PhD Thesis in Computer Vision

February 2016

Wide-Baseline Image Change Detection

Zygmunt Jones

Department of Electrical and Electronic Engineering, Imperial College, Exhibition Road, London, UK

z.jones11@imperial.ac.uk

Supervisors:

Mike Brookes

Pier-Luigi Dragotti
© The copyright of this thesis rests with the author and is made available under a Creative Commons Attribution Non-Commercial No Derivatives licence. Researchers are free to copy, distribute or transmit the thesis on the condition that they attribute it, that they do not use it for commercial purposes and that they do not alter, transform or build upon it. For any reuse or redistribution, researchers must make clear to others the licence terms of this work.

All work contained in this thesis is the work of the author unless otherwise referenced.
Abstract

Growth in the prevalence of cameras has resulted in larger amounts of available image data. This has resulted in demand for automated methods of analysing this data. One key area of demand is automated change detection, the automated detection of changes in a scene, as recorded by a reference and sample image. Established methods of change detection tend to rely on the reference and sample image being captured from the same position, but much of the available data does not fit this criteria. This thesis presents novel approaches to key challenges in wide-baseline cases involving differences in viewing angle of up to 30°, including registration and the image region matching that are robust to the inherent registration errors. The developed algorithms are then combined into an end-to-end system.

This thesis presents novel registration approaches including the use of a Delaunay triangulation mask that enables registration of each component triangle, a method of finding local planes in scenes by clustering matched feature points, the use of edge detection to register the edges of objects, and a method for registering planes that are orthogonal to a defined image plane and to the camera line. These techniques allow for the registration of complex 3D scenes with viewing angles of up to 30°. The density of the available correspondences obtained using feature points is a key limiting factor in these methods and so ASIFT, a extension to the SIFT feature point that improves performance at wide angles is also introduced. ASIFT is shown to have an order of magnitude increase in correctly matches feature point density at 30°.

Though robust to wide differences in viewing angle, these registration techniques do nonetheless introduce registration errors of up to a few dozen pixels. For this reason the dense SIFT and shifted dense SIFT image comparison algorithms which are robust to registration errors of a few dozen pixels are also developed. The development of these comparison methods includes an analysis of SIFT descriptor statistics and their correlation. Finally these techniques are combined to form an end-to-end change detection system which is evaluated on a number of test datasets.
# Contents

1 Introduction .............................................. 6
   1.1 Change detection ........................................ 6
   1.2 Motivation and applications .......................... 7
   1.3 Problem statement ..................................... 7
   1.4 Contributions .......................................... 9

2 Background Material and Literature Review .............. 11
   2.1 Projective Geometry ................................... 11
   2.2 Registration ............................................ 18
   2.3 Feature points .......................................... 20
   2.4 Edge detection and segmentation ..................... 27
   2.5 2D Change detection – Image based ................... 29
   2.6 Change detection – Feature and object classification based .... 34
   2.7 Similar applications ................................... 41
   2.8 Summary ................................................. 41

3 Project Approach ........................................ 43
   3.1 Addressing key challenges ............................ 43
   3.2 Evaluation .............................................. 45

4 Feature Matching ........................................ 50
   4.1 Affine Compensated SIFT ............................. 50
   4.2 Increasing Feature Point Density ..................... 62
   4.3 Feature points summary ................................ 68
5 Region Comparison

5.1 Probabilistic assessment of the SIFT descriptor .............................................. 71
5.2 Dense SIFT ........................................................................................................ 79
5.3 Shifted Dense SIFT using Markov random fields ............................................... 90
5.4 Markov random field change localisation ......................................................... 98
5.5 Visual texture based matching ........................................................................ 103
5.6 Summary ........................................................................................................... 109

6 Segmentation and registration

6.1 Delaunay triangulation and affine compensation ............................................. 111
6.2 Local Plane Clusters ........................................................................................ 114
6.3 Object Edge Hypothesis Testing ...................................................................... 118
6.4 Standing Object Matching ................................................................................ 125

7 End-to-end change detection system

7.1 Delaunay triangulation and dense SIFT matching ............................................. 137
7.2 OEHT and SDSIFT matching .......................................................................... 138
7.3 LPC and SDSIFT .............................................................................................. 142
7.4 Standing wall matching .................................................................................... 142
7.5 CDnet video change detection data .................................................................. 144

8 Conclusion

8.1 Summary of key contributions ......................................................................... 146
8.2 Further work .................................................................................................... 147

9 List of Symbols
### List of Abbreviations

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ASIFT</td>
<td>Affine Compensated Scale-Independent Feature Transform</td>
<td>56</td>
</tr>
<tr>
<td>BoW</td>
<td>Bag of Words</td>
<td>35</td>
</tr>
<tr>
<td>BRIEF</td>
<td>Binary Robust Independent Elementary Features</td>
<td>23</td>
</tr>
<tr>
<td>DoG</td>
<td>Difference of Gaussians</td>
<td>21</td>
</tr>
<tr>
<td>FAST</td>
<td>Features from Accelerated Segment Test</td>
<td>23</td>
</tr>
<tr>
<td>GLOH</td>
<td>Gradient location and orientation histogram</td>
<td>22</td>
</tr>
<tr>
<td>HOG</td>
<td>Histograms of Oriented Gradients</td>
<td>24</td>
</tr>
<tr>
<td>MSER</td>
<td>Maximally Stable Extremal Regions</td>
<td>25</td>
</tr>
<tr>
<td>OEHT</td>
<td>Object Edge Hypothesis Testing</td>
<td>118</td>
</tr>
<tr>
<td>ORB</td>
<td>Oriented FAST and Rotated BRIEF</td>
<td>23</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal Component Analysis</td>
<td>22</td>
</tr>
<tr>
<td>RANSAC</td>
<td>Random Sample Consensus</td>
<td>18</td>
</tr>
<tr>
<td>DSIFT</td>
<td>Dense SIFT</td>
<td>70</td>
</tr>
<tr>
<td>SDSIFT</td>
<td>Shifted DSIFT</td>
<td>70</td>
</tr>
<tr>
<td>SIFT</td>
<td>Scale-Independent Feature Transform</td>
<td>9</td>
</tr>
<tr>
<td>SURF</td>
<td>Speeded-Up Robust Features</td>
<td>22</td>
</tr>
<tr>
<td>U-SURF</td>
<td>Upright Speeded-Up Robust Features</td>
<td>22</td>
</tr>
</tbody>
</table>
Chapter 1

Introduction

1.1 Change detection

The aim of this thesis is to present novel techniques for the detection of changes within a scene using a single reference image and a single sample image. A reference image is used to record the baseline state of the scene which is then compared with a sample image taken at a later time in order to detect any changes in the scene. Changes between the images can consist of different classes of change for example

- The addition, removal or displacement of objects
- Internal changes to objects such as their size, shape or colour
- Changes in the lighting or visibility
- Changes in the properties of the camera used

When detecting changes in a scene from still camera images the last two example types of change are no typically of interest. Changes that fall into the first two may be of interest depending on the application, for example the addition, removal or displacement of a bag may not be of interest for the purpose of surveying the building in an area but may be of interest to security personnel searching for suspicious items. Similarly changes in the size, shape or colour of a lake may be of interest to an environmental scientist but may not necessarily be of interest to security personnel. Knowing the types of changes of interest is important when designing a change detection system. This thesis focuses on any changes in the first two categories as long as the scale of the change is above the sensitivity of the system. The categorisation of change as of interest or not of interest is excluded.

The comparison between the reference and sample image can be carried out manually by human operators or automatically with the aid of computers; the topic of this project is the latter. Automated change
Chapter 1. Introduction

detection using reference and sample images taken from the same position is a well established topic with many established techniques. The majority of these techniques rely on the reference and sample images being captured from a constant perspective. This project aims to present new methods for situations where the reference and sample images of the scene are taken from different perspectives.

The performance of a change detection system can be measured by the false positive and false negative rate of the system. The false positive rate is the rate of misclassification of non-change as change while the false negative rate is the rate of misclassification of change as non-change, also known as the miss rate.

1.2 Motivation and applications

Automated change detection is of increasing importance in the modern world due to the growing availability of imaging data which makes it impractical to detect changes manually. While there are many established techniques for change detection the majority are applied to images taken from the same perspective. Developing methods of change detection that can be used on images captured from different perspectives would allow automated change detection to be used in a much wider range of applications. Important applications of change detection include surveillance, remote sensing, medical diagnosis, civilian infrastructure, underwater surveying, and driver aids in vehicles [RAAKR05].

A key application area which would benefit greatly from automated change detection is the detection of potential threats in a military or security context; changes in a scene may indicate activity that presents a threat. This application area is the focus of this thesis. In this context aerial, satellite, vehicle and person mounted camera are all a potential source of imagery for change detection. Change detection using aerial or satellite images reference and sample images where the camera is facing vertically downwards is an established area of research [RAAKR05, Sin89]. Likewise other scenarios where there reference and sample images are captured from the same perspective can use the same established techniques.

1.3 Problem statement

This thesis focuses on applications where the reference and sample images are taken from different perspectives. Vehicle or personnel mounted camera imagery where the camera is oriented near horizontally presents a challenging problem, especially in urban situations due to occlusion from buildings which would result in drastically different areas of the environment being visible with relatively small changes in viewing angle. For this reason angled aerial imagery where the camera is oriented at approximately $45^\circ$ to the horizontal. The techniques developed could later be extended to be applied to other wide baseline image change detection scenarios.
The goal of automated change detection is to identify significant changes in the scene using reference and sample images, this presents two key challenges. The first challenge is to align areas of the sample image to their corresponding position in the reference image through registration. The second challenge is to compare the registered areas in a way that is robust to registration errors as well as changes in environmental conditions which can affect light intensity and blur.

The way changes in perspective distort the image can be illustrated by the simple example of a cube imaged from two different perspectives in Fig. 1.1. The side of the cube displaying an eye and the side showing a woman illustrate surfaces that have been distorted and shifted due to being imaged from a different position while the aeroplane has been occluded and the monitor with a fish has been dis-occluded. Changes in the way objects occlude each other and their own surfaces provide an problem for change detection that cannot be solved completely as the state of these areas is not recorded in both the reference and sample image. The problem can however be detected and its effects reduced. Detecting changes in occlusion enables the classification of these areas as knowns while localising them minimises their effect on the performance of a change detection system on neighbouring areas. Areas of the scene that appear distorted between the reference image and sample image due to changes in viewing position need to be aligned before they are compared. In order to align them, points or areas of each image that correspond to each other need to be found and the geometry of the shapes in the scene and the changes in viewing position need to be understood.

This illustrates that methods of segmenting and equating areas of the scene consistently in the reference and sample images are needed. Additionally, even if the areas of the images are aligned between the two images, the areas may appear very different when there is no change due to registration errors or other effects such changes in lighting or blur. This presents the two key challenges for wide-baseline image change detection, firstly the ability to align images of complex shapes across images taken from different viewing angles, and secondly ways of comparing the aligned image areas that are not only robust in
environmental effects but also to larger registration errors.

1.4 Contributions

The key contributions of this thesis can be split into three key areas. Correspondences between the reference and sample images found from feature points are used as a starting point for registering the images. The performance of established feature points at wide differences of viewing angle was not found to be adequate and so techniques for improving performance at differences in viewing angle of up to 40° are introduced in Chapter 2.3:

- Affine compensated Scale-Independent Feature Transform (SIFT) [Low99, Low04] feature points that add greatly increased robustness to wide viewing angles compared to SIFT
- Ground plane pre-compensation to improve feature point densities
- Affine compensated SIFT descriptors used with Harris corner points

Secondly region comparison methods that are robust to registration error are introduced in Chapter 5. These are required because of the difficulties in accurately registering images in wide baseline image change detection. Methods based on a probabilistic assessment of the SIFT descriptor introduced:

- Using dense SIFT to match regions with registration error up to the scale of the SIFT descriptor
- Shifted dense SIFT which tracks the variation of registration error over the images in order to extend the robustness to registration error up to a few dozen pixels
- Change localisation using Markov random fields to classify areas as change or not change

Thirdly segmentation techniques that consistently segment the reference and sample images and inherently enable registration are introduced in Chapter 6. These techniques are designed for the specific challenges presented by wide baseline change detection and focus on segmenting regions so that each region can be registered into the corresponding region in the other image using a single transformation. The presented techniques are

- Segmentation based on a Delaunay [Del34, LS80] triangulation, using the triangle corners as correspondences to enable registration
- The clustering of the resulting triangular segments based on the similarity of the affine transformations formed by their corner points
Chapter 1. Introduction

- Possible object edge layouts around image edges are found using object edge hypothesis testing, including the location of any occlusions

- An approach to registering objects that are perpendicular to a defined plane is introduced as standing object matching. It is shown to be able to register vertical walls

Additionally Chapter 7 shows how the techniques introduced can be combined into a change detection system based on the above three key areas. Affine compensated SIFT, the Delaunay triangulation segmentation and registration approach and the clustering of the resulting triangular segments has been published in [JZ13] and [ ].
Chapter 2

Background Material and Literature Review

This chapter of the thesis covers background material required to address the challenges of wide-baseline image change detection. It first includes sections on projective geometry, registration and feature points which provide the required background for understanding how the imaged geometry changes with viewing angle and how to compensate for those changes. The chapter then reviews edge detection and segmentation approaches which can be used to segment the images into sections that are likely to shift and distort in consistent ways. Next methods for comparing images or segments of images are reviewed, including both 2D methods that involve direct pixel comparison and more abstracted approaches that compare images based on the features and objects they contain. Finally similar approaches and systems are reviewed.

2.1 Projective Geometry

Homogeneous coordinates are typically used in projective geometry geometry to simplify the mathematics describing projections. Homogeneous coordinates include an additional non-zero scale parameter, \( s \), with the many-to-one mapping between Homogeneous and 2D Cartesian coordinates defined as \([HZ04]\).

\[
\begin{pmatrix}
  x \\
  y \\
  s
\end{pmatrix}
\rightarrow
\begin{pmatrix}
  x/s \\
  y/s
\end{pmatrix}
\]
2.1.1 Camera matrix

A camera matrix is a 3x4 (3 rows, 4 columns) matrix that represents the internal and external parameters of a single camera. The internal parameters include the focal length of the camera, the pixel shape on the photo-sensor, the nature of the lens and other parameters of the camera itself. The external parameters represent the location and orientation of the camera relative to the coordinate system. The camera matrix can be used to project a 4x1 homogeneous coordinate in the 3D scene to a 3x1 2D imaged coordinate, this represents the imaging of a point in 3D space onto the 2D image. A camera matrix can be understood by constructing it from a simple form. At its simplest a camera is located at the origin of the coordinate system and is aligned the coordinate frame facing down the z axis and with a focal length of 1.

\[
P_o = \begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{pmatrix}
\]

A graphical representation of this can be seen in Fig. 2.1. This camera matrix projects and point along a line in the 3D scene defined by \( \left( \begin{array}{c} x \\ y \\ z \\ a \end{array} \right) ^T \) where \( a \) can take any value to the 2D imaged point \( \left( \begin{array}{c} x/z \\ y/z \\ 1 \end{array} \right) ^T \).

\[
\begin{pmatrix}
1 & 0 & 0 & 0 \\
0 & 1 & 0 & 0 \\
0 & 0 & 1 & 0
\end{pmatrix}
\begin{pmatrix}
x \\
y \\
z \\
a
\end{pmatrix}
= \begin{pmatrix}
x \\
y \\
z
\end{pmatrix}
\rightarrow \begin{pmatrix}
x/z \\
y/z \\
1
\end{pmatrix}
\]

(2.1)

The camera can be rotated, translated and the focal length can be changed by multiplying it by \( \begin{pmatrix}
R & 0 \\
0 & 1
\end{pmatrix} \), \( \begin{pmatrix}
I & c \\
0 & 1
\end{pmatrix} \) and \( \begin{pmatrix}
K & 0 \\
0 & 1
\end{pmatrix} \) respectively where \( R \) is a 3x3 rotation matrix representing the rotation of the camera relative to the coordinate axis, \( c \) is a 3x1 non-homogeneous 3D translation vector representing the position of the camera and \( K \) represents the internal parameters of the camera. If the camera is not skewed, has square pixels and has its optical axis aligned with the origin, \( K \) is a diagonal matrix

\[
K = \begin{bmatrix}
f_d & 0 & 0 \\
0 & f_d & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

where \( f_d \) is the focal length of the camera. combined this gives the camera matrix
Figure 2.1: Two view projective geometry of a point $A$ imaged to $A_1$ and $A_2$ in images 1 and 2. The epi-polar lines are $l_1$ and $l_2$ in images 1 and 2. $x'$, $y'$ and $z'$ are the axes of camera $P_2$. The components of the $P_2$ camera matrix are $R$, $c$ and $f$ where $P_1$ also has a focal length $f$ and is at the origin and aligned to the coordinate system. $e_1$ and $e_2$ are the epi-poles in images 1 and 2.

$$
P = \begin{pmatrix}
I & 0 \\
0 & 1
\end{pmatrix}
\begin{pmatrix}
K & 0 \\
0 & 1
\end{pmatrix}
\begin{pmatrix}
R & 0 \\
0 & 1
\end{pmatrix}
\begin{pmatrix}
I & c \\
0 & 1
\end{pmatrix}

P = \begin{pmatrix}
KR & KRe
\end{pmatrix}
$$

If the focal length is 1 this gives

$$
P = \begin{pmatrix}
r_1^T & r_1^T c \\
r_2^T & r_2^T c \\
r_3^T & r_3^T c
\end{pmatrix}
$$

where $r_n^T$ represent row $n$ of $R$.

2.1.2 Geometric transformations

This section describes and defines a number of 2D geometric transformations that are used within the thesis.
2.1.2.1 Similarity transform

The similarity transform shown below which has 4 degrees of freedom allows for the rotation, translation and scaling of an image [HZ04]. Rotation is represented by \( \theta \), translations in the \( x \) and \( y \) direction are represented by \( t_x \) and \( t_y \) and scaling is represented by \( s \) which can be negative to invert the image. A similarity transform preserves angles and length ratios but not lengths.

\[
k \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} s \cos \theta & -s \sin \theta & t_y \\ s \sin \theta & s \cos \theta & t_x \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}
\]

(2.2)

2.1.2.2 Affine transforms

The trigonometric constraints imposed by the sine and cosine functions in the isometric and similarity transforms ensure that the object proportions remain the same as the image is rotated and scaled. Removing the trigonometric constraints allows for shapes to be compressed or stretched in any direction. The result is known as the affine transformation which has 6 degrees of freedom where the object can be compressed and stretched in any direction. Under this transform the angles between lines change while length ratios in the same direction remain the same and parallel lines remain parallel.

\[
k \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}
\]

An affine transformation, \( A \), can be decomposed as follows

\[
A = kR(\theta)R(-\phi)DR(\phi)
\]

\[
D = \begin{bmatrix}
\lambda_1 & 0 & 0 \\
0 & \lambda_2 & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

\[
R(\phi) = \begin{bmatrix}
\cos \phi & -\sin \phi & 0 \\
\sin \phi & \cos \phi & 0 \\
0 & 0 & 1
\end{bmatrix}
\]

\[
k \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{bmatrix} a_{11} & a_{12} & t_x \\ a_{21} & a_{22} & t_y \\ 0 & 0 & 1 \end{bmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}
\]
\( R(\varphi) \) rotates the image so the required compression direction aligns with the \( x \) or \( y \) axis. \( D \) then compresses by \( \lambda_1 \) in the \( x \) axis and \( \lambda_2 \) in the \( y \) axis before \( R(\varphi) \) rotates the image back to the original position. \( R(\theta) \) is the desired rotational component of the Affine transformation [DLK83].

### 2.1.2.3 Projective homography

The projective homography builds on the affine transform by allowing for perspective effects. In a projection the bottom row of the transform varies.

\[
k \begin{pmatrix}
x' \\
y' \\
1
\end{pmatrix} = H \begin{pmatrix}
a_{11} & a_{12} & t_x \\
a_{21} & a_{22} & t_y \\
v_x & v_y & v
\end{pmatrix} \begin{pmatrix}
x \\
y \\
1
\end{pmatrix}
\]

Typically \( v \) is set to 1 unless it is zero by scaling the matrix at the matrix is scale invariant. \( v_x \) and \( v_y \) cause the \( x \) and \( y \) axis to compress in inverse proportion to the \( x \) and \( y \) coordinates at each point. If \( v_x \) and \( v_y \) are positive, points near the origin are compressed while points towards higher values of \( x \) and \( y \) are stretched. By controlling \( v_x \) and \( v_y \) independently the shrinking due to perspective can be set in any direction [DLK83].

### 2.1.2.4 Calculating transformations

Geometric transformations that describe the transformation between two images can be calculated from point correspondences between the two images. In the case of a projective homography

\[
k \begin{pmatrix}
x' \\
y' \\
1
\end{pmatrix} = \begin{pmatrix}
a_{11} & a_{12} & t_x \\
a_{21} & a_{22} & t_y \\
v_x & v_y & v
\end{pmatrix} \begin{pmatrix}
x \\
y \\
1
\end{pmatrix}
\]

the first row gives the relationship

\[a_{11}x + a_{12}y + t_x = kx'\]

which can be divided by the third row, \( v_xx + v_yx + v = k \) to give
\[ (a_{11}x + a_{12}y + t_x) / (v_x x + v_y y + v) = (kx') / k \]
\[ a_{11}x + a_{12}y + t_x = x'(v_x x + v_y y + v) \]
\[ -a_{11}x - a_{12}y - t_x + x'(v_x x + v_y y + v) = 0 \]

and similarly for the second row divided by the third row
\[ -a_{21}x - a_{22}y - t_y + x'(v_x x + v_y y + v) = 0. \]

Now \( H \) can be found by solving
\[
\begin{bmatrix}
-x & -y & -1 & 0 & 0 & x'x & x'y & x' \\
0 & 0 & 0 & -x & -y & -1 & y'x & y'y & y'
\end{bmatrix}
\begin{bmatrix}
a_{11} & a_{12} & t_x & a_{21} & a_{22} & t_y & v_x & v_y & v
\end{bmatrix}^T = 0
\]

(2.3)

using 4 correspondences which solve the 8 degrees of freedom as each correspondence provides 2 equations as long as no three points are collinear [HZ04]. Transformations with fewer degrees of freedom may be determined using the same method but require fewer correspondences.

The numerical stability of the procedure can be improved [HZ04] by first normalizing both sets of coordinates \((x, y)\) and \((x', y')\) to have their centroid at the origin and an average distance from the centroid of sqrt(2). The similarity, \( T \) is found using equation 2.2 with no rotation and the homography between the transformed points, \( \tilde{H} \) is found using equation 2.3. \( H \) can now be found using the relationship
\[ H = (T^T)^{-1} \tilde{H} T. \]

### 2.1.3 Two-view geometry

Two view geometry describes the geometry of imaging a 3D scene from two camera viewpoints to form two 2D images of the same scene. It describes the position, orientation as well as the internal parameters of each camera in the form of camera matrices and the position of the imaged objects in the 3D scene relative to the cameras. The 2D position of the objects in either of the images can be related to their 3D position via each camera matrix.

\[ x'_4 = P_1 x \]
where $P_1$ represents the first 3x4 camera matrix, $x'_1$ represents the 2D imaged position in image 1 as a 3x1 homogeneous vector and $x$ represents the 3D position as a 4x1 homogeneous vector.

2.1.3.1 Epi-polar geometry

Two view geometry relates to the relative positions of two cameras and the point in 3D space being imaged by the two cameras as shown in Fig. 2.1. A point with a known location in one image produced using a known camera matrix has a single degree of freedom in 3D space as its location along the line from the imaged point through the camera centre is not known. Projecting this line onto the second camera gives a line in the second image that represents the possible positions of the point in that image. This line in the second image is known as the point’s epi-polar line. Finding the epi-polar line can be used to confirm the accuracy of matched correspondences and help in the reconstruction of 3D positions.

How to find the epi-polar line can be illustrated using the camera $P_0$ located at the origin and facing down the $z$ axis as well as a camera $P_1$ located at a different location. From Eq. 2.1, all points along a line $x = \begin{pmatrix} x \\ y \\ z \\ s \end{pmatrix}^T$ where $s$ represents the position along the line to the image of the point $x_0 = \begin{pmatrix} x/z \\ y/z \\ 1 \end{pmatrix}^T$ and so the true location of and point at $\begin{pmatrix} x/z \\ y/z \\ 1 \end{pmatrix}^T$ may lie anywhere along $\begin{pmatrix} x \\ y \\ z \\ s \end{pmatrix}^T$. This line passes through the camera centre of $P_0$ as does the line representing the 3D position of any point in the image. Because of this the position of the camera centre of the other camera in both images in a two view geometry is key to finding epi-polar lines and defining the system.

The image point of the other camera in a two view system is known as the epi-pole or $e$ and is given by $P_1c$ where $c$ is the location of the first camera. As can be seen in Fig. 2.1, the line in 3D space linking the two epi-poles as well as both camera centres is known as the baseline. An epi-polar line in the image of $P_1$ also has to pass through the image of camera $P_0$ at the point where $x$ is imaged which can be found by multiplying $x_0$ by the pseudo-inverse of $P_0$ , $P_0^+$. Projecting onto the second image gives a point in the second image $P_1P_0^+x_0$ which together with the epipole can define the epi-polar line as a cross product of homogeneous points

$$l = e \times P_1P_0^+x_0$$

The fundamental matrix, $F$ is a 3x3 matrix relating $l$ to $x_0$ and so

$$F = e \times P_1P_0^+$$

As the point imaged in by $P_1$ lies on $l$, $x_1^Tl = 0$ and as $l = Fx_0$
\[ x^T_i F x_o = 0 \]

### 2.1.4 RANSAC

Often many more correspondences are available than are required to find the geometry of a camera, transformation or other unknown but the set of correspondences include incorrect matches and have localisation errors. Random Sample Consensus (RANSAC) is a technique used to find the solution that provides the best fit result while excluding the effects of outliers [FB81]. First multiple subsets of correspondences, each with enough members to find a solution, are selected. The solution is found for each subset. The solution that agrees with the most correspondences in the entire set to within a chosen threshold is chosen. Points that do not agree with that solution are removed from the set. The final output is found using the remaining points using a least squared error solution. The RANSAC approach provides an approach that removes the effects of outliers while still allowing for the utilisation of the majority of the points when calculating the final solution in order to reduce the effects of noise.

### 2.2 Registration

The majority of change detection techniques require two or more images to be registered. Image registration is the alignment of the images from one viewing angle and set of internal camera characteristics to another. Typically registration consists of the following steps:

- Feature detection and matching - Finding distinct points in both images and matching them using descriptors calculated at each point.
- Image geometry estimation - Using the matched points as correspondences to calculate the geometric transformations between the two views.
- **Image interpolation** - Transforming each pixel position using the defined geometric transformations and interpolating to give a regular pixel grid [ZF03].

Feature detection and matching and image geometry estimation are covered separately in more detail in Sec. 2.3 and Sec. 2.1 respectively. The remaining interpolation step as well as the types of errors encountered in registration are reviewed in this section.

#### 2.2.1 Pixel interpolation

Once the geometric transform has been found it is applied to re-align each pixel in the image. After re-alignment the pixels are unlikely to lie in a regular grid and so a method of producing a regular grid
of pixels from the irregular transformed pixels is required. The discrete 2D function represented by the
re-aligned pixels needs to be used to produce a continuous function from which the regular grid of pixel
values can be sampled. Discrete pixels represent the average image intensity and colour over an area and
so form a stepped function in 2D. An ideal low pass filter can be used to smooth the stepped function
while retaining all available image data. In practice this is not realisable and also will result in high
frequency edges being smoothed. Common methods for interpolating pixel values include:

- Nearest neighbour [PKT83]
- Linear interpolation [PKT83]
- Cubic B-spline [PKT83]

Of importance in this project is whether the interpolation technique used has any effect on other tech-
niques applied, for example feature points or textons. Any techniques that first applies any smoothing,
for example Gaussian filters used in SIFT, should not be greatly affected by these.

2.2.1.1 Nearest neighbour interpolation

Nearest neighbour is the least computationally intensive but results in blocky images and can result in
location errors of features by up to half the pixel spacing. One the other hand strong edges which many
interpolation methods tend to smooth out are retained. Although the resulting image may appear blocky
to a human observer the retained strong edges mean that it may be a good candidate if the purpose of
the processing is to produce an image for edge or corner detection as long as localisation errors are not
of importance [PKT83].

2.2.1.2 Linear interpolation

Linear interpolation or the use of a triangle filter is in effect a weighted average of the nearby pixels,
weighted by distance from the continuous point. The pixel weightings follow a triangle shape with the
top of the triangle at the point being found. It in effect interpolates with a straight line if only two pixels
are within the scope of the filter [PKT83].

2.2.1.3 Cubic B-spline interpolation

Cubic splines are 3rd order polynomials with constants selected so that they pass through a set of control
points and are set to have a second derivative of zero at the start and end points. Splines can be used
to model the physical process of drawing a curve with minimum curvature through a set of points. In
this way they can be used to form a continuous signal from discrete data points and so provide a method
for interpolation. A tutorial in using splines in a variety of signal processing applications including interpolation can be found in [Uns99].

2.2.2 Registration error

Registration error is the extent to which each pixel has been mis-registered and the likely extent of the mis-registration. Measures of this are often quoted to the user of a process that involves registration. Registration error can consist of a combination of localisation error, matching error and alignment error. Localisation error is the error due to the uncertainty of the location of the feature points used; this is often quoted together with the algorithm defining the feature point algorithm. Matching error is the possibility of an error due to false feature point matching. To reduce the possibility of this error occurring various strategies can be applied:

- Use two matching algorithms and only accept matches obtainable from both
- Exclude a pair of points from each step of the process and then check that all points fall within a threshold distance of their matched location in the final image [ZF03]
- Have two sets of points, a training set and a test set. The registration is conducted using the training set and tested using the test set
- Use a least squared error approach with a much larger number of points than necessary, rejecting points that are far from the resulting transform

An iterative approach which converges on a solution that provides a close fit to the largest possible set of samples known as RANSAC is often used [FB81]

Alignment error is the component of registration error that represents the difference between the mapping model used in the geometric transformation and the actual transform. This can be a knock on effect of poor feature point localisation or higher order distortive effects than those used in the transform [ZF03]

2.3 Feature points

Feature points are used in a large number of computer vision applications and specifically both in image registration and in change detection. In the former the feature points are used in the process of aligning images and obtaining pixel level image registration. In the later the feature points can be used as the actual quantities that are compared in the change detection process. An overview of the various methods of extracting and matching feature points can be found in [TM08].
The concept of feature points is quite simple. The aim is to find points in an image that are consistently identifiable in other images of the same scene. These can be designed to tolerate effects such as scaling, rotation, projective transformation, illumination variation and white noise. Feature point methods can be seen as having two steps, firstly a method for detecting feature points is required and secondly a method for describing the feature point is needed to allow for matching between images.

2.3.1 SIFT

SIFT is a local image feature detector and descriptor that is invariant to image scaling, translation and rotation and also partially invariant to illumination changes, affine or projective transformations. SIFT is claimed to tolerate a 60° change in viewing angle on flat surfaces or up to a 20° rotation of a 3D object [Low04]. SIFT feature points are maximum and minimum intensity points on Difference of Gaussians (DoG) abstractions of an image. A number of Gaussian filters are applied to the image such that a continuous set of filtered images is produced with a constant multiple separating the standard deviation between each filter used. Filtered images adjacent in the Gaussian scale used are subtracted from each other to produce DoG images. The Gaussian filtered images are then down-sampled by a factor of 2 and a new set of DoG images is found. Maxima and minima are found in the DoG images by finding pixels that are bigger or smaller than the 8 surrounding pixels in the DoG image. Pixels that are not also bigger or smaller than the 9 pixel square centred at the same coordinate in the DoG images obtained from the same sampled image and the sampled images immediately larger and smaller are rejected. Points that correspond to edges are removed as the gradients around such points are not stable across viewpoints. Also points of low contrast are removed as these are hard to detect and are less stable across images [Low04].

The second function of SIFT is to assign descriptors to each point to allow for feature point matching across images. As the descriptors are orientation-specific, a way to identify an orientation for each point is required. The image is smoothed at the scale of the feature point, again using a Gaussian filter. The orientation of each point is assigned to be the dominant intensity gradient direction of the Gaussian smoothed image at the scale of the point. This orientation is used to align an 8 × 8 pixel square grid around the feature point. The size and direction of the gradient in the Gaussian smoothed image within each grid box is found as a vector with direction and magnitude.

These vectors are consolidated into 16 sets of histograms consisting of eight rotation bins each as illustrated in Fig. 2.2 (which only shows four histograms for simplicity). The amount each vector in the eight by eight grid contributes to each of the 16 histograms is defined by the Gaussian weighting of its distance from the centre of each of the 16 sets of eight vectors. The magnitude of each vector is assigned to a bin within each histogram according to it’s orientation using a triangular filter to split the magnitude

21
between adjacent bins. The descriptor is normalised and the 16 histograms are concatenated to form a 128-element vector. **SIFT point matching** is based on finding the points with the lowest Euclidean distance between descriptors.

### 2.3.2 SURF

Bay et al. introduced an evolution of the SIFT algorithm tuned for speed known as Speeded-Up Robust Features (SURF) [BTG06]. The Gaussian filter is approximated with a box filter that gives results close to the Gaussian but greatly reduces the computational load which is key to performance due to the large number of times the Gaussian is applied. This also allows all scales to be processed without the need for sub-sampling of the image. Otherwise the detection of the feature point follows the SIFT process described in Sec. 2.3.1.

The orientation of the feature point is recorded by finding the angle around the local area of the interest point that gives the highest Haar-wavelet response. If the orientation of the image is already known or is not expected to vary this step can be removed and the simplified algorithm is known as Upright Speeded-Up Robust Features (U-SURF). The descriptors are produced by finding the wavelet response in x and y directions in each square of a 4x4 grid around the feature point. Another version known as SURF-128 records the total +ve and –ve components of the wavelet responses and further split depending on whether the perpendicular response is +ve and –ve magnitude responses for each section. SURF-128 does not add significant computation time as the components are already available in the basic version of SURF but the large number of resulting vector components slows down matching and requires more data to be stored. The experimental results quoted compare favourably with other SURF based feature points such as SURF, Principal Component Analysis (PCA)-SURF and Gradient location and orientation histogram (GLOH) [BTG06].
As with SIFT, SURF compensates for scale, orientation and luminance but not for affine or projective effects.

### 2.3.3 FAST

Features from Accelerated Segment Test (FAST) is a corner detector introduced by [RD05, RD06] designed for computational efficiency and suitability to real-time video processing. The algorithm uses a continuous circle of 16 pixels around a candidate point. If at least \( n \) contiguous pixels have an intensity over a threshold above, or under a threshold below the candidate point, the point is defined as a corner. It is suggested that \( n = 12 \) which allows for an efficient implementation is presented in [RD06]. When \( n = 12 \), candidate points can be filtered out in a preliminary step by first taking into account 4 equally spaced points around the 16 pixel circle as at least 3 of these must pass the threshold for the point to be accepted. This greatly speeds up computation.

### 2.3.4 BRIEF

Binary Robust Independent Elementary Features (BRIEF) is a feature point descriptor consisting of a bit string where each bit represents the result of a logical test \( p(x) < p(y) \) where \( p(x) \) and \( p(x) \) represent the pixel intensity of a pixel at positions \( x \) and \( y \) of a smoothed image [CLSF12]. A Gaussian smoothing kernel with a variance of 2 is suggested or alternatively a discrete kernel windows of \( 9 \times 9 \) pixels can be used. Variants with 128, 256 and 512 pairs of pixel positions, resulting in 128, 256 and 512 bit descriptor vectors are shown. A number of methods of selecting the \( x \) and \( y \) pixel positions are tested in [CLSF12] and the random selection of points from a Gaussian distribution \( G(0, \frac{1}{25}S^2) \) where \( S \) is the width and height of the patch being described is shown to perform best. Descriptors can be matched using a Hamming distance.

Overall BRIEF is designed to be a computationally efficient descriptor that can be used with a feature point that provides the position, scale and orientation of detected points.

### 2.3.5 Oriented FAST and Rotated BRIEF

Oriented FAST and Rotated BRIEF (ORB) is an open source complete feature point detector, descriptor and matching algorithm based on the FAST feature point detector and the BRIEF descriptor [RR11]. It claims to be two orders of magnitude faster than SIFT while providing similar levels of performance. As it uses a corner detector instead of a blob detector it will not detect similar points in the scene and its performance will vary compared to SIFT across different scenes. A variant of FAST that uses a circle of pixels with a radius of 9 to test for corners is used, the points are then ranked using the Harris corner
measure [HS88]. The detector is applied to a scale pyramid of sub-sampled images to produce multi-scaled features. The authors introduce a rotational component to the FAST feature point detector based on the intensity centroid [Ros99]. The position coordinates of the BRIEF pixel sample pairs are rotated in increments of 12° to produce 30 sets of coordinates. The set for which the rotation best matches the point orientation is used to calculate the descriptor.

2.3.6 PCA-SIFT

PCA-SIFT is an adaptation of SIFT that uses the same method for locating the feature point as SIFT but uses a PCA-based approach for producing the feature point descriptor. After the feature points are identified a 41x41 grid is placed around the feature point which is sized and oriented in the same way as in SIFT. The local horizontal and vertical intensity gradients within each grid box are recorded in a vector. The vector is normalised to unit size. PCA is applied to the set of vectors to extract the principle gradients and dimensionally reduce the descriptor, it is suggested that around 20 dimensions should be retained. The aim is that distortions not compensated for in SIFT such as perspective effects can be modelled as low dimensional Gaussian effects and so will form part of the low eigenvalue dimension components, Those will not be included in the 20 or so dimensions retained and so will prevent these effects from affecting matching adversely [KS04].

PCA-SIFT is shown to perform significantly better under noise than standard SIFT. While it is also shown to perform significantly better under similarity and projective transforms it still does not successfully match significantly more than 50% of the points if thresholds are selected to allow for an acceptably low false positive rate. Under changes in illumination, both PCA-SIFT and SIFT perform very well with SIFT having a slight advantage [KS04].

2.3.7 Histograms of Oriented Gradients

Histograms of Oriented Gradients (HOG) is a descriptor designed for use in robust visual object recognition, in particular in the support vector machine (SVM) based detection of human figures [DT05]. It is designed for use on a dense or overlapping grid of small spatial regions. In each region, a local 1D histogram of gradient directions is accumulated. For better robustness to variations in lighting levels the histograms are normalised over larger spatial regions. In the example implementation given in [DT05] the dense grid of histograms is then used as a combined feature vector in a SVM based window classifier for human detection.
2.3.8 Corner, edge and area based techniques

A common form of edge and corner detection is the method published by Harris and Stephens [HS88] known as the Harris point. This technique expanded on previous techniques in three ways:

- By conducting filtering so that edges and corners in all alignments are detected
- Using a smooth circular window, for example a Gaussian kernel minimises the number of false edges and corners due to noise and allows for the detection of edges along lines of high gradient
- The components of the filter response are used to distinguish edges and corners and they also give information about the direction of edges and the edge strength in each direction

By setting low and high thresholds to the detected edges and applying edge contour hysteresis greater edge continuity can be achieved.

Mikolajczyk and Schmid suggest an extension of Harris point detector that improves its robustness under scale and affine transformations [MS04]. The Harris point detector is combined with automatic scale selection. The second moment matrix is used to detect the Harris points; to make it scale adaptive the second moment matrix is scaled to the Gaussian local scale and the derivatives are averaged using a Gaussian window. The points that stand out compared to points across scales at the locality are picked so that they are easily separated over a wide range of scales. A local approximation for the effect of the affine transformation on the second moment matrix is found for a neighbourhood which allows for the matching of all points within the neighbourhood.

2.3.9 Maximally stable extremal regions

A method of identifying and matching distinguished regions is known as Maximally Stable Extremal Regions (MSER) [JMTUP01]. This is similar in purpose to feature points as it attempts to identify features that are recognisable under a wide range of conditions, but relate to a region of the image rather than a point. The method for selecting regions depends on finding areas of the image that are uniformly dark or light in relation to the rest of the image. If a grey scale image is thresholded to black and white at all possible intensity thresholds differing areas of black and white will be present. The author claims that some of these areas will be stable for a large range of thresholds as they have a boundary with a large intensity gradient; these areas are selected as the maximally stable. Matching is conducted via a descriptor that maps the region onto a unit disk, mapping to a unit disk eliminates all affine effects other than rotation, this idea was introduced by Baumberg [Bau00], a vector descriptor of this unit disk can be used for index matching. Tuyltelaars describes a similar technique [TG04]. Schaffalitzky et al. give an example application of wide angle feature recognition using MSER where holiday pictures are grouped
together where they include the same unit disk representation of the MSER and so must be of the same scene [SZ01].

2.3.10 Spatial methods

Instead of using feature descriptors the spatial arrangement of the feature points can be used. The sorted inter-point distance vector, sorted nearest neighbour vector or the minimum spanning tree can be used to match clusters of points between images [DLK83]. The vector representations of these measures can be matched using standard matching techniques such as finding pairs with a small Euclidean distance. Similarly points connected by abstract edges, lines or shapes can be matched between images by modelling the shape of the interconnecting lines or shapes and matching them between images. Measuring properties that are preserved under projective transformation such as straight lines and the intersections between them can make these measures independent to the perspective of the image [ZF03].

2.3.11 Comparison of feature point techniques

Mikolajczyk and Schmid conducted an evaluation of various local descriptors to help clarify which methods are more appropriate for various scenarios. Shape context, steerable filters, PCA-SIFT, differential invariants, spin images, SIFT, complex filters, moment invariants and cross correlation were investigated. In general, over the tests conducted, SIFT is shown as the best performing complex algorithm while moments and steerable filters perform the best among the low dimension algorithms. An extension of the SIFT descriptor is also proposed called GLOH which is shown to outperform SIFT [MS05].

2.3.12 Object recognition by using the object edge profile

The phase offset of different scales of the dual-tree complex wavelet [SBK05] of an area of an image can be used to categorise the object within that area of the image. The dual-tree complex wavelet gives a description of edge properties around a point in an image, how these compare with the edge properties at that point at different scales can be used to characterise an object. It is argued that this method is more robust to noise and has the advantage of not relying of the detection of interest points such as Harris points or difference of Gaussian maxima or minima but instead gives an overall description of the object [RAF06].
2.4 Edge detection and segmentation

It is often useful to obtain the structure of the scene in terms of object boundaries and object regions. Two approaches to this are edge detection and segmentation. Edge detection attempts to find boundaries between areas of differing pixel intensity using a filter to detect intensity boundaries while segmentation attempts to group pixels with similar characteristics. Although the two methods work very differently they can both be used to detect boundaries between objects and the shapes of objects.

In this section the mean-shift segmentation algorithm [CM02, GC03] as well as the Canny edge detection approach [Can86] and the Sobel filter [Sob68] that can be used as part of the approach is reviewed.

2.4.1 Mean-shift segmentation

Mean-shift [Che95] is an algorithm for finding local maxima in a feature space that can be applied to the problem of image segmentation. Each point in the feature space is assigned to a local maxima. All the points assigned to a local maxima form a cluster. In the application of image segmentation the pixels forming a cluster can be defined as a segment and so mean-shift can be used in image segmentation. Mean-shift finds maxima by iteratively shifting each data point in the feature space using gradient ascent, towards its local feature space maxima. This process is repeated until the points converge into a number of stable maxima. The feature space density is locally weighted around each point using a kernel defined for the application. Typically a Gaussian or a stepped kernel is used together with dimensional weightings. While a specific scale need to be defined for the weighted kernel, this kernel does not define the scale of the clusters detected. The algorithm shifts data points towards local density maxima, using the kernel to remove the effects of small density variations due to noise, and so finding all maxima at a scale larger than the kernel used. Areas of image that contain the original position of the image pixels assigned to the same cluster before the clustering process is applied are defined as a segment. In this way the image is split into a number of segments equal to the number of clusters found. Using mean-shift for image analysis is described in [CM02] and an implementation for image segmentation is introduced in [GC03]. The benefits of mean-shift over other clustering algorithms such as k-means or Gaussian mixture models is that the number as well as the scale of centres does not need to be known beforehand. The larger number of centres recalculated at each iteration however results in a larger computational load.

2.4.2 Sobel filter

The Sobel filter operator [Sob68] combines a 1D differentiation filter with an orthogonal 1D smoothing filter to form a 2D filter that can be orientated in either direction. The smoothing filter
and differentiation filter

\[
\begin{bmatrix}
1 \\
2 \\
1 \\
\end{bmatrix}
\]

are convolved to give the filter in the \( x \) axis

\[
E_x = \begin{bmatrix}
-1 & 0 & 1 \\
-2 & 0 & 2 \\
-1 & 0 & 1 \\
\end{bmatrix} \ast \begin{bmatrix}
1 \\
2 \\
1 \\
\end{bmatrix}
\]

and similarly in the \( y \) axis

\[
E_y = \begin{bmatrix}
-1 & -2 & -1 \\
0 & 0 & 0 \\
1 & 2 & 1 \\
\end{bmatrix} \ast \begin{bmatrix}
-1 \\
0 \\
1 \\
\end{bmatrix}
\]

The response from these two filters, \( R_x \) and \( R_y \) can be used to find the 2D gradient magnitude \( R = \sqrt{R_x^2 + R_y^2} \) and direction \( \theta = \arctan2(R_y, R_x) \). A Sobel filter can be used as an edge detector by thresholding on the resulting magnitude. The combination of a differentiation filter with a smoothing filter allows for the detection of changes in intensity while rejecting those responses that do not form a line. The ability to detect a gradient allows for the tracing and connection of edge pixels to form a continuous line rather than a series of individual edge pixels.

### 2.4.3 Canny edge detection

Edge detection using the Canny edge detection method [Can86] can be broken down into 5 steps:

1. Smooth the image using a Gaussian kernel to remove noise.

2. Use a filter such as a Sobel filter to detect the edge strength in 4 directions, horizontal, vertical and both diagonals.

3. Non-maximum suppression is applied. This thins the edges by removing responses that are not local maxima.
4. The edge responses are thresholded to remove weak responses. Two thresholds are used to separate strong and weak edges.

5. Weak edges that are not connected to a strong edge are rejected. In this way only edges giving a strong response are detected but discontinuities due to parts of the edge producing weak response can be completed if a hint of an edge exists.

2.5 2D Change detection – Image based

There are a number of well-established change detection techniques used to compare two images taken by identical cameras from identical viewpoints. A large motivation for the development of these techniques was the availability of satellite imagery, CCTV and other forms of automated image capture which provided images in larger quantities then it was practical to study manually but which fitted these criteria closely. The techniques can be split into two main types

- Differencing the intensity of pixels
- Classifying pixels and detecting the change in classification

Differencing the intensity of individual pixels is often a simpler process as it only requires a single step, this also means that there is only one step in which an error might occur which generally results in more reliable change detection however it does not give any information about the type of change that has occurred other than the level of intensity change. Classification before change detection can make the differencing step simpler but requires a more complex classification step beforehand and so in many situations performs less well then pixel differencing. On the other hand as the before and after state are classified already, more can be directly interpreted from the results without additional processing steps. Registration errors and occlusion effects present after registration mean that post-classification change detection may be preferable when registration errors are present.

2.5.1 Illumination adjustments

Change detection methods are often highly dependent on the images being compared having consistent properties. Two of the properties of particular importance are spatial alignment and illumination. There are several techniques used to pre-process the images to reduce these effects and sensitivity to them can vary between different change detection techniques [RAAKR05].

Illumination differences can be compensated for by using image intensity histogram normalisation techniques. Additionally some techniques such as least squares or polynomial mapping can incorporate the differences between images assuming the whole scene has the same illumination change [RAAKR05].
2.5.2 Pixel differencing

The intensity of pixels can be subtracted from each other to produce a difference distribution. Pixels with values around the mean show no change while pixels with values in the tails tend to show change. A key element of this method is to decide on a threshold boundary used to categorise each pixel as being a change or a no-change pixel. The threshold can be denoted by the number of standard deviations from the mean level of change and applied to different images, of course this results in images with high levels of change achieving false negatives and low levels false positives.

Prepossessing methods using edge-preserving smoothing filters have been found to be useful to minimise the effects of registration noise [GLM07] and intensity distribution normalisation (mean and standard deviation) have been found useful in some situations in situations where lighting is inconsistent [RAAKR05]. Some information may be destroyed that corresponds to significant image changes, for example pixels that changed from intensities of 110 to 70 and 180 to 140 appear the same on the difference distribution [Sin89].

2.5.3 Ratio or least squares differencing

Least squares linear regression or ratioing can be used as alternatives to differencing. Linear regression is used to find a linear relationship between the pixel intensities in the reference and sample images. Pixels that are more than a threshold away from their predicted values are marked as change. This is useful when environmental effects affect pixel intensities in no-change regions in a significant but linear way. Ratioing takes the ratio of the pixel intensity values between corresponding pixels in the reference and sample images and results above a threshold away from 1 are marked as change. The advantage of taking the ratio instead of the difference is that some environmental effects such as changes in ambient light can cause a multiplicative effect on pixel intensity values and so modelling those effects can be easier when using the ratio between pixel values [Sin89, RAAKR05].

The advantage of using regression is that more complex but still linear distortive effects can be removed without the requirements to know and understand what those effects are. Ratioing is an alternative to differencing where the intensity of a pixel is divided by the intensity of the corresponding pixel in another image; results away from 1 indicate change. The results produced can be interpreted in the same way as a log of the pixel intensity change. Ratioing can make changes in illumination easier to detect as they result in multiplicative effects which have the effect of shifting the centre of the no-change threshold away from 1 to the mean of the change ratio for the image. Ratioing however results in a non-normal distribution of results which complicates further operations for example when applying symmetrical thresholds [Sin89, RAAKR05].
2.5.4 Area differencing

Due to registration errors and possible local disturbances to the image it is often preferable to incorporate the level of change in blocks of pixels rather than at the individual pixel level. The two main approaches both input blocks of pixels into the change detection algorithm, in the first the result is applied to the entire block while in the second the result is applied to the centre pixel. Area differencing gives a similar effect to passing a low pass filter over the difference distribution before thresholding. The negative effects of this are that small levels of change only affecting one or two pixels will be missed [RAAKR05].

2.5.5 Probabilistic models

In probabilistic models the threshold level of change across the pixels within the view (or single pixel) can be chosen to give a desired false positive and false negative rate by using a noise model. An easy example is the application of a Gaussian noise model to the simple differencing of a single pixel. The threshold for classifying the pixel as having changed can be set on the lower and upper portion of the Gaussian describing the expected level of noise so that the proportion of pixels under the noise condition that are wrongly classified as having changed is set, for example 0.1%. This sets the false positive rate. Assuming that a changed pixel becomes a random intensity between white and black, the proportion of the intensity spectrum of the image that falls within the thresholds chosen gives the false negative rate. Other models can be applied that implement more complex hypotheses then random Gaussian noise, also the expected level of noise across spatial frequencies can be used in area differencing [RAAKR05].

If multiple reference images are available, more complex models can be produced with more than one mode. For example a single pixel on an area of road that is sometimes obscured by a swaying branch can have one threshold around the expected intensity and colour of the branch and another around the expected colour of the road. In this way the model can be trained to ignore predictable changes [FMS+05].

2.5.6 Post classification differencing

In this technique, classification of the pixels or areas within an image is made before the images are compared. An area of the image would be classified as for example jungle, building or farm land and then a difference distribution is found of the classified images. The classification can be done algorithmically by grouping pixels with similar properties or by grouping areas with low levels of change between them. This technique has the advantage or being able to discount changes in environmental conditions and imaging techniques between the two images. The disadvantages are that there is the opportunity for error in the classification of either image and that classification criteria which act consistently on both images are necessary for change detection to be carried out [Sin89].
2.5.7 PCA

Principle component analysis (PCA) is a process of breaking down data into a series of orthogonal vectors with progressively lower correlations with the data which can be used to reconstruct the original data set, so no information is lost. Vectors with low correlations can then be removed to reduce dimensionality while losing the minimum amount of information. A data set is a vector holding the measurements of an output of the system being studied, all of the data sets must be the same length and at least two must be available. The data sets must be captured against a consistent input, for example time, or position in the case of images. The first step is to normalise each data set by subtracting the mean of the data set from each value in the data set. Next a matrix of the covariances between every data set is produced where $x$ and $y$ are two example data sets

$$
\begin{bmatrix}
cov(x, x) & cov(x, y) \\
cov(y, x) & cov(y, y)
\end{bmatrix}
$$

The eigenvalues and eigenvectors for the covariance matrix are found

$$
\begin{bmatrix}
\lambda_1 \\
\lambda_2 \\
a_{11} & a_{21} \\
a_{12} & a_{22}
\end{bmatrix}
$$

Where $\lambda_x$ are the eigenvalues and $a_{xi}$ are the eigenvector components for the eigenvectors $e = \{1, 2, 3\}$. If the matrix formed of the eigenvectors is multiplied with the combined data set the data set is transformed onto a space where the eigenvectors are the axes. If some of the eigenvectors are removed the result omits the dimension that eigenvector represents. In this way eigenvectors with low eigenvalues can be removed without much impact on the data, these removed eigenvectors often represent noise.

With real world change detection problems where differing images correlate to some extent, similar images do not correlate perfectly and a finite data set the process is not as clean. Ideally the first eigenvector should reproduce the reference image, the second the change and the remainder should represent unimportant variations between the reference image set for example noise and changing luminance which can be called the noise images. The imperfect correlation between reference images and the partial correlation with the change image mean that the primary eigenvector is partly influenced by the change and so will not perfectly ‘point’ in the direction of the correlation between the images and will not reproduce a perfect reference image. This will have the knock on effect that some of the change and reference information will
be present in the secondary change image and noise images due to the orthogonal nature of eigenvectors. Another issue is if there is more than one image with change, ideally this would be represented on the third image in the series but in reality will correlate to some degree with the reference image and first change image, further changing the eigenvector alignment.

Rafael Wiemker et al. attempt to solve these real world issues mentioned above in [RWB97]. A method of training the primary eigenvector to the non-change image data is attempted using an iterative approach. After the first iteration the pixels are weighted according to their distance to the primary eigenvector representing the reference image and the analysis is repeated, an example is shown in Fig. 2.3 where the primary eigenvector is stabilised in as little as 2 or 3 iterations.

Additionally the assumption that change pixels are surrounded by other change pixels is used to weigh the feature space points by the inverse of the standard deviation of the pixels in the local region when finding the primary eigenvector. The change component of the image represented by the second eigenvector is divided by the standard deviation of the region to minimise the effect of high frequency noise on the change image. The change image is then smoothed; this represents the increased probability that a pixel represents change if neighbouring pixels have similar change [RWB97].

### 2.5.8 Vector techniques

The intensity values for each spectral band captured (red, blue, green, infra-red etc.) for each pixel can be represented as a vector and the change between pixels can be represented as a change vector with
its magnitude related to the likelihood of change occurring. The direction of the vector can be used to classify the type of change, for example a change from green growth to bare soil [Sin89].

2.5.9 Background subtraction

Background subtraction can be used when detecting changes in features rather than surface coverage, the change in the distribution of features that are not part of the background can be measured. A low pass filtered version of the image can be used to give a background image [Sin89].

2.5.10 Model fitting

Polynomials or other mathematical constructs can be fitted to the variation of pixel intensity or colour over time between several images. This can be useful to represent the nature of the change, for example an area might be getting gradually lighter, and highlight unusual change that does not fall within the level of expected change captured within the model [RAAKR05]. Alternatively a function can be fitted to the reference image or section of the image and the if the function coefficients are close to those found when fitted to the sensed image no change is detected, if the coefficients do change then a change is detected [RAAKR05].

2.6 Change detection – Feature and object classification based

Change detection does not necessarily require full registration but can be conducted on a set of characteristics collected from the images. Object recognition and classification techniques give the tools needed to capture the key characteristics of images. This review has already looked at feature points which can be used as one method of capturing characteristics for example SIFT is used in [Low99].

Due to the nature of object recognition techniques not all of the characteristics of an object will always be captured successfully, because of this, a Bayesian approach is often used to determine the probability of an object being present by attaching a relative weight to each characteristic [SK98].

2.6.1 Recognition of specific objects in images

Techniques for the recognition of specific objects such as those reviewed in [BSB+96] and [LSW90] require the development of characteristics for each type of specific object using images of the object taken from multiple perspectives without occlusion. The object can then be detected in sample images despite occlusion, perspective and noise effects. This has been applied to the identification of weapons in live
scenes [Out09]. As the catalogue of objects in the type of scenes proposed for this study will not be
known in advance, it is unlikely that these techniques can be applied, although they have been shown to
tolerate large changes in viewing angle.

### 2.6.2 Partial depth reconstruction

Object recognition can be conducted via partial depth reconstruction using inputs such as stereo images,
motion, shading, texture and other inputs to build a model of the object between 2D and 3D known
as 2.5D. It is argued that complete depth information is not a vital component of object recognition in
humans as illustrated by similar recognition times for images and line drawings in psychological experi-
ments. The object models can be matched between images and then changes to the catalogue of images
or their location or other properties can be identified [Low87].

### 2.6.3 Colour based approaches

Colour based object recognition introduces an additional characteristic that is claimed to be highly in-
variant to viewing angle and hue and saturation are, in addition, invariant to illumination level changes.
A method of normalising the colour has been proposed which performs well under changes in the illu-
mination colour balance and has been applied to the recognition of objects in images from library of 500
objects [GS98].

### 2.6.4 SIFT bag of words

The idea behind the Bag of Words (BoW) approach is analogous to characterizing a document by forming
a histogram of the frequencies with which words occur in the document [SZ03, JDS10, CLVZ11]. Thus
two documents on the same subject might be expected to use similar words and would therefore have
similar usage histograms.

To represent an image as a combination of visual ‘words’ a way of capturing the ‘words’ is needed.
Typically a method of representing local image intensity gradients as vectors is used. The ‘words’ are
defined by clustering a large number of gradient vectors from a set of training image with each cluster
centre defining a ‘word’. The set of ‘words’ is known as a dictionary and each image can be represented as
a combination of a number of each ‘word’. In this way an image patch can be represented as a histogram
of visual ‘words’. The histogram of two image patches can be compared to determine if the images match.
2.6.5 Textons

An alternative matching approach based on measuring the response of a section of the image from a set of textons or visual 'words' is presented in [WCM05]. A training data set is convolved with a filter bank, the filter responses are aggregated over all the images in the training set and clustered. Textons are defined as the centre of each cluster and so represent the characteristic filter response for a set of features that produce a similar response to the filter bank. Together all the textons obtained are known as the universal visual dictionary. An area of the image with a similar distribution of textons is said to have a similar texture. Objects as diverse as cows, bicycles, cars, grass and trees are said to produce a distinct texture which can be recognised even with high occlusion differing perspectives as shown in Fig. 2.4.

Fogel and Sagi [FS89] show that Gabor filters can discriminate simple shapes with similar performance characteristics to human tests. In these tests human subjects are asked to differentiate simple shapes that are flashed on a screen. It was shown that Gabor filters and human observers performed well and poorly on similar groups of test objects. The results show that Gabor filters seem to discriminate in a similar way to the human test subjects. The Gabor filter is defined as

\[ G(x, y|W, \theta, \phi, X, Y) = \exp \left( -\frac{(x - X)^2 + (y - Y)^2}{2\sigma^2} \right) \sin(Wx \cos \theta - y \sin \theta + \phi) \]

where \((x, y)\) is the position, \((X, Y)\) is the support, \(\theta\) is the filter orientation, \(W\) is the frequency of the filter and \(\phi\) is the phase shift. The responses at phases \(\phi = 0\) and \(\phi = \pi/2\) are used to characterise the image, \(A(x, y)\) to form the descriptor \(Y\).

\[
G_{A,0}(X, Y|W, \theta) = \sum_{xy} G(x, y|W, \theta, 0, X, Y) \times A(x, y)
\]

\[
G_{A,\pi/2}(X, Y|W, \theta) = \sum_{xy} G(x, y|W, \theta, \pi/2, X, Y) \times A(x, y)
\]

\[
Y_A^2(X, Y|W, \theta) = G_{A,0}^2(X, Y|W, \theta) + G_{A,\pi/2}^2(X, Y|W, \theta)
\]
The descriptor is also collected from image $B$ and to form $Y^2_B(X, Y|W, \theta)$. The two descriptors collected at a number of angles, $\theta$ can be used to form the discriminator $D_G$:

$$D^2_G = \left| \sum_{X,Y,\theta} Y^2_A(X, Y|W, \theta) - \sum_{X,Y,\theta} Y^2_B(X, Y|W, \theta) \right| \left| \sum_{X,Y,\theta} Y^2_A(X, Y|W, \theta) + \sum_{X,Y,\theta} Y^2_A(X, Y|W, \theta) \right|$$

[WCM05] goes on to describe a method of segmentation based on texture response. The image is divided into a grid and the texture response of each grid square is captured in an image. Smoothing, thresholding and pruning steps are then used to produce clean segmentation around areas that contain the texture shape searched for as shown in Fig. 2.5.

Leung and Malik use a texton approach to visually recognise different surfaces such as concrete, rug, marble, or leather under different lighting conditions and viewing angles [LM01]. It is hypothesised that at a local scale a material consists of a limited number of micro-structures known as textons which could represent ridges, grooves, bumps and hollows, each of these could be thought of as a texton. Learning a dictionary of around 100 of these micro-structures should allow for the description of any texture. The
paper creates view and illumination invariant textons by collecting the filter responses of a bank of 48 filters from each pixel in a set of 400 100x100 pixel images producing a 48 length vector. The set of images consists of 20 images taken from different angles and lighting conditions of 20 textures. The responses from each pixel in each of the 20 images of each texture are concatenated to form a 960 length vector. The paper uses a 2 step clustering method based on k-means clustering to cluster the 960 length vectors into textons. First 400 cluster centres are found for each texture which in total across the 20 textures produces 8000 cluster centres. Centres too close to other centres or ones with too few data samples are removed until 100 centres remain. These 100 points are then used as the initialisation points for a second round of k-means clustering conducted on the concatenated filter responses from the 20 images for the full set of 20 textures.

Each material is characterised by a histogram recording the frequency of the occurrences of each texton. Sample textures can be classified using images of them. To successfully recognise each texton in the image texture 3 images of the texture taken with different conditions are used which results in a matching rate of 97%. If only one sample of the texture is available the matching problem becomes more difficult. Here an iterative approach is used, this assigns the probability of each texton being categorised correctly according to how well the overall histogram agrees to one of the known texture histograms; a 87% matching rate is achieved.

The weakness of this technique is that multiple reference images of a texture are required to capture its characteristics and, for best results, multiple sample images of the texture are also required. Neither of these is available in the current problem statement for this project.

Rubner and Tomasi [RT99] present a way of searching for textures within images based on a texton approach that uses different textons for each image. k-d tree [Ben75] clustering is used to split the pixel textures of the image into clusters of a roughly equal size; cluster centres close to each other are then combined. This results in a different number of clusters for each image, each with a weight depending on the number of pixels assigned to that cluster. This produces image descriptions using different texton dictionaries for each image which does not allow for the usual histogram comparison. A novel approach for comparing the texton descriptions of the two images is introduced. The distance between the histograms is found based on the distance between the texton centres and the number of pixels assigned to each texton centre.

Alternatively a filter bank defined in [VZ02] and shown in Fig. 2.6 can be applied at a range of scales. Each filter type detects a different type of image texture, filters that are rotationally variant are applied at a number of orientations and the entire filter bank is applied at a range of scales. The leftmost column in Fig. 2.6 shows the impulse responses of the Laplacian of Gaussian (LoG) texton filters defined by
The remaining texton filters in Fig. 2.6 detect either edges (upper row) or bars (lower row). Their impulse responses at an orientation of 0 degrees are given by

\[
\begin{align*}
\text{Edge}(x, y, \sigma) &= \frac{3}{\sigma \sqrt{2\pi}} e^{-\frac{y^2}{2\sigma^2}} \times -\frac{x}{\sigma^3 \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \ast I \\
\text{Bar}(x, y, \sigma) &= \frac{3}{\sigma \sqrt{2\pi}} e^{-\frac{y^2}{2\sigma^2}} \times \frac{x^2 - \sigma^2}{\sigma^5 \sqrt{2\pi}} e^{-\frac{x^2}{2\sigma^2}} \ast I
\end{align*}
\]

The standard deviations of the Gaussian used to construct the filters are \(\sigma_n = 2^{(n-3)/3}\) where \(n = [1, 2, \ldots 10]\). The filters are normalised to have a mean of 0 and a norm of 1.

Twelve rotational variants of the bar type filter in increments of 15° over 180° are used to capture the response over all orientations, also the negative versions of the bar filters give another set of twelve filters. As the edge filters have only one axis of symmetry, twenty four variants, again in increments of 15° over 360°, are needed. This results in five filter types, positive LoG, negative LoG, positive bar, negative bar and edge.

Each filter type is used to construct description vectors, the LoG filters are used to produce one each while the bar and edge filters produce two each, a scale and rotational component.
where \( n \) represents the scale and \( \theta \) represents the orientation.

### 2.6.6 Recognition of a specified type of object in an image

Techniques have been developed that rapidly identify a specific class of object in an image. It has been shown that the technique can be trained to recognise faces and cars [VJ01, BBEW13].

Statistical approaches known as cascade or boosting are commonly applied, there are a variety of alternatives and various versions of each, for example AdaBoost is used in [VJ01]. They all involve the combination of indicators to produce an overall probability. Each indicator is assigned a probability of correctly indicating the unknown quantity. Depending on the algorithm used it is not necessary to apply all indicators as they can be applied one at a time and if the combined probability at any point drops below a threshold the sample can be rejected as false. Simple filters as shown in Fig. 2.7 can be used as indicators, the choice of indicators and the probabilities assigned to each also need to be determined for the object type being detected. This method has been shown to work in real time for the recognition of faces in video due to the simple nature of the filters and the efficient combination of probabilities [VJ01].

The main weakness of these techniques for the purpose of change detection is the requirement to train the system to be able to detect every possible type of object of interest. This would involve a very large development effort and the testing for the existence of every possible object at every position in the sample and reference image would involve a large processing effort.
Figure 2.7: Filters used as indicators in face recognition [VJ01].

2.7 Similar applications

2.7.1 Vehicle mounted route change detection

Change detection systems which successfully compensate for small angle perspective changes and successfully integrate positioning and orientation data have been deployed in military applications. An example is a system mounted on a ground vehicle with a video camera integrated with GPS positioning, 3 axis magnetic compass and a computer mounted in the vehicle. A database of images is produced each time the vehicle is driven down a specific route. The positioning information is used to match images and transform them to compensate for perspective offset. The camera operates at 7.5 frames per second travelling at 40km/h. Objects as small as drinks cans at distances of 200 meters as well as vehicle tracks and disturbed ground are claimed to be spotted by the system [Val10].

2.7.2 Airborne multi-view change detection

Techniques for novel view synthesis and object detection have been suggested to be compatible with change detection. The synthesis of a 3D model using images obtained from a flight path as mentioned in section 2.2.2 of this report has been used to conduct change detection using standard pixel differencing techniques [DP10]. The vehicle recognition method described by [BBEW13] is also suggested to be useful in change detection.

2.8 Summary

In this chapter a number of approaches to change detection and similar problems have been reviewed including established 2D change detection methods and object recognition. A number of areas that could
be applied when developing a wide-baseline image change detection system have also been reviewed. Overall, although established change detection techniques exist, they are not well suited to the specific case of wide-baseline image change detection. In the remaining chapters some of these areas reviewed, such as projective geometry, registration, feature points and segmentation, will be referred to and used in the development of new methods for solving specific problems within the area of wide-baseline image change detection as well as the development of an end-to-end change detection system.
Chapter 3

Project Approach

This chapter presents the approach the thesis takes for both addressing the challenges involved in wide-baseline image change detection and also the evaluation of the solutions presented. The overall approach used is to attempt to register and match areas of the reference image and sample image and to then mark any areas that cannot be matched as change or if occlusion is detected to mark the areas as unknown.

3.1 Addressing key challenges

The introduction of this thesis highlighted the two key challenges involved in wide-baseline image change detection, the alignment of possibly match areas and the comparison of those areas to determine if they match or contain any changes. In this section the approach the thesis presents for addressing those challenges is presented and illustrated in Fig. 3.1. This thesis focuses on presenting novel approaches to solving these challenges to illustrate how these challenges can be overcome. The implementations of the approaches are proof of concepts rather than polished implementations and as such, computational efficiency in particularly in the choice of lower level algorithmic tools is not a key focus. Similarly the end-to-end system presented in the final chapter focuses on showing the potential when combining these approaches rather than a commercial implementation.

3.1.1 Understanding the scene geometry and image alignment

The approach the thesis takes to compensate for changes in viewing position and the resulting changes in the relative geometry of the camera and the scene is to use a combination of feature points and segmentation to align areas of the scene between the reference and sample image. Feature points are used to define points that correspond to the same position in the scene in the reference and sample images.
Once correspondences are found they are used to define geometric transformations which can then be used to register the areas of the images that lie between detected feature points. They are also used to understand the two view geometry which in turn restrains the transformations used to register the images in order to increase accuracy and to reduce the number of correspondences requires to calculate the transformations. The reliability of the feature points and their density have a large influence on the performance of subsequent steps and so methods are introduced to improve both these aspects of their performance. Object edges are a key challenge in the registration of two images of a scene as they form discontinuities in the registration offset between the images and also result in changes in occlusion. Segmentation can be used find where these boundaries lie by segmenting the images into similar areas. This will separate the images into separate areas of wall, ground roof and various other objects and areas. The boundaries between these areas will likely form discontinuities in the registration offset. The corresponding locations of edges in the reference and sample images are also used when defining the geometric transformations used for registration.

3.1.2 Robust comparison of aligned regions

As feature points are used to find correspondences between the images any errors in the corresponding positions on each image, known as their localisation error, will introduce errors in the resulting registration. Changes in lighting and other environmental effects will also create differences between the registered areas of the reference image and sample image. To distinguish between these differences and differences caused by changes in the scene the methods used to attempt to match the registered areas need to be robust to both registration errors and environmental changes.
Chapter 3. Project Approach

The SIFT descriptor is used as the basis of the region comparison techniques introduced in this thesis. The SIFT descriptor algorithm includes intensity normalisation and uses the local image gradient to construct its descriptor vector, this makes it robust to environmental effects, while the spatial and directional binning used make it robust to small registration errors. In this thesis the SIFT descriptor’s robustness to registration error is greatly increased by finding the registration compensation that result in the best match and tracking possible variations across the image areas compared. The possible registration offsets at each point in the image is used together with the expected range in rates of change of registration offset to a construct graph. The best global solution for the graph defines the best fit registration compensation across the whole image areas being compared. Graphs are also used to find the most likely locations of boundaries between change an no-change areas by using the likelihood of neighbouring areas having the same change classification together with local matching distances.

3.2 Evaluation

This thesis uses 5 datasets containing photographed objects used to evaluate algorithms, this section introduces them. The five datasets are (1) flat scene, (2) rotating scene, (3) rotating scene with trees, (4) larger rotating scene and (5) CDnet video change detection dataset. All images are captured using a SLR camera with a resolution of 3008x2000 pixels. The datasets present different challenges as explained in the sections below. The images are collected in studio conditions with controlled lighting. This minimises the effects of variation in lighting and contrast between images. Compensating for these factors outside the scope of this thesis as it is assumed that the images can be cleaned before being presented to the change detection system.

3.2.1 Flat scene

The flat scene image data is a set of images of a flat poster which is rotated in an axis at a known angle to the camera line. This provides images of a plane that undergoes a projective transformation without overall scale or rotation changes. It can be used to test the robustness of algorithms to geometric transformations. The advantage of using images of a plane is that the level of distortion is known across the image, unlike in more complex scenes where different areas undergo different levels of distortion depending on their shape and position relative to the camera. The central image is shown in Fig. 3.2 and the complete dataset contains a total of 80 with images rotated by 0°, 5°, 10°, 15°, 20°, 25°, 30°, 35°, 40°, 45°, 50°, 55°, 60°, 65°, 70° and 75° around an axis that is 90°, 75°, 60°, 45° and 30° to the camera optical axis and aligned to the y-axis of the image. All images have a resolution of 3008x2000 pixels.
Figure 3.2: Central image from flat scene dataset.

3.2.2 Rotating scene

The rotating scene dataset consists of images of a scene constructed using scale models such as those used by model railway hobbyists. The scene is rotated on a turntable and images are captured at 5° intervals of rotation. The scene contains 4 buildings, 6 cars and 5 people. This dataset is used to evaluate techniques on a 3D scene containing areas that are not flat and objects that provide a level of occlusion and self-occlusion. The level of occlusion between objects is relatively low as the buildings are relatively spaced out. The set of images is collected twice around 180° to give 72 images with a resolution of 3008x2000 pixels. The angle between the optical axis and the turntable axis is approximately 60°. The following changes occur between the image sets as can be seen in Fig. 3.3:

- House removed
- Shed removed
- Red car removed and replaced with a grey Jeep that is removed from elsewhere in the scene
- Green Jeep removed
3.2.3 Rotating scene with trees

As with the rotating scene dataset in Sec. 3.2.2 there are two sets of images of a model scene mounted on a turntable and images are collected at 5° intervals. This scene contains trees which are more irregular in shape than images in the rotating scene dataset and whose lack of planar surfaces makes consistent features more difficult to extract and the resulting occlusion and self-occlusion is more complex. Again the set of images is collected twice around 180° to give 72 images with a resolution of 3008x2000 pixels and changes made between the two image sets as can be seen in Fig. 3.4. The changes are as follows:

- Green Jeep and red car switched
- Black car moved
- White car moved
- Sand pile added
- One worker removed and box placed on ground
- Shed between trees removed
3.2.4 Larger rotating scene

The larger rotated scene datasets similar to the rotating scene dataset but contains a larger scene with more objects that are more closely spaced. This provides a more challenging scenario than the rotating scene while still mainly excluding object with very irregular shapes such as trees. The levels of occlusion between objects is higher as the buildings are close enough to each other to occlude each other. The scene is rotated in around 360° and the position of the camera relative to the centre of scene is similar to the other datasets to give a total of 144 images with a resolution of 3008x2000 pixels. Sample images from both sets of images are shown in Fig. 3.5. The changes are as follows:

- Removed barn
- Removed workers
- Fire engine moved
- Red and white cars switched
- Shed removed
- Black car removed
- Garage removed

3.2.5 CDnet video change detection data

The CDnet [GJFP12] dataset consists of a number of videos that present different challenges in the area of video change detection. Each pixel in each frame of each video has been assigned a ground truth change classification. While most of the videos involve a static camera and so no significant registration between frames is required, the pan tilt zoom (PTZ) video requires significant registration. Although the
camera used in the PTZ video has a static position and so the registration task is simpler than in the datasets presented above, the dataset does provide real world, outdoor images and allows for comparison with other algorithms that have been tested against the dataset.
Chapter 4

Feature Matching

Image change detection relies on the ability to compare images of the same scene taken at different times. In order to compare two unregistered images it is necessary to locate points in the two images that correspond to the same positions in the scene. This provides a starting point from which regions of the scene can be aligned in order to register and match areas. It also provides correspondences between the two images for the calculation of elements of the two view geometry such as homographies of any planes in the scene and the fundamental matrix which describes the camera motion between the two images. These in turn can be used to define other elements of the geometry of the scene itself such as the positions of objects and planes as described in in Sec. 6.

4.1 Affine Compensated SIFT

Established methods of finding correspondences using feature point detection and matching perform poorly at wide viewing angles. For example the number of SIFT points matched can drop from over 1500 at a viewing angle of 5° to under 150 at 20°. Viewpoint-robust methods based on region detection, for example MSER, suffer from the lower density of feature points available before descriptors are calculated or matched, in an example image around 500 are detected compared to over 30000 with SIFT. In this section we therefore propose an extension to the SIFT algorithm that makes matching less sensitive to viewpoint changes while retaining or improving the density of available feature points.

4.1.1 Aim

As previously stated, the approach used in this thesis requires the ability to reliably collect corresponding points, known as feature points, between the sample and reference images when there are wide angled
viewing differences between the images. Increasing the number and density of the feature point correspondences gives several advantages. It allows the camera geometry of the two view system to be more accurately defined; it is also important when using the segmentation and registration algorithms in Chapter 6 Segmentation and Registration. It also allows for more accurate calculation of the fundamental matrix and camera parameters. These are used for the verification of point correspondences and so has a feedback effect on their reliability.

Current feature point techniques either do not perform well at wide angles or do not provide a high density of points. In either case the overall density of points is low. The aim is to find a novel method of obtaining and matching feature points that results in a large number of matched points at wide baselines.

4.1.2 The use of SIFT to compare images

A widely used technique for finding point correspondences is SIFT [Low04] reviewed in Sec. 2.3.1 which both identifies distinctive points within an image and provides a mechanism for matching them between images. The SIFT algorithm is robust to translation, rotation, scale changes and lighting changes, however, while it is robust to smaller changes in viewing angle, such as reasonable differences from one frame to another of a video, it is not robust to larger changes in camera viewpoint between two images without the availability of intermediate images. As will be shown in Sec. 4.1.5, the number of matched SIFT points decreases rapidly with a change in viewing angle from approximately 10000 at 5° to under 100 at 30°. While 100 feature points provides enough data to find the geometry of the two camera system a higher density is required to find the geometry in 3D space of different objects in the scene and to find local estimates of the geometry of non-planar surfaces.

4.1.3 Problem and Approach

The relationship between two images of a plane taken from different viewing angles is described by a homography which has 4 degrees of freedom as described in Sec. 2.1.2 This distortion as well as the effects of occlusion are a principle cause of the reduction in matched points at wide viewing angles. The resulting projective distortion can be approximated with an affine transformation that has 3 degrees of freedom, this results in small approximation errors provided that

- The depth relief is small compared to the average depth.
- The distance of the point from the principle ray is small [HZ04].

This can be shown by first defining the situation where the behaviour of the projective camera behaves as an affine camera. A projective camera shown in Fig. 4.1 can be defined as
Figure 4.1: Projective camera imaging a plane where $d$ is the distance of the plane from the camera along the camera line, $x$ is the width of the plane, $f_d$ is the focal length and $c$ is the camera position.

$$P_0 = K \begin{bmatrix} r_1^T & r_1^T c \\ r_2^T & r_2^T c \\ r_3^T & r_3^T c \end{bmatrix}$$

(4.1)

while an affine camera can be defined as

$$P_A = K \begin{bmatrix} r_1^T & r_1^T c \\ r_2^T & r_2^T c \\ 0 & r_3^T c \end{bmatrix}.$$  

(4.2)

$K$ represents the internal parameters of the camera and the matrix on the right represents the external geometry where $r_n$ represent the rows of a 3D rotation matrix, representing the camera’s orientation relative to the coordinate frame, and $c$ represents the position of the camera. The principle ray or camera line of the camera is parallel to $r_3$. The term $-r_3^T c$ represents the distance of the camera from the point, denoted $d_0 = -r_3^T c$ where the camera line crosses the plane, this crossing point is also defined as the world origin. Changing the value of $-r_3^T c$ represents moving the camera towards or away from the plane without changing the angle to the plane or where the camera line crosses the plane. $d_0$ can be multiplied by a scaling factor, $d$ to represent this motion along the camera line. An alternative method of scaling the resulting image is to use a zoom which changes the lens focus and in turn the focal length.
of the camera. By using a zoom the focal length of the camera can be scaled by a scaling factor, \( f_d \). This also changes the overall scale of the image but results in an image different to the one resulting from moving the camera; this difference is due to perspective effects. This perspective effect is represented in a projective camera but not in an affine camera [HZ04]. Including the scaling factors \( d \) and \( f_d \) results in the projective camera matrix;

\[
P_0 = \begin{bmatrix}
  f_d & 0 & 0 \\
  0 & f_d & 0 \\
  0 & 0 & 1
\end{bmatrix} \begin{bmatrix}
  r_1^T \\
  r_2^T \\
  r_3^T
\end{bmatrix} = f_d K \begin{bmatrix}
  r_1^T \\
  r_2^T \\
  r_3^T / f_d
\end{bmatrix} \begin{bmatrix}
  r_1^T c \\
  r_2^T c \\
  -dd_0 / f_d
\end{bmatrix}
\]

where \( d_0 = (-r_1^T c) \). By altering \( d \) and \( f_d \) together the overall scale of the image is retained but projective effects cause the resulting images to differ, this difference represents the error when using an affine approximation. When they both tend towards infinity the result is an affine camera

\[
P_\infty = f_d K \begin{bmatrix}
  r_1^T \\
  r_2^T \\
  0
\end{bmatrix} \begin{bmatrix}
  -r_1^T c \\
  -r_2^T c \\
  -r_3^T c
\end{bmatrix}.
\]

We now know that a projective camera at infinite distance from a plane behaves as an affine camera. Next we show that a point on a plane orthogonal to the camera line images to the same point when using an affine and projective camera. Any point, \( x \) on the plane that passes through the origin and is orthogonal to \( r_3 \) can be written as \( x = \begin{bmatrix} \alpha r_1^T + \beta r_2^T & 1 \end{bmatrix}^T \). Projecting \( x \) using \( P_0 \) gives

\[
P_0 x = K \begin{bmatrix}
  r_1^T \\
  r_2^T \\
  r_3^T
\end{bmatrix} \begin{bmatrix}
  \alpha r_1 + \beta r_2 \\
  1
\end{bmatrix} = K \begin{bmatrix}
  u \\
  v
\end{bmatrix}
\]

\[
P_0 x = K \begin{bmatrix}
  u \\
  v
\end{bmatrix} = K \begin{bmatrix}
  r_1^T (\alpha r_1 + \beta r_2) + dr_3^T c
\end{bmatrix}
\]

since \( r_3^T (\alpha r_1 + \beta r_2) \equiv 0 \)

\[
P_0 x = K \begin{bmatrix}
  u \\
  v \\
  dr_3^T c
\end{bmatrix}.
\]
Chapter 4. Feature Matching

Projecting $x$ with an affine camera $P_A$ gives the same result

$$P_A x = K \begin{bmatrix} r_1^T & r_1^T c \\ r_2^T & r_2^T c \\ 0 & dr_3^T c \end{bmatrix} \begin{bmatrix} \alpha r_1 + \beta r_2 \\ 1 \end{bmatrix}$$

$$P_A x = K \begin{bmatrix} u \\ v \\ dr_3^T c \end{bmatrix}$$

Imaging a point on a plane orthogonal to the camera line with a projective or affine camera yields the same result. The affine approximation error can be found by considering the displacement of the point from this plane in the direction of $r^3$ by a distance $\gamma$, otherwise known as the depth relief. The difference between the resulting imaged point when using a projective or an affine camera gives the error, $\epsilon_r$. Using equations 4.1 and 4.2:

$$P_0 x = P_A x + \epsilon_r$$

$$K \begin{bmatrix} r_1^T & -r_1^T c \\ r_2^T & -r_2^T c \\ r_3^T & -r_3^T c \end{bmatrix} \begin{bmatrix} \alpha r_1 + \beta r_2 + \gamma r_3 \\ 1 \end{bmatrix} = K \begin{bmatrix} r_1^T & -r_1^T c \\ r_2^T & -r_2^T c \\ 0 & -r_3^T c \end{bmatrix} \begin{bmatrix} \alpha r_1 + \beta r_2 + \gamma r_3 \\ 1 \end{bmatrix} \epsilon_r$$

$$\begin{bmatrix} f_d & 0 & 0 \\ 0 & f_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha - r_1^T c \\ \beta - r_2^T c \\ \gamma - r_3^T c \end{bmatrix} = \begin{bmatrix} f_d & 0 & 0 \\ 0 & f_d & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \alpha - r_1^T c \\ \beta - r_2^T c \\ -r_3^T c \end{bmatrix} \epsilon_r$$

Converting into Euclidean space and re-arranging yields the value of $\epsilon_r$: 
\[
\begin{bmatrix}
    f_d (\alpha - r_1^T c) / (\gamma - r_1^T c) \\
    f_d (\beta - r_2^T c) / (\gamma - r_2^T c)
\end{bmatrix}
= \begin{bmatrix}
    -f_d (\alpha - r_1^T c) / r_1^T c \\
    -f_d (\beta - r_2^T c) / r_2^T c
\end{bmatrix} \epsilon_r
\]

\[-r_3^T c \begin{bmatrix}
    f_d (\alpha - r_1^T c) \\
    f_d (\beta - r_2^T c)
\end{bmatrix}
= (\gamma - r_3^T c) \begin{bmatrix}
    f_d (\alpha - r_1^T c) \\
    f_d (\beta - r_2^T c)
\end{bmatrix} \epsilon_r
\]

\[-r_3^T c / (\gamma - r_4^T c) = \epsilon_r
\]

The term \( \gamma \) represents the error caused by the affine approximation. The smaller \( \gamma \) is the smaller the error and the larger \(-r_3^T c\), which represents the distance to the plane from the camera along the camera line, the smaller the impact of the affine approximation.

So if the depth of the plane from the camera along the camera line \(-r_3^T c\) is large compared to \( \gamma \) the affine approximation error is small. When using images that show a scene taken at a distance where there is no foreground of interest and there are no extreme variations in depth the affine approach gives a good approximation. In order to find a way of making SIFT robust to affine distortions, the way SIFT is not robust to affine transformations needs to be understood. As explained in Sec. 2.1.2.2, using an affine transformation the image is translated, scaled, rotated and foreshortened along an axis. This can be illustrated by decomposing the affine transform \( A \) into rotation matrices represented by \( R \) and a diagonal scaling matrix \( D \) that changes the overall scale by a factor \( s \) and foreshortens it in the rotated \( x \)-axis by \( t \). Where the angle of foreshortening is given by \( \varphi \) and the rotation after foreshortening is given by \( \theta \)

\[
A = R (\theta) R (-\varphi) DR (\varphi)
\]

\[
D = \begin{bmatrix}
    \frac{1}{s} & 0 & 0 \\
    0 & 1 & 0 \\
    0 & 0 & \frac{1}{s}
\end{bmatrix}
\]

SIFT is robust to changes in the overall translation, scale and rotation and so if it can also be made to also be robust to the additional effect of foreshortening along an axis, it can be made to be robust to affine distortions. As SIFT is robust to rotation, \( R (\theta) \) can be ignored, also as SIFT is robust to changes in scale only the foreshortening components, \( t \) of \( D \) needs to be included which leaves
Figure 4.2: Change in imaged size of plane based on viewing angle of $\theta$ in an affine camera.

\[
A' = R(-\varphi) \begin{bmatrix} \frac{1}{t} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} R(\varphi)
\]

Applying $A'$ to a planar surface results in a foreshortening of the surface by $t$ in a direction $\varphi$ from the $x$-axis. The direction of foreshortening aligns with the direction of the camera motion between the two camera positions and the magnitude depends on the magnitude of the motion and the original orientation of the plane as illustrated in Fig. 4.2.

**Figure 4.2 illustrates the affine approximation of the effect of imaging a plane** from an angle $\varphi$ from the plane normal when the scale of the image remains constant (the equivalent of having a focal length equal to the distance of the plane from the camera in a projective model). It can been seen that the plane is foreshortened by $\cos\varphi$. A further viewing angle change of $\alpha$ will foreshorten the plane by $\cos(\varphi + \alpha)$ from its original size, less than an additional $\cos\alpha$. It can also be seen that the foreshortening occurs in the direction of the camera’s displacement from the line normal to the plane at the point the camera line intersects the plane. If the camera viewpoint is moved the value of $\varphi$, and hence the amount of foreshortening will change. The foreshortening will depend on the the original viewing angle as well as the motion and will differ for each plane in the image, but the direction of foreshortening will follow the movement of the camera.

### 4.1.4 ASIFT algorithm

The foreshortening component of the affine transformation can be modelled in the reference image by foreshortening the image by a number of foreshortening factors in the direction $\varphi$, the direction the camera moves between image captures. If $\varphi$ is not known a hemisphere of foreshortened versions can be found that will provide a close match to all possible foreshortening directions. This produces a number of adjusted reference images and is the basis of the proposed extension of SIFT denoted Affine Compensated
Scale-Independent Feature Transform (ASIFT).

The first step is to choose the foreshortening directions. If the direction of the viewpoint change is known in advance, this direction can be used otherwise a range of possible directions is used. In the general situation it is assumed here that intervals of $\pi/4$ are used unless otherwise stated. The original reference image is rotated by $\pi/4$ to give a second image. The images are both foreshortened in both the $x$-axis and $y$-axis by $\cos \varphi$ where $\varphi \in \{\pi/9, 2\pi/9, \pi/3, 4\pi/9\}$ to give a total 17 images. It is not necessary to consider stretching the image since a stretch in one direction is equivalent to a foreshortening in the orthogonal direction to within a scale change and SIFT is invariant under scale changes.

SIFT points and descriptors are collected from the extended set of reference images to form an extended set of dictionaries. The formation of these dictionaries can be carried out offline before the sample image is available. Each SIFT descriptor from the sample image is matched to each of the dictionaries following the same euclidean squared procedure as in the SIFT algorithm. The descriptor with the closest Euclidean squared distance in each dictionary is found. As with standard SIFT, if the distance to this descriptor is below a decision threshold $d_{\text{thresh}}$ times closer than the next closest descriptor match, it is accepted as a match. When a sample point is matched in more than one of the altered reference image dictionaries the match with the smallest matching distance is used.

### 4.1.5 Evaluation of ASIFT

ASIFT is evaluated using the 5 data sets introduced in Sec. 3.2. The performance is compared to SIFT, SURF and ORB, which are all reviewed in Sec. 2.3, across viewing angles up to $40^\circ$ after which point the performance of ASIFT also degrades. In order to maintain a high matching precision, the matching threshold is set so that the squared euclidean distance of the second closest match must be at least 3 times the closest match for the closest match to be accepted. Errors are estimated by testing for consistency with an estimate of the fundamental matrix. A fundamental matrix is found using a RANSAC based approach so that outliers can be excluded as errors while including as many good matches to minimise the effects of localisation errors. Although this will not detect errors that are displaced from the correct position along the direction of the epipolar line, the number of errors found should remain proportional to the actual number of errors. If the number of the correspondences is low or the rate if errors is high an accurate fundamental matrix may not be found which may result in poorly performing feature point detectors being handicapped at large angles. Results will be shown with the number of matches after removing matches classified as erroneous. In all the results collected the number of false matches was found to be low and consistent across viewing angle and feature point type. For this reason the number of correct matches was chosen as the measure used to evaluate performance.

The first image dataset used is the flat scene dataset which contains images of a poster rotated through
Chapter 4. Feature Matching

![Graph showing number of matches vs scene rotation angle]

**Figure 4.3:** ASIFT performance compared to common feature point detectors using the flat scene dataset.

increments of $5^\circ$. The images where the poster is rotated on an axis orthogonal to the camera axis are used. As the images in this dataset only contain a single plane the entire image area undergoes the same homography resulting from a known change in the angle between the camera line and plane normal. This allows for a very controlled assessment of the density of the matched feature points using each algorithm at each viewing angle. The rotation of the plane is such that the direction of affine compensation aligns with the direction that the plane appears to be foreshortened in the image. This is the optimum situation for ASIFT as it means that the direction in which the algorithm compensation acts is aligned with the principle direction of the distortion.

**ASIFT is configured so that there are 2 compensation axis for 90° ASIFT, 4 for 45° ASIFT and 8 for 22.5° ASIFT.** The number of correct matches are presented in Fig. 4.3 on a logarithmic scale and show that ASIFT significantly outperforms SIFT, SURF and ORB at all angles with the performance gap increasing with viewing angle. SIFT also significantly outperforms SURF and ORB which justifies the use of the SIFT feature point as the basis of ASIFT. ASIFT with 45° intervals between compensation directions slightly outperforms the version with 90° intervals which was not expected as the main distortion in the image is aligned to one of the compensation directions in both these versions. This may be explained by better tolerance to projective effects which will distort the scene locally in some areas of the image as parts of the scene move towards or away from the camera.

The optimum number of compensation directions is now assessed by comparing performance on the flat scene dataset. The reference image is rotated so that the rotation axis is aligned to the worst case orientation for the compensation axes configuration, where the apparent foreshortening of the imaged plane lies at an angle half way between compensation axes. The results shown in Fig. 4.4 show an increase in performance as the number of compensation axes increases. The performance gain decreases
Figure 4.4: Comparison of performance of ASIFT configured to compensate in directions separated by 90°, 45° and 22.5° degrees. Tested on the flat scene dataset.

as the number of axes increases while the computational load doubles with each step as points and descriptors have to be calculated and matched against twice as many images. The number of errors also increases as the number of compensation axis increases, for example with a 10° viewing angle 90° ASIFT results in 106 errors while 45° ASIFT results in 155 and 22.5° ASIFT results in 246. This shows that the performance can be boosted by increasing the number of compensation axis but this needs to be weighed against an increase in the number of errors and the increased computational overhead. The results are also compared to other common feature point algorithms in Fig. 4.5 and show significantly better performance.

In the following figures (Fig. 4.6, Fig. 4.6 and Fig. 4.8) more complex images containing three dimensional scenes are used to illustrate performance in more realistic scenarios. These scenes contain many approximately planar areas with different orientations as well as curved surfaces and complex textured areas. As reviewed in Sec. 2.1 the homography acting on planes within a scene between camera views varies with the orientation and depth of the plane and so the performance of a feature point algorithm will be affected to different extents in different places. It is hoped that robustness to curved and textured areas that will undergo more complex distortions which are not modelled explicitly in ASIFT will also see performance gains.

It can be seen that ASIFT significantly outperforms the other descriptors in all three datasets. Again SIFT also significantly outperforms SURF and ORB with reasonably consistent performance across the three datasets. The benefit of ASIFT with 45° intervals over 90° intervals can now more clearly be seen, indeed the performance gain over standard SIFT approximately doubles. Although ASIFT aims to increase the performance of SIFT at wide angles we see that even at small rotation angles ASIFT
Figure 4.5: ASIFT performance compared to common feature point detectors using the flat scene dataset tilted by 22.5°.

Figure 4.6: ASIFT performance compared to common feature point detectors using the rotated scene dataset.
Chapter 4. Feature Matching

Figure 4.7: ASIFT performance compared to common feature point detectors using the larger rotated scene dataset.

Figure 4.8: ASIFT performance compared to common feature point detectors using the rotated scene with trees dataset.
achieves more matches than SIFT. The benefit of ASIFT increases with the viewing angle difference.

4.2 Increasing Feature Point Density

In addition to implementing ASIFT which provides an improved tool for providing correspondences between images at wide angles, there are additional methods that can be used to improve the point density. Improving point density is especially important particularly in areas with textures that produce few correspondences and in areas with more complicated scene geometry. A larger number of points also reduces the impact of any incorrect matches when using methods, such as random sample consensus (RANSAC) [FB81], that reject inconsistent matches. In this section we look at ways of improving the density of matched points.

4.2.1 Ground plane pre-compensation

This thesis mainly focuses on situations where the scene of interest is an outdoor scene consisting of a ground plane with objects such as cars or buildings positioned on it. In these situations a large proportion of the imaged scene will lie on the ground plane or be displaced by a relatively small distance from it. In two-view geometry, the change in the imaged position of points lying on a plane between the two images can be described by a homography with points close to the plane also closely following the homography. If the image is compensated using this homography, any feature points on or close to the plane are more likely to match, as the majority of the projective distortion in the area of the descriptor window will be removed.

Using an initial set of ASIFT points a ground plane homography can be found that agrees with the largest number of point correspondences. A RANSAC based approach is used as a large number of points not on the plane are expected. In many cases there will be more points not on the plane than on the plane. In these situations a least squared error or other similar cost function will not fit well. It is assumed a large number of correspondences are available from an initial uncompensated application of ASIFT and that a large subset lie on the same plane. A large number of candidate homographies are calculated (around 100) each using a different combination of correspondences with four point correspondences required to define each homography. To improve performance it is ensured that the minimum distance between two correspondences used to calculate each homography is no less than 1/4 of the image’s smallest dimension. This ensures that localisation errors in correspondences have a reduced impact on each degree of freedom of the homography and that local planes in areas with high point densities are not found. Homographies are calculated using each of the set of four point correspondences, the homographies are then used to project the entire set of correspondences from the reference image to the sample image. Each point
projected to within 4 pixels of it’s matched position is marked as agreeing with the homography. The homography that agrees with the largest number of points is used to exclude points not on the plane from the final calculation. Finally all points marked as being on the plane are used to calculate the output homography.

Once the best fitting homography is found, the sample image can be transformed and interpolated so that the ground plane of the sample image matches that of the reference image, an example of a reference image, sample image and transformed sample image from the larger rotating scene dataset is shown in Fig. 4.9. You can see that the orientation of areas such as the central road align closer between the sample and compensated image than between the reference and sample image. Also the shapes of areas such as the warehouse roof are more similar. A new set of feature points can now be found and matched between the reference image and transformed sample image using ASIFT. The new set of sample image points can be transformed to the coordinate frame of the original sample image using the inverse of the ground plane homography. The set of points found using the transformed sample image can now be combined with the original set found before the image is transformed to further increase the matched feature point density. This combination is required because although the intention is that the transformed image should match better, some areas, for example vertical planes within the scene may in fact match better before the image is transformed. Results are shown in Sec. 4.2.3.

4.2.2 Affine Harris feature points

When finding and matching feature points using the SIFT, SURF, ORB or a similar algorithm, the feature point detection step can be separated from the descriptor calculation step. For example, in SIFT a descriptor is attached to each feature point detected by finding maxima and minima in a DoG filtered, as described in Sec. 2.3.1, version of the image at a number of scales as described in Sec. 2.3. It is possible to use a feature point detector from one algorithm with a descriptor from another algorithm. The aim of this section is to increase feature point density, particularly areas where density is low. The density of discovered points will naturally vary in different areas of scenes, using more than one type of feature point detector that detect different types of feature point can even out this variation. Corner point detectors
such as Harris points [HS88] complement SIFT points well as they detect different types of points that are likely to occur in different areas. However the Harris point was not implemented with a descriptor. SIFT points tend to locate in the centre of image blobs at various scales while Harris corner points find corners which tend to locate at object edges. The SIFT descriptor can be adapted to be robust to changes in viewing angle as shown in Sec. 4.1 and so if this same adapted SIFT algorithm is attached to the Harris point the resulting hybrid feature point should remain robust to changes in viewing angle, this will be referred to as A-Harris.

In order to combine a SIFT descriptor with a Harris point detector, the descriptor requires the location, scale and orientation of the feature point. The Harris corner point detector only provides the location. The orientation can be found using the direction of the local gradient in the same way as in the SIFT algorithm. If the image is pre-compensated with the ground plane homography as shown in Sec. 4.2.1 and the point is close to the plane relative to the image depth, the scale will not vary significantly between the images. The results of combining the ASIFT feature point with with the A-Harris feature point is shown in Sec. 4.2.3.

4.2.3 Combined results

In this section the effects of combining ASIFT with compensating the image using the ground plane homography and with A-Harris is measured and shown in graphs with a logarithmic y-axis. Because more than one set of feature points are combined, duplicate points can exist. For this reason, when more than one point is within an area of 3 pixels, all but one point is removed using an arbitrary selection. A threshold of 3 pixels was used as this was found to be the maximum localisation error found when using SIFT when the match still seemed to be correct. Across all results a large number of points are removed due to this criteria, although this results in a reduction in feature point density closely located points are not useful when calculating scene or camera geometry as any localisation error, even sub pixel error, will be large compared to inter pixel distances. As with the assessment of ASIFT in Sec. 4.1.5 the fundamental matrix is found between the two images and points that contradict it are are classified as erroneous matches.

The performance on the flat and tilted flat scene dataset is shown in Fig. 4.10 and Fig. 4.11 respectively. They show a significant improvement from pre-compensating using the homography. This is to be expected as the entire scene lies on a single plane and so the whole scene is well compensated for by the ground plane homography. Not many additional points are found by combining with A-Harris, this may be because the image doesn’t contain many well defined corners. There is no significant difference between the performance on the aligned and tilted flat scenes.

In Fig. 4.12 the performance is illustrated on the rotating scene dataset. In these results an improvement
Figure 4.10: Evaluation of A-Harris and compensating the ground plane on the number of captured feature points on the flat scene dataset. The plot labelled ASIFT shows the performance when only ASIFT is used, ASIFT + H-Comp shows the performance shows the combined performance of ASIFT and ground plane compensation, ASIFT + A-Harris shows the combined performance of ASIFT and A-Harris points and Combined shows the performance when all 3 methods are used.

Figure 4.11: Evaluation of A-Harris and compensating the ground plane on the number of captured feature points on the tilted flat scene dataset.
of combining A-Harris together with ASIFT can now be seen. This may be because the scene now contains well defined corners. The total number of points at 30° on the rotating scene increases from 485 with ASIFT to 879 when combined with ground plane compensation and A-Harris.

The larger rotating scene and the rotating scene with trees present more difficult tests as shown by the more rapid decrease in feature point matches with an increase in viewing angle, results are shown in Fig. 4.13 and Fig. 4.14 respectively. These more difficult scenes show a larger performance gain. A reduction in performance occurs at 35° in Fig. 4.13 which is likely due to the ground plane homography being inaccurately calculated, but even in this case the performance is still an improvement over basic ASIFT.

Next we will look at an example image showing point matches on the large rotating scene dataset with a 30° angle between views. In Fig. 4.15, examples of point matches using ASIFT on the larger rotating scene dataset with a 30° azimuth rotation between images is shown and in Fig. 4.16 the same is shown with the addition of A-Harris and ground plane compensation. As before, matches that are incompatible with the fundamental matrix are classified as erroneous and are removed. It can be seen that significantly more points are found with the number of matches (after errors are removed) increasing from 193 to 400. Using standard SIFT only 39 matches are detected of which 25 are classified as errors, although with 39 available points it is difficult to find the fundamental matrix reliably. As expected a large number of the additional points are on the ground plane where the ground plane compensation has the largest effect, but there are also significantly more points on buildings and other areas not on the ground plane.
Figure 4.13: Evaluation of A-Harris and compensating the ground plane on the number of captured feature points on the larger scene dataset.

Figure 4.14: Evaluation of A-Harris and compensating the ground plane on the number of captured feature points on the rotating scene with trees dataset.
4.3 Feature points summary

In this chapter SIFT, a well established feature point detection and matching method, has been extended so that the number of correctly matched feature points is significantly increased when matching between images with large differences in viewing angle, for example at 30° from ~10 to ~100. This was achieved by affine compensating the reference image before the collection of SIFT descriptors to give ASIFT and also by increasing the number of points available for matching by using Harris points and by pre-compensating the reference image using the dominant ground plane. This improves the performance in applications where there is a wide viewing difference between images. In particular this is important when using still images rather than video where the change in viewing angle between frames is typically very small. Wide baseline change detection using a single reference and sample image is an application area where increasing robustness to wide viewing angles is important in the approach presented in this thesis as feature point correspondences form a starting point for registering and comparing regions and are also used to find the camera system and target scene geometry. If the density of points drop the robustness of camera system geometry calculations to incorrect matching and localisation errors reduces. When finding the scene geometry the density of points needs to be high enough to provide enough points to estimate the local geometry. For example each plane in the scene requires 3 points to define an affine
transformation to register the plane between the two images. The more complex the scene the greater
the required density of points. If a greater density feature points is available to the system the system
can be applied to more complex scenes.
Chapter 5

Region Comparison

In order to use images to detect changes in a scene, a method of comparing sections of images to determine if they match is needed. In this chapter, methods for conducting this comparison are presented. In particular the methods are designed to be robust to the registration errors that normally result from the registration techniques introduced in Chapter 6.

Most of the methods presented here make use of SIFT descriptors [Low04]. The aim is to develop techniques for determining if two regions match that are robust to the presence of pixel noise, interpolation noise and registration error ($E_r$) as well as other forms of noise. In the Dense SIFT (DSIFT) and Shifted DSIFT (SDSIFT), matching algorithms described in Secs. 5.2 and 5.3 below, the SIFT descriptor is used to form a dense grid of descriptors known as DSIFT. The dense grid is formed by calculating the SIFT descriptor at scale $D_s$ at a defined shift, $D_\Delta$, along the $x$ and $y$ axes of the region. Decreasing the value of $D_\Delta$ increases the number of descriptors available but will increase the similarity between neighbouring descriptors due to the descriptor windows overlapping and the increase in any correlation between descriptors. Once DSIFT descriptors are obtained over the pair of image regions they can be compared pairwise using the euclidean squared distance between their vectors to give a dense grid of descriptor matching distances. The design and performance of these algorithms depend on the performance of the SIFT descriptor, the chapter therefore begins with an evaluation of SIFT descriptor statistics. The findings of this evaluation are then used in the design of the subsequent region comparison methods.

The performance of the algorithms is determined by their ability to determine whether the matching hypothesis $H_1$ or non-matching hypothesis $H_0$ is the correct hypothesis, given the state $S$ which can be matching, $S = 1$, and non-matching $S = 0$. Their performance is measured by the proportion of pixels incorrectly classified as false positives where an area without change is not matched and as false negatives where an area with change is matched.
Chapter 5. Region Comparison

5.1 Probabilistic assessment of the SIFT descriptor

The purpose of this section is to investigate the statistical properties of SIFT descriptors collected from an image over a dense grid. The findings are used in the later sections of this chapter in the design of novel area matching techniques and to determine the performance of the approaches in terms of their expected error rates and the robustness of their performance to registration error.

First the separability of a single pair of SIFT descriptors drawn from reference and sample images is measured. When they are compared they result in a single Euclidean squared matching distance in the descriptor vector space. Next the correlation of how the resulting matching distances across a pair of images vary with respect to their separation in pixels is found. The correlation of the resulting distances illustrates neighbouring descriptors independence when used together to classify a region as matching or non-matching. The distance measure’s independence determines the extent to which the performance is improved by combining matching distanced from neighbouring descriptors. Finally the correlation between the neighbouring descriptors themselves, within a single image, with respect to pixel separation is found. As registration error will cause descriptors to not align accurately between the reference and sample images the similarity of neighbouring descriptors can be used to model the robustness of the matching distance to registration error.

5.1.1 Separability of single SIFT descriptor

The matching performance of a single SIFT descriptor is the probability of correctly of assigning the correct hypotheses, \( H_1 \) or \( H_0 \), for an image using a single SIFT descriptor. In this section the separability is measured by finding the matching distance probability density function in the case of the state, \( S \).

In order to determine the separability of pairs of images or image regions using a single SIFT descriptor the distribution of matching scores when comparing pairs of SIFT descriptors gained from matching and non-matching pairs of images needs to be found. The distribution of matching distances is found experimentally using pairs of images from \( 10^6 \) azimuth rotations from the datasets introduced in Sec. 3.2. 10 flat regions, each consisting of around 10 000 pixels, are selected from the images and are registered using homographies defined by 4 manually confirmed SIFT feature point matches. As correctly matched SIFT points can have a registration error of up to 2 pixels the registration will not be perfect. A dense grid of SIFT descriptors with a 1 pixel separation between descriptors are computed from both image regions and corresponding descriptors are compared to find the matching distances. A 1 pixel separation is used to generate the largest possible set of data. When the distances are collected from a number of images the distribution of scores can be used to estimate the probability density function (PDF). To find the distribution of distances from non-matching images a dense grid of SIFT descriptors is computed.
on pairs of non-matching images. Each descriptor from one image can be matched to each descriptor in the other image and visa versa to produce a large number of matching distances. Again the matching distances can be used to estimate a PDF. The separability of the two distributions can be found using a Bayesian formulation where \( i \) is a single matching distance.

\[
P(H_1 \mid i) = \frac{P(S = 1) p(i \mid S = 1)}{P(S = 1) p(i \mid S = 1) + P(S = 0) p(i \mid S = 0)}
\]

\[
P(H_0 \mid i) = \frac{P(S = 0) p(i \mid S = 0)}{P(S = 1) p(i \mid S = 1) + P(S = 1) p(i \mid S = 0)}
\]

Fig. 5.1 shows the recorded probability densities \( p(i \mid S = 1) \) and \( p(i \mid S = 0) \) at SIFT scale 5. The horizontal axis is limited to 52500, the maximum distance between two SIFT descriptors. There is a small spike in \( p(i \mid S = 0) \) at this value. The PDF in Fig. 5.1 shows that the mode of \( p(i \mid S = 0) \) is not zero but is around 50 000 which indicates that sources of noise introduce a significant variation between descriptors captured at the same point. \( p(i \mid S = 1) \) has a long tail which extends to 300 000 so that, even in registered images, large matching distances are quite common. There is a significant overlap between the distributions which means that a single pair of descriptors cannot be reliably used to assign the correct hypothesis.

To use the SIFT matching distance for classification a decision threshold for the likelihood ratio, \( D_t \) needs to be chosen. The predicted false negative and false positive rate can then be defined from the likelihood ratio

![Probability density of scale 5 SIFT matching score from matching and non-matching pairs of images](image)

Figure 5.1: Probability densities of SIFT matching scores for matching and non-matching images at SIFT scale 5.
\[
PR = \frac{P(H_1 \mid i)}{P(H_0 \mid i)}
\]

\[
P(H_0 \mid S = 1) = p(\lceil P_R \rceil S = 1 \leq D_1)
\]

\[
P(H_1 \mid S = 0) = p(\lceil P_R \rceil S = 0 > D_1)
\]

The chosen decision threshold \(D_1\) is set depending on the desired error rates as well as the priors \(P(S = 1)\) and \(P(S = 0)\). By varying \(D_1\) a ROC curve can be produced for each scale as shown for scale 2, 4, 6 and 8 descriptors in Fig. 5.2. The equal error rate for scales 1 to 10 are also shown in Fig. 5.2 with performance increasing with scale. This suggests that larger scales produce better separability as fewer errors are produced. This may be because larger scales are less sensitive to any registration errors that may remain. Also a larger descriptor window encompasses more of the image and so includes more data. This results in larger scale descriptors on average being less sparse as there is a higher likelihood of a larger variation of image gradient within the window. A larger descriptor is calculated from a larger region of the image but typically a finite region is being compared. The ability of different descriptor scales to distinguish between matching and non-matching regions with a finite size depends on the performance per unit area which is evaluated next.

5.1.2 SIFT matching distance correlations

In this section we will find how the correlation between matching distances varies with pixel separation. SIFT descriptors can be calculated across an image region to form an overlapping regular grid with shifts
of a minimum of one pixel between grid nodes. Two image regions with the same size can then be compared by matching descriptors at corresponding positions between the image regions to form a grid of matching distances. Larger descriptor scales will result in a larger overlap between descriptor windows which is likely to increase the correlation between the matching distances. Neighbouring areas of an image are also more likely to be similar than more distant areas and so descriptors are more likely to be similar than more distant areas. This means that neighbouring matching distances are not independent indicators even when the descriptor window do not overlap. Because of this the level of SIFT matching distance correlation at different pixel separations is important. To test the relationship the SIFT descriptor matching distance correlation needs to be found. In this section two types of correlation are measured, the correlation between matching distances at different pixel separations and the correlation between descriptors at different pixel separations. The first is used to determine how independent matching distances are when using them together to determine if areas match and the second is used to determine the expected effect of registration error on matching distance.  

The separability of a binary classification measure depends on the separation of its two probability densities. This can be represented as a likelihood ratio. The likelihood ratio is defined as

\[ P_R = \frac{P(H_1 | i)}{P(H_0 | i)} = \frac{P(S = 1) p(i | S = 1)}{P(S = 0) p(i | S = 0)} \]

where \( P(S=1) \) is the prior probability ratio and \( i \) is the SIFT matching distance, the probability ratio based on any previously known information such as the expected proportion of the image containing change. Taking the log of the ratio allows for the calculation of the expected mean and variance of the sum large number of individual measures through the application of central limit theory if the probabilities are independent or as central limit theory for mixing variables [Bil95] if the variables can be modelled as mixing. If the combined separability of a large number of descriptors at different scales can be modelled the expected performance of a region matching technique using dense grids of SIFT descriptors at different scales can be found. If the priors are taken to be equal for the time being in order to simplify calculations the log probability ratio becomes, \( L = \ln \left( \frac{p(i|S=1)}{p(i|S=0)} \right) \).

Training data for the evaluation of the correlation of SIFT matching distances is collected from the datasets introduced in Sec. 3.2 in the following forms:

- Pairs of images of 10 flat regions, each consisting of around 40 000 pixels taken from the datasets in Sec. 3.2 viewed from different perspectives and registered using a homography determined from manually verified correspondences. This dataset represents matching image regions.

- 10 pairs of images of different regions, each consisting of around 250 000 pixels taken from the datasets in Sec. 3.2. This dataset represents non-matching image regions.
A dense grid of SIFT descriptors is collected from each pair of images with \( D_\Delta \), the shift between neighbouring descriptor windows set to 1 pixel. The matching distance between the two resulting sets of descriptors is found, resulting in a \( N \times M \) grid of distances denoted \( i_{n,m} \) where \( n \) and \( m \) denote the position in the grid. The log likelihood ratio for each resulting matching distance, \( L_{n,m} = \ln \left( \frac{p(i_{n,m}|S=1)}{p(i_{n,m}|S=0)} \right) \), is calculated using the probability densities found in Sec. 5.1.1 to form a 2D grid of \( N \times M \) log probabilities.

The grid can be split into \( N \) row vectors of length \( M \) and \( M \) column vectors of length \( N \). Each 1D vector can be auto-correlated to measure the correlation at various displacements \( x \). The auto-correlation along a row \( n \) at an offset \( x \) is

\[
R_{n,*}(x) = \sum_{m=x+1}^{M-1} L_{n,m} L_{n,(m-x)}
\]

and along a column \( m \) is

\[
R_{*,m}(x) = \sum_{n=x+1}^{N-1} L_{n,m} L_{(n-x),m}
\]

The covariance with respect to pixel separation can be found from the auto-correlation by multiplying by the variance of the distances as the auto-correlation for a column \( m \) is defined as

\[
R(L_{0,m}, L_{*,m}) = \frac{1}{N} \sum_{n=0}^{N-1} \frac{(L_{0,m} - \mu_{*,m})(L_{n,m} - \mu_{*,m})}{\sigma_{*,m}^2} \tag{5.3}
\]

where \( \mu_{*,m} \) is the mean and \( \sigma_{*,m}^2 \) is the variance of column \( m \) of the of the \( N \times M \) grid of log likelihood ratios. The covariance of the column is defined as

\[
\sigma(L_{0,m}, L_{*,m}) = \frac{1}{N} \sum_{n=0}^{N-1} ((L_{0,m} - \mu_{*,m})(L_{n,m} - \mu_{*,m})) \tag{5.4}
\]

and so by combining equations 5.3 and 5.4 we can find the covariance in terms of the auto-correlation

\[
\sigma(L_{0,m}, L_{*,m}) = \sigma_{*,m}^2 R(L_{0,m}, L_{*,m}).
\]

The results in Fig. 5.3 show that covariance between different \( L_{n,m} \) tends to zero as the pixel separation increases and is only significant at below \( 5 \times D_\Delta \). The correlation initially decreases rapidly before tailing off and decreasing more rapidly at small scales in Fig. 5.3(a). The trends for matching regions are less clearcut. The rate of decrease in covariance with offset for matching areas in Fig. 5.3(b) appears to initially be much steeper and seems to be less dependant on descriptor scale. This may point to the covariance
in matching distances from matching image regions being the result of image features. Certain features may match well between images while others may not which causes matching distances to vary at a rate dependant on the scale of those features. This can be seen in Fig. 5.23 on page 101 where matching distances vary significantly even in areas with no change depending on the scene features. In particular this can be see near object edges where the strong local image intensity gradients cause darker regions as the effects of noise have a smaller impact. In both cases the covariance tends to zero as the offset tends to infinity and so the matching distance can be modelled as a random mixing process. The results also show that covariance is larger when using larger scale descriptors but this effect is less significant when the images match. Larger descriptors create stronger distance measure covariance as the proportion of overlap in the descriptor window is larger when shifting the descriptor window by one pixel between descriptors.

The sum of the covariance between a matching distance and matching distances at all pixel separations along a line in both directions at a descriptor scale \( D_s \) is \( T_\sigma (D_s) = 2 \sum_{i=1}^{\infty} \sigma (L_{0,m}, L_{i,m}) \), this can then be found which is used in Sec. 5.2.2

\[
T_\sigma (D_s, m) = 2\sigma^2_{*,m} \sum_{n=1}^{\infty} n R (L_{0,m}, L_{n,m}).
\]

The average value of \( T_\sigma (D_s, m) \) across all rows and columns for a specific scale, \( T_\sigma (D_s) \) is found. To extend to two dimensions the auto-correlation can be integrated rotationally though \( 2\pi \)

\[
T_{\sigma,2D} (D_s, m) = 2\pi \sigma^2_{*,m} \sum_{n=1}^{\infty} n R (L_{0,m}, L_{n,m}). \tag{5.5}
\]

which again can be averaged across all rows and columns for a specific scale to find \( T_{\sigma,2D} (D_s) \). The sample points form a square, two dimensional grid and so do not form a continuous distribution. An alternative approach is to sum the result of the auto-regression for the displacements from the point \((n,m)\) of each point in the grid. The displacements are given using Pythagoras’ Theorem as \( \{1, \sqrt{2}, 2, \sqrt{5}, \sqrt{8}, 9, ...\} \). As
these points are not all integers the value of $R(L_{0,m}, L_{n,m})$ for the non-integers displacements are found using a linear interpolation. The resulting values of $\mathcal{R}_{\pi,2D} (D_s, m)$ are very close to the values found using rotational integration as shown in Fig. 5.4.

The results from the test images are shown in Fig. 5.5 showing that correlation increases with scale for non-matching regions. The results for non-matching descriptors seems to plateau above scale 6 and decrease slightly. The results for matching areas do not seem to vary as significantly which is in line with the matching distance correlation result in Fig. 5.3.
5.1.3 SIFT descriptor correlation

We have seen how the correlation between SIFT matching distances at different pixel separations are correlated. In this section we look at the correlation between descriptors themselves at different pixel separations. This correlation determines the effect of registration error on the matching distance. If descriptors 2 pixels apart are highly correlated then a registration error of 2 pixels will not have a large impact on the matching distance. It also has an effect on the ability to determine the registration offset required to compensate for the registration error in shifted DSIFT in Sec. 5.3.

Neighbouring SIFT descriptors selected from a dense grid of descriptor windows separated by a small number of pixel are more similar than two picked at random from two unrelated images or sections of an image, particularly at larger scales where there will be a large overlap between windows and image features will cover a large number of pixels. This correlation affects the dependence of SIFT matching distance on registration error and also the sensitivity of SIFT descriptors when compensating for registration error. The mean and standard deviation of the SIFT matching distance between descriptors separated by a range of pixel separations can be found for each SIFT descriptor scale. The variation with pixel separation of the mean matching distance of SIFT descriptors at scales 1 to 10 is shown in Fig. 5.6. As the pixel separation increases, the mean and standard deviation will tend to the mean and standard deviation of unrelated regions shown in Sec. 5.1.1 and the rate at which this occurs decreases with \( D_s \).

The value that the matching distance tends to as the pixel separation tends to infinity also decreases with scale and equals mean of the non-matching log likelihood ratio.

In this example at a pixel separation of 0 the descriptor is compared to itself and so has a matching distance of 0. In the presence of other sources of noise for example lighting differences or interpolation errors the mean matching distance will not tend to 0 but will instead tend to the average distance between
Chapter 5. Region Comparison

Figure 5.7: Mean matching distance of SIFT descriptors with pixel separation in the presence of noise.

matching descriptors as found in Sec. 5.1.1. The test can be re-run using the registered images used for the matching distances distributions for matching regions in Sec. 5.1.1 to find the trend in the presence of other effects. The result of this is shown in Fig. 5.7. This result should be used in the comparison of registered images as it includes the effects of other sources of noise. Here the mean matching distance at a separation of 0 pixels is not 0, due to environmental effects such as lighting, noise and registration error but then tend to values similar to those shown in Fig. 5.6. The mean and standard deviation of matching distances at a pixel separation of $D_p$ found here is denoted $\mu_D (D_p)$ and $\sigma_D (D_p)$.

5.2 Dense SIFT

In this section two methods of using a DSIFT descriptor grid to match images are presented and the expected separability based on the statistical measures are found. DSIFT descriptor grids have been used as ways to collect information about images for applications such as the statistical representation of images [CLVZ11] and the tracking of optical flow [CL08]. The novelty of this section includes the application of DSIFT to change detection, the method of application, as part of a Markov random field and the calculation of the statistical properties of the dense grids when used in change detection. The first method discussed in this section compares corresponding descriptors collected from DSIFT descriptor grids from the pair of images being matched in order to classify the images as matching or non-matching. This method works well when the registration error is small compared to the SIFT descriptor size. The second method discussed in Sec. 5.3 also classified the images of matching or non-matching but also finds the optimum smooth registration offset that minimises the matching distance across an image region and uses this to improve matching performance at higher registration errors.
5.2.1 Implementation

DSIFT uses SIFT descriptors to compare a region of a reference image to a section of a sample image that have been registered to determine if they match. The SIFT descriptors are collected over a dense rectangular grid with 1 pixel spacing across the pair of regions being matched. The descriptors at corresponding positions across the two images are compared using their Euclidean squared distance to produce a matching distance for each position in the dense grid. The areas can then be determined to match or not match using the statistical properties found in Sec. 5.1 or alternatively by setting a matching threshold and marking the areas as matching if the number of non-matching descriptor pairs exceeds a second threshold. Larger regions are less likely to match when using this second method as they will include more descriptors and so the probability that a specific number of them produce a matching distance above a threshold is higher. This introduces a bias towards the non-matching hypothesis when comparing larger regions. As the segmentation and registration methods in Sec. 6 result in segments sized inversely proportional to the density of detected feature points in a region this in turn introduces a bias towards the non-matching hypothesis in areas with few feature points. As feature points should not match in areas containing change this bias improves performance. For this reason the latter method using two thresholds is used.

5.2.2 Performance of a large number of SIFT descriptors

In this section the statistical properties of SIFT determined in Sec. 5.1 are used to find the separation of a large number of SIFT descriptors. This determines the sensitivity of the detector in terms of the area of the image needed in to determine the matching state with a desired error rate. We find this by applying central limit theory to the log likelihoods under the assumption that the independent before this by modelling the log likelihoods as random mixing. When using multiple independent descriptors the likelihood ratio is

\[
\frac{P(H_1 | i_1, i_2, \ldots, i_W)}{P(H_0 | i_1, i_2, \ldots, i_W)} = \frac{P(S = 1)}{P(S = 0)} \prod_{w=1}^{W} \frac{p(i_w | S = 1)}{p(i_w | S = 0)}
\]

where \(W\) is the number of matching distances collected from the DSIFT grid and \(i_w\) denotes an individual matching distance. We now write this using log likelihood ratios so that central limit theory can be applied

\[
\ln \left( \frac{P(S = 1 | i_1, i_2, \ldots, i_W)}{P(S = 0 | i_1, i_2, \ldots, i_W)} \right) = \ln \left( \frac{P(S = 1)}{P(S = 0)} \right) + \sum_{w=1}^{W} L_w
\]

where \(L_w = \ln \left( \frac{p(i_w | S = 1)}{p(i_w | S = 0)} \right)\). To use the log probability as a classification measure a threshold \(D_t\) for the mean matching distance needs to be chosen as shown in the case for a single descriptor in equation 5.1
and 5.2. This gives the probability of correctly detecting a matching hypothesis as

$$P(H_1 \mid S = 1) = P \left( \frac{1}{W} \sum_{w=1}^{W} L_w > \frac{1}{W} \ln (D_t) - \frac{1}{W} \ln \left( \frac{P(S = 1)}{P(S = 0)} \right) \mid S = 1 \right)$$

Altering the priors, \(P(S = 1)\) and \(P(S = 0)\) varies the prior probability ratio \(P_p = \frac{P(S = 1)}{P(S = 0)}\). \(P_p\) can be combined with \(D_t\)

$$P(H_1 \mid S = 1) = P \left( \frac{1}{W} \sum_{w=1}^{W} L_w > \frac{1}{W} \ln (D_t) - \frac{1}{W} \ln \left( P_p \right) \mid S = 1 \right)$$

$$P(H_1 \mid S = 1) = P \left( \frac{1}{W} \sum_{w=1}^{W} L_w > \frac{1}{W} \ln \left( \frac{D_t}{P_p} \right) \mid S = 1 \right)$$

Defining a composite threshold \(T = \frac{1}{W} \ln \left( \frac{D_t P(S = 1)}{P(S = 0)} \right)\) and the mean log probability ratio \(\mu_W = \frac{1}{W} \sum_{w=1}^{W} L_w\) gives

$$P(H_1 \mid S = 1) = P (\mu_W > T) \quad (5.6)$$

In other words the probability of correctly matching a region depends on the mean value of the log probability ratio.

The probability that the mean of a set of independent random variables gives a certain value can be defined by **central limit theory** if the mean and standard deviation of the set of random variables is known. Using the mean and standard deviations of the matching and non-matching distance found in Sec. 5.2 the expected value of the log probability ratios, \(\mu\) and their variance \(\sigma^2\) can be found. The expected value of \(\mu_W\) and the variance of \(\mu_W\) in matching and non-matching cases with for different values of \(W\) can be used to find the expected error rates.

From central limit theory if \(L_w\) are independent the distribution of \(\sqrt{W} (\mu_W - \mu)\) tends to the normal distribution \(N(0, \sigma^2)\) as \(W \to \infty\) [Bil95],

$$\sqrt{W} (\mu_W - \mu) \xrightarrow{d} N(0, \sigma^2).$$

However if the matching distances, \(i_w\) are modelled more accurately as random mixing as shown in Fig. 5.3 as correlation between descriptors tends to 0 as the pixel separation increases. In this case the distribution of \(\mu_W\) can be found using
\[
\sqrt{W} \left( \frac{\mu_W - \mu}{\sqrt{V}} \right) \xrightarrow{d} N(0, 1)
\]

where \( V \) is defined by

\[
V = \lim_{W \to \infty} n \text{Var}(\mu_W)
\]  

(5.7)

[Bil95]

which in the 2D case gives

\[
V_{2D} = \sigma^2 + 2\pi \sum_{i=1}^{\infty} \sigma(L_0, L_i).
\]

Equation 5.5 from Sec. 5.1 gives \( \gamma_{\sigma,2D}(D_s, m) = 2\pi \sigma_{x,m}^2 \sum_{n=1}^{\infty} nR(L_{0,m}, L_{n,m}) \), where \( D_s \) is the SIFT descriptor scale and \( \sigma(x,y) \) is the covariance between \( x \) and \( y \). \( \gamma_{\sigma,2D}(D_s) \) was found experimentally in Sec. 5.1. Substituting this into equation 5.7 gives

\[
V_{2D} = \sigma^2 + \gamma_{\sigma,2D}(D_s)
\]

and

\[
\sqrt{W} \left( \frac{\mu_W - \mu}{\sqrt{V_{2D}}} \right) \xrightarrow{d} N(0, 1).
\]

As the value of \( \mu_W \) determines the decision from equation 5.6 we re-arrange

\[
\mu_W \xrightarrow{d} \frac{\sqrt{V_{2D}}}{\sqrt{W}} N(0,1) + \mu
\]

So if we define \( \phi \) as a random variable sampled from \( N(0,1) \) we and combine with equation 5.6 we get

\[
P(\mu_W > T) = P \left( \frac{\sqrt{V_{2D}}}{\sqrt{W}} \phi > T - \mu \right)
\]

\[
P(\mu_W > T) = P \left( \phi > \frac{T - \mu}{\sqrt{V_{2D}}} \right).
\]

So \( P(\mu_W > T) \) can be found from the cumulative normal distribution \( \Phi(x) \) using

\[
P(\mu_W > T) = 1 - \Phi \left( \frac{T - \mu}{\sqrt{V_{2D}}} \right)
\]

82
The expected performance can be used to derive a ROC curve for the classification of pairs of images as matching or non-matching for a given number of descriptors by varying the value of $T$. Figure 5.8 shows a sample predicted ROC curve produced in this way with 1600 scale 5 SIFT descriptors. This represents a 40 by 40 pixel area of the image. Figure 5.9 shows how the equal error rate varies with the number of descriptors at scales 3, 5, 7 and 9 decreases as the number of descriptors used increases. It shows that scale 5 is more accurate at higher sensitivities, requiring around 1500 descriptors to get an equal error rate below 5%, this represents approximately a $39 \times 39$ pixel area. This illustrates the performance when comparing small areas with DSIFT or SDSIFT in Sec. 5.3.

### 5.2.3 Evaluation

The performance of scale 5 DSIFT is compared to established pixel differencing and pixel ratioing matching techniques, these techniques were reviewed in Sec. 2.5. The performance is evaluated in the presence of the following types of noise and interference by matching an original image with a version that has undergone simulated interference. The 10 flat regions, each consisting of around 10 000 pixels, and registered using homographies defined by 4 manually confirmed SIFT feature point used in Sec. 5.1.1 and Sec. 5.1.2 are used. The following simulated interferences are introduced:

- Interpolation noise
- A uniform registration error over the entire image
- Uniform stretching of the image
- Additive Gaussian white noise
Figure 5.9: Equal error rate with increasing number of SIFT descriptors at different SIFT descriptor scales.

- Loss of resolution

Interpolation is a common source of noise when registering images or sections of images and so it is important that matching techniques are robust to it. Registered images often have global registration errors and also can have local variations in registration error, particularly if the image is non-planar, measuring the robustness to a constant registration shows the performance in both circumstances. To exclude interpolation error from the registration error test, the constant registration error used will take an integer value. Stretching an image combines a registration error that varies over the image with interpolation noise due to the need to interpolate back to a regular grid. Additive Gaussian white noise provides a general test that simulates many different variations between two images taken of the same scene or object for example changes in lighting and self-occlusion within a texture. Additive Gaussian white noise can also be used to approximate processing steps that result in information loss, for example rounding to integer values. Registering an image often results in the image being locally or globally scaled to match the required reference frame, some processing steps may also result in the downscaling of the image which would result in a loss of resolution.

The second test involves applying the comparison methods to triangular segments registered by the Delaunay triangulation and affine compensation registration method change detection method introduced in Sec. 6.1. The resulting error rates are compared to show performance differences in the intended application.

The performance is assessed by finding the false positive rate, the proportion of pixels in the image that do not indicate a match when two matching images are compared with each type of noise and interference. The threshold is set so that 5% of the non-matching distances are below it in each experiment, to give
an expected false negative rate of 5%. This threshold is selected as it is used to set the threshold used in
the Delaunay triangulation change detection step in Sec. 8.

Robustness to interpolation noise is tested by comparing an original image to a version that has been
modified by interpolating the original image to a stretched reference frame and then back to the original
reference frame. The two images are then matched using each matching method and the false positive
rate for each is found. Bilinear interpolation and bi-cubic interpolation [Key81] are evaluated. Bilinear
interpolation uses the nearest 4 available pixel values, \( Q_{11} = (x_1, y_1) \), \( Q_{12} = (x_1, y_2) \), \( Q_{21} = (x_2, y_1) \)
and \( Q_{22} = (x_2, y_2) \) to interpolate the value at the position \((x, y)\). Where \( f(X) \) is the image intensity at
position \( X \) bilinear interpolation is calculated first in one axis and then the other

\[
\begin{align*}
f(x, y_1) &\approx \frac{x_2 - x}{x_2 - x_1} f(Q_{11}) + \frac{x - x_1}{x_2 - x_1} f(Q_{21}) \quad f(x, y) \\
f(x, y_1) &\approx \frac{x_2 - x}{x_2 - x_1} f(Q_{12}) + \frac{x - x_1}{x_2 - x_1} f(Q_{22}) \quad f(x, y) \\
f(x, y) &\approx \frac{y_2 - y}{y_2 - y_1} f(x, y_1) + \frac{y - y_1}{y_2 - y_1} f(x, y_2).
\end{align*}
\]

Bi-cubic interpolation for digital image processing defined by [Key81] by applying a cubic convolution
interpolation kernel in one axis and then the other. The method uses 4 pixels to interpolate a continuous
function from which the new pixel values can be taken from. The continuous function with a pixel spacing
of 1 is

\[
g(x) = \sum_{k=-1}^{1} c_k u(x - k)
\]

where \( c_k \) is the original pixel value at position \( x_k \), \( x_k \) are the positions of the original set of pixels where
\( k \) indicates their order in the \( x \)-axis, \( x \) is the position of the continuous output function \( g(x) \) along the
\( x \)-axis and \( u(x) \) is the cubic convolution interpolation kernel. The cubic convolution interpolation kernel
takes the values

\[
u(s) = \begin{cases} 
\frac{3}{2} |s|^3 - \frac{5}{2} |s|^2 + 1 & 0 < |s| < 1 \\
\frac{3}{2} |s|^3 + \frac{5}{2} |s|^2 - 4 |s| + 2 & 1 < |s| < 2 \\
0 & 2 < |s|.
\end{cases}
\]

The results shown in Table 5.1 show that DSIFT is the most robust to interpolation noise. It also shows
that bi-cubic interpolation performs better than bilinear interpolation.
Table 5.1: False positive rate from interpolation noise with a threshold set to give a 5% false negative rate.

<table>
<thead>
<tr>
<th></th>
<th>Bilinear Interpolation</th>
<th>Bi-cubic Interpolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>DSIFT</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Pixel Differencing</td>
<td>5.0x10^-5</td>
<td>0</td>
</tr>
<tr>
<td>Pixel Ratioing</td>
<td>0.17</td>
<td>0.068</td>
</tr>
</tbody>
</table>

Figure 5.10: False positive rate against registration error with threshold selected to give a 5% false negative rate.

Robustness to a constant registration error over the image is tested by comparing an original image to a version of the original image that has been altered by translating the original by a number of pixels in the x-axis and then cropping the images so that they are the same size with no white space. The two images are then matched using each matching method and the false positive rate for each is found. The magnitude of the translation takes an integer value so that interpolation is not needed.

DSIFT performs best with small registration errors but deteriorates rapidly as the size of the error approaches the scale of the descriptor window. Pixel differencing deteriorates almost linearly with registration error. The pixel differencing error is related to the probability of a neighbouring pixel having the same or similar intensity at a pixel separation equal to the registration error, while this may change rapidly in some situations, in most situations it varies slowly or not at all which gives the average error rate shown. Pixel ratioing performs very poorly. SIFT relies on intensity gradients after the image undergoes Gaussian filtering at the scale of the SIFT descriptor which is more distinct across the image. The combination of these two effects mean that DSIFT performs well for registration errors that are less than the scale of the SIFT features used and then rapidly deteriorates for larger registration errors. SIFT points are used to register images in many of the techniques used in this thesis. The registration error when using these techniques does not normally exceed the 2 or 3 pixels localisation error of SIFT points and so DSIFT provides the best performance in terms of robustness to registration error.

86
Figure 5.11: False positive rate against image stretch error with threshold selected to give a 5% false negative rate.

Next robustness to image stretching is measured. Stretching an image results in a varying registration error across the image. Because the resulting registration error will not always take integer values interpolation is required and so this test combines registration error with interpolation noise. The test involves comparing an image to a stretched version of itself. The original image is stretched by a number of pixels in the $x$-axis and the variation in matching distance with increased stretch is measured. A stretch of $k$ pixels is defined as interpolation from an image with $N$ pixels in the $x$-axis to an image with $N + k$ pixels in the $x$-axis which is then cropped to form an image with $N$ pixels in the $x$-axis which can then be compared to the original image. Stretching introduces two distortions to the image. Firstly a registration error that varies from 0 to $k$ across the image is introduced. Secondly the size of image features is increased by $(N+k)/N$ in the $x$-axis which changes the intensity and orientation of the image gradient. An absolute size change is used rather than a relative size change so that the maximum registration error introduced remains constant across images tested. The maximum registration error was prioritised as pixel differencing and pixel ratioing do not rely on image intensity gradients and the intensity gradients used to form the SIFT descriptor are binned which minimises the effect of small changes. Bi-cubic interpolation is used as this was previously shown to give the best performance so the effects of interpolation noise are minimised.

The results shown in Fig. 5.11 are similar to the consistent registration error results as the registration error is the main component of the error. Again pixel ratioing performs very poorly and DSIFT outperforms pixel differencing up to stretches of 8 pixels. Again as errors of 2-3 pixels are expected, DSIFT performs better overall.

The additive white noise test provides a different type of challenge to registration error which dominated the last two tests. In this test the original image is matched to a noisy version of itself. White Gaussian noise is added to the image and the sensitivity of the matching distance to the noise variance is found
for each matching technique. The pixel intensity values are normalised to be in the range 0 to 1 and Gaussian white additive noise with a variance between 0 and 1 is applied. The resulting pixel values are clipped so that no pixel has a value more than 1 or less than 0. The results are shown in Fig. 5.12 and show that DSIFT outperforms pixel differencing which in turn outperforms pixel ratioing. The relative performance is consistent at different levels of noise.

The final simulated test measures the robustness to loss in resolution. To model the affect of a loss in resolution on each matching technique the original image is down-scaled to reduce resolution and then up-scaled back to its original size and matched to the original image. Again bi-cubic interpolation is used. The downscaling will remove information which will not be recovered when interpolating back to the original size. The results are shown in Fig. 5.13. The average matching distance for each matching technique is measured against the scaling factor. DSIFT outperforms pixel differencing for resolution reduction factors of less than 7 which outperforms pixel ratioing.

Overall DSIFT has been shown to perform the best and outperforms the other methods across all tests. It has been shown to be very robust to errors based on registration compared to the other methods as long as registration error is below the scale of the SIFT descriptors used. Pixel ratioing is shown to perform very poorly.

The pixel differencing technique is compared to the DSIFT technique using the Delaunay triangulation and affine compensation method from Sec. 6.1 to register image regions. The images used are separated by a azimuth rotation of 10° and are taken from the rotating scene dataset. 2147 feature points are matched between the images using ASIFT from Sec. 4.1. Delaunay triangulation is conducted on the matched points to form 4265 triangular segments and each triangular segment is then compared independently. The triangular segment in the second image is transformed to match the shape of the corresponding
Figure 5.13: False positive rate against loss of resolution with threshold selected to give a 5% false negative rate.

Figure 5.14: Comparison of DSIFT and pixel differencing used in Delaunay triangulation based change detection.

triangular segment in the first image and interpolated using bi-cubic interpolation. Full details of the registration technique are in Sec. 6.1. The triangular segment are then compared using both DSIFT and pixel differencing; pixel ratioing is not used as it has been shown to perform very poorly.

The results in Fig. 5.14 show that for false negative rates below 7% DSIFT provides a better false positive rate. However, for false negative rates above 8% pixel differencing gives a slightly better performance. Overall DSIFT performs better. DSIFT provides a more reliable match when the image is well registered and is less likely to give false matches as individual descriptors are more distinct than pixel intensity values. On the other hand pixel differencing relies on the average value of the pixel which can often remain approximately constant over a homogeneous area. This means that when triangles are well registered, DSIFT is more sensitive to changes and so false negatives are less likely to occur but is also
more likely to produce false positives as the registration error approaches the scale of the SIFT descriptor. Conversely pixel differencing matches relatively well with large registration errors and so reduces false positives caused by them but is also more likely to miss changes due to poorer performance on the majority of triangles that have low registration errors.

5.3 Shifted Dense SIFT using Markov random fields

5.3.1 Introduction

SDSIFT is an alternative to DSIFT that is also based on a dense grid of SIFT descriptors. The method is designed for matching images when the registration error is up to a few dozen pixels, significantly larger than the levels DSIFT is able to tolerate. SDSIFT uses the known probability distributions of SIFT descriptor matching distances to find the most likely registration error over the area covered by the two images. Registration error is likely to vary smoothly over the areas being compared and this can be used to track the registration over image region. A Markov random field is used to model the relationship in the registration error between neighbouring SIFT descriptors. A Markov random field is an undirected graph consistent of nodes representing random variables and edges connecting the nodes that represent the relationship between the variables. The nodes of the graph represent the SIFT descriptor sample points, forming a regular two dimensional grid, while the edges connect neighbouring nodes and represent the relationship in the registration error between neighbouring sample points. The mean matching distances obtained after compensating using the registration errors at each sample point giving the most likely solution to the random field are then used as the matching score for the images being compared.

5.3.2 Calculating the probabilities of likely registration errors

A dense grid of SIFT descriptors is collected from two image patches both of size \((X,Y)\) at scale \(D_s\) and grid separation \(D_\Delta\). The descriptors calculated from the first image are denoted \(f_A(x,y)\) and the descriptors calculated from the second image are denoted \(f_B(x,y)\), where \(x\) and \(y\) are the position from which the on the image that the descriptor is calculated. The descriptor \(f_A(x,y)\) is matched against the descriptors at an offset in image \(B\), \(f_B(x+q,y+r)\) where \(-Q/2 \leq q \leq Q/2\) and \(-R/2 \leq r \leq R/2\) to form a grid of \(Q \times R\) (where \(Q = R\)) matching distances denoted \(d_{x,y}(q,r)\) around \((x,y)\). The size of the grid should be set so that \(QD_\Delta\) is at least twice the maximum registration error \((E_r)\) expected, so that the maximum \(E_r\) in any direction will fall within the grid. These matching distances can be used to form hypotheses for the registration error at position \((x,y)\), \(E_r(x,y)\). The hypothesis that the image matches at the a registration error \((q,r)\) at a position \((x,y)\) on image \(A\) is denoted \(H_{1;x,y}(q,r)\) while the probability that it does not match is denoted \(H_{0;x,y}(q,r)\). The probability of the hypotheses can
be found using the PDFs for matching distance from matching, $p(i \mid S = 1)$ and non-matching image regions, $p(i \mid S = 0)$ found in Sec. 5.1.1. The matching state at a position $(x, y)$ and registration error $(q, r)$ are now denoted $S_{x,y}(q, r)$ which takes the value of 1 if the area matching and 0 if it does not match. The probability of $H_{1;x,y}(q, r)$ can now be found using

$$P(H_{1;x,y}(q,r) \mid d_{x,y}(q,r)) = \frac{p(d_{x,y}(q,r) \mid S_{x,y}(q,r) = 1) P(S_{x,y}(q,r) = 1)}{p(d_{x,y}(q,r) \mid S_{x,y}(q,r) = 1) P(S_{x,y}(q,r) = 1) + p(d_{x,y}(q,r) \mid S_{x,y}(q,r) = 0) P(S_{x,y}(q,r) = 0)}$$

The prior probabilities $P(S_{x,y}(q,r) = 1)$ and $P(S_{x,y}(q,r) = 0)$ can be defined using external knowledge or the registration error at a neighbouring position on the image. If the regions being compared show an area of the scene that is approximately planar and not containing changes, the distribution of the variation in the registration error from a descriptor position $f_A(x, y)$ to $f_A(x - D_\Delta, y)$ could be approximated using a normal distribution. As the regions are already registered the mean registration error when using this matching technique will be 0 and so the mean of the normal distribution, $u_e$ should equal 0. The standard deviation, $\sigma_e$ depends on the expected maximum registration error variation $\max(\Delta E_r)$, the size of the region, $(X,Y)$ and the descriptor spacing, $D_\Delta$. If the scene area is not smooth and flat it could also depend on the scene texture and shape. For example when using a smooth image $\sigma_e$ can be defined as

$$\sigma_e = \frac{2D_\Delta \max(\Delta E_r)}{(X + Y)}.$$

The distribution modelling the registration error variation between neighbouring points in the descriptor grid can be applied to the matching hypothesis probabilities, $P(H_{1;x,y}(q,r) \mid d_{x,y}(q,r))$ to find the prior probabilities at a neighbouring point, $P(S_{x+D_\Delta,y}(q,r) = 1)$,

$$P(S_{x+D_\Delta,y}(q,r) = 1) = N(q, r; u_e, \sigma_e) * P(H_{1;x,y}(q,r) \mid d_{x,y}(q,r)).$$

To apply this approach the hypotheses at an initial point on image $A$ need to be found. The central point on image $A$, $([X/2,Y/2])$ can be used. The descriptor at the central point in the first image, $f_A(X/2, Y/2)$ is compared against the descriptors in a $Q \times R$ square neighbourhood around the central point in a second image to find a $Q \times R$ grid of matching distances. For this initial point the priors are not available from a neighbouring position and so $P(S_{x,y}(q,r) = 1)$ and $P(S_{x,y}(q,r) = 0)$ are both set to 0.5. Once the hypothesis probabilities at this initial position are known the priors for the next position can be found and the technique can be repeated to form a vertical line along the y-axis of points on image $A$ for which registration error hypothesis are known. The initial $Q \times R$ square neighbourhood is extended
as the convolution is applied at each step. This line can then be extended horizontally along the x-axis to form a complete set of hypothesis at every position on image A.

The mean matching distance at the highest $P (H_{1,x,y} (q,r) \mid d_{x,y} (q,r))$ across image A can be used as the matching scores for the two image regions. The mean is used rather than the number of distances over a threshold, as in DSIFT, as SDSIFT will identify the best matches even when the image regions being compared are non-matching which will avoid very high matching distances.

In order to increase computational efficiency the number of registration hypotheses can be reduced by only calculating the probability of the registration hypothesis for hypotheses with a prior probability above a threshold. This threshold is applied at each step while also ensuring that the number of tracked hypotheses remains about 20 and below 100, with this overruling the threshold if necessary.

### 5.3.3 Re-calculate outliers

After the calculation of the initial vertical line of registration error hypotheses along the y-axis the registration errors across the image are calculated along the x-axis for the remainder of the images. Because of this the majority of the image is only calculated using priors from the neighbouring point on the x-axis. This can result in inconsistencies in the hypotheses along the y-axis which may result in rapid jumps in registration error between neighbouring positions. This is unlikely to be due to a real misalignment between the images but instead may be due to an error propagated along the x-axis. To identify outliers, shifts in the location of the most likely hypothesis between descriptor positions that are above a threshold along the y-axis are found. The hypotheses can then be recalculated using the neighbouring position along the y-axis as the prior. In cases where the location of registration error with the highest probability is changed the corresponding hypotheses in the same row along the x-axis is also recalculated. This process is repeated until the result stabilises.

### 5.3.4 Estimation of the probability ratio

As the PDF of matching and non-matching SIFT matching distances is obtained experimentally the probability of unlikely occurrences will often be recorded as 0. This is caused by unlikely matching distances not occurring in the training data. If all values of $p(i \mid S = 0)$ for a set of given values of $i$ are 0, the resulting set of Bayesian probabilities will be 1 regardless of the exact value of the $i$.

$$P(H_1 \mid i) = \frac{p(i \mid S = 1)}{p(i \mid S = 1) + p(i \mid S = 0)} = \frac{a}{a + 0} = 1$$

This can lead to set of registration errors with the same probability distributed around the location of the true registration error. This stops the algorithm tracking the point with the smallest matching distance
which is the best fit. A separate problem is that the experimentally obtained PDFs may not be smooth due to the nature of the training images. This can cause the matching probability to reduce slightly or not increase as expected as the matching distance decreases. A solution to these issues is to map a function to the general shape of the Bayesian probability function which follows its general shape but results in a smooth curve, the following can be used

\[
P^*(H_1 \mid i) = \frac{1}{1 + k^{d_i - d_0}}
\]

where \(d_\mu\) is the matching distance at which \(p(i \mid S = 1) = p(i \mid S = 0)\) and \(k\) is a constant. \(k\) is selected so that the sum of the approximation error of the upper and lower quartiles of \(P^*(H_1 \mid i) - P(H_1 \mid i)\) is minimised.

### 5.3.5 Multi-modal tracking

Due to repetitive characteristics in textures high probability hypotheses can often cluster around multiple, non-adjacent registration errors across a section of the image due to repeated patterns in the scene texture. The cluster centres that do not give the correct registration error should not give the best hypothesis once the method is repeated across all rows and columns of the descriptor grids but may in some situations result in the correct result being dropped due to pruning. To resolve this a multi-modal tracking element can be added to the algorithm. Multiple local probability maxima are found at different registration errors. The previous method of pruning that ensures that the number of hypothesis points did not exceed a set limit is replaced with a maximum per local maxima. The minimum number of tracked hypotheses pruning criteria and the probability threshold pruning criteria are retained for overall pruning to reject unlikely local maxima.

### 5.3.6 Increasing accuracy using SIFT descriptor correlation

Rather than assigning a matching probability based on the matching distance of a single pair of descriptors at a local maximum, the expected trend in matching distances in an \(N_x \times N_y\) area around the maximum as found in the SIFT descriptor correlation in Sec. 5.1.3 can be used to find the maximum region that best matches the expected trend. The probability of a matching distance given a displacement \((n_x, n_y)\), where \(-N_x/2 \leq n_x \leq N_x/2\) and \(-N_y/2 \leq n_y \leq N_y/2\), from the candidate maximum, \((q, r)\) is denoted \(p(d_{x,y} \mid (q - n_x, r - n_y) \mid E_r(x, y) = (q, r))\) and can be found by modelling it as a normally distributed independent variable with a mean, \(\mu_D(D_p)\) and standard deviation, \(\sigma_D(D_p)\) where \(D_p\) is the pixel separation from the local maximum, \(D_p = (n_x^2 + n_y^2)^{0.5}\). The probability of a matching distance can be found using
\[
p (d_{x,y} (q - n_x, r - n_y) \mid E_r(x,y) = (q,r)) = \frac{N (d_{x,y} (q - n_x, r - n_y); \mu_D (D_p), \sigma_D (D_p))}{N (d_{x,y} (q - n_x, r - n_y); \mu_D (D_p), \sigma_D (D_p)) + p (d_{x,y} (q - n_x, r - n_y) \mid S_{x,y} (q,r) = 1)}.
\]

Combining the probabilities from an area \((N_x, N_y)\) around \((q, r)\) can be used to find the probability of \(H_{1;x,y} (q,r)\) and thus \(E_r (x,y) = (q,r)\) given the neighbouring set of matching distances \(\{d_{x,y}\}\) as

\[
P (H_{1;x,y} (q,r) \mid \{d_{x,y}\}) = \prod_{n_x = -N_x/2}^{N_x/2} \prod_{n_y = -N_y/2}^{N_y/2} [p (d_{x,y} (q - n_x, r - n_y) \mid E_r (x,y) = (q,r))] .
\]

### 5.3.7 Model offsets as a Markov random field

Each descriptor position in the reference image can be modelled as a node of a graph. The possible states of each node in the graph represent the possible registration errors at that descriptor position given as an integer pixel value in the \(x\)-axis and \(y\)-axis. The matching probability for each registration error at that descriptor position is used as the probability of each node state. All states with a probability below a threshold (such as 0.01) within the search limit are set to the threshold value to stop outlier results at a single node causing the probability of the whole graph being 0. The edge states which represent the registration error variation between nodes can again be represented by a Gaussian distribution where the standard deviation depends on the expected variation of registration error in the image. Image segmentation such as mean-shift segmentation [GC03] can be used to detect likely object boundaries that might cause the registration error to shift significantly between neighbouring nodes. Edges near segment boundaries can be set to have an equal probability for all state combinations. If the epipolar geometry of the images is reliably known, the edge probabilities can be adjusted. For example in a stereo rig the epipolar lines are horizontal and so the registration error can only vary in the \(x\)-axis.

An iterated conditional modes solver [Bes86] can be used to find the optimal decoding of the graph. The iterated conditional modes solver is initiated by setting each node to its most likely state, ignoring edges. The joint probability of the graph including all nodes and edges is found by multiplying their state probabilities together. Each node of the graph is then iterated through and its state is changed or left the same to maximise the joint probability. This is repeated until a full cycle is completed without any state changes, which indicates a local maximum has been found [Bes86].

Instead of cycling though the entire image, block coding can be used. Segments are likely to have a consistent state across all of their nodes and so each segment can be solved independently before a global cycle is carried out, this speed up the calculation and also reduces the probability of the local maximum being the global maximum [Bes86].
5.3.8 Results

SDSIFT is tested on images with simulated errors in the same way as with DSIFT in Sec. 5.2. Descriptors of scale $D_s = 5$ and spacing $D_\Delta = 1$ are used. The size of the initial $Q \times R$ grid is set to 41 $\times$ 41 which is designed to tolerate registration errors of up to 20 pixels. The distribution of registration error changes between descriptor positions is set to $\sigma_e = \frac{2D_\Delta \max(\Delta E_v)}{(X+Y)}$ where $\Delta E_v$ is assumed to equal 20. Areas of the rotating scene and larger rotating scene datasets that contain different objects are selected. The average matching distance between different areas will be lower than for DSIFT as the SDSIFT will tend to select registration offsets that result in a smaller matching distance. Even when the areas do not match this will decrease the average matching distance by some extent. Because of this the average matching distance between different areas and the variance in matching distance needs to be found.

Fig. 5.15 shows an example of the results for the $x$-axis registration offset over a pair images of a scene region. Here the overall offset variation can be seen as well as local variations due to the scene not being a smooth plane.

Each area shown in Fig. 5.16 is matched to each other area to find the average matching distance between two different areas. The standard deviation of the matching score is also found. From this the threshold matching distance required to achieve a 5% false negative rate is found.

The expected performance for the types of distortion that do not involve registration error is expected to be in line with the result for DSIFT while those involving registration error should be substantially better. The performance is shown along side DSIFT as well as pixel differencing and pixel ratioing.

Figure 5.17 shows that SDSIFT outperforms DSIFT in robustness to noise while Fig. 5.18 shows that
Chapter 5. Region Comparison

Figure 5.16: Masked 200x200 pixel areas on the larger rotating scene (left) and rotating scene (right) datasets used for SDSIFT testing.

Figure 5.17: False positive rate of SDSIFT compared to DSIFT, pixel differencing and pixel ratioing against noise variance with threshold selected to give a 5% false negative rate.

DSIFT performs better with resolution reduction. Both perform equally well when tested for robustness to interpolation noise with no detected errors in Table 5.2 for SDSIFT as well as DSIFT. None of these sources of image distortion involve registration error. The difference in performance must therefore come from differences from the distancing method post adjustment for registration offset. DSIFT uses the number of matching distances above a threshold to determine if the region matches while SDSIFT uses the average matching distance. SDSIFT performs well up to a level before degrading sharply to an even greater extent than DSIFT.

Next the robustness to registration error is tested. Figure 5.19 shows the false positive rate for SDSIFT

<table>
<thead>
<tr>
<th>Method</th>
<th>Bilinear Interpolation</th>
<th>Bi-cubic Interpolation</th>
</tr>
</thead>
<tbody>
<tr>
<td>SDSIFT</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>DSIFT</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Pixel Differencing</td>
<td>5.0x10-5</td>
<td>0</td>
</tr>
<tr>
<td>Pixel Ratioing</td>
<td>0.17</td>
<td>0.068</td>
</tr>
</tbody>
</table>

Table 5.2: False positive rate from interpolation noise with a threshold set to give a 5% false negative rate.
Figure 5.18: False positive rate of SDSIFT compared to DSIFT, pixel differencing and pixel ratioing against image resolution reduction with threshold selected to give a 5% false negative rate.

Figure 5.19: False positive rate of SDSIFT compared to DSIFT, pixel differencing and pixel ratioing against registration error simulated by shifting the pixels of the sample image by 1 in the x-axis with threshold selected to give a 5% false negative rate.

against simulated registration error. Registration error is simulated by using a copy of the reference image offset by an integer number of pixels in the x-axis as the sample image. SDSIFT results in false positive rates of less than 0.001 up to a registration error of 22 pixels compared to registration errors of up to around 4 pixels for DSIFT. At registration errors above 22 pixels the error degrades rapidly. This is due to the initial 41x41 pixel search window used to find the registration error at the central point in SDSIFT. This search window means that a registration error in one axis of over 20 pixels results in the correct matching position not being found. The algorithm still performs well up to 22 pixels due to matching distance correlation as described in Sec. 5.1 which means that matching distances within a few pixels of the correct registration offset will be lower than most non-matching distances and will still form a local probability maximum. This allows the search window to centre on the correct offset after a few steps. The window size must be set to twice the expected level of registration as the algorithm performs well up to this limit.

The performance when matching a stretched version of an image against itself is shown in Fig. 5.20.
Figure 5.20: False positive rate of SDSIFT compared to DSIFT, pixel differencing and pixel ratioing against image stretch with threshold selected to give a 5% false negative rate.

In this case the registration error varies over the image, does not take a integer value across the image and also involves interpolation noise. SDSIFT performs well up to the 20 pixel registration error the initial search area is designed for. The additional of a varying, non integer registration error as well as interpolation noise does not affect the performance.

As SDSIFT compensates for registration error using a copy of the reference image that is offset by an integer number of pixels as the sample image will often result in identical descriptors being compared which will result in a zero matching distance. To avoid identical pixels being compared larger, 400x200 pixel image regions are selected and alternate pixels in the x-axis are used to form the reference and sample image so that two 200x200 pixel regions with no shared pixels are formed. The resolution of the resulting images is halved and they have a half pixel registration error between them. The registration error can be increased by shifting the sample image in the x-axis. The results are shown in Fig. 5.21 and show a similar performance to the previous registration error test in Fig. 5.19. In this case performance degrades after an error of 21 pixels (on top of the inherent 1/2 pixel registration error due to the sub-sampling) which is not a significant reduction in performance.

5.4 Markov random field change localisation

So far the SIFT based region matching techniques introduced in this chapter determine if the entire regions compared match. In this section Markov random field change localisation is used to attempt to classify which areas of the regions match and which do not. As with DSIFT a dense grid of SIFT descriptors is obtained from the images being compared and descriptors from corresponding position are matched between the images. The resulting matching distances are used to define node probabilities at each grid position using the matching and non-matching distance probability distributions found in Sec. 5.1. The probability of neighbouring SIFT descriptors having the same state is used to relate neighbouring nodes to
form the edges probabilities of a Markov random field. Additionally an image segmentation method such as mean-shift segmentation [CM02] can be used to adjust the edge probabilities using the segment edge location. This Markov random field can be numerically solved to find the most likely change classification for each point in the dense grid of SIFT descriptors.

A dense grid of scale 5 SIFT descriptors with 1 pixel separation is found for the two images being matched. The matching distances between corresponding descriptors are used to form the probabilities of the nodes of the graph. The probabilities are calculated using the probability distributions to form a Bayesian formulation as explained in Sec. 5.1.

\[
P(H_1 \mid i) = \frac{p(i \mid S = 1)}{p(i \mid S = 1) + p(i \mid S = 0)}
\]

\[
P(H_0 \mid i) = \frac{p(i \mid S = 0)}{p(i \mid S = 1) + p(i \mid S = 0)}
\]

The edge probabilities control the sensitivity of the detector to objects of different sizes. The higher the probability that neighbouring nodes have the same state, the less sensitive the detector is to small objects but the more reliable the classification is. The choice of sensitivity depends on the expected size of changes in the scene and the reliability of the DSIFT matching measure. The reliability depends on the separability of the probability distributions for matching and non-matching distances and also the correlation between neighbouring nodes. The edge probabilities are set to the values shown in Table 5.3.

As matching areas are expected to be larger than non-matching areas, the probability of a matching node neighbouring another matching node is higher than for non-matching. The probabilities used were calculated by using a training set of images where the ground truth is known that have the type, size and
Table 5.3: Proportions of edge configurations where rows and columns represent the nodes on either end of the edge.

<table>
<thead>
<tr>
<th>Without Segment Boundary</th>
<th>Matching</th>
<th>Non-Matching</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matching</td>
<td>0.88</td>
<td>0.0012</td>
</tr>
<tr>
<td>Non-Matching</td>
<td>0.0012</td>
<td>0.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>With Segment Boundary</th>
<th>Matching</th>
<th>Non-Matching</th>
</tr>
</thead>
<tbody>
<tr>
<td>Matching</td>
<td>0.4</td>
<td>0.1</td>
</tr>
<tr>
<td>Non-Matching</td>
<td>0.25</td>
<td>0.25</td>
</tr>
</tbody>
</table>

Figure 5.22: Larger rotating scene dataset images taken from the same angle in the two scene configurations.

density of matching and non-matching areas expected. A grid was defined across each of the training images and the proportion of edges that cross boundaries between matching and non-matching areas was found. The proportions can be used to find probabilities using a Bayesian formulation. Similarly the proportion of edges that cross a segmentation boundary with each configuration can be found. The result of this training on the rotating scene dataset used is shown in Table 5.3.

5.4.1 Results

Markov random field change localisation is tested on two images from the larger rotating scene data, one from each scene version. The changes introduced between capturing the scene versions are described in Sec 3.2.4 and shown in Fig 5.22. Both images are taken from the same angle. The background is removed to only leave the round scene platform and the objects lying on it. The registration error between the two images is around 5 pixels. The images are registered using a global homography calculated on SIFT feature points by using a RANSAC-based approach [FB81]. Due to SIFT point localisation errors and because the scene is not planar, registration errors of around 2 pixels remain and the occlusion remains.

A dense grid of scale 5 SIFT descriptors with a 1 pixel offset are collected over both of the images. Corresponding SIFT descriptors are compared using a euclidean squared distance to give a matching distance for each pixel. The matching distances are illustrated in Fig. 5.23.

The matching distances are then used to find the independent probability of change at each pixel point.
Figure 5.23: Left: Test image from the larger rotating scene with changes between the two scene configurations shown in white. Right: DSIFT matching distances with high distances shown in white and low distances shown in black. The ellipse of white around the end is due to edge effects when removing the scene background.

Figure 5.24: Mean-shift segmentations of images of both of the scene configurations in the larger rotating scene dataset using images taken from the same angle.

Based only on the matching distance at that pixel position using the findings in Sec. 5.1. This probability is used as the node probability when constructing the Markov random field. The mean-shift segmentations shown in Fig. 5.24 are used to define edges in the scene. Any pixels that are in a different segment in either of the segmentations are counted as being in different segments. Edge probabilities are set to the values in Table 5.3.

The resulting solution to the Markov random field is shown in Fig. 5.25. All areas of change are detected and the only significant false positives are due to edge effects introduced when the background is removed. The false positive rate is 0.83% and the false negative rate is 6.5%.
Figure 5.25: Result of Markov random field change localisation on the two scene configurations of the larger rotating scene dataset. Changes are shown in white and areas with no change are shown in black. Ellipse outline is due to edge effects when removing the background of the image.
5.5 Visual texture based matching

5.5.1 Introduction

Visual texture based matching approaches use a texture descriptor based on visual ‘words’ to match areas that are difficult to register because they are far from planar or include self-occlusion or occlusion, for example areas containing trees or uneven ground. If areas are not locally close to planar, registration methods such as those in Sec. 6 do not perform well. If those areas cannot be accurately registered they cannot be matched using established change detection techniques such as those reviewed in Sec. 2.5. If they cannot be registered even with large errors matching is not possible with dense SIFT which is robust to registration errors of up to 3 to 5 pixels which is introduced in Sec. 5.2 or even shifted dense SIFT that can match regions with registration errors of up to 20 pixels which is introduced in Sec. 5.3. It is proposed that visual texture matching can be used to match regions based on the statistics of descriptors obtain from them rather than by directly comparing pixel values or descriptors.

Visual texture matching involves the comparison of two images and the classification of regions within the images as either matching or differing under widely differing viewing angles. Two methods of obtaining a higher level representation of a region calculated from descriptors obtained from the region that is robust to large variations in viewing angle are proposed. When viewed from wide angles objects often have completely different surfaces in view, this makes traditional matching techniques impossible. The approaches developed here attempt to capture the characteristic of an object or region with the assumption that this will remain consistent across the object surfaces. Occluding and occluded objects will often have similar textures; for example a tree will still have branches and leaves, a house bricks and a car metal panels and windows. Also in crowded areas where occlusion is high the occluding object will often have a similar visual texture, for example in a forest the tree may be obscured but the tree obscuring the view is likely to have a similar visual texture.

5.5.2 Defining separate areas for matching

Although visual texture approaches do not require registration they still require a method for defining regions and determining which regions in the reference image correspond to which regions in the sample image. This requires a method of segmentation as well as approximate alignment of segments between images. In the context of this thesis, visual texture change detection will typically be used after an initial approach based on registration using Delaunay triangulation as described in Chapter 6 has matched areas that are close to planar. The same triangular segments could be used to define regions for visual texture matching. These triangular segments correspond between images and so further alignment is not required.
Another approach is to segment the reference and sample images using a method such as mean-shift segmentation [CM02]. Segments in the reference and sample images that include areas that have not yet been matched can then be compared. If the epi-polar geometry is known the number of comparisons can be reduced so that segments in the reference image are only compared to segments in the sample image that are on the epi-polar line.

The aim is for segmenting to occur along object boundaries so that each segment is homogeneous and encompasses only pixels from a single object or type of region, for example a car or a forested region. Segmentation will often split a conceptionally homogeneous area of an image, for example a tree, into multiple segments, or conversely a single segment may consist of multiple objects. This split will normally not be consistent between the sample and reference images. For example in one image there may be 5 segments of the same building and in another image containing the same building but viewed from the other side there may be 7 segments, none of which show the same surfaces. The premise is that although the surfaces in view are not identical, the fact that they are from the same object or type of region will mean that their texture will be similar. While in this situation the method cannot detect small changes on the building, it can detect larger changes such as regions changing from forest to road or regions appearing or being removed. There is still a possibility that an entire area type or object is obscured in one view and in this situation will produce false positives.

5.5.3 Area categorisation

5.5.3.1 SIFT Bag of Visual Words

The principle of a SIFT bag of visual words (BoW) comes from a basic method of text document indexing and searching [Har54]. A text document consists of a number of individual words and so can be represented by the variety and frequency of the words contained within it. This is used in simple methods of document index and search. Each word in a dictionary represents a dimension of a vector and the number of each of those words in a document represents the document’s magnitude in that dimension. In this way each document can be represented as a vector and documents can be compared by finding a distance between the two vectors. Similarly a image can be considered as a grouping of component parts which can be used to represent the image as a whole as introduced by [JDS10]. If a dictionary of all possible components is available a vector can be produced and used for comparison. A method of representing a part of an image in a way that makes it possible to assign it to one of a number of ‘words’ in a dictionary as well as a method for producing a dictionary is required. The key difference between applying the idea to text documents and images is that different instances of a word in a document will match exactly (particularly after stemming is used to remove grammatical variations of words) while descriptors from an image are very unlikely to be the same as each other. Because of this visual ‘words’ need to be defined so that
descriptors with similar characteristics are defined as the same word.

In this implementation we use dense SIFT [Low04, CLVZ11] descriptors to represent the image, where each descriptor represents a part of the image or a ‘word’. As described in Sec. 2.3.1, SIFT descriptors are 128 dimensional vectors which represents the image intensity gradients within the descriptor window. Dense SIFT descriptors are extracted by calculating the descriptor window across the image. A dictionary of ‘words’ where each word represents a dimension in a description vector can be formed by clustering a set of descriptors gathered from training images. The k-means [HW79] clustering algorithm is used to define the visual ‘words’. Through experimentation it was found that 500 cluster centres gave the best performance.

Once the dictionary is defined, we can characterise a segment using a vector whose magnitude in each dimension represents the relative frequency of each cluster centre ‘word’ in the segment. To construct the vector, dense SIFT descriptors are extracted at a number of scales and assigned to the nearest ‘word’ as defined by the vector distance. To compare image patches, vectors of the number of each ‘word’ present within the patch are compared, again using a vector distance.

5.5.3.1.1 Synonyms The quality of matches can be improved when indexing and searching text documents using knowledge of word synonyms. Text documents can be said to be similar if words that have the same meaning are shared between them regardless of if the same precise word is used. In the same way visual ‘words’ that represent a similar part of an image can be represented as synonyms. Image parts that are oriented in different ways can be seen as perceptually similar and grouped together in this way. A reflected version of each descriptor obtained from a region is found. The original and reflected versions are then rotated in increments of 90°, resulting in 8 version of the descriptor. The rotations and reflections can be computed directly from the SIFT descriptor vector. The 128 bin vector is split into its 16 component 8 bin rotation histograms. These histograms can be rotated by circularly shifting each 8 bin vector by 2 to achieve a 90° rotation. The location of each 8 bin histogram in the 128 bin SIFT vector can then be re-arranged to rotate or reflect the histogram position as illustrated in Fig. 5.26 where the bin positions as numbered on the left are reflected to give the arrangement on the right. The visual ‘words’ are still defined with k-means clustering without taking the synonyms concept into account as it was found that it is unlikely for reflections of cluster centres to be close to existing centres. The 8 versions of each of the descriptors obtained from the region being characterised are then matched to the k-mean centres and the best match across all 8 versions is used to represent the descriptor.

5.5.3.1.2 Results To test how well the approach captures the characteristic of an image patch nineteen test images from the rotating scene with trees dataset are segmented using the mean-shift algorithm [CM02] using segmentation settings that are manually tuned for each image to produce well
defined segments. Each segment is manually classified according to the type of objects or texture within view. Segments that contained more than one category or none of the categories are rejected. The categories are illustrated in Table 5.4.

BoW histogram vectors are obtained for each segment and normalised to length 1. The distribution of matching distances being BoW histogram vectors obtained from regions within a group is compared with the distribution of matching distances between groups. The average matching distances between categories $a$ and $b$ are defined as

$$D_B(a, b) = \frac{1}{n_an_b} \sum_{i=1}^{n_a} \sum_{j=1}^{n_b} 1 - U_i.V_j$$

where $n_a$ and $n_b$ are the number of images in each category and $U_i$ and $V_j$ are the sets of histograms in the two categories being compared. $U_i$ and $V_j$ are the same when finding within group means. The average matching distances between categories are compared to the distances between segments within a category to determine if the method is a good differentiator. The standard deviations and means of the distances are also found. Ideally the distances between segments within a category would be much smaller than the distances between segments within that category and other categories. For evaluation purposes this
### Figure 5.27: BoW using dense SIFT descriptors with 500 ‘words’ using ‘word’ synonyms.

is defined as average distances between segment categories being at least one standard deviation more than distances within the category. The results are shown in Fig. 5.27, note that the table is symmetrical. Mean matching distances . The diagonal elements of the table in green show the mean distance between histograms from the same category. Ideally these should be low and all of them should be lower than the off-diagonal distances. Inter category average distances that are lower than the average distance with the category of that row are shown in red and those less than one standard deviation above are in yellow. Some categories (e.g. Background and Sandpit) have a very much lower intra-class distance than others (e.g. Side of Shop). In one case the diagonal element is not even the lowest value within a row indicating that the intra-class distance is actually higher than the inter-class distance (Side of Shop to Background). In many other cases (highlighted in the table) off-diagonal elements are within one standard deviation of the on-diagonal element in the same row. The poor differentiation between ‘Main forest’ and ‘Other trees’ is a positive result as the categories contain the same object type and so should match well. Categories which are likely to contain other objects within them and so are not a single homogeneous texture (e.g. Grass, Gravel and Road) are likely to produce large matching distances within their category with a high standard deviation and so provide a challenge. To implement BoW using SIFT descriptors in a change detection system a matching threshold needs to be defined that does not require prior knowledge of the category of the region in the reference image as opposed to the results in 5.27 where the threshold is set by category. Two categories with equal standard deviations with means one standard deviation apart will on average produce an equal error rate of over 30°. If there are multiple candidate segments in the reference image this error rate will be further compounded. For these reasons, while the technique does have promise and is impressively robust to viewing angle and does not require registration, it is not integrated into the end to end change detection system in Sec. 7.
5.5.3.2 Textons bag of words

Textons provide an alternative method for capturing the characteristic of an image region. Textons are a group of filters that capture the shape, orientation and scale of shapes in the region. The filters are convolved with the image region and the filter response to each filter captured from a dense grid of points across the region is can be used to characterise the region. This characterisation can be used in place of the SIFT descriptor in Sec. 5.5.3.1 to implement a BoW visual texture matching algorithm. The term texton was proposed by Julesz in 1981 [Jul81] before later being used to describe filter banks that attempted to capture the perceived texture of a region [LM01]. The implementation of the filter bank used here used was first introduced by [VZ02], which is reviewed in the literature review in Sec. 2.6.5 and illustrated in Fig. 2.6. The definition of the filters that are included in the filter bank are

\[
\text{LoG}_{\sigma_n} = \sum_{y=-\infty}^{\infty} \sum_{x=-\infty}^{\infty} \text{LoG}^+(x, y, \sigma)
\]

\[
\text{Edge}_{\sigma_n} = \sum_{\theta=0}^{2\pi} \left[ \left( \sum_{y=-\infty}^{\infty} \sum_{x=-\infty}^{\infty} \text{Edge}^+(x, y, \sigma, \theta) \right)^2 - \prod_{\theta=0}^{2\pi} \sum_{y=-\infty}^{\infty} \sum_{x=-\infty}^{\infty} \text{Edge}^+(x, y, \sigma, \theta) \right]
\]

\[
\text{Bar}_{\sigma_n} = \sum_{\theta=0}^{\pi} \left[ \left( \sum_{y=-\infty}^{\infty} \sum_{x=-\infty}^{\infty} \text{Bar}^+(x, y, \sigma, \theta) \right)^2 - \prod_{\theta=0}^{\pi} \sum_{y=-\infty}^{\infty} \sum_{x=-\infty}^{\infty} \text{Bar}^+(x, y, \sigma, \theta) \right]
\]

where \( n = [1, 2, 10] \) represents the scales. Varying \( \theta_1 \) and \( \theta_2 \) changes the orientation of the filter; filters are calculated for each 15° internal over 360° for the edge filter and over 180° for the bar filter as it is symmetrical. The LoG and bar filters also have negative versions. 5 vectors are constructed, the first vector has length 20 and consists of the positive sum of the response of the positive and the negative LoG filter at each of the 10 scales. The second vector consists of the positive sum of the response of all rotations of the bar filter at each of the 10 scales, concatenating the positive and negative bar responses gives a length 20 vector. The same is done for the edge filter but as there is no negative version the result in a length 10 vector. The last two length 10 vectors give the positive sum of the response of the positive and negative LoG filters. Each of the 5 vectors is combined into a single description vector and normalised to have a Euclidean length of 1.

The inner product of vectors from two regions is used to give a matching distance of the two image regions

\[
\text{Distance} = \text{Descriptor}_a \cdot \text{Descriptor}_b
\]

The resulting descriptor vectors are used to form a visual ‘word’ dictionary in the same way as Sec. 5.5.3.1 by using k-means clustering to find 500 cluster centres. Once descriptors are obtained from the region
being characterised synonyms are again found for 4 90° rotated versions of the original descriptor and 4 90° rotated versions of the reflected versions by circularly shifting the component vectors. As before the cluster centre closest to any of the synonyms is used to represent the region.

5.5.3.2.1 Results The technique is evaluated using the same data as used in Sec. 5.5.3.1 shown in Fig. 5.27. The results are evaluated in the same way and shown in Fig. 5.28. The results show a similar level of performance to SIFT BoW which again is not good enough for use in a change detection system.

5.6 Summary

In this chapter the statistical properties of SIFT including the correlation between neighbouring matching distances and the correlation between neighbouring descriptors were found. These were used to define the following methods for the matching of regions of registered pairs of images. Firstly DSIFT was introduced which was shown to perform well the the registration error is smaller than the scale of the SIFT descriptor used to form the grid of SIFT descriptors. Secondly SDSIFT was introduced which compensates for registration error and was shown to perform well with registration errors of a few dozen pixels when using SIFT descriptors with a scale of 4 pixels. Thirdly it was shown how Markov random fields can be used to localise change within the region being compared. Finally the use of visual texture descriptors was explored for the application of region matching. Although their performance in terms of error rates was not found to be good enough for the purpose the approach did show promise and would give good robustness to registration errors and certain types of occlusion where the occulting area has a similar visual texture to the occulted area.
Chapter 6

Segmentation and registration

In order to detect changes in a scene by comparing a sample image with a reference image the images first need to be registered. Unless the scene is planar or the viewing angle remains the same between images there is no single linear transformation that can register the images. However, the datasets used in this thesis, described in Sec. 3.2, include many areas that are approximately planar. These locally planar areas can be registered if a method can be found for consistently segmenting them and defining a homography that describes the projective distortion between the views. Although a non-planar region of the scene cannot be precisely registered using a single homography a curved surface can be approximated with a number of planar facets each with an associated registering homography. The larger the number of planes the smaller the average resulting registration error across the surface will be. As described in Sec. 4.1 an affine approximation can be used to approximate a homography if the depth relief of the plane is small compared to its average depth and the size of the plane relative to the scene depth is small.

We would like to divide the two views of the scene up into corresponding regions and to establish the correspondences between images. A natural way to do this would be to use a method of image segmentation on the basis of intensity, colour or texture; this would have the advantage that the segment boundaries would tend to align themselves with object boundaries [Che95, CM02, GC03, BVZ01]. Unfortunately, the segmentations of two different images are normally completely different and it would be necessary to split and merge regions; even then there is no straightforward way of establishing reliable correspondences. The structure of the scene can be inferred from the feature points, for example for the purpose of object matching as in [DCC14]. We have developed novel approaches for segmenting the images into corresponding approximately planar regions, using feature points and edges to infer the structure of the scene. These methods allow for the definition of either projective or affine transformations for the registration of each other those regions. The methods attempt to define the regions so that they lie on approximately planar regions of the scene so that the transformations provide good registration.
6.1 Delaunay triangulation and affine compensation

In this section a method that segments the reference and sample images into approximately consistent triangular regions defined by three corner points is described. If the region of the scene within the segment is approximately planar it can be registered using an affine transformation. As an affine transformation has 6 degrees of freedom it can be defined by the three points that define the triangle as described in Sec. 2.1.2.4. Segmenting the scene into triangles defined by correspondences between the reference and sample images will therefore allow an affine approximation to be found for areas of the scene where the planar triangular segments closely match planar areas of the scene. Segmenting into triangles also provides the most granular segmentation possible, purely using point correspondences.

The Delaunay triangulation and affine compensation approach introduced here uses correspondences found using the ASIFT feature points introduced in Sec. 4.1 together with A-Harris feature points described in Sec. 4.2.2 and the method of increasing feature point densities using ground plane compensation introduced in Sec. 4.2. The resulting correspondences are used to define a triangular mesh found using Delaunay triangulation [Del34]. Delaunay triangulation defines a triangular mesh such that no other point lies within the circle defined by the three points of any triangle. This property maximises the minimum angle of the triangles in the mesh and so tends to avoid skinny triangles. This property is of benefit when calculating geometric transformations as the corner points of skinny triangles are close to being collinear. The triangulation is defined by the feature point locations in the reference image and the resulting triangular mesh is then applied to the sample image by placing the nodes of the mesh onto the corresponding locations of the feature points in the sample image. This results in a number of triangular segment that correspond across both images. As each triangle within the mesh is defined by points that have been matched between the reference and sample images the areas of the images within the triangles will be approximately consistent if the area of the scene imaged within the triangle is approximately planar. Each triangular segment can then be registered using an affine transformation defined by the corner points. This will result in low registration errors across the images in regions where the scene is locally close to planar as long as the conditions where affine transformation provides a good approximation of a projective transformation described in Sec. 4.1 hold.

If a feature point is located on an object displaced significantly from the plane it is possible for segments of the resulting triangular mesh in the sample image to overlap. Matching errors when finding correspondences may also cause this to happen. In both circumstances the mesh is adjusted to remove any overlaps, this is done by removing the points that cause overlaps in order to produce a mesh that does not overlap in either image. In order to find the best points to remove the triangular segments each pixel falls within are found. Triangular segment pairs that share at least one pixel segment are found and labelled as errors. The proportion of triangular segments attached to each feature point that have
Figure 6.1: Sample images from the rotating scene, rotating scene with trees and larger rotating scene datasets, shown left, middle and right respectively showing the scene from the same viewpoint as the reference images.

Figure 6.2: Delaunay triangular masks applied to the central image of the rotating scene dataset using correspondences found when matching against images at 10°, 20° and 30° shown in images left, middle and right respectively.

errors is found and the point with the highest proportion is removed. The triangulation is re-calculated on the reference image and re-applied to the sample image and re-checked for overlaps. If more than one feature point is only connected to triangular segments with errors or if the proportion of errors in more than feature point is the same the triangulation is re-calculated with each of those points removed in turn. The version of the mesh that results in the lowest number of triangular segments that overlap on the sample image is retained. This process is repeated until no more triangular segments overlap which usually requires around a dozen iterations when constructing a mesh using 1000 correspondences.

6.1.1 Results

The image datasets introduced in Sec. 3.2 are used to illustrate the Delaunay triangulation segmentation and registration approach. The rotating scene, rotating scene with trees and larger rotating scene datasets are used. Correspondences found using the algorithms in Sec. 4 are used to construct the triangulations. The results are shown on the central reference images, for easier comparison the central sample images showing the changes in the three datasets are shown in Fig. 6.1.
Chapter 6. Segmentation and registration

Figure 6.3: Delaunay triangular masks applied to the central image of the rotating scene with trees dataset using correspondences found when matching against sample images at 10°, 20° and 30° shown in images left, middle and right respectively.

The Delaunay triangulation masks in Fig. 6.2 show the resulting mask on the rotating scene dataset at 10°, 20° and 30°. The triangular segments are mostly small on the flat surface of the scene. As ground plane compensation in Sec. 4.2.1 boosts the number of points found on the flat round surface of the scene the density of the triangular segments is higher. The largest triangular segments are are over the bottom left building; this is because the building is not present in the sample images as shown in Fig. 6.1 and so no correspondences can be found within that area. Triangular segments will be larger when covering areas that contain change, occlusion/dis-occlusion or self-occlusion/dis-occlusion as these areas cannot include any valid correspondences. The triangular segments covering, or partially covering, other objects are on average larger than those on the ground and certain areas like the roof of the garage to the bottom right of the scene do not contain any points.

Figure 6.3 shows the masks applied to the rotating scene with trees dataset. Again results are shown at 10°, 20° and 30°. This scene is more crowded with less of the ground plane visible which leads to a lower density of correspondences and so also leads to larger triangular segments and worse coverage of the scene, particularly at 30°. Self-occlusion within the trees results in far fewer correspondences being available on them. The barn does not produce any matches with the sample image at 30° as the barn roof in the sample image as shown in Fig. 6.4 is at a very acute angle on one side and the other side is very acute in the reference image as shown in Fig. 6.3. These acute angles mean that even if the projective distortion in this area is compensated for the image will be highly blurred after interpolation and so will not match well.

Images with superimposed triangular segments from the larger rotating scene dataset are shown in Fig. 6.5. The importance of dense correspondences is illustrated in Fig. 6.5 on the roof of the large building between the roads and closest to the camera. The roof has a relatively low density of correspondences but is formed from two almost planar surfaces and so it should be possible to register the area using affine transformations. In order to achieve a good registration, triangular segments need to lie on planar areas. Triangles lying wholly on the roof will provide allow for the calculation of an affine transformation that
Figure 6.4: Rotating scene with trees dataset sample image at 30° to the central image.

Figure 6.5: Delaunay triangular masks applied to the central image of the larger rotating scene dataset using correspondences found when matching against reference images at 10°, 20° and 30° shown in images left, middle and right respectively.

results in low registration error while those with one or two corners lying off the roof will not. Figure 6.5 shows that at 10° the majority of the roof is covered by triangular segments that lie wholly on the roof, this decreases as the azimuth rotation is increased to 20°, and at 30° no triangular segments lie on the roof. This illustrates how registration accuracy can degrade in areas with a low densities of correspondences.

6.2 Local Plane Clusters

The triangular plane segments found using Delaunay triangulation in Sec. 6.1 provide good registration when the area of the scene within the triangle lies close to the plane defined by the three corners of the triangular segment. At the edges of locally planar regions of the scene, the scene will not align with the triangular segment’s plane unless the feature points used to form the triangulation happen to fall precisely on the plane boundaries. For example, if there are a number of triangular segments on a flat roof, they will allow for good registration but the triangular segments at the edge of the roof consisting of corners both on and off the roof will not provide good registration. This will result in areas near plane boundaries with poor registration of the reference and sample image. If the segment contains a change the segments will not match but the change will not always make up the entire segment area which result
Chapter 6. Segmentation and registration

in areas of the segments being falsely classified as areas containing change.

This section presents a method of reducing these errors by attempting to match areas using transformations defined by neighbouring triangular segments where the areas have not already been successfully matched. The first step is to find groups of adjacent triangular segments in the scene that are approximately coplanar. This is done using a growing cluster algorithm. Triangular segments that share two feature points are defined as being neighbouring. For each neighbour the point that is not shared between the two segments is found. The three feature points that form the vertices of a triangular segment are used to define an affine transformation between the reference and sample images. The unshared point is transformed from the reference to the sample image using this affine transformation. The Euclidean distance between the transformed location and the feature point’s corresponding position in the reference image, $d_r$ is found. This represents the registration error introduced when using the neighbouring triangular segment’s affine transformation to register the image at that point. If the triangular segments are not on the same plane this error will be large.

If the triangular segments are on the same plane errors can still occur due to three things. Firstly using an affine transformation to approximate the homography the plane undergoes will introduce an error proportional to the size of the two triangular segments. Secondly localisation errors in the feature points can cause errors when defining the affine transformation, the error in the resulting point position will also be proportional to the size of the two triangular segments. The third source of error is the error caused by areas within the triangular segment not lying on the same plane as the triangular segment used to define the affine transformation. This third error is the error of interest when determining if neighbouring triangular segments lie on the same plane. To find the errors caused by this third source of error we attempt to remove the first two sets of error. To account for localisation errors, 3 is subtracted from the error while ensuring that no error is less than 0. As the other errors dependant on the dimensions of the triangular segments the error is divided by the distance between the central points of the segments, $d_c$, squared. This gives the error distance

$$d_e = \frac{\max (d_r - 3, 0)}{d_c^2}$$

Errors between all neighbouring triangular segments are found and, starting with the two segments with the least error distance, segments with errors less than a threshold (in experiments a threshold of 0.0005 is found to work best) are joined as a cluster.

If one of the triangular segments is already assigned to a cluster the other joins the same cluster and if both are already assigned to a cluster the two clusters are merged. In this way local ground plane clusters are formed. Once the clustering is complete homographies are calculated for clusters formed of more than two segments using the approach described in Sec. 6.1. In cases where the majority of the area
of the scene within a triangular segment lies close to the plane defined by the corners of the triangular segment but some areas do not the triangular segment will not match between the reference and sample images. However the areas not on the same plane as the triangular segment may lie on the same plane as a neighbouring local plane cluster. For this reason triangular segments that share two corners with a triangular segment that is in a cluster are soft assigned to that cluster which indicates that the triangular segment does not lie within the local plane cluster but some areas within it might.

The next step involves segmenting the reference image in a way that is more likely to adhere to object boundaries, and thus local planes. Mean-shift [CM02], reviewed in Sec. 2.4, can be used. Mean-shift segments that contain areas that were not successfully matched when registered using Delaunay triangulation and affine compensation are identified. Triangular segments that overlap the identified mean-shift segments are found. If the triangular segments are assigned or soft assigned to a local plane cluster, the homography of that cluster can be used to register the mean-shift segment.

### 6.2.1 Results

Figure 6.6 shows all local plane clusters containing more than 10 triangular segments on the larger rotating scene dataset. The majority of local plane clusters lie on the ground plane of the scene but some are also found on buildings. Smaller clusters that are not shown here are also found on the roofs of the large warehouse labelled ‘1’ and the closest building to the right of the roads near the label ‘2’. Some areas within certain clusters do not seem to lie on the same plane, for example clusters labelled ‘1’, ‘2’ and ‘3’. Areas ‘1’ and ‘2’ show errors which have been caused by feature point localisation inaccuracies. The cluster at label ‘3’ shows a cluster consisting of an area that does not seem to lie on a single plane. This has occurred because the component triangular segments do not align well with the area contained within them but happen to align well to each other. At label ‘4’ a building falls almost completely within two local plane clusters. This has occurred because the building is only present in the reference image which means that no feature points lie on the building.

These errors will cause the homography of the cluster not to provide good registration in certain areas of the local plane cluster. As the homography is only used to match triangular segments within or adjacent to the cluster after an initial match has already been attempted using the Delaunay triangulation approach in Sec. 6.1 a failure to match the areas due to these errors will not cause additional false negatives.

The local plane cluster registration approach is assessed further as part of an end to end change detection system in Sec. 7.
Figure 6.6: Illustration of local plane clusters containing more than 10 triangular segments on the larger rotating scene dataset. Local plane clusters are highlighted as lighter regions with red lines denoting the boundaries of each cluster. Label ‘1’, ‘2’ and ‘3’ show triangular segments that do not seem to fall on the same plane as the remainder of the cluster. Label ‘4’ shows an object that is only present in one of the images.
6.3 Object Edge Hypothesis Testing

As mentioned at the start of Sec. 6.2 the Delaunay segmentation and registration technique described in Sec. 6.1 can result in registration errors around the boundary of two planes where the boundary bisects the triangular segment. This error is addressed in cases where the area is adjacent to a local ground plane cluster that lies on the same plane as the erroneous area as described in Sec. 6.2, but in many cases either one or both sides of the boundary will not be near a local ground plane cluster. In this section a solution called Object Edge Hypothesis Testing (OEHT) is presented that attempts to register the areas separately uses the epi-polar geometry of the two camera positions together with the matched feature points and an edge-map to define possible edge configurations. Each configuration is then tested in turn to attempt to find a match. This allows the triangulation approach to be extended to complex scenes that involve edges as well as occlusions which are common in real world outdoor scenes. Without this capability the triangulation approach is restricted to approximately flat scenes.

The epi-polar geometry is defined by a fundamental matrix and can be found using a RANSAC [FB81] approach. The fundamental matrix can be used to find the line that a point in one image must lie on in the other image. This is known as its epi-polar line as reviewed in Sec. 2.1.2. An edge map can be found using an edge detector but the boundaries of a mean-shift segmentation were found to be more consistent between viewing angles. Although mean-shift involves greater computational complexity than standard filter based edge detectors, the resulting edges follow object boundaries well. As the aim of the thesis is to illustrate possible solutions to the challenges of wide-baseline image change detection rather than efficient implementations mean-shift is used to generate the edge map used.

In many cases an edge that separates two approximately planar regions may bisect a triangular segment. In this section the scene geometry configurations shown in Fig. 6.7 and Fig. 6.8 are used to attempt to match each triangular segment that has not already been successfully matched. Each needs to be applied in every orientation of the triangular segment except in situations where one or more corner point falls on the same local plane cluster from Sec. 6.2. In this case the edge that bisects the triangular segment must separate points in the same cluster from the other points.

Fig. 6.7 illustrates the case where the triangular segment encompasses two planes that are attached at an edge where the edge forms a bisecting line. In this case one of the planes will contain two of the corners of the triangular segment and the other will contain one. The bisection of the planes will form a straight line, shown as a dotted black line. The points where the triangular segment edges cross the dotted line of the plane edges in the reference image can be used to re-define the triangular segment as a triangle and a quadrilateral. The cases shown in Fig. 6.8 show a similar problem but in this case one of the planes is occluding the other. When viewed from a different angle the dotted line will still fall in a consistent position on the upper plane but the amount occluded on the lower plane may vary.
Figure 6.7: Triangular segment geometry when the image within the segment contains two connected planes at the edge of an object with no occlusion with dotted line showing the plane boundary. Left (A): The case where two corners of the triangular segment lie on the object. Right (B): The case where one corner of the triangular segment lies on the object.

In the configurations shown in Fig. 6.7 the plane edge in the reference image can be used together with epi-polar geometry to find the corresponding position in the sample image. In the configurations shown in Fig. 6.8 additional information is needed to define a transformation used to register the ground plane but the initial steps to define the bisecting edge are the same.

The first step is to find the points on both the reference and sample image where the edge of the triangular segment intersect the line bisecting the segments. These are the points in Fig. 6.7 and Fig. 6.8 where the sold black triangle edges intersect the dashed black plane edges. It is quite common to have multiple segments within each plane and so there may be more than one crossing point on each line triangular segment edge. In these cases each candidate bisecting edge can be used in the following steps in turn to attempt to register and match the triangular segment, only the correct choice should result in accurate registration and a successful match. In order to find these crossing points a straight line along the edge-map needs to be detected. A crossing point from one triangular segment edge needs to be paired with a crossing point on another edge to define a line, each possible pairing is tested in turn. Pairs that are not joined by a straight edge need to be excluded as a straight line is needed to define the bisecting edge shown as a dashed black line in the figures. Edge-maps are often inconsistently localised and an apparently straight line in a scene may not result in a straight line of pixels in the edge-map. To compensate for this the edge map is convolved with a $3 \times 3$ flat filter with intensity 1 to produce a range of possible edge positions. The proportion of possible edge pixels along the straight line from one crossing point to the other must be over a threshold for the line to be confirmed, a threshold of 0.7 is used.

As shown in Fig. 6.7 the triangular segments are bisected by the dashed edge line to form a triangle and a
quadrilateral sub-region. In the reference image the discovered crossing points are used as correspondences to define the triangle and the quadrilateral sub-region of the triangular segment. The crossing points in the sample image are not the same points on the scene as the crossing points in the reference image. Instead they can be used to define the edge line, shown as dashed black line. The crossing points in the reference image are now used together with the fundamental matrix to find their epi-polar lines in the sample image. The points where the epi-polar lines intersect the dashed black edge line in the sample image are the corresponding positions of the reference image crossing points as illustrated in Fig. 6.9. In situations where the epi-polar line and the edge line are parallel or close to parallel the corresponding key points may not be found or may be poorly localised.

The triangular segment corner points together with the crossing points can now be used as point correspondences to define transformations for the two planes within the triangular segment. The plane with two corner points will now have 4 available points which can be used to define a homography while the other plane has 3 points which can be used to define an affine transformation.

If both of the planes can be matched when registered using the new transformations the configuration must not contain occlusion as shown in Fig. 6.7. Otherwise if one of the planes match but the other does not the configuration may contain occlusion as shown in Fig. 6.8 and so these configurations are now also tested for. If neither plane matches then OEHT cannot register the triangular segment and it remains marked as not matching. The positions of the edge lines on the lower planes in Fig. 6.10 do not remain consistent between viewing angles if there is any occlusion, for example the area occluded in Fig. 6.10(G) is not occluded and shown as a red dotted line and red shaded area in Fig. 6.10(H). Because of this its position cannot be used to register the lower plane. In this configuration a neighbouring triangular segment positioned away from the edge, for example the adjacent triangular segments shown in Fig. 6.10 is used to define the registration transformation. This adjacent triangular segment can be used to
define an affine transformation or alternatively if the edge points on the lower plane are part of a local plane cluster the cluster’s homography can be used. When registering the lower plane using the new transformation, areas of the lower plane visible in one viewpoint may be obscured in the other viewpoint as we have seen in Fig. 6.10. As the obscured area is only visible from one viewpoint it is impossible to tell if a change is present in that area without further information. This area is marked as being an unknown: an area that we cannot classify.

6.3.1 Results

OEHT is illustrated using the triangular segment highlighted in white in the reference image shown in Fig. 6.11. The plane of this triangular segment clearly does not align well with the scene but as it consists of two planes bisected by a straight edge it is a good candidate for OEHT. The edge-map of the mean-shift [CM02, GC03] segmentation of the image shown in Fig. 6.12 is used.

The points where the edges in the edge-map intersect the edges of the triangular segment in the reference image are shown in Fig. 6.13. In this situation the edges of the triangular segment that intersect the edge of the building are known as two of the triangular segment corner points lie on the same local plane cluster. The multiple edge boundaries points in the reference image are due to shadows as the wall of the building is not visible while in the sample image the wall of the building becomes visible and crossing
Figure 6.10: Triangular segment geometry when the region within the segment contains two planes overlapping at the edge of an object with occlusion, the black dotted line in the left figure shows the plane boundary. Left (G): View in reference image. Right (H): View in sample image with points marked with a cross representing the corresponding locations of the points where the triangular segment edge intersects with the edge-map in the reference image. The red dotted line represents the boundary of the area of the lower plane that is in view in the reference image with the area of the triangular segment that is occluded in the reference image but dis-occluded in the sample image is shaded red.

Figure 6.11: Image from the larger rotated scene dataset showing the triangular segment used to illustrate OEHT highlighted in white.
points are found at the base as well as the top of the wall. The crossing points produce two candidate lines in both the reference and sample images, these are shown in Fig. 6.14. The right edge caused by the shadow does not follow a straight enough path to be detected as a line. In the reference image the line defined by the blue diamonds do not define an object edge as they are caused by a shadow but in the sample image both lines represent an edge in the object. The upper green diamonds in the sample image correspond to the green diamonds in the reference image as they are both on the edge of the roof.

The algorithm determines which of the configurations shown in Fig. 6.7 and Fig. 6.8 are the correct configuration for this triangular segment by trying each in turn. A matching algorithm, for example shifted dense SIFT introduced in Sec. 5.3, is used to find which if any of the configurations result in a registration that give a matching score below a threshold. The result shown in Fig. 6.15 is the only configuration that produces a match for this triangular segment. Here the area on the roof successfully match when using the corner point on the roof and the edge line to register the region, the quadrilateral on the ground matches when using a transformation defined by the local plane cluster. This leaves a strip of the wall of the building that is occluded in the reference image but not in the sample image which is marked as unknown.
Figure 6.13: Example triangular segment for OEHT with edge overlays in white. The reference image is on the left and sample image right. Triangular segment corner points are shown as red circles and the points where the edges intersect with the triangular segment edges are shown as blue diamonds.

Figure 6.14: Example triangular segment for OEHT with edge overlays in white. The reference image is on the left and sample image right. Triangular segment corner points are shown as red circles, endpoints of one candidate line are shown as blue diamonds and the endpoints of the other are shown as green diamonds.

Figure 6.15: Lightened and darkened areas of the reference image on the left are registered to the correspondingly coloured areas in the sample image on the right. Darkened area on the roof is registered using the triangular segment corner point and the line following the edge of the roof defined by edge detection. Lower lightened area is registered using a homography defined by a neighbouring local plane cluster.
6.4 Standing Object Matching

Outdoor scenes in both urban and rural environments typically consist of a global ground plane as defined in Sec. 4.2.1 or a number of local ground planes as defined in Sec. 6.2 with a number of objects situated on top. The objects might include walls, buildings, cars, people, trees or posts. OEHT in Sec. 6.3 registers regions where the objects consist of approximately planar sides. Many objects sit vertically on the plane, for example walls, lamp posts, people, bins and trees. The geometry of the visible surface of these objects can be modelled as planes that are orthogonal to the ground plane and intersecting the ground plane at the base of the object. In this section a method of finding a homography that can be used to register this plane is introduced. First the camera matrices of the two views need to be found. Using these together with the ground plane homography and a direction vector parallel to the ground plane that determines the vertical plane’s orientation, the homography of the vertical plane can be found. These steps are explained in the next two subsections, Secs. 6.4.1 and 6.4.2.

6.4.1 Finding the camera matrices from two views

The fundamental matrix of a two view camera system can be found as explained in Sec. 2.1.3. The focal length can be found using the relationship

\[ 2FQF^TQF - \text{trace}(FQF^TQ)F = 0 \]

according to Sturm et al. [PSP05, Stu01] assuming that the camera internal matrices, \( K_1 \) and \( K_2 \) are the same and are equal to \( Q \) where \( Q = \text{diag} \left( 1, 1, 1/f_d^2 \right) \) and \( f_d \) is the focal length which is assumed to be the same for both cameras. The essential matrix can then be found as \( E = QFQ \) if the camera is not skewed. If the first camera matrix is defined to be at the origin and aligned to the coordinate system then \( P_0 = \begin{bmatrix} 1 & 0 \end{bmatrix} \) and the second camera matrix can take one of the following 4 possible choices as shown by [Stu01, PSP05, HZ04]

\[
P_1 \in \left\{ \begin{bmatrix} U & V^T & u_3 \end{bmatrix}, \begin{bmatrix} U & V^T & -u_3 \end{bmatrix}, \begin{bmatrix} U & V^T & u_3 \end{bmatrix}, \begin{bmatrix} U & V^T & -u_3 \end{bmatrix} \right\}
\]

where \( U \) and \( V \) are found from the singular value decomposition

\[
E = U \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix} V^T
\]
and

\[
W = \begin{bmatrix}
0 & -1 & 0 \\
1 & 0 & 0 \\
0 & 0 & 1
\end{bmatrix}.
\]

A single correspondence between the images from the two cameras can be used to determine the correct camera matrices from the possible set.

### 6.4.2 Finding the vertical plane homography

The homography, \( H \) of a plane can be found using the camera matrices of the two views \( P_0 \) and \( P_1 \) and the ground plane

\[
\pi = \begin{bmatrix}
w^T \\
\omega 
\end{bmatrix}^T
\]  
(6.1)

if the origin does not lie on the plane. The coordinate system is defined so that \( P_0 \), the reference camera matrix, is at the origin and aligned with the coordinate system with \( K_0 \) representing its internal parameters.

\[
P_0 = K_0 \begin{bmatrix}
I & 0
\end{bmatrix}.
\]  
(6.2)

The sample image camera matrix is defined as

\[
P_1 = K_1 R_1 \begin{bmatrix}
I & -c_1
\end{bmatrix}
\]  
(6.3)

where \( K_1 \) represents the internal characteristics, \( R_1 \) represents the orientation and \( c_1 \) represents the position of the sample image camera. The point on the reference image \( x_0 \) that lies on the plane \( \pi \) can be back projected to find the first 3 elements of its 3D position, \( x \) leaving its fourth element as a scale ambiguity. \( x \) and \( x_0 \) are related by the first camera matrix and then the first three elements of \( x \), denoted \( x_{1,3} \) are found in terms of \( K_0 \) and \( x_0 \) by using equation 6.2 and re-arranging:
\[
P_0 \mathbf{x} = \mathbf{x}_0
\]
\[
\mathbf{K}_0 \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \mathbf{x} = \mathbf{x}_0
\]
\[
\mathbf{K}_0 \begin{bmatrix} 1 & 0 \\ 1 & 0 \end{bmatrix} \mathbf{x} = \mathbf{K}_0^{-1} \mathbf{x}_0
\]
\[
x_{1-3} = \mathbf{K}_0^{-1} \mathbf{x}_0.
\]  

(6.4)

This leaves the fourth element of \( \mathbf{x} \), \( \rho \) undefined. As \( \mathbf{x} \) must lie on \( \pi \) we know that \( \mathbf{\pi}^T \mathbf{x} = 0 \) and using equation 6.1 and 6.4 we can find \( \mathbf{x} \) in terms of \( \mathbf{x}_0, \mathbf{K}_0 \) and the ground plane:

\[
\mathbf{\pi}^T \mathbf{x} = 0
\]
\[
\begin{bmatrix} w^T & \omega \end{bmatrix} \begin{bmatrix} \mathbf{K}_0^{-1} \mathbf{x}_0 \\ \rho \end{bmatrix} = 0
\]
\[
(w^T \mathbf{K}_0^{-1} \mathbf{x}_0 + \omega \rho) = 0
\]
\[
-w^T \mathbf{K}_0^{-1} \mathbf{x}_0 / \omega = \rho
\]
\[
\mathbf{x} = \begin{bmatrix} \mathbf{K}_0^{-1} \mathbf{x}_0 \\ -w^T \mathbf{K}_0^{-1} \mathbf{x}_0 / \omega \end{bmatrix}
\]
\[
\mathbf{x} = \begin{bmatrix} \omega \mathbf{x}_0 \\ -w^T \mathbf{x}_0 \end{bmatrix} \mathbf{K}_0^{-1}
\]  

(6.5)

We can relate the ground plane, \( \mathbf{\pi} = \begin{bmatrix} w^T & \omega \end{bmatrix}^T \) to the ground plane homography, \( \mathbf{H} \) and the camera parameters by projecting \( \mathbf{x} \) onto the sample image using the sample camera defined by equation 6.3 and equation 6.5:

\[
x_1 = P_1 \mathbf{x}
\]
\[
x_1 = \mathbf{K}_1 \mathbf{R}_1 \begin{bmatrix} I & -c_1 \end{bmatrix} \begin{bmatrix} \omega \mathbf{x}_0 \\ -w^T \mathbf{x}_0 \end{bmatrix} \mathbf{K}_0^{-1}
\]
\[
x_1 = \mathbf{K}_1 \mathbf{R}_1 (\omega \mathbf{I} \mathbf{x}_0 + c_1 w^T \mathbf{x}_0) \mathbf{K}_0^{-1}
\]
\[
x_1 = \mathbf{K}_1 \mathbf{R}_1 (\mathbf{I} + c_1 w^T) \mathbf{x}_0 \mathbf{K}_0^{-1}
\]

as \( \mathbf{K}_0^{-1} \) is diagonal we can write
\[ x_1 = K_1 R_1 (\omega I + c_1 w^T) K_0^{-1} x_0 \]

and as \( x_1 = H x_0 \)

\[
H = K_1 R_1 (\omega I + c_1 w^T) K_0^{-1} \\
H = \omega K_1 R_1 K_0^{-1} + K_1 R_1 c_1 w^T K_0^{-1}.
\]

As the internal parameters of the cameras \( K_0 \) and \( K_1 \) are the same this can be written as

\[ H = \omega K_1 R_1 K_1^{-1} + K_1 R_1 c_1 w^T K_1^{-1}. \]

We can now re-arrange to find \( \pi \) in terms of \( H \) using \( \pi = \left[ \begin{array}{c} w^T \\ \omega \end{array} \right]^T \) and the pseudo-inverse of the camera parameters

\[
H = \left[ \begin{array}{cc} K_1 R_1 c_1 K_1^{-1} & K_1 R_1 c_1 K_1^{-1} \\
K_1 R_1 c_1 K_1^{-1} & K_1 R_1 c_1 K_1^{-1} \end{array} \right] \pi \\
\pi = \left[ \begin{array}{cc} K_1 R_1 c_1 K_1^{-1} & K_1 R_1 c_1 K_1^{-1} \end{array} \right]^+ H.
\]

As \( \left[ \begin{array}{cc} K_1 R_1 c_1 K_1^{-1} & K_1 R_1 c_1 K_1^{-1} \end{array} \right] \) is full rank the pseudo-inverse can be found using \( A^+ = (A^T A)^{-1} A^T \).

Using this relationship, if we know the camera matrices and can find the homography of the ground plane we can find \( \pi \), the ground plane in 3D. As \( \pi^T x = 0 \) for any \( x \) on the ground plane the 3D position of the point on the ground plane at the base of the object, \( x_g \) can be found. We now have the location of the point on the ground plane and the position of the ground plane in 3D, which together with the location of the cameras can be used to find the 3D position of the plane that is orthogonal to the ground plane, parallel to the baseline, and intersects the point at the base of the object on the ground plane, \( \pi_b = \left[ \begin{array}{c} u_b^T \\ u_b \end{array} \right]^T \). The plane normal, \( u_b \) is orthogonal to \( w \) and the baseline, \( c_1 - c_0 \) and so \( u_b \) can be defined as

\[ u_b = w \times (c_1 - c_0) \]

and since \( c_0 = 0 \) we obtain
\[ \mathbf{u}_b = \mathbf{w} \times \mathbf{c}_1. \]

This leaves \( u_b \) to be found which can be found using \( x_g \) as \( x_g \) lies on \( \pi_b \):

\[
\begin{bmatrix}
\mathbf{u}_b \\
u_b
\end{bmatrix} x_g = 0.
\]

The vertical plane homography, \( \mathbf{G}_b \), can be found using the camera parameters and \( \pi_b \):

\[
\mathbf{G}_b = \begin{bmatrix}
K_1 \mathbf{R}_1 \mathbf{c}_1 K_1^{-1} & K_1 \mathbf{R}_1 \mathbf{c}_1 K_1^{-1}
\end{bmatrix} \pi_b.
\]

Alternatively instead of defining the vertical plane as being parallel to the baseline, its orientation can be defined by an additional point that lies on the line intersecting the ground and vertical plane. For example if the visible region of the standing object approximately lies on the vertical plane \( \pi_w \), for example in the case of a wall, and at least two ground points, \( x_g \) and \( x'_g \) along the base of the wall can be found, the vertical plane can be defined to cross both those points instead of being defined as parallel to the baseline.

\[
\pi_w = \begin{bmatrix}
\mathbf{u}_w^T \\
u_w
\end{bmatrix}^T
\]

\[
\begin{bmatrix}
\mathbf{u}_w \\
u_w
\end{bmatrix} x_g = 0
\]

\[
\mathbf{G}_w = \begin{bmatrix}
K_1 \mathbf{R}_1 \mathbf{c}_1 K_1^{-1} & K_1 \mathbf{R}_1 \mathbf{c}_1 K_1^{-1}
\end{bmatrix} \pi_w
\]

Another approach is to orientate the vertical plane, \( \pi_c = \begin{bmatrix}
\mathbf{u}_c^T \\
u_c
\end{bmatrix}^T \) so that the line between \( x_g \) and the point on the ground plane nearest to the halfway point between the two cameras, \( c_g \) is orthogonal to the vertical plane. The orientation of the vertical plane, \( \mathbf{u}_c \) can be found from \( c_g \) and \( x_g \)

\[
\mathbf{u}_c = c_g - x_g
\]

and \( u_c \) can be found using

\[
\begin{bmatrix}
\mathbf{u}_w \\
u_w
\end{bmatrix} x_g = 0.
\]
\( c_g \) can be found from the halfway point between the cameras

\[
\mathbf{c}_m = \frac{\mathbf{c}_1 + \mathbf{c}_0}{2}
\]

as the line from \( \mathbf{c}_m \) to \( c_g \) must be orthogonal to the line between \( c_g \) and \( \mathbf{x}_g \) because \((c_g - \mathbf{x}_g)\) lies in the ground plane:

\[
(c_g - \mathbf{c}_m)(\mathbf{c}_g - \mathbf{x}_g) = 0
\]

### 6.4.3 Registration walls using the vertical plane homography

Registration using the vertical plane homography is illustrated as part of an algorithm to match vertical planar sections. The reference image is segmented using mean-shift segmentation introduced in Sec. 2.4.1. The hypothesis that each segment that covers an area of the reference image that has not been matched is a vertical plane is tested. It is assumed that the ground forms an approximately horizontal plane. If the segment is approximately a vertical plane it will intersect with a ground plane along a straight line and will be approximately orthogonal to it. Two points along this line are needed together with the camera matrices of the reference and sample view-points and the ground plane homography to determine the vertical plane homography. If the segment is a plane intersecting the ground plane along that line the homography will register the image correctly and it should match.

The method of defining the vertical homography was presented in the previous section. The intersecting line between the vertical plane and the ground plane can be found from the mean-shift segment. If the camera is approximately horizontal, one end of the intersecting line will lie on the bottom point of the vertical plane’s segment and the intersecting line will form a straight line from that point. As mean sift segmentation is unlikely to produce an edge aligned perfectly with the intersection of the two planes a method tolerant to misalignments is needed. A method for determining the intersecting edge when the candidate segment forms an approximate plane with straight vertical edges at the ends of the intersecting line in now described.

The edge mask of the mean-shift segment is filtered using a Gaussian filter. This accounts for the uncertainty of the position of the edge by representing the true edge position as a probability over an area. The Harris corner detector is applied to detect possible corners of the edge mask. Detected corners within a 50 pixel threshold distance from the lowest point of the segment in the \( y \)-axis are selected as candidate intersecting line endpoints. The candidates are tested to determine if probable lines can be formed between them and other corner points through the filtered edge map. The best performing candidate is selected as the corner of the segment polygon. For each candidate point the sum of the
filtered edge-map pixel values between the candidate point and all other detected corner points is found. This value is divided by the log of the distance between the points to give measure of how well the two points define a line in the edge-map. The candidate intersecting line endpoint for which the second strongest line that is at least 30° from the strongest line is selected. The two strongest lines that are at least 30° apart are selected as the two lines that meet at the candidate intersecting line endpoint. If both lines are within 20° of the horizontal the candidate point is likely to be along the intersecting line but not at its end. In this case the corner point at the end of the line that is closest to the horizontal is used as the new candidate intersecting line endpoint and the process is repeated. Once two acceptable lines from a corner point are found the line closest to horizontal is selected as the intersecting line and the two corner points defining it can be used to define the vertical homography.

Now that the vertical plane segment can be registered it can be matched using shifted dense SIFT from Sec. 5.3 which can tolerate the expected levels of registration error when registering using this method.

### 6.4.4 Results

The technique is illustrated on the 4 vertical segments highlighted in red and numbered in Fig. 6.16. These segments are found using the mean shift segmentation algorithm.

The dominant ground plane is defined using the points marked as white circles in Fig. 6.17. The lowest
Figure 6.17: Reference image top and sample image bottom showing the ground points used to define the dominant plane homography as white circles. The corner points defining the edge lines that meet at the lowest corner point in the y-axis of each of the four segments are shown as black squares, diamonds, asterisks and crosses.
corner point in the y-axis in each of the four segments along with the endpoints of the edge lines meeting at that point are shown as black squares, diamonds, asterisks and crosses respectively. These corners are found using the method described in Sec. 6.4.3. The 2 lowest points in each set of 3 points is used together with the camera matrices and the dominant ground plane to define a vertical homography for each plane. The registered segments from the sample image are shown alongside the segments from the reference image when registered using the vertical plane homography are shown in Fig. 6.18. Segments 1 to 3 register well with no significant registration errors between the sample and reference images. Segment 4 has an obvious error with the sample image including an areas to the right of the segment in the registered image. This error is due to the segment not lying directly on the ground plane as the building lies on a raised mound. Even in this case the error is not very large and shifted dense SIFT from Sec. 5.3 configured to allow for large registration errors would still be expected to provide a correct match.
Figure 6.18: Reference image segments left and registered sample image right when registered using the vertical plane homography. Segments from the top to the bottom correspond to segments 1 to 4 in Fig. 6.16.
Chapter 7

End-to-end change detection system

This chapter presents an end-to-end change detection system which utilises the techniques introduced in Chapters 4, 5 and 6. The chapter presents a system architecture, the reasoning behind the architecture choices and an evaluation of the system’s performance using the datasets introduced in Sec. 3.2. The system attempts to match areas of the reference image to areas of the sample image, using each method in turn to match areas that are unmatched. Areas that cannot be matched are marked as change while matched areas are marked as not containing change. Any occlusion detected is also marked as such.

The main steps as illustrated in Fig. 7.1 are:

1. Find point correspondences using the techniques presented in Sec. 4 and define the fundamental matrix and as well as the dominant ground plane homography.

2. Attempt to match relatively large planar areas of the scene using the Delaunay triangulation registration method from Sec. 6.1 and the DSIFT matching method from Sec. 5.2.

3. Attempt to register areas that have not yet been matched using OEHT as described in Sec. 6.3 which are then matched using the SDSIFT matching method introduced in Sec. 5.3.

4. Find local plane clusters as described in Sec. 6.2, using the homography defined by each cluster to register areas within or adjacent to each cluster that have not yet been successfully matched. SDSIFT is used to attempt to match the areas.

5. Attempt to register unmatched segments using the vertical plane homography defined at the point the segment intersects the dominant ground plane to register the segment as described in Sec. 6.4. SDSIFT is then used to match the segments.

The principle applied is to use methods that produce low false negatives and which are less computationally intensive to exclude as many areas that do not contain change before using less reliable and more
1. Find correspondencies, fundamental matrix and dominant ground plane

2. Delaunay triangulation registration with dense SIFT matching
   - No match

3. OEHT registration with shifted dense SIFT matching
   - No match

4. LPC registration with shifted dense SIFT matching
   - No match

5. Vertical homography registration with shifted dense SIFT matching
   - No match

- Match
- No change

Figure 7.1: End-to-end change detection system architecture
computationally intensive methods on the remaining areas to attempt to separate the positives and false positives. This approach maintains a low false negative rate and so also a low miss rate while reducing the false positive rate and minimising the risk that possibly pertinent changes are missed.

The correspondences, fundamental matrix and ground plane homograph collected in the feature point evaluation results for the combined ASIFT, Affine-Harris with ground plane pre-compensation approach in Sec. 4.2.3 form a prerequisite first step in the end-to-end change detection system, providing geometric information used in the subsequent steps.

7.1 Delaunay triangulation and dense SIFT matching

The second step is to use Delaunay triangulation to register regions of the images as described in Sec. 6.1 and to then compare those regions using the DSIFT matching technique from Sec. 5.2. The step is evaluated using the rotating scene, rotating scene with trees and two reference viewing angles from the larger rotating scene dataset. The reference images for each dataset are shown in Fig. 7.2. ASIFT feature points as well as Affine-Harris and the ground plane pre-compensation technique introduced in Sec. 4 on page 50 are used as correspondences to form the Delaunay triangulation. Scale 4 SIFT descriptors are used in the DSIFT matching technique with a one pixel spacing in the dense descriptor grid of the
Figure 7.3: Equal error rates of the Delaunay triangulation registration and dense SIFT matching step of the end-to-end change detection system.

DSIFT matching technique. The number of matching distances above 300,000 resulting from comparing the grids collected from the reference image and sample image determines whether the triangular segment is successfully matched. If areas are successfully matched they must not contain change and so those areas are marked as negatives, while other areas are marked as positives. This classification together with a ground truth change classification of each pixel in the reference image is used to determine the false positive and false negative rate. The resulting equal error rates are shown in Fig. 7.3. As expected the performance decreases with rotation angle and also varies significantly between datasets. The first step produces the lowest error rates on the rotating scene dataset as the scene is relatively simple with large amounts of flat space and little occlusion. The system performs worst on the rotating scene with trees as there is less flat ground visible and many objects occlude each other. In particular the trees cannot be registered well using affine transformations as their shapes cannot be well approximated as a small number of discrete planes. The performance using the two different viewing angles of the larger rotating scene dataset vary significantly. The same variations in performance can be seen in the ROC curves at an azimuth rotation of 10° between viewing angles in Fig. 7.4. The result mask in Fig. 7.5 shows a larger number of false positives in the first view compared to the second view, these false positives are concentrated on the buildings near to the camera. The errors are caused by the triangular segments not lying on a single plane and so they are not correctly registered. This explains the poorer performance on the first viewing angle of the larger rotating scene dataset.

### 7.2 OEHT and SDSIFT matching

The next step of the end-to-end change detection system, as shown in Fig. 7.1, is to use OEHT from Sec. 6.3 for registration, and SDSIFT from Sec. 5.3 for matching. This is carried out on the areas that have not yet been successfully matched and so are currently marked as positives. This includes true positives
Figure 7.4: ROC curves of the Delaunay triangulation registration and dense SIFT matching step of the end-to-end change detection system at an azimuth rotation of 10°

Figure 7.5: Result mask of the Delaunay triangulation registration and dense SIFT matching step of the end-to-end change detection system at an azimuth rotation of 10° on the first view of the larger rotating scene dataset, left, and the second view, right. White areas are correct negatives, red are false positives and blue are false negatives. Unshaded areas are outside of the Delauney mask.

Figure 7.6: Result mask of the Delaunay triangulation registration and dense SIFT matching step of the end-to-end change detection system at an azimuth rotation of 10° on the rotating scene dataset, left, and the rotating scene with trees view, right. White areas are correct negatives, red are false positives and blue are false negatives. Unshaded areas are outside of the Delauney mask.
Chapter 7. End-to-end change detection system

<table>
<thead>
<tr>
<th>Data</th>
<th>Delaunay Triangulation FP</th>
<th>Delauney Triangulation FN</th>
<th>OEHT FP</th>
<th>OEHT FN</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotating scene</td>
<td>4.2%</td>
<td>11.8%</td>
<td>3.9%</td>
<td>11.8%</td>
</tr>
<tr>
<td>Larger rotating scene 1</td>
<td>23.1%</td>
<td>15.0%</td>
<td>12.4%</td>
<td>15.0%</td>
</tr>
<tr>
<td>Larger rotating scene 2</td>
<td>15.2%</td>
<td>27.1%</td>
<td>13.2%</td>
<td>31.1%</td>
</tr>
</tbody>
</table>

Table 7.1: False positive and false negative rates of of the second step of the end-to-end change detection system on the rotating scene and the two viewing angles of the larger rotating scene where the reference image and sample images are separated by an azimuth rotation of 10°.

![Image](image.png)

Figure 7.7: Result from the OEHT and SDSIFT on the rotating scene dataset at a difference in viewing angle of 10°. Left: Ground truth change areas highlighted in yellow. Areas reclassified from false positives to true negatives highlighted in green.

and false positives. Areas that are successfully matched using OEHT and SDSIFT can be re-assigned as negatives and so the false positive rate can be reduced. SDSIFT is used rather than DSIFT as the registration errors are larger when using OEHT to register areas compared to Delaunay triangulation due to edge localisation errors and inaccuracies when finding the fundamental matrix.

Each configuration of OEHT is applied as explained in Sec. 5.3 and each registration attempt is compared using SDSIFT. If any of the configurations successfully match the triangular segment is marked as a negative. This step is conducted on the rotating scene dataset, and both viewing angles of the larger rotating scene dataset. The second step is not applied to the rotating scene with trees as the false positives within that scene occur mainly on triangular segments containing trees. As trees do not consist of approximately planar regions, OEHT will not provide good registration. The step is applied to the results of the first step at an azimuth viewing angle between the reference and sample images of 10°. The resulting error rates are shown in Table 7.1.

It can be seen in particular in Fig. 7.8 that OEHT combined with SDSIFT works well when there are large planar area separated by definite straight edges. It does not work on more complex shapes or where the planes are small and so do not contain enough feature points, for example on vehicles.
Figure 7.8: Result from the OEHT and SDSIFT on the first viewing angle of the larger rotating scene dataset at a difference in viewing angle of 10°. Left: Ground truth change areas highlighted in yellow. Areas reclassified from false positives to true negatives highlighted in green.

Figure 7.9: Result from the OEHT and SDSIFT on the second viewing angle of the larger rotating scene dataset at a difference in viewing angle of 10°. Left: Ground truth change areas highlighted in yellow. Areas reclassified from false positives to true negatives highlighted in green.
<table>
<thead>
<tr>
<th>Data</th>
<th>Delaunay Triangulation FP</th>
<th>Delauney Triangulation FN</th>
<th>OEHT FP</th>
<th>OEHT FN</th>
<th>LPC</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rotating scene</td>
<td>4.2%</td>
<td>11.8%</td>
<td>3.9%</td>
<td>11.8%</td>
<td>0.4%</td>
</tr>
<tr>
<td>Larger rotating scene 1</td>
<td>23.1%</td>
<td>15.0%</td>
<td>12.4%</td>
<td>15.0%</td>
<td>3.0%</td>
</tr>
<tr>
<td>Larger rotating scene 2</td>
<td>15.2%</td>
<td>27.1%</td>
<td>13.2%</td>
<td>31.1%</td>
<td>9.4%</td>
</tr>
</tbody>
</table>

Table 7.2: False positive and false negative rates after the Delaunay triangulation step, OEHT step and LPC step.

### 7.3 LPC and SDSIFT

LPC from Sec. 6.2 is used to register regions of the reference image and sample image using geometric transformations defined by neighbouring planar areas. Previous steps identify triangular segments that contain change, they also result in false positive triangular segments those segments they include regions that cannot be well registered as they do consist solely of planes. In both of these cases each triangular segment may also contain large areas that lie on a plane which currently results in a false positive, for example the ground around an object that is removed. These areas increases the false positive rate. The false positives can be reduced by attempting to match mean-shift segments [Che95, CM02] using the projective transformation transformations defined by LPC found within or adjacent to the mean-shift segments for registration. SDSIFT is again used to compare the registered areas. The false positive and false negative rates are shown in Fig. 7.2. The results show a substantial decrease in false positive rates dropping from 3.9%, 12.4% and 13.2% to 0.4%, 3.0% and 9.4% for the rotating scene, larger rotating scene angle 1 and larger rotating scene angle 2 respectively. The false negative rates has increased by an average of 2% as well as show in Fig. 7.10, Fig. 7.11 and Fig. 7.12 no areas of change are missed completely and the false negatives are restricted to localisation of the boundaries of the areas containing change. Because of this the overall performance is substantially increased by the application of the LPC step of the change detection system.

This step has been shown to reduce false positives measured by the number of pixels classified by matches areas at the edges of defined planes. This produces better defined areas of change but is unlikely to match objects or regions that are not previously at least partially matched.

### 7.4 Standing wall matching

Standing wall matching as introduced in Sec. 6.4 is a method for registering vertical planes within a scene. As such is can be used to reduce false positives caused by areas consisting of vertical planes not matching. Again SDSIFT is used to match candidate matches. When this is applied as the next step after the LPC results show in Fig. 7.10, Fig. 7.11 and Fig. 7.12 there are no successful matches as there are no suitable false positives containing vertical planes.
Figure 7.10: Result from the LPC step on the rotating scene dataset at a difference in viewing angle of 10°. Left: Ground truth change areas highlighted in yellow. Areas reclassified from false positives to true negatives are highlighted in green.

Figure 7.11: Result from the LPC step on the first viewing angle of the larger rotating scene dataset at a difference in viewing angle of 10°. Left: Ground truth change areas highlighted in yellow. Areas reclassified from false positives to true negatives are highlighted in green.

Figure 7.12: Result from the LPC step on the second viewing angle of the larger rotating scene dataset at a difference in viewing angle of 10°. Left: Ground truth change areas highlighted in yellow. Areas reclassified from false positives to true negatives are highlighted in green.
7.5 CDnet video change detection data

Available image datasets used for benchmarking change detection in the literature consist mainly of images that are top down onto scenes, either using satellite imagery or top down aerial photographs. These datasets explicitly exclude issues related to varying viewing angles over the scene and so do not provide a good test for the techniques presented in this thesis. The CDnet video dataset [YWI14] however does contain a number of videos used for the testing of various challenges in video change detection. Although video change detection involves many challenges that are not applicable to still image change detection, some of the datasets can be used to test the performance of the techniques presented in this thesis. The CDnet dataset includes videos for testing for robustness to bad weather, low frame-rate, night videos, panning, tilting and zooming (PTZ), turbulence, dynamic backgrounds, camera jitter, intermittent object motion, shadow and thermal effects. None of the datasets involve the camera position changing between frames and so do not involve occlusions or dis-occlusions, however the PTZ video does require the change detection method to register the image between frames.

Change detection in video requires comparison between adjacent frames in the video. This requires substantially more work, however the changes between adjacent frames is small. The PTZ video doesn’t involve any changes in viewing angle and so there is no occlusion or dis-occlusion, this also means that only a similarity transformation as described in Sec. 2.1.2.1 is needed. In general this will lead to lower registration errors. As the CDnet video dataset requires a computationally faster but less robust method the end-to-end approach is simplified. Only the first step of the change detection process is applied, using standard SIFT instead of ASIFT feature points together with DSIFT comparison. Every 10th frame is compared to detect changes. The resulting ROC figure is shown in Fig. 7.13. The results marked as whole area were obtained by extending the Delaunay mask to the edges of each frame by treating the four corners of each frame as matched feature points. This results in large amounts of false positive areas around the edges of frames. The more points, whole area results use a denser Delaunay triangular mask by reducing the SIFT matching threshold. This will also on average extend closer to the frame boundary. This does reduce false positives but the results are still significantly worse than when only the area lying within the Delaunay grid is taken into account, as shown by the triangle area results. When only taking the area within the mask into account, using a denser grid does not affect performance, as shown by the more points, whole area results. The equal error rate for the triangle area is 13% which compares well with the best performing specialised video change detection techniques such as M4CD [WGLW15] which obtains a false positive rate of 9% and a false negative rate of 14%, IUTIS [BCS15] which obtains a false positive rate of 32% and a false negative rate of 12%, and SOBS CF [LM10] which obtains a false positive rate of 32% and a false negative rate of 14%.
Figure 7.13: ROC applying Delaunay change detection using standard SIFT applied to the CDnet PTZ video dataset.
Chapter 8

Conclusion

8.1 Summary of key contributions

This thesis has presented three key contributions, it has presented methods of feature point matching that perform better with larger differences in viewing angle, methods for registering complex scenes between two images with across wide viewing angles and image matching methods that are robust to large registration errors.

ASIFT was introduced in Sec. 4.1 which greatly improves the performance of feature points to differences in viewing angles. ASIFT extends SIFT by applying affine transformations to the reference image to form a hemisphere of reference images before the collection of SIFT descriptors. Using ASIFT it is possible to match a high density feature points between images with up to 30° differences in viewing angles. It was also shown that densities can be further increased by also using Harris points with ASIFT descriptors attached and also by using ground plane compensation to align the dominant ground plane between the two images in Sec. 4.2.

Wide baseline registration techniques that use correspondences provided by feature points and also edges provided by segmentation were introduced. Delaunay Triangulation registration, introduced in Sec. 6.1, segments the image into triangular segments that have corners defined by matched feature points. Each triangular segment can then be registered by using the three corner points to define an affine transformation between the reference and sample image. It is shown in Sec. 6.3 that triangular segments that are incorrectly registered by the Delaunay triangulation step because of bisecting by object edges can be registered by using segmentation to define the location of the object edge in object edge hypothesis testing. The location of the edge together with the fundamental matrix defined using matched feature points can be used to find corresponding points on the bisecting line. These points can then be used to register the two parts of the triangular segment separately and can also define any areas of occlusion.
Chapter 8. Conclusion

Errors can be reduced further by finding local planes using local plane clustering and registering them using homographies defined by feature points on that plane as shown in Sec. 6.2. It was also shown that vertical objects could be matched using standing object matching if the two view geometry and the ground plane could be defined.

Methods of matching image areas that are robust to registration errors are introduced. Dense SIFT, introduced in Sec. 5.2 applies a dense grid of SIFT descriptors to the reference and sample images and obtains matching distances between corresponding positions of the reference and sample image. The number of the resulting matching distances across the grid that are above a threshold value is used to determine if the areas match. The statistics of the matching distances of SIFT descriptors when areas contain and do not contain change are found including the mean and distribution of the matching distances. How the correlation between matching distances of neighbouring pairs of descriptors varies with the pixel distance is found as well as the variation in the correlation of descriptors with pixel distance. These are used to design the shifted dense SIFT and Markov random field change localisation in Sec.5.3 and Sec. 5.4.

Finally the feature point, registration and image comparison techniques are then combined to produce an end-to-end change detection system in Chapter 7 which is evaluated on a range of datasets.

8.2 Further work

The end-to-end change detection system presented works well on images containing objects that have clear geometric shapes such as buildings, as illustrated on the rotating scene with trees dataset the system struggles with complex shapes that cannot be registered well using a number of planes and have large depth variations over small areas. A texture based approach was attempted in Sec. 5.5 but did not perform well enough to be included in the end-to-end system but with further improvements could provide an avenue for improving the system performance in image areas such as these.

The techniques presented in this thesis illustrate approaches for addresses some of the challenges involved in wide-baseline image change detection. In order to create practical tools from these approaches the implementations need to be improved in terms of their computational complexity and the integration of the different steps. Currently the implementation takes of the order of hours to compare two images. The use of relatively intensive tools such as SIFT feature points and descriptors could be addressed. Less intensive feature points and descriptors are available [CLSF12, RR11, BTG06]. Also the different steps used in the end-to-end system are not deeply integrated, for example corner points are used as feature points in affine Harris in Chapter 4 and then a separate edge map is found in OEHT in Chapter 6. Both the edge map and corner point could be found simultaneously [HS88], saving computational complexity.
Areas that are registered as being planar can be compared using shifted dense SIFT as it provides a method for dealing with small changes in the registration offset while OEHT provides a method for compensating for large shifts in registration offset caused by corners and object edges. The two approaches could be combined into a single step that simultaneously tracks small changes in registration offset while also incorporating knowledge of the two view geometry and detected scene geometry, such as the location of edges. These and other changes could be investigated and their impact on the performance of the end-to-end system assessed.
# Chapter 9

## List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>First Used</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$x$</td>
<td>2.1.2</td>
<td>x-axis coordinate</td>
</tr>
<tr>
<td>$y$</td>
<td>2.1.2</td>
<td>y-axis coordinate</td>
</tr>
<tr>
<td>$z$</td>
<td>2.1.3</td>
<td>z-axis coordinate</td>
</tr>
<tr>
<td>$s$</td>
<td>2.1.2</td>
<td>Scaling factor</td>
</tr>
<tr>
<td>$x'$</td>
<td>2.1.2</td>
<td>Transformed position of $x$</td>
</tr>
<tr>
<td>$\theta$</td>
<td>2.1.2</td>
<td>Rotation angle</td>
</tr>
<tr>
<td>$\xi$</td>
<td>2.1.2</td>
<td>Indicates reflection if set to -1</td>
</tr>
<tr>
<td>$t_x,t_y$</td>
<td>2.1.2</td>
<td>Scalar transformation distance in $x$-axis and $y$-axis respectively</td>
</tr>
<tr>
<td>$A$</td>
<td>2.1.2</td>
<td>Affine transformation</td>
</tr>
<tr>
<td>$R$</td>
<td>2.1.2</td>
<td>Rotation matrix</td>
</tr>
<tr>
<td>$\phi$</td>
<td>2.1.2</td>
<td>Affine stretching/foreshortening axis angle</td>
</tr>
<tr>
<td>Symbol</td>
<td>First Used</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>$a_{xy}$</td>
<td>2.1.2</td>
<td>Scalers of upper left 2x2 area of the affine transformation</td>
</tr>
<tr>
<td>D</td>
<td>2.1.2</td>
<td>Diagonal matrix, often used to represent the scaling component of a transformation</td>
</tr>
<tr>
<td>$\lambda_1$</td>
<td>2.1.2</td>
<td>Diagonal components of affine component scaling matrix</td>
</tr>
<tr>
<td>$v_x$</td>
<td>2.1.2</td>
<td>Projective component of homography</td>
</tr>
<tr>
<td>$P_0$ &amp; $P_1$</td>
<td>2.1.2</td>
<td>Reference and sample image camera matrices</td>
</tr>
<tr>
<td>$K$, $K_0$ &amp; $K_1$</td>
<td>2.1.2</td>
<td>Internal component of camera matrix, reference camera matrix and sample camera matrix</td>
</tr>
<tr>
<td>c</td>
<td>2.1.2</td>
<td>Camera centre position</td>
</tr>
<tr>
<td>$f_d$</td>
<td>2.1.2</td>
<td>Focal length</td>
</tr>
<tr>
<td>$e$</td>
<td>2.1.2</td>
<td>The epipole, the imaged position of the other camera in two view geometry</td>
</tr>
<tr>
<td>l</td>
<td>2.1.2</td>
<td>The epipolar line</td>
</tr>
<tr>
<td>$F$</td>
<td>2.1.2</td>
<td>The fundamental matrix</td>
</tr>
<tr>
<td>$r_n$</td>
<td>2.1.2</td>
<td>Row vector of rotation matrix $R$</td>
</tr>
<tr>
<td>$G$</td>
<td>2.6</td>
<td>Gabor filter</td>
</tr>
<tr>
<td>$D_G$</td>
<td>2.6</td>
<td>Gabor filter discriminator</td>
</tr>
<tr>
<td>$A, B$</td>
<td>2.6</td>
<td>Reference and sample images respectively.</td>
</tr>
<tr>
<td>$L_G$</td>
<td>2.6</td>
<td>Laplacian of Gaussian</td>
</tr>
<tr>
<td>$P_A$</td>
<td>4.1</td>
<td>Affine camera</td>
</tr>
</tbody>
</table>
## Chapter 9. List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>First Used</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>$d_0$</td>
<td>4.1</td>
<td>Distance of plane from the camera</td>
</tr>
<tr>
<td>$d$</td>
<td>4.1</td>
<td>Distance of plane from the camera scaling factor</td>
</tr>
<tr>
<td>$u, v$</td>
<td>4.1</td>
<td>Position of point in the $x$ and $y$</td>
</tr>
<tr>
<td>$E_r$</td>
<td>5</td>
<td>Registration error</td>
</tr>
<tr>
<td>$D_{\Delta}$</td>
<td>5</td>
<td>Seperation distance between neighbouring descriptors in a dense grid of SIFT descriptors in pixels</td>
</tr>
<tr>
<td>$D_s$</td>
<td>5</td>
<td>SIFT descriptor scale</td>
</tr>
<tr>
<td>$H_0 &amp; H_1$</td>
<td>5</td>
<td>Not matching and matching hypothesis</td>
</tr>
<tr>
<td>$S$</td>
<td>5</td>
<td>Matching state that makes the value of 0 for non-matching and 1 for matching</td>
</tr>
<tr>
<td>$i$</td>
<td>5.1</td>
<td>Matching distance</td>
</tr>
<tr>
<td>$P_R$</td>
<td>5.1</td>
<td>Hypothesis probability ratio $P_R = \frac{P(H_1</td>
</tr>
<tr>
<td>$D_l$</td>
<td>5.1</td>
<td>Likelihood ratio probability threshold</td>
</tr>
<tr>
<td>$L$</td>
<td>5.1</td>
<td>Hypothesis log probability ratio $L = \ln \left( \frac{p(i</td>
</tr>
<tr>
<td>$i_{nm}$</td>
<td>5.1</td>
<td>Matching distance at position $n$ and $m$ in the $x$-axis and $y$-axis respectively of the image.</td>
</tr>
<tr>
<td>$R$</td>
<td>5.1</td>
<td>Correlation of a row of matching probabilities</td>
</tr>
<tr>
<td>$\mu_m$</td>
<td>5.1</td>
<td>Mean of matching distances for column $m$ of the image</td>
</tr>
<tr>
<td>$\sigma_m^2$</td>
<td>5.1</td>
<td>Variance of matching distances for column $m$ of the image</td>
</tr>
<tr>
<td>$\sigma_{(x,y)}$</td>
<td>5.1</td>
<td>Covariance between the variables $x$ and $y$</td>
</tr>
<tr>
<td>Symbol</td>
<td>First Used</td>
<td>Description</td>
</tr>
<tr>
<td>-----------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>$T_\sigma (D_s)$</td>
<td>5.1</td>
<td>Sum of auto-correlations across all displacements at SIFT descriptor scale $D_s$</td>
</tr>
<tr>
<td>$T_{\sigma,2D} (D_s)$</td>
<td>5.1</td>
<td>2D sum of auto-correlations across all displacements at SIFT descriptor scale $D_s$</td>
</tr>
<tr>
<td>$D_p$</td>
<td>5.1</td>
<td>Distance in pixels between two SIFT descriptors</td>
</tr>
<tr>
<td>$\mu_D (D_p)$</td>
<td>5.1</td>
<td>Mean matching distances between SIFT descriptors separated by $D_p$</td>
</tr>
<tr>
<td>$\sigma_D (D_p)$</td>
<td>5.1</td>
<td>Standard deviation of matching distances between SIFT descriptors separated by $D_p$</td>
</tr>
<tr>
<td>$i_w$</td>
<td>5.2</td>
<td>Member of a set of $W$ SIFT matching distances</td>
</tr>
<tr>
<td>$W$</td>
<td>5.2</td>
<td>Size of set of SIFT matching distances, $i_w$</td>
</tr>
<tr>
<td>$L_{i_w}$</td>
<td>5.2</td>
<td>Log likelihood ratio of matching state calculated from the matching distance $i_w$</td>
</tr>
<tr>
<td>$P_p$</td>
<td>5.2</td>
<td>Ratio of the matching state prior probability $P_p = \frac{P(S=1)}{P(S=0)}$</td>
</tr>
<tr>
<td>$T$</td>
<td>5.2</td>
<td>Matching threshold combined with priors,</td>
</tr>
<tr>
<td></td>
<td></td>
<td>$T = \frac{1}{W} \ln \left( \frac{D_x P(S=1)}{P(S=0)} \right)$</td>
</tr>
<tr>
<td>$\mu_W$</td>
<td>5.2</td>
<td>Mean log probability ratio gained from a set of $W$ matching distances</td>
</tr>
<tr>
<td>$\mu$</td>
<td>5.2</td>
<td>Expected log probability ratio</td>
</tr>
<tr>
<td>$V_{2D}$</td>
<td>5.2</td>
<td>Effective variance of the mean values of sets of $W$ log probability ratios</td>
</tr>
<tr>
<td>Symbol</td>
<td>First Used</td>
<td>Description</td>
</tr>
<tr>
<td>------------------------</td>
<td>------------</td>
<td>-----------------------------------------------------------------------------</td>
</tr>
<tr>
<td>$f_A(x, y)$ $f_B(x, y)$</td>
<td>5.3</td>
<td>SIFT descriptors obtained from position $(x, y)$ of images $A$ and $B$ respectively</td>
</tr>
<tr>
<td>$q &amp; r$</td>
<td>5.3</td>
<td>Offset from position $(x, y)$ in image $B$</td>
</tr>
<tr>
<td>$Q &amp; R$</td>
<td>5.3</td>
<td>Size of grid around position $(x, y)$ in image $B$</td>
</tr>
<tr>
<td>$d_{x,y}(q,r)$</td>
<td>5.3</td>
<td>Matching distance between the SIFT descriptors at position $(x, y)$ in image $A$ and position $(x + q, y + r)$ in image $B$</td>
</tr>
<tr>
<td>$H_{0,x,y}(q,r)$ &amp; $H_{1,x,y}(q,r)$</td>
<td>5.3</td>
<td>Matching and non-matching hypothesis between position $(x, y)$ in image $A$ and position $(x + q, y + r)$ in image $B$</td>
</tr>
<tr>
<td>$S_{x,y}(q,r)$</td>
<td>5.3</td>
<td>Matching state between position $(x, y)$ in image $A$ and position $(x + q, y + r)$ in image $B$</td>
</tr>
<tr>
<td>$u_e$</td>
<td>5.3</td>
<td>Mean of registration error variations at pixel shifts $D_{\Delta}$ in the images being compared</td>
</tr>
<tr>
<td>$\sigma_e$</td>
<td>5.3</td>
<td>Standard deviation of registration error variation at pixel shifts $D_{\Delta}$ in the images being compared</td>
</tr>
<tr>
<td>$n_x &amp; n_y$</td>
<td>5.3</td>
<td>Offset from the point $(x, y)$ in image $A$ and $(x + q, y + r)$ in image $B$ in the $x$-axis and $y$-axis</td>
</tr>
<tr>
<td>$d_c$</td>
<td>6.2</td>
<td>Central point of triangular segment</td>
</tr>
<tr>
<td>$d_r$</td>
<td>6.2</td>
<td>Euclidean distance between the matched position and the transformed location when transformed using an affine transformation defined by the neighbouring triangular segment</td>
</tr>
<tr>
<td>Symbol</td>
<td>First Used</td>
<td>Description</td>
</tr>
<tr>
<td>--------</td>
<td>------------</td>
<td>-------------</td>
</tr>
<tr>
<td>$d_s$</td>
<td>6.2</td>
<td>Clustering distance between two neighbouring triangular segments</td>
</tr>
<tr>
<td>$Q$</td>
<td>6.2</td>
<td>$Q = \text{diag} \left( \begin{array}{ccc} 1 &amp; 1 &amp; 1/f_d^2 \end{array} \right)$</td>
</tr>
<tr>
<td>$E$</td>
<td>6.4</td>
<td>Essential matrix</td>
</tr>
<tr>
<td>$\pi = \begin{bmatrix} w^T &amp; \omega \end{bmatrix}^T$</td>
<td>6.4</td>
<td>Ground plane, $\pi = \begin{bmatrix} w^T &amp; \omega \end{bmatrix}^T$</td>
</tr>
<tr>
<td>$R_0 &amp; R_1$</td>
<td>6.4</td>
<td>Rotational component of reference and sample image camera matrices respectively</td>
</tr>
<tr>
<td>$c_0 &amp; c_1$</td>
<td>6.4</td>
<td>Position component of the reference and sample camera matrix</td>
</tr>
<tr>
<td>$x_0 &amp; x_1$</td>
<td>6.4</td>
<td>Position of point $x$ in the reference and sample image respectively</td>
</tr>
<tr>
<td>$x_{1-3} &amp; \rho$</td>
<td>6.4</td>
<td>Components of point $x = \begin{bmatrix} x_{1-3} &amp; \rho \end{bmatrix}$</td>
</tr>
<tr>
<td>$x_g$</td>
<td>6.4</td>
<td>Intersection point between the ground plane, $\pi$ and the vertical plane, either $\pi_w$, $\pi_c$ or $\pi_b$</td>
</tr>
<tr>
<td>$\pi_b = \begin{bmatrix} u_b^T &amp; u_b \end{bmatrix}^T$</td>
<td>6.4</td>
<td>Vertical plane defined as orthogonal to the ground plane, $\pi$, and passing through the point, $x_a$ and parallel to the baseline. $\pi_b = \begin{bmatrix} u_b^T &amp; u_b \end{bmatrix}^T$</td>
</tr>
<tr>
<td>$G_b$</td>
<td>6.4</td>
<td>Homography of plane, $\pi_b$ between camera matrices $P_0$ and $P_1$</td>
</tr>
<tr>
<td>$x'_g$</td>
<td>6.4</td>
<td>Second intersection point between the ground plane, $\pi$ and the vertical plane $\pi_w$</td>
</tr>
</tbody>
</table>
### Chapter 9. List of Symbols

<table>
<thead>
<tr>
<th>Symbol</th>
<th>First Used</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>(\pi_w = \begin{bmatrix} u_w^T &amp; u_w \end{bmatrix}^T)</td>
<td>6.4</td>
<td>Vertical plane defined as orthogonal to the ground plane, (\pi), and passing through the points, (x_y) and (x'_y).</td>
</tr>
<tr>
<td>(\pi_c = \begin{bmatrix} u_c^T &amp; u_c \end{bmatrix}^T)</td>
<td>6.4</td>
<td>Vertical plane defined as orthogonal to the ground plane, (\pi), and orthogonal to the line between the point, (c_y) on the ground plane nearest to the halfway point between the two cameras and the point intersecting the ground plane and vertical plane, (x_y).</td>
</tr>
<tr>
<td>(c_m)</td>
<td>6.4</td>
<td>Point half way between the two cameras</td>
</tr>
<tr>
<td>(c_y)</td>
<td>6.4</td>
<td>Point on the ground plane closest to the point half way between the two cameras</td>
</tr>
</tbody>
</table>
Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


Bibliography


