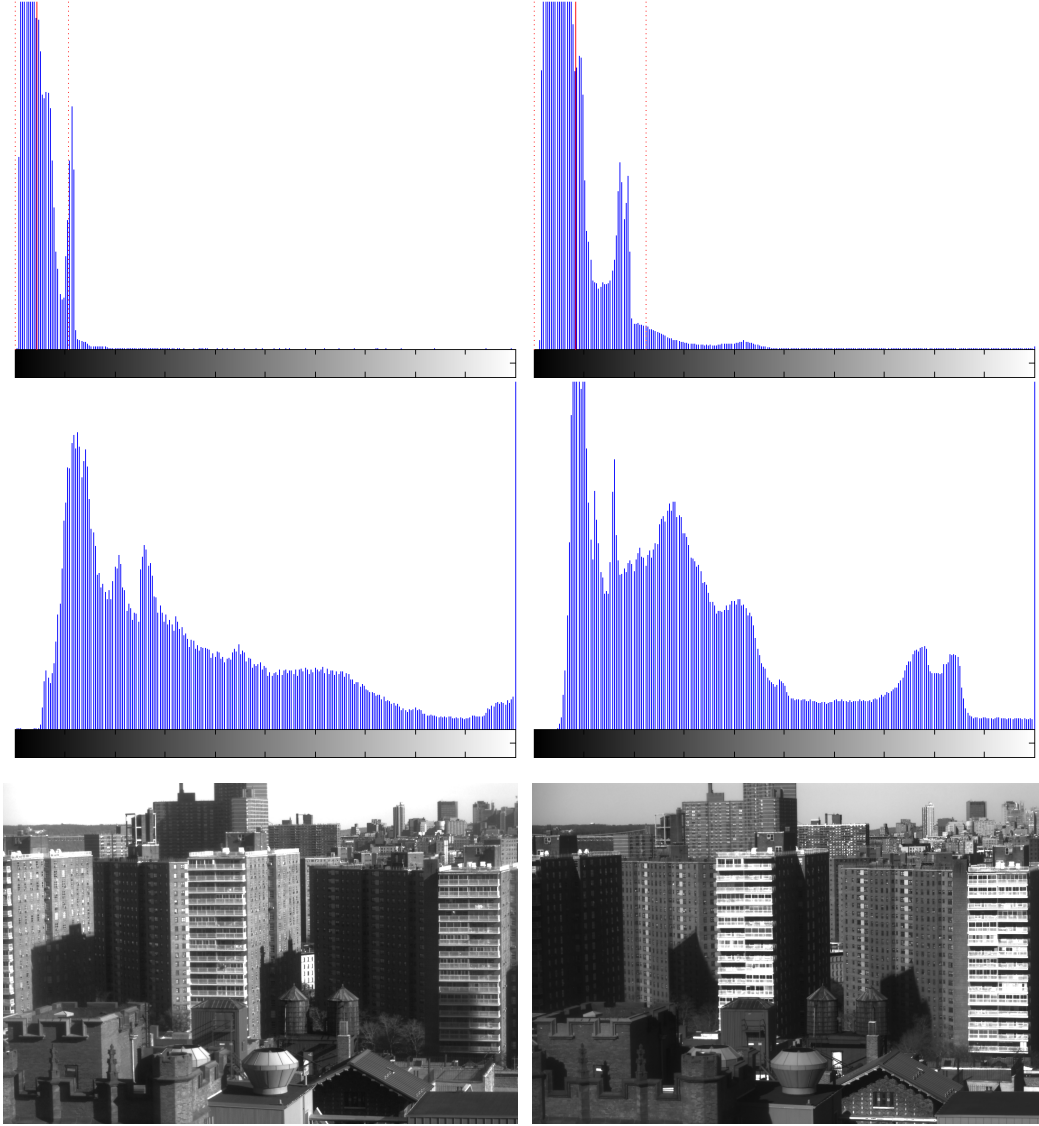


Figure A.2: Histogram truncation results when utilising 2σ to correct the $9am$ and $2pm$ WILD images. The left column shows the histograms and resulting $9am$ image, while the right column shows those for the $2pm$ image.

This is despite a similar number of correct matches up until $10am$, indicating the drop in performance stems from an increase in the number of correspondences, with no equal increase in correct matches. At $2pm$, however, HESAFF, HARAFF, EBR, and IBR obtain a large increase in the number of correct matches compared with 3σ . HESAFF obtains 150 more correct matches, however, to put this in context, 550 additional correspondences are found at this time, meaning that HESAFF suffers a drop in recall score. MSER obtains few correct matches from $10am$ to $6pm$ despite obtaining more correspondences, resulting in drops in recall performance of up to 12%.

Examining the 1-precision results shown in Figure A.3(c), shows that many of

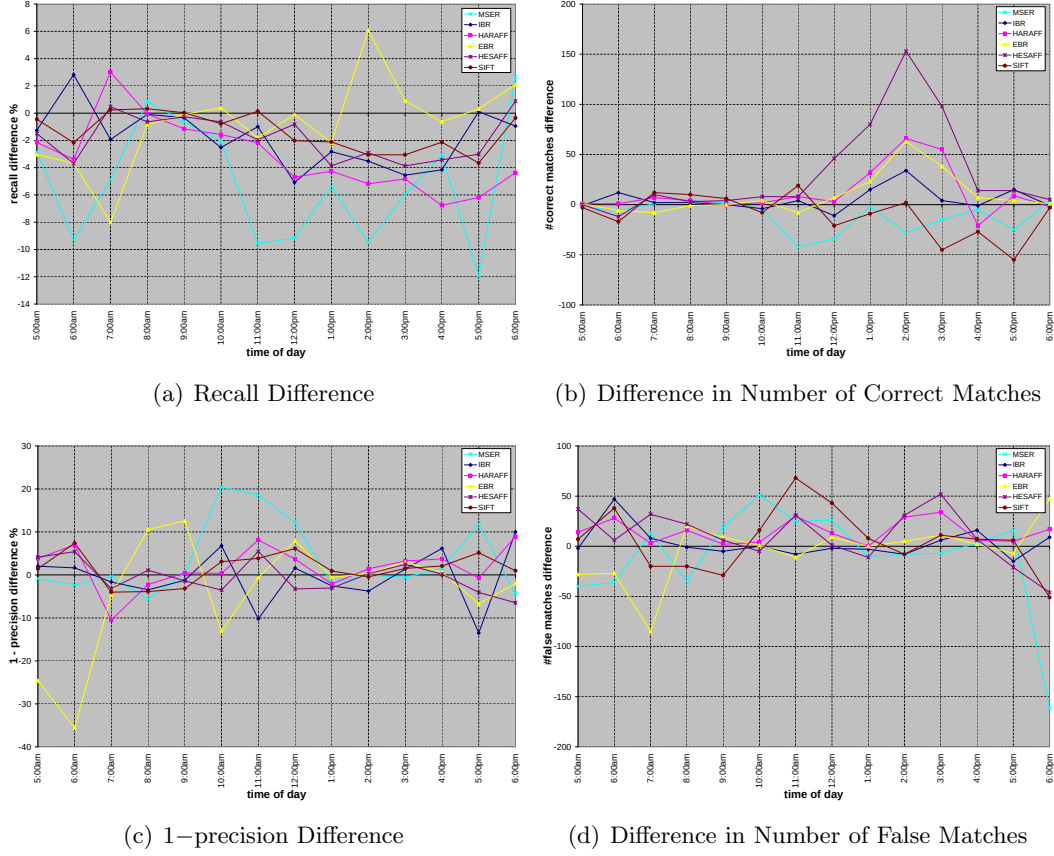


Figure A.3: Difference in matching performance between WILD images constructed using 3σ and 2σ

the algorithms obtain a similar level of precision, with the exception of EBR and MSER. At 6am EBR obtains a 35% improvement in precision when using 2σ over 3σ . At 8am and 9am, however, EBR obtains worse performance due to an increase in the number of false matches. Examining Figure A.3(d) shows a difference in the total number of false matches. EBR obtains the biggest increase at 7am where 85 fewer false matches are obtained when using 2σ . Despite this, an improvement in precision of only 4% is obtained as the ratio of correct to false matches did not improve much. This is because the number of correct matches at this time, dropped from 11 to 3.

Figure A.4 shows the difference in performance when computing a homography. At times when homography computation is successful, MSER obtains few inliers consistent with the estimated model. HESAFF however, obtains a higher number, mirroring the performance seen in the recall and precision results. This indicates that performance differences are algorithm specific, with HESAFF benefitting from using 2σ while MSER is disadvantaged by it. Importantly, however, is that the

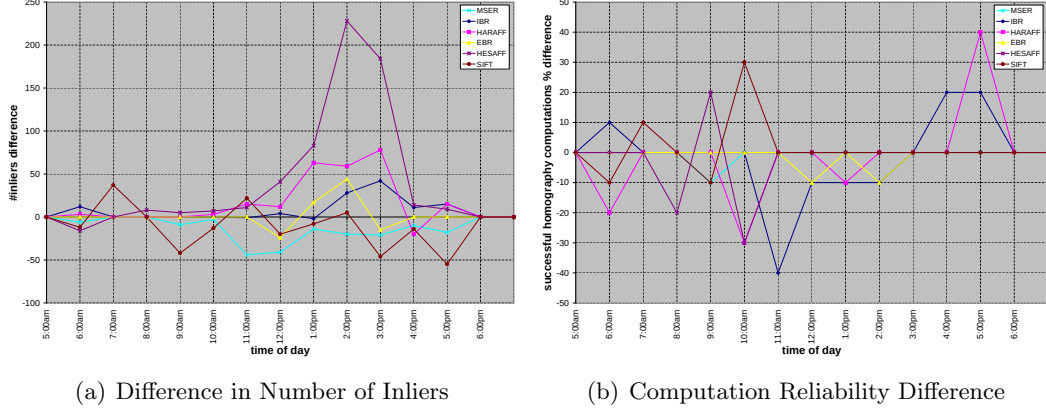


Figure A.4: Difference in homography computation performance between WILD images constructed using 3σ and 2σ

performance for the early images does not differ much between either 2σ or 3σ . Reliability also does not change much throughout the entire sequence and typically does not deviate by more than 5%. Between 12–3pm, when performance is the best under 3σ , there is no advantage in reliability to using 2σ , and some algorithms obtain slightly worse performance. Images up until 11am suffer from using 2σ more than they benefit.

A.2 Performance of 3std vs 4std

Figure A.5 shows the difference in detection performance between the clear weather WILD selection transformed using 3σ and 4σ . These results demonstrate a trend opposite to that seen between 2σ and 3σ , where there is an upward trend from the early images towards the later images. In contrast to the results seen between 2σ and 3σ , the performance of all algorithms is generally worse when using 4σ compared with 3σ , especially for early images. EBR and HARAFF obtain better repeatability scores from 11am to 1pm, whereas algorithms such as MSER and SIFT consistently obtain worse performance over the entire sequence. It should be noted, however, that the results typically differ by less than 5% from the performance using 3σ .

These performance drops are reflected by the large reductions in both the number of correspondences and the total number of features, as shown in Figures A.5(b) and A.5(c) respectively. The reductions in repeatability are the result of a disproportionate decrease in the number of correspondences compared with the total number of features detected. The drop in the number of features detected indicates that the reduction in intensity value resolution when using 4σ rather than three, limits the

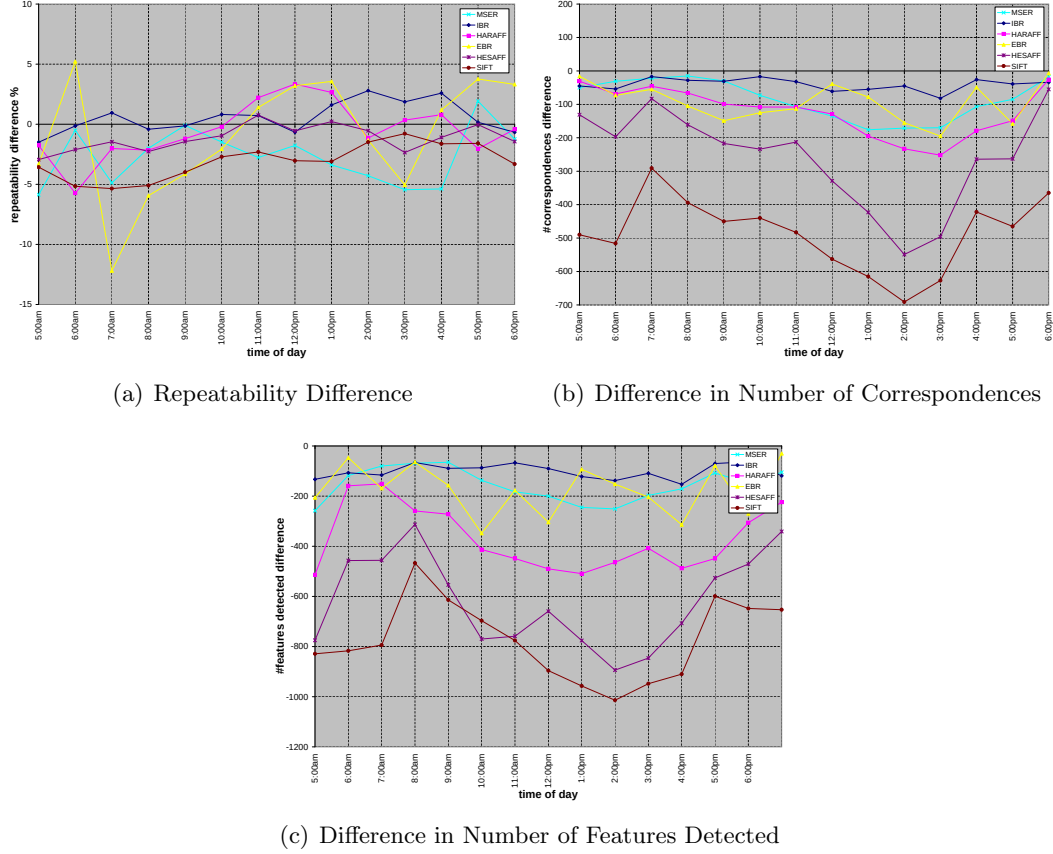


Figure A.5: Differences in repeatability results between WILD images constructed using 3σ and 4σ

number of points that can be detected.

Analysing the histograms for the $9am$ and $2pm$ images in Figure A.6, shows that utilising 4σ produces images without the over-saturated regions seen when using 2σ . For example, in the $9am$ image, the rails at the top of the centre building retain their detail and are not washed out by the histogram truncation process. Conversely, the $2pm$ image appears darker when using 4σ rather than 2σ . This is because large portions of the intensity data are now pushed further to the left of the final histogram, resulting in lower intensities for many regions, which subsequently makes them look darker. Notice also that the peak on the far right of the final histogram is significantly smaller than that seen when using 2σ . This is due to a larger range of intensities covered by the bounds of the truncation operation, resulting in fewer clipped intensity values.

The matching results shown in Figure A.7 demonstrate similar trends to the detection performance. Figure A.7(b) depicts the absolute number of correct matches and shows opposite results compared with 2σ *vs* 3σ . The early images

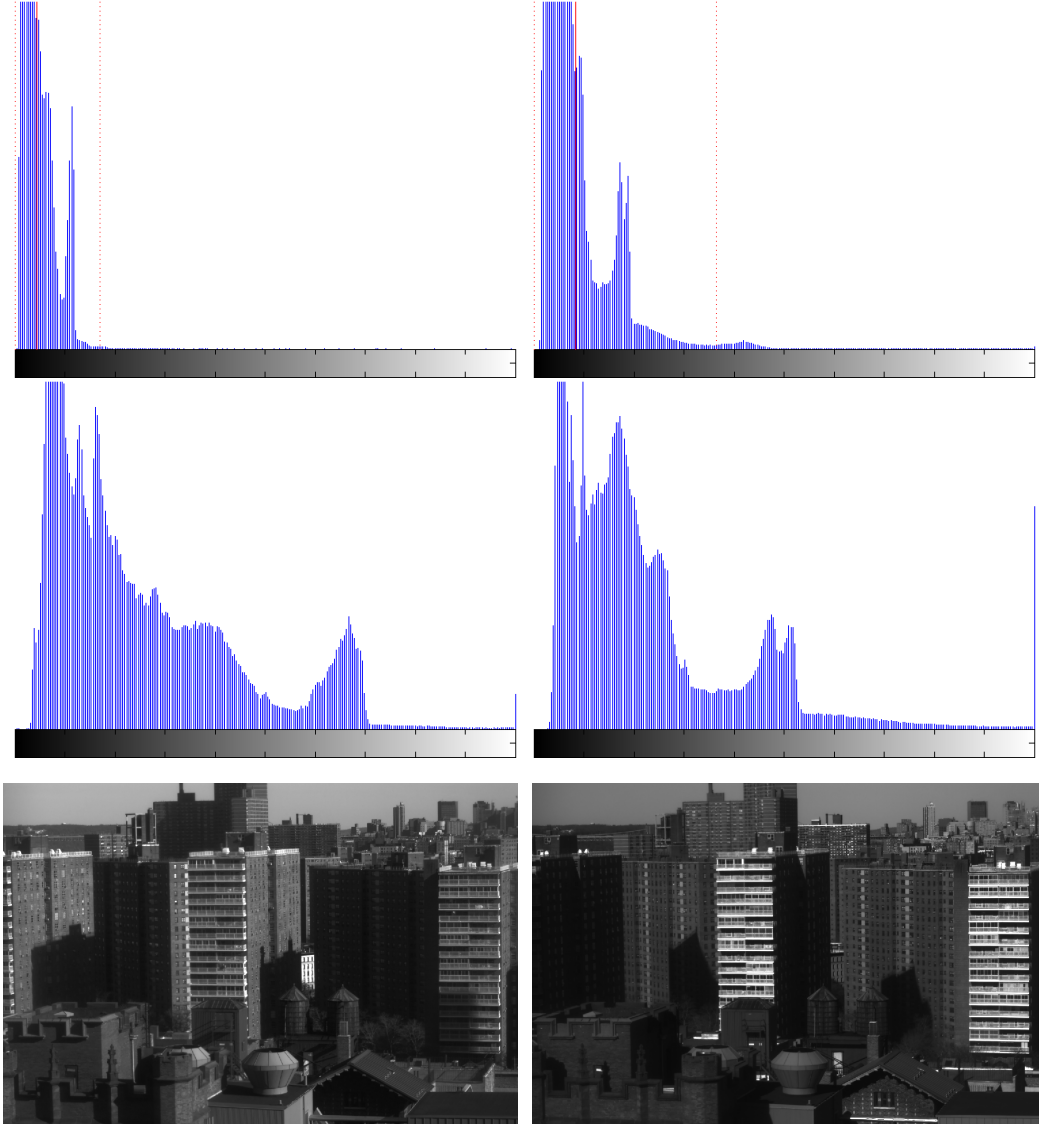


Figure A.6: Histogram truncation results when utilising 4σ to correct the $9am$ and $2pm$ WILD images. The left column shows the histograms and resulting $9am$ image, while the right column shows those for the $2pm$ image.

receive a slight reduction in the number of correct matches, whereas from $10am$ onwards the loss increases and reaches its lowest value at $2pm$. The results then progressively converge to values identical to those obtained using 3σ at $6pm$.

The 1-precision scores show mixed results with most algorithms generally obtaining worse performance when using 4σ compared to three. MSER obtains the most significant performance drop of any algorithm from $10am$ to $12pm$. This mirrors the trend seen between 2σ and 3σ , indicating that performance for these times is best for MSER when using 3σ . This is verified by examining the number of false matches in Figure A.7(d). This shows that during this time MSER obtains a higher number of false matches when using both 2σ and 4σ .

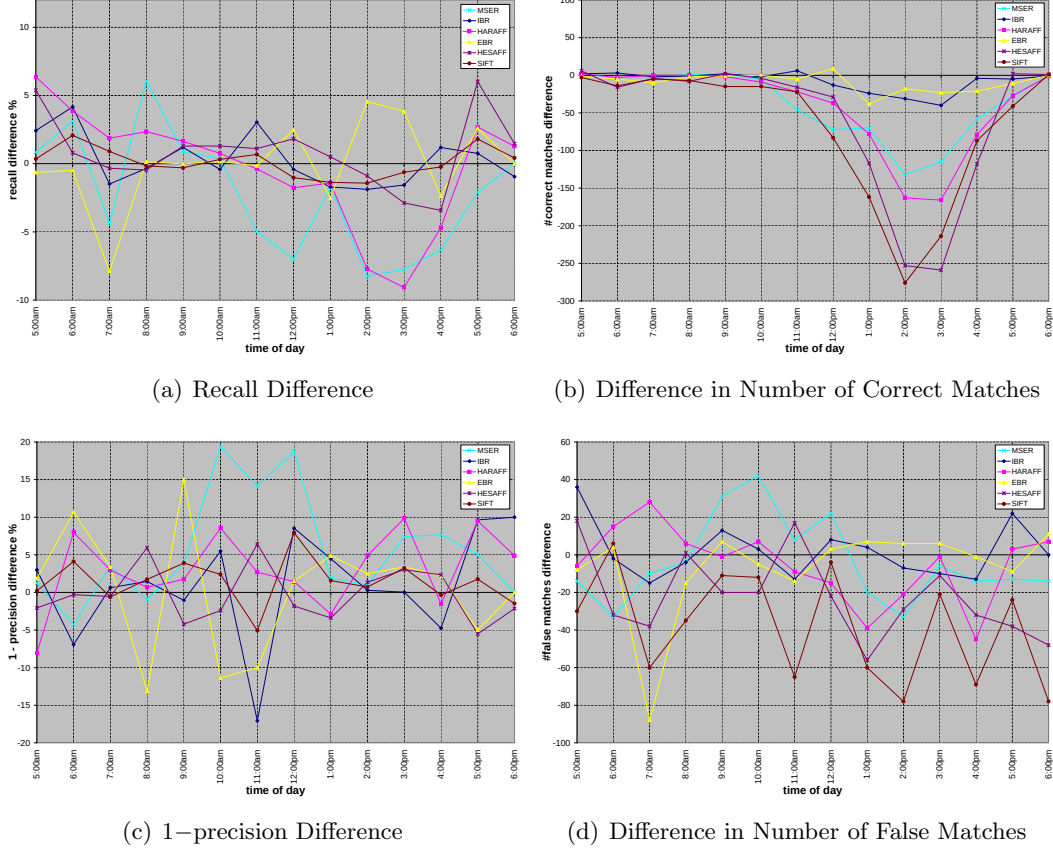
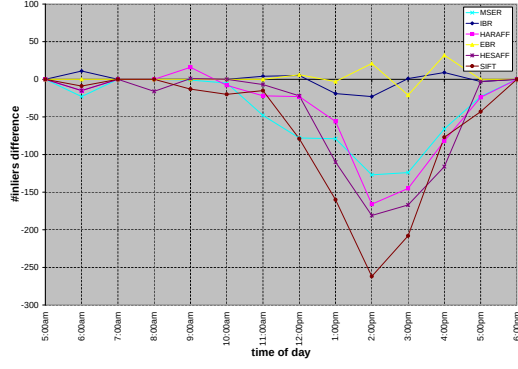


Figure A.7: Difference in matching performance between WILD images constructed using 3σ and 4σ

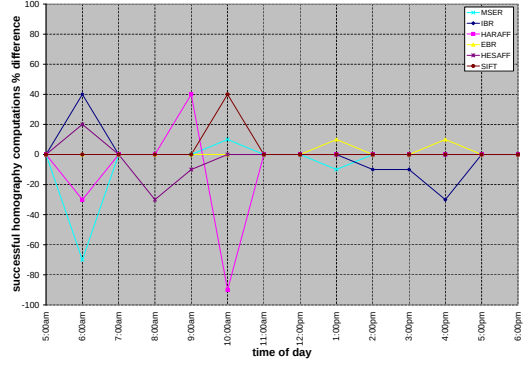
The other algorithms vary between enhancements in 1-precision of up to 5%, and reductions in precision up to 10%. The absolute number of false matches generally decreases when using 4σ , however, an accompanying decrease in the number of correct matches eliminates any potential performance gain by obtaining fewer false matches.

The difference in performance when computing a homography mirrors that seen for the number of correct matches. The performance of most algorithms is slightly lower for the early images, and then drops substantially, reaching its lowest point at $2pm$.

In terms of reliability, IBR obtains a 20% increase at $6am$ due to a small improvement in both the number of correct and false matches. MSER on the other hand, drops by 70% at $6am$ thus obtaining a reliability score of 0%, meaning homography computation fails at this time. At $10am$ HARAFF's reliability drops 90% and only succeeds in computing a homography 10% of the time. From $11am$ to $6pm$ reliability remains at similar levels to 3σ for most algorithms except IBR



(a) Difference in Number of Inliers



(b) Computation Reliability Difference

Figure A.8: Difference in homography computation performance between WILD images constructed using 3σ and 4σ

which drops 30% at 4pm.

Appendix B

Matching Score Results

This appendix provides the matching score results for the experiments in Chapters 5 and 6. It is intended that these results be considered with respect to the original detection and matching results already presented.

B.1 Methodology

The matching score metric was utilised by Mikolajczyk *et al.* in their comparison of affine covariant detectors [MTS⁺05] and evaluates the performance of the matching process. It is computed using:

$$\text{matching score} = \frac{\# \text{correct matches}}{\min(\# \text{features in 1st image}, \# \text{features in 2nd image})} \quad (\text{B.1})$$

In essence it is the ratio of the number of correct matches with the smaller of the number of features detected in the images being matched. Matches are selected as the nearest neighbour in descriptor space. The matches are verified as being correct by ensuring these features are also correspondences, where their overlap error is less than e which is 40%.

Both the repeatability and matching score are computed as a ratio of the total number of features extracted from either the reference or live frame. Therefore, the maximum value the matching score can achieve equals the repeatability score. A matching score similar to the repeatability indicates a high percentage of correspondences are correctly matched.

Like the repeatability score, the matching score cannot be taken alone without the absolute number of correct matches. A high matching score could be obtained

with a low number of correct matches if the total number of features extracted in one of the images was low, giving a misleading impression of algorithm performance in a practical situation.

B.2 Test Results

B.2.1 Clear Weather WILD Selection

Figure B.1 shows the matching score results for the matching test in Figure 5.10. Figure B.1(a) and B.1(b) show the matching score and the number of correct matches respectively. A similar, but more pronounced trend to that seen in the repeatability results of Figure 5.9(a), is witnessed here. Low matching scores are obtained from 5–9am with a small peak at 6am. From 10am onwards performance begins to increase along with the number of correct matches shown in Figure B.1(b), reaching the maximum at 2pm. Performance then drops off and obtains the worst performance at 6pm where all algorithms obtain matching scores less than 1%.

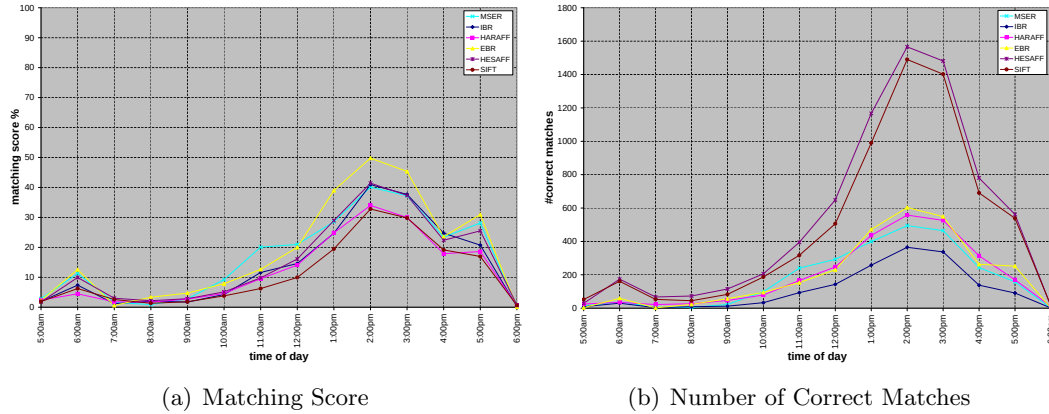


Figure B.1: Matching score results for WILD selection

Comparing the matching score to the recall values shown in Figure 5.10, it can be seen that similar results are obtained. The most significant difference between the two metrics is the ranking of the algorithms. In the matching score, EBR consistently performs best. However, in terms of recall it performs the worst. This suggests that a higher percentage of the total number of features detected by EBR are classified as correct, compared with the other algorithms. The recall results, however, indicate that given the number of corresponding regions obtained by EBR, the number of correspondences also classified as matches is low compared with the other approaches. This suggests inadequacies in matching rather than

detection performance. Considering EBR does not detect as many features as SIFT or HESAFF, the percentage of correct matches is high, however given the number of correct matches that could be obtained, the numbers are comparatively low.

A significant difference between the recall and matching score results is seen at 6am where MSER obtains a recall rate almost double the matching score. This indicates that as a percentage of the total number of features found, MSER obtains similar results to the other algorithms; however, given the number of potential matches, it performs significantly better.

The number of correct matches shows HESAFF and SIFT with the most matches, indicating that although SIFT is obtaining a lot of correct matches—by virtue of the fact SIFT extracts a lot more regions and consequently produces more correspondences—the number of false matches is increasing at an equal rate, producing matching scores similar to the other algorithms.

B.2.2 Cloudy Selection

Figure B.2(a) shows the matching score results for the cloudy selection of images. Comparing these results to those of the clear weather selection shown in Figure B.1(a), demonstrates a substantial increase in performance up until 1pm. Between 1-3pm performance is better in the clear weather selection as the similarity of the shadow configurations between the clear weather reference and live images produces near identical feature sets, and thus a higher proportion of correspondences. From 4pm onwards the cloudy selection performs better.

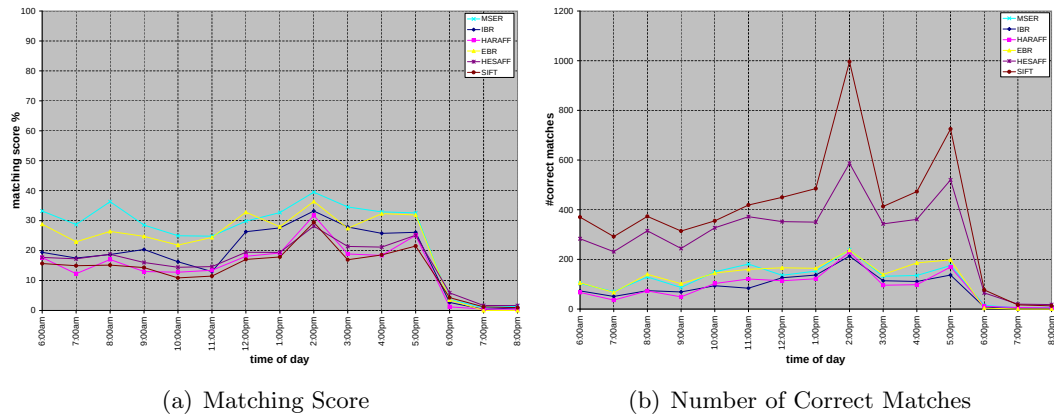


Figure B.2: Matching score results for selection of cloudy WILD images

EBR particularly suffers at 2pm when its matching score drops from 49.8% in the clear weather set, to 36.4% under cloudy conditions. MSER's ranking benefits over

the clear weather results and outperforms the other algorithms. At 8am, MSER's matching score increases from 0.88% in the clear weather selection, to 36.3% in the cloudy images.

The matching score improvements for the early images are achieved through a substantial increase in the number of correct matches, despite a decrease in the total number of features extracted from the cloudy images for many algorithms. The increase in the number of correct matches is brought about by the more uniform distribution of features extracted from the cloudy images. The most important attribute of performance under cloudy conditions is the consistent results obtained, which do not fluctuate as they do when shadows are present.

B.2.3 Seasonal Changes

The average matching score results shown in Figure B.3(a) mirror the downward trend shown in the repeatability results of Figure 5.23 and the recall results of Figure 5.24. The change in matching score for all algorithms increases the larger the seasonal variation becomes, with many of the algorithms sustaining a 40% drop in matching score performance. EBR produces the least variation as is demonstrated by the flatness of the line compared with the other algorithms, particularly from 3–4 to 5–6 months where the line for EBR is almost flat. The flatness of this line indicates EBR is potentially more robust to seasonal changes than the other algorithms.

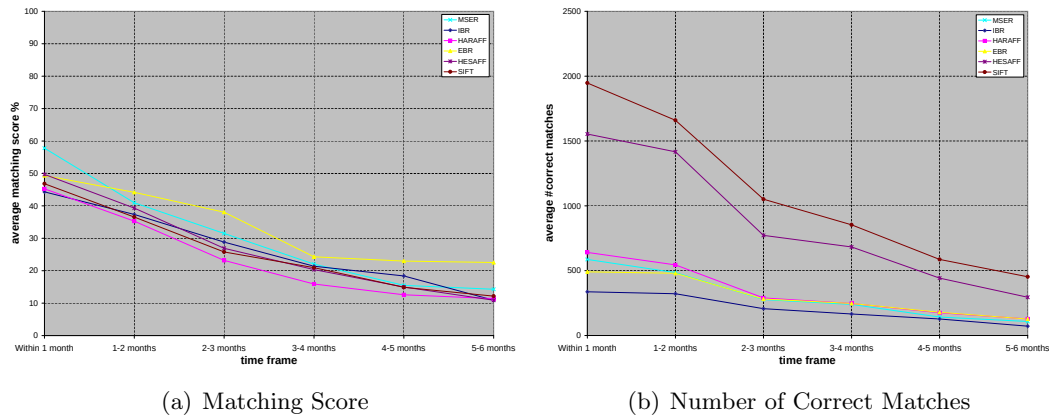


Figure B.3: Matching score changes for seasonal WILD selection

B.2.4 SloDB Selection

The matching score results shown in Figure B.4 show a substantial difference in algorithm ranking compared with the repeatability results of Figure 5.28(a). SIFT,

which performed best in repeatability tests, often obtains the worst matching score results indicating that it obtains the lowest number of correct matches considering how many features were detected. This indicates the detector performance is high as it re-detects the same features in this scene, however, those regions are not as reliable for matching, resulting in lower performance. IBR consistently obtains the highest scores throughout the sequence with MSER ranked in the middle of the group. This ranking differs from the recall results of Figure 5.30(a), where MSER consistently achieves significantly higher recall scores than the other algorithms. This indicates a discrepancy between the matching score and recall metrics. In this instance, the recall results for MSER are more representative of the true performance of the algorithm. The recall scores demonstrate that MSER's matching results are closer to its best possible performance than the other algorithms, meaning a higher percentage of re-detected MSER points are correctly matched than any other algorithm.

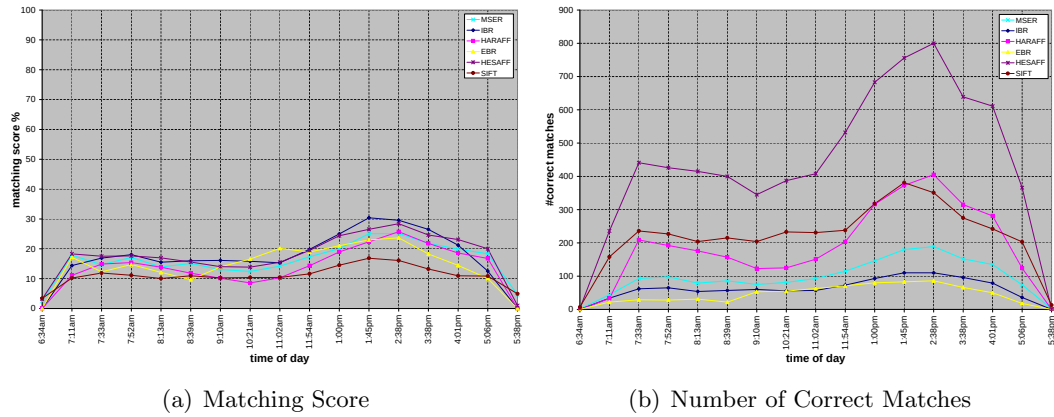


Figure B.4: Matching score results for SloDB selection

The number of correct matches of Figure B.4(b) paints a similar picture to the correspondence results. The exception to this is SIFT, which obtains similar numbers of correspondences to HESAFF, but drops considerably in terms of the number of correct matches, and obtains results more similar to HARAFF. This results in HESAFF obtaining better matching performance despite SIFT's advantage in detector performance shown in the repeatability results.

B.2.5 Descriptor Performance

The matching score results for different descriptor types shown in Figure B.5 are similar to the recall results of Figure 5.32. The same trends are exhibited by each of these metrics with the only difference being the number of matches that are deemed

correct. The number of correct matches obtained during recall computation are considerably less than those utilised for the matching score. This is a result of more a restrictive process in determining correct matches during recall computation than that used for the matching score.

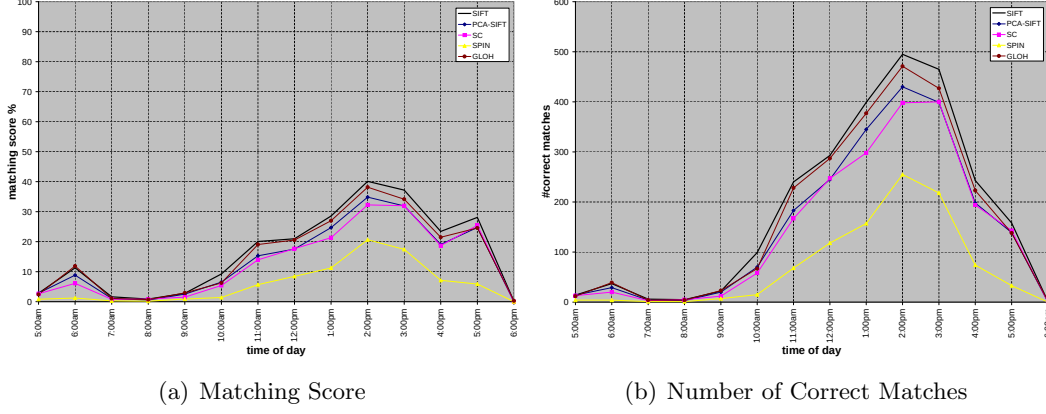


Figure B.5: Matching score results for Different Descriptors on WILD selection

During matching score computation correct matches are determined as the closest match, without any restriction on how close the second closest match must be. Therefore, the number of correct matches obtained from this process are similar to those using the distance ratio technique with the ratio set to one, as shown in Figure 5.34(b). Another difference between the matching score and recall results, is that SPIN images obtain recall performance similar to the other algorithms, whereas, matching score performance, and the number of correct matches is substantially lower than the other approaches.

B.2.6 Relaxed Thresholds

Figure B.6 shows the matching score results obtained when using relaxed threshold parameters for MSER. Similar results to the repeatability scores in Figure 6.2(a) are obtained for each of the parameter combinations. For instance in both the repeatability and matching scores, enabling the *rel* parameter produces significant increases in performance between 6–9am.

When comparing the matching score to the recall results of Figure 5.24(a), the matching score better demonstrates the improvement in matching afforded by toggling the *rel* parameter. For example, between 6–9am the recall results show no improvement in matching performance despite obtaining more correct matches over this time. The matching scores, however, show an improvement over these times,

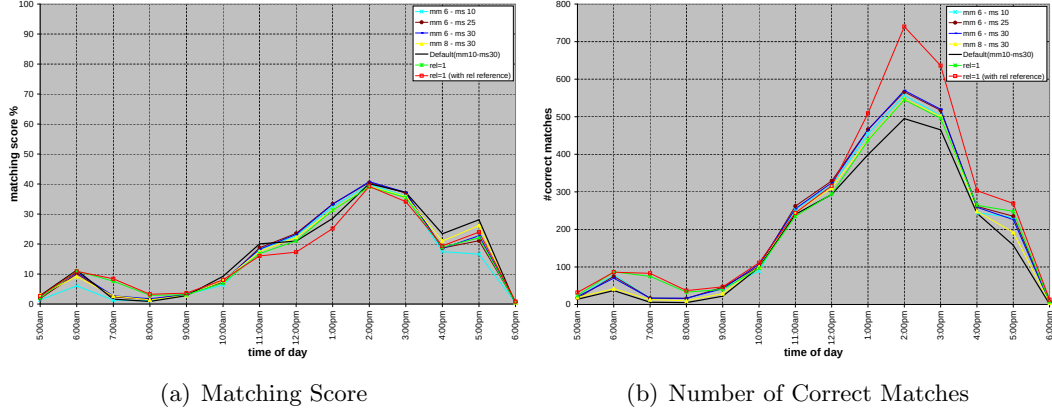


Figure B.6: Matching score results using relaxed thresholds

in addition to more correct matches as shown in Figure B.6(b). Improvements in matching score are obtained as the number of correct matches increases more than the total number of detected features. The recall results do not reflect the increase in the number of correct matches as the number of correspondences increase at a similar rate. The improvement in matching performance is demonstrated however, in the 1-precision scores shown in Figure 5.24(c), where the improvement in precision demonstrates the ratio of correct to false matches improves when the *rel* parameter is enabled.

B.2.7 Image Enhancement

The following experiments present the matching score results for the image enhancement tests of Chapter 6, Section 6.1.2.

Contrast Adjustment

Figure B.7 shows the difference in the matching score and the number of correct matches between the clear weather set, and the contrast adjusted versions. The matching score results with a contrast adjustment applied to the shadow regions of the image, show some improvement over the unadjusted images. From 6-8am, performance for all algorithms increases by 0-5% with the largest gain obtained at 7am. From 1pm onwards, however, performance drops by up to 4%. This is not surprising given that the shadow configurations around 2pm are nearly identical to the unadjusted reference frame. This means any new features detected in the shadow regions due to the enhancement cannot be matched, and increases the probability of obtaining false matches.

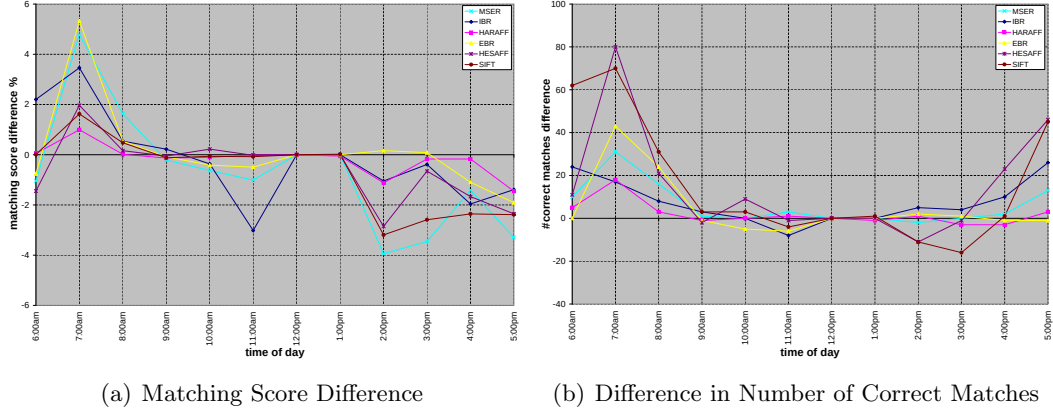


Figure B.7: Difference in matching score results using contrast enhancement

Some similar trends to the repeatability results are seen. For example, both the detection and matching results for MSER, IBR and EBR benefit the most from the contrast enhancement in the early images, whereas HESAFF, HARAFF, and SIFT obtain little or no benefit. Despite this, the difference in the number of correct matches shown in Figure B.7(b), demonstrate that HESAFF and SIFT receive the largest gain at 7am. This means their relatively low matching score increases are the result of an even greater increase in the number of features detected.

Gamma Adjustment

When applying a gamma adjustment EBR, IBR and MSER again obtain the largest gains followed by HESAFF and SIFT. Like the repeatability results shown in Figure 6.8(a), HARAFF obtains virtually identical matching scores to the unadjusted set throughout the sequence, indicating it is invariant to gamma changes. As was seen for the contrast adjustment, performance gains are only obtained between 6–8am with the largest gain at 7am. Furthermore, the performance from 2pm onwards is lower than the unadjusted images.

The differences in the number of correct matches shown in Figure B.8(b) also follow the same trends seen in the contrast adjustment. HESAFF and SIFT obtain the largest increase in correct matches at 7am. From 9am to 3pm the number of correct matches varies little from the unadjusted results, from 3pm onwards the results begin to diverge.

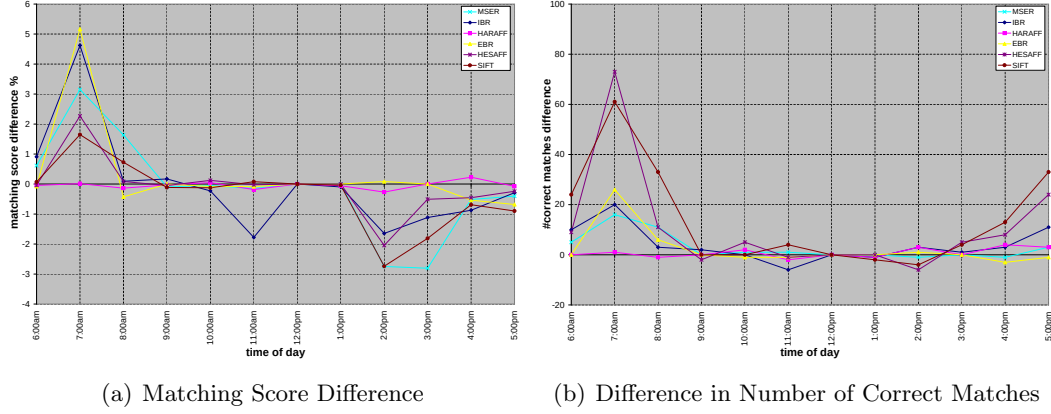


Figure B.8: Difference in matching score results using gamma enhancement

Homomorphic Filtering

Homomorphic filtering produces the smallest differences in matching score of the three enhancement algorithms. A increase in performance of 3% is obtained by MSER at 7am. At 2pm a drop in performance of 2.5% is obtained, while many of the other algorithms achieve close to baseline performance. The number of correct matches also show comparatively small differences compared with the contrast and gamma adjustments. SIFT obtains the largest gain at 7am, finding 17 additional matches compared with the 70 and 60 obtained under contrast and gamma adjustments respectively.

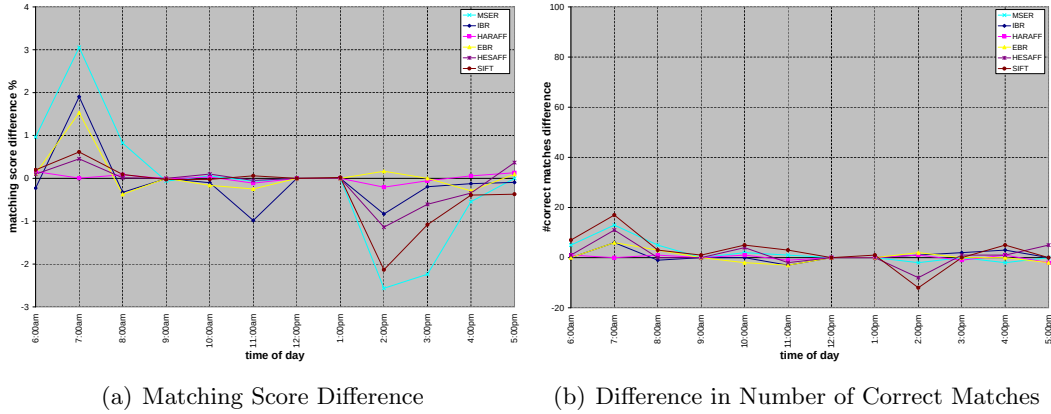


Figure B.9: Difference in matching score results using homomorphic filtering

B.2.8 PILLAGE

The following graphs provide the matching score results when using the PILLAGE algorithm. The corresponding detection, recall and precision results are presented in Chapter 6, Section 6.2.2.

Clear Weather WILD Images

Figure B.10 shows the matching score results when testing PILLAGE reference data with the selection of clear weather WILD images. The results show more consistent matching performance when utilising reference data constructed using the PILLAGE algorithm. The early images show a marked improvement compared with the low performance obtained when using a single $2pm$ reference frame, as was shown in Figure B.1. For some algorithms such as HESAFF, optimal matching performance is even obtained at $7am$ instead of the more common $2pm$.

Another interesting result is that the matching scores when *rel* is enabled on MSER, are lower than when it is disabled, despite a higher number of correct matches with *rel* enabled. This indicates that enabling the *rel* parameter increases the number of features detected more than it increases in the number of correct matches, thus producing lower matching scores.

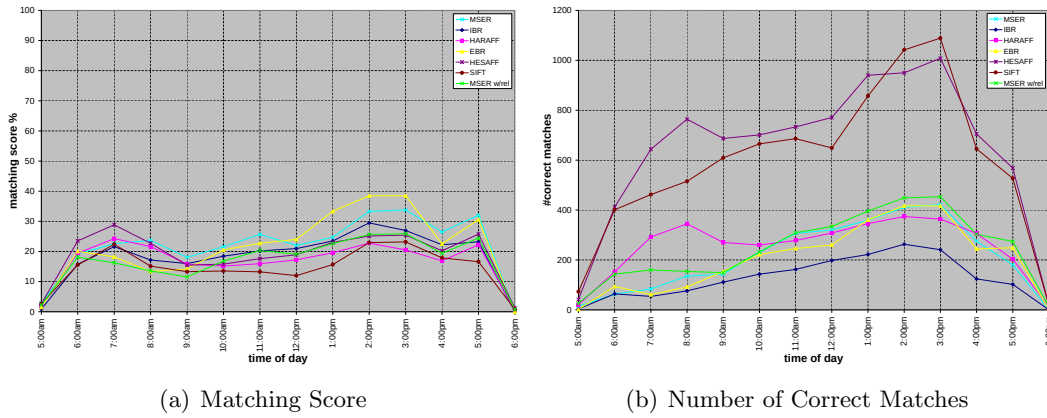


Figure B.10: Matching score results for WILD selection in clear weather using reference data constructed from PILLAGE

Comparing these results to the recall data, indicates that EBR continues to obtain a higher ranking using the matching score metric than is obtained when using recall. This follows the trend seen when using a single reference frame, meaning that even when using PILLAGE, EBR obtains a higher percentage of correct matches given the total number of features extracted than the other algorithms. Conversely, considering how many re-detected features exist, the percentage of corresponding features that are also correctly matched is lower than other approaches.

Cloudy WILD Images

Figure B.11 shows the matching score results using the cloudy WILD selection and reference data constructed using the PILLAGE algorithm. Comparing Figure B.11(a) to the matching score results when using the 2pm reference frame shown in Figure B.2, indicates that the PILLAGE algorithm produces a higher degree of consistency even under cloudy conditions. In addition, the matching performance for all algorithms is improved by PILLAGE, with the exception of images after 6pm where results are similar to those obtained using a single reference frame.

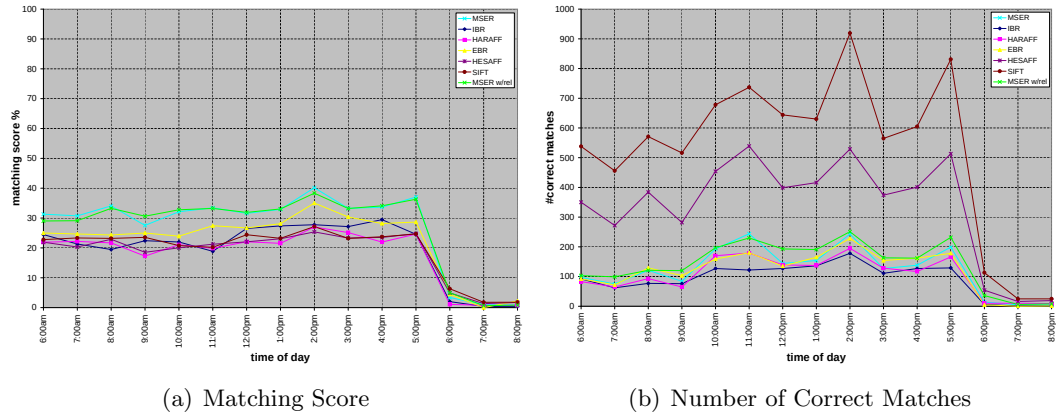


Figure B.11: Matching score results for WILD selection in cloudy weather using reference data constructed from PILLAGE

The algorithm ranking differs compared with those seen with the clear weather set. Here, MSER obtains the highest performance over the entire sequence, except for the dark images after 6pm. EBR then follows, and the remaining algorithms all obtain similar performance.

It can also be seen that MSER's *rel* parameter provides little benefit, and performance remains at similar levels to when *rel* is disabled. This is expected, as the adaptive approach provided by the *rel* parameter improves performance in dark regions of an image, such as in shadows. The cloudy images, however, do not contain shadows, and exhibit more uniform illumination, hence the *rel* parameter provides no benefit.

Arizona Test Set

Figure B.12 shows the matching score results using the Arizona test set shown in Figures 4.11 and 4.12, and the reference images shown in Figure 4.13 combined using the PILLAGE algorithm. The matching scores show very similar results

to the repeatability and recall results shown in Chapter 6, Section 6.2.5. The best performance for each image from the test set is obtained by the reference image captured at the most similar time. This produces highly variable matching performance where early reference images produce high performance for early test images but poor performance for later images. Conversely, late reference images produce poor performance for early test images but high performance for late test images. The PILLAGE frame produces the most consistent performance of all the reference frames. It should be noted that the dip in PILLAGE performance for test images 10–14 was also present in the repeatability and recall results of Chapter 6. For a detailed analysis of this see Chapter 6, Section 6.2.5.

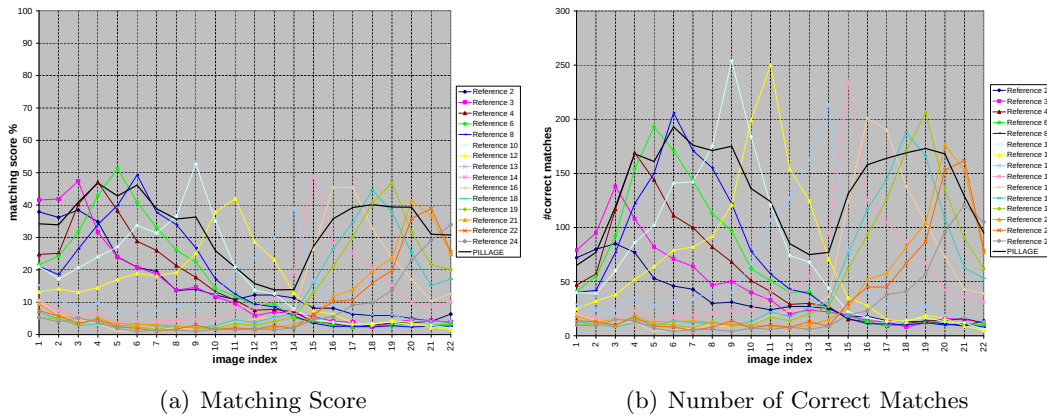
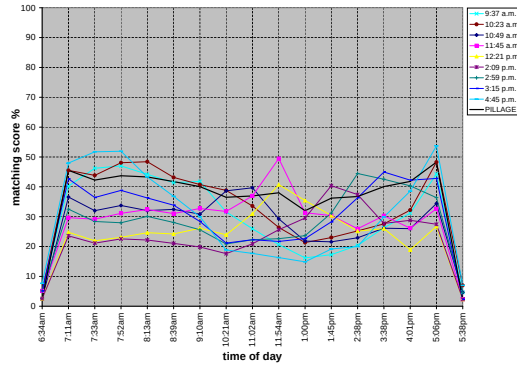


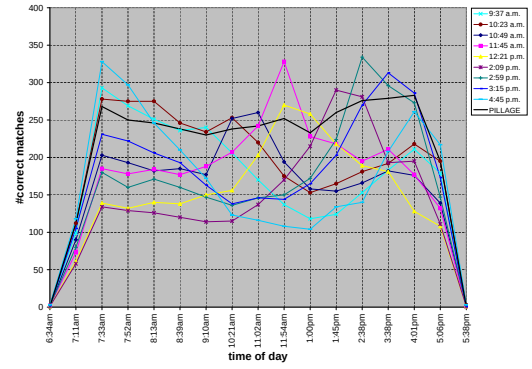
Figure B.12: Matching score results for Arizona test set including results using reference data constructed from PILLAGE

SloDB Selection

Figure B.13 shows the matching score results for a range of different reference images including one constructed using the PILLAGE algorithm. Like the repeatability and recall results of Chapter 6, Section 6.2.6 showed, the matching performance here is more consistent than the other data sets as the SloDB images do not contain such strong shadows. Test images still produce the best performance when matching against reference images taken at similar times, however, the effect is not as pronounced as in the WILD or Arizona data sets. The results also show that the PILLAGE frame produces the most consistent results.



(a) Matching Score



(b) Number of Correct Matches

Figure B.13: Matching score results for the SloDB image selection including results using reference data constructed from PILLAGE

Bibliography

- [ABB⁺01] Ronald Azuma, Yohan Baillot, Reinhold Behringer, Steven Feiner, Simon Julier, and Blair MacIntyre. Recent Advances in Augmented Reality. *IEEE Computer Graphics and Applications*, 21(6):34–47, November/December 2001.
- [AHIS99] Ronald Azuma, Bruce Hoff, Howard Neely III, and Ron Sarfaty. A Motion-Stabilized Outdoor Augmented Reality System. In *IEEE VR '99*, pages 252–259, Houston, Texas, 13-17 March 1999.
- [ALCS03] Kusuma Agusanto, Li Li, Zhu Chuangui, and Ng Wan Sing. Photorealistic Rendering for Augmented Reality using Environment Illumination. In *ISMAR '03: Proceedings of the The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality*, page 208, Washington, DC, USA, 2003. IEEE Computer Society.
- [ALJ⁺99] Ronald Azuma, Jong Weon Lee, Bolan Jiang, Jun Park, Suyu You, and Ulrich Neumann. Tracking in Unprepared Environments for Augmented Reality Systems. *Computers & Graphics*, 23(6):787–793, December 1999.
- [AMU97] Yael Adini, Yael Moses, and Shimon Ullman. Face Recognition: The Problem of Compensating for Changes in Illumination Direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(7):721–732, 1997.
- [Azu95] Ronald Azuma. Predictive Tracking for Augmented Reality. Technical report TR95-007, University of North Carolina, Chapel Hill, 1995.
- [Azu97] Ronald Azuma. A Survey of Augmented Reality. *Presence: Teleoperators and Virtual Environments*, 6(4):355–385, August 1997.

- [Azu99] Ronald Azuma. The Challenge of Making Augmented Reality Work Outdoors. In *First International Symposium on Mixed Reality*, Mixed Reality: Merging Real and Virtual Worlds, pages 379–390, Yokohama, Japan, 1999. Springer-Verlag.
- [Beh98] Reinhold Behringer. Improving the Registration Precision by Visual Horizon Silhouette Matching. In *First IEEE Workshop on Augmented Reality*, San Francisco, CA, November 1998.
- [Beh99] Reinhold Behringer. Registration for Outdoor Augmented Reality Applications Using Computer Vision Techniques and Hybrid Sensors. In *IEEE Virtual Reality*, page 244, Houston, Texas, March 13 - 17 1999.
- [Ber94] Marie-Odile Berger. How to Track Efficiently Piecewise Curved Contours with a View to Reconstructing 3D Objects. In *International Conference on Pattern Recognition (ICPR)*, pages 32–36, 1994.
- [BFY00] Michael J. Black, David J. Fleet, and Yaser Yacoob. Robustly Estimating Changes in Image Appearance. *Computer Vision and Image Understanding: CVIU*, 78(1):8–31, 2000.
- [BIF04] Hrvoje Benko, Edward W. Ishak, and Steven Feiner. Collaborative Mixed Reality Visualization of an Archaeological Excavation. In *ISMAR '04: Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)*, pages 132–140, Washington, DC, USA, 2004. IEEE Computer Society.
- [BKP01] Mark Billingham, Hirkazu Kato, and Ivan Poupyrev. The MagicBook—Moving Seamlessly between Reality and Virtuality. *IEEE Computer Graphics and Applications*, 21(3):6–8, May/June 2001.
- [BN95] Michael Bajura and Ulrich Neumann. Dynamic Registration Correction in Augmented Reality Systems. In *IEEE VRAIS*, pages 189–196, 1995.
- [BNR02] Gregory Baratoff, Alexander Neubeck, and Holger Regenbrecht. Interactive Multi-Marker Calibration for Augmented Reality Applications. In *International Symposium on Mixed and Augmented Reality*

(*ISMAR'02*), page 107, Darmstadt, Germany, September 30 - October 01 2002. IEEE.

- [Bou] Jean-Yves Bouguet. Camera Calibration Toolbox. http://www.vision.caltech.edu/bouguetj/calib_doc/.
- [BPS02] Reinhold Behringer, Jun Park, and Venkataraman Sundareswaran. Model-Based Visual Tracking for Outdoor Augmented Reality Applications. In *International Symposium on Mixed and Augmented Reality (ISMAR'02)*, page 277, Darmstadt, Germany, September 30 - October 01 2002. IEEE.
- [Bro] Christopher M. Brown. University of Rochester Computer Science. http://www.cs.rochester.edu/u/brown/home_orig.html.
- [Can86] John Canny. A Computational Approach to Edge Detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 8(6):679–698, November 1986.
- [CCP02] Kar Wee Chia, Adrian David Cheok, and Simon J.D. Prince. Online 6 DOF Augmented Reality Registration from Natural Features. In *International Symposium on Mixed and Augmented Reality (ISMAR'02)*, page 305, Darmstadt, Germany, September 30 - October 01 2002. IEEE.
- [CHR04] Ozan Cakmakci, Yonggang Ha, and Jannick P. Rolland. A Compact Optical See-Through Head-Worn Display with Occlusion Support. In *ISMAR '04: Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)*, pages 16–25, Washington, DC, USA, 2004. IEEE Computer Society.
- [CJ02] G. Carneiro and Allan D. Jepson. Phase-Based Local Features. In *ECCV '02: Proceedings of the 7th European Conference on Computer Vision-Part I*, pages 282–296, London, UK, 2002. Springer-Verlag.
- [CJ03] G. Carneiro and A. D. Jepson. Multi-scale Phase-based Local Features. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '03)*, volume 1, pages I–736–I–743 vol.1, 2003.

- [CJK⁺92] Artur V. Cideciyan, Samuel G. Jacobson, Colin M. Kemp, Robert W. Knighton, and Joachim H. Nagel. Registration of High-Resolution Images of the Retina. *SPIE: Medical Imaging VI*, 1652:310–322, 1992.
- [CM92] Thomas P. Caudell and David W. Mizell. Augmented reality: An application of heads-up display technology to manual manufacturing processes. In *Proceedings of Hawaii International Conference on System Sciences*, pages 659–669, January 1992.
- [CM03] Haifeng Chen and Peter Meer. Robust Regression with Projection Based M-estimators. In *ICCV '03: Proceedings of the Ninth IEEE International Conference on Computer Vision*, page 878, Washington, DC, USA, 2003. IEEE Computer Society.
- [CM06] Cristina Cabani and W. James MacLean. A Proposed Pipelined-Architecture for FPGA-Based Affine-Invariant Feature Detectors. In *CVPRW '06: Proceedings of the 2006 Conference on Computer Vision and Pattern Recognition Workshop*, page 121, Washington, DC, USA, 2006. IEEE Computer Society.
- [Cor] Myvu Corporation. Myvu. <http://www.myvu.com>.
- [CP83] James L Crowley and Alice C. Parker. Representation for Shape based on Peaks and Ridges in the Difference of Low-pass Transform. Technical Report CMU-RI-TR-83-04, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, May 1983.
- [CPVG01] Kurt Cornelis, Marc Pollefeys, Maarten Vergauwen, and Luc Van Gool. Augmented Reality using Uncalibrated Video Sequences. *Proc. Second Workshop on Structure from Multiple Images of Large Scale Environments*, 2010:144–160, 2000 2001.
- [DCK04] A.J. Davison, Y. González Cid, and N. Kita. Real-Time 3D SLAM with Wide-Angle Vision. In *Proc. IFAC Symposium on Intelligent Autonomous Vehicles, Lisbon*, July 2004.
- [DD98] M. Daum and G. Dudek. On 3-D Surface Reconstruction Using Shape from Shadows. In *CVPR '98: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, page 461, Washington, DC, USA, 1998. IEEE Computer Society.

- [DMM03] Andrew J. Davison, Walterio W. Mayol, and David W. Murray. Real-Time Localisation and Mapping with Wearable Active Vision. In *ISMAR '03: Proceedings of the The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality*, page 18, Washington, DC, USA, 2003. IEEE Computer Society.
- [ECR92] B. Espiau, F. Chaumette, and P. Rives. A New Approach to Visual Servoing in Robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, June 1992.
- [EHH05] Jochen Ehnes, Koichi Hirota, and Michitaka Hirose. Projected Augmentation II - A Scalable Architecture for Multi Projector Based AR-Systems Based on 'Projected Applications'. In *ISMAR '05: Proceedings of the Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 190–191, Washington, DC, USA, 2005. IEEE Computer Society.
- [FANH04] Eric Foxlin, Yury Altshuler, Leonid Naimark, and Mike Harrington. FlightTracker: A Novel Optical/Inertial Tracker for Cockpit Enhanced Vision. In *ISMAR '04: Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)*, pages 212–221, Washington, DC, USA, 2004. IEEE Computer Society.
- [FB81] Martin A. Fischler and Robert C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Communications of the ACM (CACM)*, 24(6):381–395, 1981.
- [FBE⁺02] Michael Figl, Wolfgang Birkfellner, Christopher Ede, Johann Hummel, Rudolf Hanel, Franz Watzinger, Felix Wanschitz, Rolf Ewers, and Helmar Bergmann. The Control Unit for a Head Mounted Operating Microscope Used for Augmented Reality Visualization in Computer Aided Sugery. In *ISMAR '02: Proceedings of the International Symposium on Mixed and Augmented Reality (ISMAR'02)*, page 69, Washington, DC, USA, 2002. IEEE Computer Society.

- [Fia04] M Fiala. ARTag Revision 1. A Fiducial Marker System Using Digital Techniques. Technical Report NRC 47419, NRC-CNRC, November 24 2004.
- [FMH97] Steven Feiner, Blair MacIntyre, and Tobias Hllerer. A touring machine: Prototyping 3D mobile augmented reality systems for exploring the urban environment. In *International Symposium on Wearable Computing*, pages 74–81, Cambridge, Massachusetts USA, October 13 - 14 1997.
- [FMS93] Steven Feiner, Blair Macintyre, and Dore Seligmann. Knowledge-based augmented reality. *Commun. ACM*, 36(7):53–62, 1993.
- [FN03] Eric Foxlin and Leonid Naimark. VIS-Tracker: A Wearable Vision-Inertial Self-Tracker. In *VR '03: Proceedings of the IEEE Virtual Reality 2003*, page 199, Washington, DC, USA, 2003. IEEE Computer Society.
- [FSVC07] Robert Fisher, Jose Santos-Victor, and James Crowley. CAVIAR: Context Aware Vision using Image-based Active Recognition, 2007. <http://homepages.inf.ed.ac.uk/rbf/CAVIAR/>.
- [FV02] Morten Fjeld and Benedikt M. Voegtli. Augmented Chemistry: An Interactive Educational Workbench. In *International Symposium on Mixed and Augmented Reality (ISMAR'02)*, page 259, Darmstadt, Germany, September 30 - October 01 2002. IEEE. POSTER.
- [FZB02] H. Foroosh, J. Zerubia, and M. Berthod. Extension of phase correlation to subpixel registration. *IEEE Trans. on Image Processing*, 11(3):188 – 200, March 2002.
- [GBB⁺02] Gaile Gordon, Mark Billingham, Melanie Bell, John Woodfill, Bill Kowalik, Alex Erendi, and Janet Tilander. The Use of Dense Stereo Range Data in Augmented Reality. In *International Symposium on Mixed and Augmented Reality (ISMAR'02)*, page 14, Darmstadt, Germany, September 30 - October 01 2002. IEEE.
- [GFM03] Jürgen Gausemeier, Jürgen Fründ, and Carsten Matysczok. AR-based Configuration and Information Retrieval of Household Appliances on

Mobile Devices. In *CHINZ*, pages 93–98, Dunedin, New Zealand, 3–4 July 2003.

- [GLPW⁺96] W. E. L. Grimson, T. Lozano-Pérez, W. M. Wells, G. J. Ettinger, S. J. White, and R. Kikinis. An Automatic Registration Method for Frameless Stereotaxy, Image Guided Surgery, and Enhanced Reality Visualization. In *Proceedings of the Computer Society Conference on Computer Vision and Pattern Recognition*, pages 430–436, 1996.
- [GMP⁺05] Maribeth Gandy, Blair MacIntyre, Peter Presti, Steven Dow, Jay Bolter, Brandon Yarbrough, and Nigel O’Rear. AR Karaoke: Acting in Your Favorite Scenes. In *ISMAR ’05: Proceedings of the Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 114–117, Washington, DC, USA, 2005. IEEE Computer Society.
- [GN07] K. Garg and S.K. Nayar. Vision and Rain. *International Journal on Computer Vision*, pages 1–25, Feb 2007.
- [GS97] Theo Gevers and Arnold W. M. Smeulders. Color Based Object Recognition. In *ICIAP ’97: Proceedings of the 9th International Conference on Image Analysis and Processing-Volume I*, pages 319–326, London, UK, 1997. Springer-Verlag.
- [GSB05] Javier-Flavio Vigueras Gomez, Gilles Simon, and Marie-Odile Berger. Calibration Errors in Augmented Reality: A Practical Study. In *ISMAR ’05: Proceedings of the Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 154–163, Washington, DC, USA, 2005. IEEE Computer Society.
- [HA00] Bruce Hoff and Ronald Azuma. Autocalibration of an Electronic Compass in an Outdoor Augmented Reality System. In *International Symposium on Augmented Reality (ISAR 2000)*, pages 159–164, Munich, Germany, 5–6 October 2000 2000.
- [HBO05] Anders Henrysson, Mark Billinghurst, and Mark Ollila. Face to Face Collaborative AR on Mobile Phones. In *ISMAR ’05: Proceedings of the Fourth IEEE and ACM International Symposium on Mixed and*

Augmented Reality, pages 80–89, Washington, DC, USA, 2005. IEEE Computer Society.

- [HFT⁺99] Tobias Hllerer, Steven Feiner, Tachio Terauchi, Gus Rashid, and Drexel Hallaway. Exploring MARS: Developing indoor and outdoor user interfaces to a mobile augmented reality system. *Computers and Graphics*, 23(6):779–785, 1999.
- [HHTF01] Tobias Hllerer, Drexel Hallaway, Navdeep Tinna, and Steven Feiner. Steps toward accommodating variable position tracking accuracy in a mobile augmented reality system. *2nd Int. Worksh. on Artificial Intelligence in Mobile Systems (AIMS '01)*, pages 31–37, 2001.
- [Hol95] Richard Holloway. *Registration errors in augmented reality systems*. Phd dissertation, University of North Carolina at Chapel Hill, Chapel Hill, 1995.
- [HS88] Chris Harris and Mike Stephens. A Combined Corner and Edge Detector. In *4th Alvey Vision Conference*, pages 147–151, 1988.
- [HS95] Richard Hartley and Peter Sturm. Triangulation. *Computer Vision and Image Understanding*, 68(2):146–157, 1995. R. Hartley and P. Sturm: Triangulation Computer Vision and Image Understanding, 68(2):146–157, 1997.
- [HS98] J.H. van Hateren and A. van der Schaaf. Independent Component Filters Of Natural Images Compared With Simple Cells In Primary Visual Cortex. In *Biological Science*, 1998.
- [HSSB96] Michael D. Heath, Sudeep Sarkar, Thomas Sanocki, and Kevin W. Bowyer. Comparison of Edge Detectors: A Methodology and Initial Study. *Computer Vision and Pattern Recognition*, 21:143–148, 1996.
- [Hum] Human Interface Technology New Zealand. Outdoor AR. <http://www.hitlabnz.org>.
- [HZ04] Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.
- [Int] Intersense. Inertiacle 2. <http://www.intersense.com/>.

- [KAI05] Yosi Keller, Amir Averbuch, and Moshe Israeli. Pseudopolar-Based Estimation of Large Translations, Rotations, and Scalings in Images. *IEEE Transactions on Image Processing*, 14(1):12–22, January 2005.
- [Kal60] Rudolph E. Kalman. A New Approach in Linear Filtering and Prediction Problems. *Transactions of the American Society of Mechanical Engineers, Journal of Basic Engineering*, pages 35–45, March 1960.
- [KB99] Hirokazu Kato and Mark Billinghurst. Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System. In *2nd IEEE and ACM International Workshop on Augmented Reality (IWAR)*, page 85, San Francisco, California, October 20 - 21 1999.
- [KBCW03] Kiyoshi Kiyokawa, Mark Billinghurst, Bruce Campbell, and Eric Woods. An Occlusion-Capable Optical See-through Head Mount Display for Supporting Co-located Collaboration. In *ISMAR '03: Proceedings of the The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality*, page 133, Washington, DC, USA, 2003. IEEE Computer Society.
- [KD03] Georg Klein and Tom Drummond. Robust Visual Tracking for Non-Instrumented Augmented Reality. In *ISMAR '03: Proceedings of the The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality*, page 113, Washington, DC, USA, 2003. IEEE Computer Society.
- [KD04] Georg Klein and Tom Drummond. Sensor fusion and occlusion refinement for tablet-based ar. In *ISMAR '04: Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)*, pages 38–47, Washington, DC, USA, 2004. IEEE Computer Society.
- [KDB⁺02] Gudrun Klinker, Allen H. Dutoit, Martin Bauer, Johannes Bayer, Vinko Novak, and Dietmar Matzke. Fata Morgana — A Presentation System for Product Design. In *International Symposium on Mixed*

and Augmented Reality (ISMAR'02), page 76, Darmstadt, Germany, September 30 - October 01 2002. IEEE.

- [KFTY00] Masayuki Kanbara, Hirofumi Fujii, Haruo Takemura, and Naokazu Yokoya. A Stereo Vision-based Augmented Reality System with an Inertial Sensor. In *IEEE and ACM International Symposium on Augmented Reality (ISAR2000)*, pages 97–100, 2000.
- [KH75] C. D. Kuglin and D. C. Hines. The phase correlation image alignment method. In *IEEE 1975 Conference on Cybernetics and Society*, pages 163–165, September 1975.
- [KjYS03] Hansung Kim, Seung jun Yang, and Kwanghoon Sohn. 3D Reconstruction of Stereo Images for Interaction between Real and Virtual Worlds. In *ISMAR '03: Proceedings of the The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality*, page 169, Washington, DC, USA, 2003. IEEE Computer Society.
- [KM07] Fredrik Kristensen and W. James MacLean. Real-Time Extraction of Maximally Stable Extremal Regions on an FPGA. In *ISCAS*, pages 165–168. IEEE, 2007.
- [Kov] P. D. Kovesi. MATLAB and Octave Functions for Computer Vision and Image Processing. School of Computer Science & Software Engineering, The University of Western Australia. Available from: <http://www.csse.uwa.edu.au/~pk/research/matlabfns/>.
- [KPT05] Gary R. King, Wayne Piekarski, and Bruce H. Thomas. ARVino - Outdoor Augmented Reality Visualisation of Viticulture GIS Data. In *ISMAR '05: Proceedings of the Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 52–55, Washington, DC, USA, 2005. IEEE Computer Society.
- [KS04] Y. Ke and R. Sukthankar. PCA-SIFT: A more distinctive representation for local image descriptors, 2004.
- [KSV98] Takeo Kanade, Hideo Saito, and Sundar Vedula. The 3D Room: Digitizing Time-Varying 3D Events by Synchronized Multiple Video Streams. Technical Report CMU-RI-TR-98-34, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, December 1998.

- [KY02] Masayuki Kanbara and Naokazu Yokoya. Geometric and Photometric Registration for Real-Time Augmented Reality. In *International Symposium on Mixed and Augmented Reality (ISMAR'02)*, page 279, Darmstadt, Germany, September 30 - October 01 2002. IEEE.
- [KYT01] Masayuki Kanbara, Naokazu Yokoya, and Haruo Takemura. A Stereo Vision-Based Augmented Reality System with Marker and Natural Feature Tracking. In *VSM '01: Proceedings of the Seventh International Conference on Virtual Systems and Multimedia (VSM '01)*, page 455, Washington, DC, USA, 2001. IEEE Computer Society.
- [KZB04] Timor Kadir, Andrew Zisserman, and Michael Brady. An affine invariant salient region detector. In *ECCV*, pages 404–416, 2004.
- [LB05] Martin D. Levine and Jisnu Bhattacharyya. Removing shadows. *Pattern Recogn. Lett.*, 26(3):251–265, 2005.
- [LBC04] Julian Looser, Mark Billingham, and Andy Cockburn. Through the looking glass: the use of lenses as an interface tool for Augmented Reality interfaces. In *GRAPHITE '04: Proceedings of the 2nd international conference on Computer graphics and interactive techniques in Australasia and South East Asia*, pages 204–211, New York, NY, USA, 2004. ACM Press.
- [LLR04] Jia Li, Robert Laganier, and Gerhard Roth. Online Estimation of Trifocal Tensors for Augmenting Live Video. In *ISMAR '04: Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)*, pages 182–190, Washington, DC, USA, 2004. IEEE Computer Society.
- [Low04] David G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.
- [LRJ⁺02] M. A. Livingston, L. Rosenblum, S. Julier, D. Brown, Y. Baillot, and J. and et al Swan. An Augmented Reality System for military operations in urban terrain. In *Proceedings of the Interservice/Industry Training, Simulation and Education Conference*, pages 868–875, 2002.

- [LVTF03] Vincent Lepetit, Luca Vacchetti, Daniel Thalmann, and Pascal Fua. Fully automated stable registration for augmented reality applications. In *International Symposium on Mixed and Augmented Reality*, Tokyo, Japan, October 2003.
- [LYN02] Jong Weon Lee, Suyu You, and Ulrich Neumann. Tracking with Omni-Directional Vision for Outdoor AR Systems. In *International Symposium on Mixed and Augmented Reality (ISMAR'02)*, page 47, Darmstadt, Germany, September 30 - October 01 2002. IEEE.
- [Mat01] Bogdan C Matei. *Heteroscedastic errors-in-variables models in computer vision*. PhD thesis, Center for Advanced Information Processing, Rutgers University, 2001. Director-Peter Meer.
- [McH] Sean McHugh. Cambridge In Colour. <http://www.cambridgeincolour.com>.
- [MCMP02] Jir Matas, Ondrej Chum, Urban Martin, and Tom Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In Paul L. Rosin and David Marshall, editors, *British Machine Vision Conference*, volume 1, pages 384–393, London, UK, September 2002.
- [MHU⁺03] Nobuyuki Matsushita, Daisuke Hihara, Teruyuki Ushiro, Shinichi Yoshimura, Jun Rekimoto, and Yoshikazu Yamamoto. ID CAM: A Smart Camera for Scene Capturing and ID Recognition. In *ISMAR '03: Proceedings of the The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality*, page 227, Washington, DC, USA, 2003. IEEE Computer Society.
- [MK94] Paul Milgram and F. Kishino. A Taxonomy of Mixed Reality Visual Displays. *IEICE Transactions on Information Systems*, E77-D(12):1321–1329, 1994.
- [MM00] B.C. Matei and P. Meer. A General Method for Errors-in-Variables Problems in Computer Vision. In *CVPR00*, pages II: 18–25, 2000.
- [Mor81] H. Moravec. Rover Visual Obstacle Avoidance. In *International Joint Conference on Artificial Intelligence*, pages 785–790, 1981.

- [MS01] Krystian Mikolajczyk and Cordelia Schmid. Indexing Based on Scale Invariant Interest Points. In *International Conference on Computer Vision*, pages 525–531, Vancouver, Canada, July 2001.
- [MS03] Krystian Mikolajczyk and Cordelia Schmid. A Performance Evaluation of Local Descriptors. In *Conference on Computer Vision and Pattern Recognition (CVPR '03)*, volume 2, 2003.
- [MS04] Krystian Mikolajczyk and Cordelia Schmid. Scale and Affine invariant interest point detectors. *IJCV*, 1(60):63–86, 2004.
- [MS05] Krystian Mikolajczyk and Cordelia Schmid. A Performance Evaluation of Local Descriptors. *IEEE Transactions on Pattern Analysis & Machine Intelligence*, 27(10):1615–1630, 2005.
- [MS06] Yoichi Motokawa and Hideo Saito. Support system for guitar playing using augmented reality display. In *ISMAR '06: Proceedings of the Fifth IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 243–244, Santa Barbara, CA, USA, Oct 2006. IEEE Computer Society.
- [MSKS03] Yi Ma, Stefano Soatto, Jana Kosecka, and S. Shankar Sastry. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer-Verlag, 2003.
- [MTS⁺05] Krystian Mikolajczyk, Tinne Tuytelaars, Cordelia Schmid, Andrew Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision*, 65(1/2):43–72, 2005.
- [Nav03] Nassir Navab. Industrial Augmented Reality(IAR): Challenges in Design and Commercialization of Killer Apps. In *ISMAR '03: Proceedings of the The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality*, page 2, Washington, DC, USA, 2003. IEEE Computer Society.
- [NF02] Leonid Naimark and Eric Foxlin. Circular Data Matrix Fiducial System and Robust Image Processing for a Wearable Vision-Inertial Self-Tracker. In *International Symposium on Mixed and Augmented*

Reality (ISMAR'02), page 27, Darmstadt, Germany, September 30 - October 01 2002. IEEE.

- [NF05] Leonid Naimark and Eric Foxlin. Encoded LED System for Optical Trackers. In *ISMAR '05: Proceedings of the Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 150–153, Washington, DC, USA, 2005. IEEE Computer Society.
- [NN02] S.G. Narasimhan and S.K. Nayar. Vision and the Atmosphere. *International Journal on Computer Vision*, 48(3):233–254, Jul 2002.
- [NN03a] S.G. Narasimhan and S.K. Nayar. Contrast Restoration of Weather Degraded Images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(6):713–724, Jun 2003.
- [NN03b] S.G. Narasimhan and S.K. Nayar. Interactive (De)weathering of an Image using Physical Models. In *ICCV Workshop on Color and Photometric Methods in Computer Vision (CPMCV)*, Oct 2003.
- [NWN02] Srinivasa G. Narasimhan, Chi Wang, and Shree K. Nayar. All the Images of an Outdoor Scene. In *ECCV*, pages 148–162, 2002.
- [NY98] Ulrich Neumann and Suya You. Integration of Region Tracking and Optical Flow for Image Motion Estimation. In *IEEE International Conference on Image Processing*, 1998.
- [NYC⁺99] Ulrich Neumann, Suya You, Youngkwan Cho, Jongweon Lee, and Jun Park. Augmented reality tracking in natural environments. In *International Symposium on Mixed Realities*, Tokyo, Japan, 1999.
- [OLG⁺05] Alex Olwal, Christoffer Lindfors, Jonny Gustafsson, Torsten Kjellberg, and Lars Mattsson. Astor: An autostereoscopic optical see-through augmented reality system. In *ISMAR '05: Proceedings of the Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 24–27, Washington, DC, USA, 2005. IEEE Computer Society.
- [OS98] John P. Oakley and Brenda L. Satherley. Improving image quality in poor visibility conditions using a physical model for contrast

- degradation. *IEEE Transactions on Image Processing*, 7(2):167–179, 1998.
- [Per] Pietro Perona. Computational Vision Archive - Mt. Wilson Images. <http://www.vision.caltech.edu/html-files/archive.html>.
- [PJN99] Jun Park, Bolan Jiang, and Ulrich Neumann. Vision-based Pose Computation: Robust and Accurate Augmented Reality Tracking. In *Proceedings of the IEEE International Workshop on Augmented Reality (IWAR'99)*, pages 3–12, 1999.
- [PLF05] Julien Pilet, Vincent Lepetit, and Pascal Fua. Augmenting Deformable Objects in Real-Time. In *ISMAR '05: Proceedings of the Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 134–137, Washington, DC, USA, 2005. IEEE Computer Society.
- [PMT03] Andrea Prati, Ivana Mikić, Mohan Trivedi, and Rita Cucchiara. Detecting moving shadows: Formulation, algorithms and evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7), July 2003.
- [PT01] Wayne Piekarski and Bruce Thomas. Tinmith-Metro: New Outdoor Techniques for Creating City Models with an Augmented Reality Wearable Computer. In *5th International Symposium on Wearable Computers*, pages 31–38, Zurich, Switzerland, October 2001.
- [PT02] Wayne Piekarski and Bruce H. Thomas. Bread Crumbs: A Technique for Modelling Large Outdoor Ground Features. In *International Symposium on Mixed and Augmented Reality (ISMAR'02)*, page 269, Darmstadt, Germany, September 30 - October 01 2002. IEEE. POSTER.
- [PVG⁺04] Marc Pollefeys, Luc Van Gool, Maarten Vergauwen, Frank Verbiest, Kurt Cornelis, Jan Tops, and Reinhard Koch. Visual Modeling with a Hand-Held Camera. *International Journal of Computer Vision*, V59(3):207–232, 2004.
- [PW03] Wouter Pasman and Charles Woodward. Implementation of an Augmented Reality System on a PDA. In *ISMAR '03: Proceedings*

of the The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality, page 276, Washington, DC, USA, 2003. IEEE Computer Society.

- [RC96] B.S Reddy and B. N. Chatterji. An FFT-based Technique for Translation, Rotation, and Scale-invariant Image Registration. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 5(8):1266–1271, 1996.
- [Rob] Robotic Systems Lab, Australian National University. Active Vision. http://axiom.anu.edu.au/~rsl/rsl_active.html.
- [SA97] Kristian T Simsarian and Karl-Petter Akesson. Windows on the World: An example of Augmented Virtuality. In *Montpellier 1997: Interfaces: Man-Machine Interaction*, 1997.
- [SB98a] Gilles Simon and Marie-Odile Berger. A Two-stage Robust Statistical Method for Temporal Registration from Features of Various Type. In *Proceedings of the 6th International Conference on Computer Vision*, pages 261–266, Bombay, India, January 1998.
- [SB98b] Ventakaraman Sundareswaran and Reinhold Behringer. Visual Servoing-based Augmented Reality. In *First IEEE International Workshop on Augmented Reality (IWAR'98)*, San Francisco, CA, USA, November 1998.
- [SB02] Gilles Simon and Marie-Odile Berger. Reconstructing While Registering: A Novel Approach for Markerless Augmented Reality. In *International Symposium on Mixed and Augmented Reality (ISMAR'02)*, page 285, Darmstadt, Germany, September 30 - October 01 2002. IEEE.
- [SBB03] Terence Sim, Simon Baker, and Maan Bsath. The CMU Pose, Illumination, and Expression Database. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(12):1615 – 1618, December 2003.
- [SBJ05] S. Se, T. Barfoot, and P. Jasiobedzki. Visual Motion Estimation and Terrain Modeling for Planetary Rovers. In *'i-SAIRAS 2005' - The*

8th International Symposium on Artificial Intelligence, Robotics and Automation in Space, volume 603 of *ESA Special Publication*, August 2005.

- [SBN05] Tobias Sielhorst, Tobias Blum, and Nassir Navab. Synchronizing 3D Movements for Quantitative Comparison and Simultaneous Visualization of Actions. In *ISMAR '05: Proceedings of the Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 38–47, Washington, DC, USA, 2005. IEEE Computer Society.
- [SCE01] E. Salvador, A. Cavallaro, and T. Ebrahimi. Shadow identification and classification using invariant color models. In *ICASSP '01: Proceedings of the Acoustics, Speech, and Signal Processing, 2001. on IEEE International Conference*, pages 1545–1548, Washington, DC, USA, 2001. IEEE Computer Society.
- [SESK05] B. Streckel, J.-F. Evers-Senne, and R. Koch. Lens Model Selection for a Markerless AR Tracking System. In *International Symposium for Mixed and Augmented Reality*, Vienna, Austria, October 2005. Uses a fish-eye lens.
- [SFZ00] Gilles Simon, Andrew W. Fitzgibbon, and Andrew Zisserman. Markerless Tracking using Planar Structures in the Scene. In *International Symposium on Augmented Reality (ISAR)*, 2000.
- [SHC⁺96] Andrei State, Gentaro Hirota, David T. Chen, William F. Garrett, and Mark A. Livingston. Superior Augmented Reality Registration by Integrating Landmark Tracking and Magnetic Tracking. In *Proceedings of SIGGRAPH '96*, volume 30, pages 429–438, 1996.
- [Sie03] Siemens. Virtual mosquito hunt: An augmented reality game, March 2003. http://w4.siemens.de/en2/html/press/newsdesk_archive/2003/foe03111.html.
- [SK92] Steven A. Shafer and Takeo Kanade. Using shadows in finding surface orientations. pages 346–377, 1992.
- [SKT03] Natsuki Sugano, Hirokazu Kato, and Keihachiro Tachibana. The Effects of Shadow Representation of Virtual Objects in Augmented

- Reality. In *ISMAR '03: Proceedings of the The 2nd IEEE and ACM International Symposium on Mixed and Augmented Reality*, page 76, Washington, DC, USA, 2003. IEEE Computer Society.
- [SL04] Iryna Skrypnik and David G. Lowe. Scene Modelling, Recognition and Tracking with Invariant Image Features. In *ISMAR'04*, pages 110–119, Washington D.C, U.S.A, 2-5 November 2004.
- [SLG⁺96] Andrei State, Mark A. Livingston, William F. Garrett, Gentaro Hirota, Mary C. Whitton, Etta D. Pisano, and Henry Fuchs. Technologies for Augmented Reality systems: realizing ultrasound-guided needle biopsies. In *SIGGRAPH '96: Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 439–446, New York, NY, USA, 1996. ACM Press.
- [SMB98] Cordelia Schmid, Roger Mohrland, and Christian Bauckhage. Comparing and Evaluating Interest Points. In *ICCV '98: Proceedings of the Sixth International Conference on Computer Vision*, page 230, Washington, DC, USA, 1998. IEEE Computer Society.
- [SMG05] Raghav Subbarao, Peter Meer, and Yakup Genc. A Balanced Approach to 3D Tracking from Image Streams. In *International Symposium for Mixed and Augmented Reality*, pages 70–78, Vienna, Austria, October 2005.
- [SNN01] Y.Y. Schechner, S.G. Narasimhan, and S.K. Nayar. Instant Dehazing of Images using Polarization. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume I, pages 325–332, Dec 2001.
- [ST94] Jianbo Shi and Carlo Tomasi. Good Features to Track. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR'94)*, pages 593–600, Seattle, Washington, June 1994.
- [SW05] Dieter Schmalstieg and Daniel Wagner. A Handheld Augmented Reality Museum Guide. In *Proceedings of IADIS International Conference on Mobile Learning*, 2005.
- [SW06] Sanni Siltanen and Charles Woodward. Augmented interiors with digital camera images. In *AUIC '06: Proceedings of the 7th*

- Australasian User interface conference*, pages 33–36, Darlinghurst, Australia, Australia, 2006. Australian Computer Society, Inc.
- [TG99] Tinne Tuytelaars and Luc Van Gool. Content-based image retrieval based on local affinity invariant regions. In *Third International Conference on Visual Information and Information Systems*, pages 493–500, 1999.
- [TG00] Tinne Tuytelaars and Luc Van Gool. Wide Baseline Stereo Matching Based on Local, Affinely Invariant Regions. In *BMVC*, pages 412–425, 2000.
- [TG04] Tinne Tuytelaars and Luc Van Gool. Matching Widely Separated Views Based on Affine Invariant Regions. *Int. J. Comput. Vision*, 59(1):61–85, 2004.
- [TGW⁺95] Mihran Tuceryan, Douglas S. Greer, Ross T. Whitaker, David Breen, Chris Crampton, Eric Rose, and Klaus H. Ahlers. Calibration Requirements and Procedures for Augmented Reality. Technical Report ECRC-95-14, European Computer-Industry Research Centre GmbH, Munich, Germany, September 1995.
- [THG02] Petra Tschirner, Bernd Hillers, and Axel Grser. A Concept for the Application of Augmented Reality in Manual Gas Metal Arc Welding. In *International Symposium on Mixed and Augmented Reality (ISMAR'02)*, page 257, Darmstadt, Germany, September 30 - October 01, 2002 2002. IEEE. POSTER.
- [TK91] Carlo Tomasi and Takeo Kanade. Detection and tracking of point features. Technical Report CMU-CS-91-132, Carnegie Mellon University, April 1991.
- [TMHF99] Bill Triggs, Philip McLauchlan, Richard Hartley, and Andrew Fitzgibbon. Bundle Adjustment - A Modern Synthesis. In *Proceedings of the international workshop on vision algorithms: Theory and Practice*, pages 298–372, September 1999.
- [TOBM02] Arthur Tang, Charles Owen, Frank Biocca, and Weimin Mou. Experimental Evaluation of Augmented Reality in Object Assembly

- Task. In *International Symposium on Mixed and Augmented Reality (ISMAR'02)*, page 265, Darmstadt, Germany, September 30 - October 01 2002. IEEE.
- [Tow] Gregg Townsend. The arizona webcam. <http://www.cs.arizona.edu/camera/>.
- [TP02] Bruce Thomas and Wayne Piekarski. ARQuake: The Outdoor Augmented Reality Gaming System. *Communications of the ACM*, 45(1):36–38, 2002.
- [TSLB05] Marcus Tonniss, Christian Sandor, Christian Lange, and Heiner Bubb. Experimental Evaluation of an Augmented Reality Visualization for Directing a Car Driver’s Attention. In *ISMAR '05: Proceedings of the Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 56–59, Washington, DC, USA, 2005. IEEE Computer Society.
- [VIK⁺02] Vassilios Vlahakis, Nikolaos Ioannidis, John Karigiannis, Manolis Tsotros, Michael Gounaris, Didier Stricker, Tim Gleue, Patrick Daehne, and Lus Almeida. Archeoguide: An Augmented Reality Guide for Archaeological Sites. *IEEE Computer Graphics and Applications*, 22(5):52–60, September/October 2002 2002.
- [VKS02] Sebastian Vogt, Ali Khamene, Frank Sauer, and Heinrich Niemann. Single Camera Tracking of Marker Clusters: Multiparameter Cluster Optimization and Experimental Verification. In *International Symposium on Mixed and Augmented Reality (ISMAR'02)*, page 127, Darmstadt, Germany, September 30 - October 01, 2002 2002. IEEE.
- [VLF04] Luca Vacchetti, Vincent Lepetit, and Pascal Fua. Combining Edge and Texture Information for Real-Time Accurate 3D Camera Tracking. In *ISMAR '04: Proceedings of the Third IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'04)*, pages 48–57, Washington, DC, USA, 2004. IEEE Computer Society.
- [WBV⁺01] Greg Welch, Gary Bishop, Leandra Vicci, Stephen Brumback, and Kurtis Keller. High-Performance Wide-Area Optical Tracking The

- HiBall Tracking System. *Presence: Teleoperators and Virtual Environments*, 10(1), February 2001.
- [Wel95] Gregory F Welch. Hybrid Self-Tracker: An Inertial/Optical Hybrid Three-Dimensional Tracking System. Dept. of Computer Science Technical Report TR95-048, UNC Chapel Hill, 1995.
- [Wel96] Greg Welch. *SCAAT: Incremental Tracking with Incomplete Information*. Phd, University of North Carolina at Chapel Hill, Chapel Hill, 1996.
- [WF02] Greg Welch and Eric Foxlin. Motion Tracking: No Silver Bullet, but a Respectable Arsenal. *IEEE Computer Graphics and Applications*, 22(6 (Special issue on Tracking)):24–38, November/December 2002.
- [WVS05] Harald Wuest, Florent Vial, and Didier Stricker. Adaptive Line Tracking with Multiple Hypotheses for Augmented Reality. In *ISMAR '05: Proceedings of the Fourth IEEE and ACM International Symposium on Mixed and Augmented Reality*, pages 62–69, Washington, DC, USA, 2005. IEEE Computer Society.
- [YN01] Suyu You and Ulrich Neumann. Fusion of Vision and GyroTracking for Robust Augmented Reality Registration. In *IEEE Conference on Virtual Reality*, pages 71–78, 2001.
- [YNA99] Suyu You, Ulrich Neumann, and Ronald Azuma. Hybrid Inertial and Vision Tracking for Augmented Reality Registration. In *IEEE VR '99, Proceedings of IEEE VR '99*, pages 260–, Houston, Texas, March 13-17 1999.
- [YP05] Ruigang Yang and Marc Pollefeys. A versatile stereo implementation on commodity graphics hardware. *Real-Time Imaging*, 11(1):7–18, 2005.
- [ZDFL94] Zhengyou Zhang, Rachid Deriche, Olivier Faugeras, and Quang-Tuan Luong. A Robust Technique for Matching Two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry. *Artificial Intelligence*, 78(1-2):87–119, 1994.

- [ZFN02] Xiang Zhang, Stephan Frons, and Nassir Navab. Visual Marker Detection and Decoding in AR Systems: A Comparative Study. In *International Symposium on Mixed and Augmented Reality (ISMAR'02)*, page 97, Darmstadt, Germany, September 30 - October 01, 2002 2002. IEEE.
- [ZT98] Djemel Ziou and Salvatore Tabbone. Edge Detection Techniques-An Overview. *International Journal of Pattern Recognition and Image Analysis*, 8(4):537–559, 1998.