Damage Detection and Monitoring for Tunnel Inspection based on Computer Vision

Krisada Chaiyasarn

Christ’s College

University of Cambridge

A thesis submitted for the degree of

Doctor of Philosophy

September 30, 2011
I hereby declare that no part of this thesis has already been or is being submitted for any other degree or qualification. This dissertation is the result of my original work, except where explicit reference has been made to the work of others. This dissertation contains approximately 59000 words and 118 figures.
Acknowledgement

First and foremost, I would like to thank Professor Kenichi Soga for his knowledge, support and generosity in giving me the opportunity to carry out this research, which has widened the horizons of my knowledge and abilities, I cannot thank him enough. I would like to thank Professor Roberto Cipolla for his generous support. I would like to thank Dr Tae-kyun Kim for his mentoring and for teaching me research skills; without him, I could not have achieved my first publication. To everyone in the Computer Vision Group, especially Tze-Ho and Fabio Viola, thank you for many interesting and insightful discussions, and research collaboration during my PhD career.

I would like to thank everyone from the Geotechnical Group for intellectual discussions at lunchtime. Special thanks go to Shun, Tina, Raksha, Echo and Ahmed and Carter, without whom my time at the office would have been very stressful.

Thanks to everyone in the Cambridge University Thai Societies (CUTS) and the Cambridge University Dancesport Team (CUDT) for making my spare time enjoyable and fruitful. Not only did I have a lot of fun, but I also have acquired many transferable skills. Special thanks go to Gate and Mew, and my coach, David Mellabone, who taught me the art of dancesport.

I would like to thank my sponsors, the Thai Government, EPSRC, Christ’s College, Lundgren and Toshiba. Without them, this thesis could not have been completed.

I would like to thank my friends for their support and my hugely enjoyable time in Cambridge, Kaoru, Diem-hang, Simon, Yang, Piing, and Chea and so many of you who made my time at Cambridge very special.

I would like to thank everyone at The Wrestlers for their authentic Thai food and generosity. Very special thanks go to Jack for his kindness and friendship.

Coming to the end of this acknowledgement, this penultimate paragraph is dedicated to my love and gratitude for my family: my parents, sisters, brother, and my little nephew. Every time I found myself needing extra support and encouragement, they were there for me, and I cannot thank them enough.

Finally, I wish to thank my boyfriend, Mark Jones, for his love and patience. Without his support, this thesis would not have been completed.
Abstract

The deterioration of the underground infrastructure of the major cities around the world, due to ageing, has become a topic of great concern among engineers. Visual inspection, as part of the routine maintenance procedures, is a common practice used in the condition assessment of infrastructure to ensure its safety and serviceability. This practice, however, is labour-intensive, costly and inaccurate and, therefore, a new system based on computer vision technology is presented in this thesis, aiming to tackle these inadequacies.

This thesis proposes a novel mosaicing system for inspection reporting, which can create an almost distortion-free mosaic of tunnels, thus allowing a large area of tunnels to be visualised. The system relies on Structure from Motion (SFM), which enables the system to cope with images with a general camera motion, in contrast to standard mosaicing software that can cope only with a strict camera motion. The system involves the automatic robust estimation of a 3D cylindrical surface using a Support Vector Machine to classify 3D points to improve the accuracy of the estimation. It is shown that some curvatures are observed in the mosaics when an inaccurate surface is used for mosaicing, while the mosaics from a surface estimated using the proposed method are almost distortion-free.

New feature matching algorithms aiming to improve the performance of SFM systems are proposed. These algorithms apply a spatial consistency constraint to match features with a similar topography, in contrast to other matching algorithms that rely on matching based on the similar appearance of local image patches. The Shape Context and Random Forest algorithms are combined in the proposed algorithm, revealing promising results.

The final contribution is a new change detection system for monitoring cracks in multi-temporal images. The system can cope with images with a general camera motion achieved by geometrical registration using SFM, unlike other systems that assume fixed or controlled cameras. The system performs photometric normalisation to cope with illumination variation in the images, and also a motion-invariant change detection algorithm is applied to handle deformable objects. It is shown that the results from the proposed change detection system are still impractical for use with tunnel images from a real environment, and further study is required.
Table of Contents

List of Figures v

List of Tables x

Chapter 1: Introduction 1

1.1 Background ........................................ 1
   1.1.1 London Underground .......................... 4
   1.1.2 Prague Metro .................................. 4
   1.1.3 Barcelona Metro ............................... 6

1.2 Motivation .......................................... 7
   1.2.1 Current approaches ........................... 7
   1.2.2 State of the art ................................ 7
   1.2.3 Potential of using Computer Vision for inspections ..... 9

1.3 Proposed approach .................................. 9

1.4 Themes and contributions .......................... 12
   1.4.1 Improved mosaicing for inspection reporting ............ 12
   1.4.2 Change detection system ........................ 13

1.5 Organisation ....................................... 13

Chapter 2: Literature Review 16

2.1 Management of infrastructure in civil engineering .......... 16
   2.1.1 Structural Health Monitoring .................... 19
   2.1.2 Roles of inspection for infrastructure management ....... 21

2.2 Inspection techniques ................................ 24

2.3 Vision-based inspection ................................ 25
   2.3.1 Visual inspection ................................ 28
   2.3.2 Automatic and interactive inspection systems .......... 30
   2.3.3 Applications of the vision-based inspection of underground infrastructure ...... 39
Chapter 3: Evaluation Datasets

3.1 Tunnel datasets ........................................ 45
  3.1.1 Acquisition procedure .................................. 45
  3.1.2 Site descriptions ........................................ 47
  3.1.3 Practical issues .......................................... 54
  3.1.4 Remarks .................................................. 56

3.2 Wide-baseline matching datasets ........................................ 56
  3.2.1 Tunnel datasets .......................................... 57
  3.2.2 Mikolajczyk datasets .................................... 58
  3.2.3 Beam images .............................................. 61

3.3 Change detection datasets ........................................ 61
  3.3.1 Experiment setup ......................................... 61
  3.3.2 Image datasets ............................................ 63
  3.3.3 Remarks .................................................. 63

3.4 Summary .................................................... 63

Chapter 4: System Background ........................................ 66

4.1 System outline .............................................. 66

4.2 Registration system ........................................... 67
  4.2.1 Image alignment .......................................... 70
  4.2.2 Image stitching ........................................... 74
  4.2.3 Experiments ............................................... 75
  4.2.4 Discussion ............................................... 76

4.3 Reconstruction system .......................................... 79
  4.3.1 Previous work ............................................ 79
  4.3.2 Structure from motion .................................... 80
  4.3.3 Experiments ............................................... 86
  4.3.4 Discussion ............................................... 90

4.4 Summary .................................................... 93

Chapter 5: Robust Surface Estimation for Mosaicing .................. 94

5.1 Introduction ................................................ 94
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.1</td>
<td>An example of an inspection report</td>
<td>5</td>
</tr>
<tr>
<td>1.2</td>
<td>The demonstration of change detection, the images of a tunnel lining obtained at Mustek Station, Prague Metro</td>
<td>6</td>
</tr>
<tr>
<td>1.3</td>
<td>System outline</td>
<td>10</td>
</tr>
<tr>
<td>1.4</td>
<td>An example of a 3D model reconstructed from images taken from Sagrada Familia Station, Barcelona Metro</td>
<td>11</td>
</tr>
<tr>
<td>1.5</td>
<td>A diagram showing the stages in Bundler</td>
<td>13</td>
</tr>
<tr>
<td>1.6</td>
<td>An example of a mosaic of Mustek Station, Prague Metro.</td>
<td>14</td>
</tr>
<tr>
<td>2.1</td>
<td>An example of deterioration curves with and without maintenance</td>
<td>18</td>
</tr>
<tr>
<td>2.2</td>
<td>A life cycle framework for infrastructure maintenance</td>
<td>18</td>
</tr>
<tr>
<td>2.3</td>
<td>Computer vision in an integrated Structural Health Monitoring framework</td>
<td>22</td>
</tr>
<tr>
<td>2.4</td>
<td>A bridge management system framework</td>
<td>23</td>
</tr>
<tr>
<td>2.5</td>
<td>The role of inspection in cost optimisation in infrastructure management</td>
<td>23</td>
</tr>
<tr>
<td>2.6</td>
<td>Example of images of defects found on tunnel linings</td>
<td>31</td>
</tr>
<tr>
<td>2.7</td>
<td>An example of a traditional inspection report</td>
<td>32</td>
</tr>
<tr>
<td>2.8</td>
<td>A panorama of a test site from Song et al. (2005).</td>
<td>37</td>
</tr>
<tr>
<td>2.9</td>
<td>An example of mosaic from Jahanshahi et al. (2011)</td>
<td>38</td>
</tr>
<tr>
<td>2.10</td>
<td>A general block diagram of image processing and ROI detection through change detection from Guo et al. (2009)</td>
<td>40</td>
</tr>
<tr>
<td>2.11</td>
<td>System configuration by Yu et al. (2007)</td>
<td>42</td>
</tr>
<tr>
<td>3.1</td>
<td>Schematic diagram showing how the image and video datasets are obtained.</td>
<td>48</td>
</tr>
<tr>
<td>3.2</td>
<td>Diagrams showing a London Underground map and Aldwych Station</td>
<td>50</td>
</tr>
<tr>
<td>3.3</td>
<td>A map showing where the Bond Street datasets are obtained (Wu et al., 2009)</td>
<td>52</td>
</tr>
<tr>
<td>3.4</td>
<td>Inside of a tunnel at Mustek Station, Prague Metro.</td>
<td>52</td>
</tr>
<tr>
<td>3.5</td>
<td>Prague Metro map, the stations and tunnels affected by the great flood in 2001 are highlighted in blue.</td>
<td>53</td>
</tr>
</tbody>
</table>
5.8 Examples of a training dataset for a Support Vector Machine ............................ 111
5.9 The plot showing ROC curves for the SVM classifier and the plot illustrating reprojection error ......................................................... 111
5.10 Diagrams showing the process of robust surface estimation ............................. 113
5.11 An example of input and warped images from the Bond Street dataset .......... 114
5.12 An example of the results from the Bond Street dataset .................................. 117
5.13 The mosaic obtained from the Bond street dataset ....................................... 117
5.14 The final mosaic obtained from the Prague dataset ....................................... 118

6.1 An example of the failure of a 3D model from Bundler due to incorrect tracks 125
6.2 An example from the Florence Duomo ........................................................... 125
6.3 An example of the cascading error .................................................................. 126
6.4 A typical pipeline of a feature matching algorithm ......................................... 127
6.5 An example of matching before and after the proposed filtering algorithm is applied ............................................................ 133
6.6 Angle and distance thresholds of the proposed filtering algorithm for the wide-baseline image datasets ................................................. 136
6.7 The effects of using the angle and distance filters ........................................... 137
6.8 A new proposed approach to feature matching, using the Random Forest and Shape Context algorithms .............................................. 138
6.9 An example of an image pair with the keypoints coloured by the leaf node indices ................................................................................. 141
6.10 An illustration of how a single randomised tree clusters a set of features ........ 142
6.11 Mapping of a feature by the Random Forest algorithm ................................ 143
6.12 Shape context, taken from Belongie et al. (2002) ........................................ 146
6.13 A modified shape context descriptor ............................................................. 146
6.14 Non-rigidly deforming bodies imaged in the beam dataset ............................. 149
6.15 Examples of the effect of increasing the number of trees in the Random Forest algorithm .......................................................... 151
6.16 An effect of different tuning parameters obtained with the Graff dataset .... 152
6.17 The results from the Graff dataset ................................................................. 153
6.18 The results from the Light dataset ................................................................. 153
6.19 The results from the Bike dataset ................................................................. 154
6.20 The results from the Boat dataset ................................................................. 154
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6.21</td>
<td>The results obtained from the Beam dataset</td>
<td>156</td>
</tr>
<tr>
<td>6.22</td>
<td>An example of matching from the beam dataset</td>
<td>158</td>
</tr>
<tr>
<td>7.1</td>
<td>An example of images taken from Mustek Station, Prague Metro</td>
<td>160</td>
</tr>
<tr>
<td>7.2</td>
<td>Images showing the results from the proposed change detection system</td>
<td>161</td>
</tr>
<tr>
<td>7.3</td>
<td>An outline of the proposed change detection system</td>
<td>165</td>
</tr>
<tr>
<td>7.4</td>
<td>The result of change detection from coral reef images from Delaunoy et al. (2008)</td>
<td>165</td>
</tr>
<tr>
<td>7.5</td>
<td>An example result from the Patch-based Multi-View System (PMVS)</td>
<td>169</td>
</tr>
<tr>
<td>7.6</td>
<td>An example of the registration of a query image</td>
<td>170</td>
</tr>
<tr>
<td>7.7</td>
<td>An example of a synthetic image based on the model-based method</td>
<td>171</td>
</tr>
<tr>
<td>7.8</td>
<td>An example of a synthetic image based on the model-based method from the Prague dataset</td>
<td>172</td>
</tr>
<tr>
<td>7.9</td>
<td>An example of images processed by intensity normalisation</td>
<td>172</td>
</tr>
<tr>
<td>7.10</td>
<td>An example of images processed by homomorphic filtering</td>
<td>174</td>
</tr>
<tr>
<td>7.11</td>
<td>Images exemplifying how ground truth change masks are created</td>
<td>179</td>
</tr>
<tr>
<td>7.12</td>
<td>An example of an image masked by the ROI</td>
<td>179</td>
</tr>
<tr>
<td>7.13</td>
<td>An example of a computed change mask vs. a ground truth change mask</td>
<td>181</td>
</tr>
<tr>
<td>7.14</td>
<td>An example of ROC curves</td>
<td>182</td>
</tr>
<tr>
<td>7.15</td>
<td>A comparison of the beam vertical displacements between Chen’s algorithm and GeoPIV.</td>
<td>184</td>
</tr>
<tr>
<td>7.16</td>
<td>The MD distances of all image patches</td>
<td>186</td>
</tr>
<tr>
<td>7.17</td>
<td>Input images and the resulting change mask</td>
<td>187</td>
</tr>
<tr>
<td>7.18</td>
<td>Illustration of the effects of varying patch sizes and thresholds in the change detection algorithm</td>
<td>189</td>
</tr>
<tr>
<td>7.19</td>
<td>A contour plot of Accuracy Rate (AR) as a function of the threshold $\lambda$ and the grid size $R$</td>
<td>190</td>
</tr>
<tr>
<td>7.20</td>
<td>An image-based structural damage monitoring framework from Chen and Hutchinson (2010).</td>
<td>190</td>
</tr>
<tr>
<td>7.21</td>
<td>Input images before and after photometric and geometric pre-processing</td>
<td>194</td>
</tr>
<tr>
<td>7.22</td>
<td>The ROC curves of change masks using two different methods of change detection</td>
<td>195</td>
</tr>
<tr>
<td>7.23</td>
<td>The AR plot from the input images with pre-processing, Chen’s algorithm is used in the change detection module.</td>
<td>196</td>
</tr>
</tbody>
</table>
7.24 An example of the results from the left camera .......... 199
7.25 An example of the results from the right camera .......... 200
7.26 Images before and after registration from the Prague dataset .... 203
7.27 The result of change masks from the Prague dataset with various thresholds τ, ranging from τ=0.1-0.6. .......... 204
8.1 Hardware prototypes to allow faster image acquisition .......... 210
List of Tables

2.1 A summary of regular and detailed inspection, taken from Yamaguchi et al. (2006) ................................................. 25
2.2 A summary of the non-destructive evaluation (NDE) techniques ........... 26
2.3 A table showing bridge inspection data in various countries .................. 33

3.1 A summary of the datasets presented in this thesis .......................... 49
3.2 A summary of the wide-baseline datasets from the tunnel images ........ 58
3.3 A summary of the imaging conditions and scene types from Mikolajczyk and Schmid (2005) ........................................ 61

4.1 Statistics obtained from Bundler for the Prague dataset ....................... 87

6.1 An overview of the feature detectors with their associated properties, taken from Tuytelaars and Mikolajczyk (2008), that can be used as a guideline when choosing a suitable detector for a required application ..................... 129
6.2 A summary of the images used in the evaluation of the beam dataset. All of the figures are shown in Figure 6.14 ................................. 149

7.1 A summary of the classifications of a change mask ............................. 181
7.2 A summary of the legends .................................................. 193
List of Algorithms

4.1 The pipeline of a typical registration system based on the feature-based alignment approach ......................................................... 68
4.2 A typical pipeline of a Structure from Motion (SFM) system ................. 82
5.1 A manual estimation of a cylindrical surface .................................. 102
6.1 Algorithm explaining the combined algorithm .................................. 142
7.1 Pre-processing steps ................................................................. 167
List of abbreviations

The following table describes the significance of the various abbreviations and acronyms used throughout the thesis. The page on which each one is defined or first used is stated.

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Meaning</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>BRIEF</td>
<td>Binary Robust Independent Elementary Features</td>
<td>131</td>
</tr>
<tr>
<td>CAD</td>
<td>Computer Aided Design</td>
<td>211</td>
</tr>
<tr>
<td>CCD</td>
<td>Charge Couple Device</td>
<td>41</td>
</tr>
<tr>
<td>DLT</td>
<td>Direct Linear Transform</td>
<td>76</td>
</tr>
<tr>
<td>DOG</td>
<td>Difference of Gaussian</td>
<td>130</td>
</tr>
<tr>
<td>ED</td>
<td>Euclidean Distance</td>
<td>178</td>
</tr>
<tr>
<td>FPR</td>
<td>False Positive Rate</td>
<td>182</td>
</tr>
<tr>
<td>LIDAR</td>
<td>Light Detection and Ranging</td>
<td>3</td>
</tr>
<tr>
<td>MEMS</td>
<td>Micro-Electro-Mechanical Systems</td>
<td>20</td>
</tr>
<tr>
<td>MD</td>
<td>Manifold Distance</td>
<td>178</td>
</tr>
<tr>
<td>MLE</td>
<td>Maximum Likelihood Estimation</td>
<td>76</td>
</tr>
<tr>
<td>MSER</td>
<td>Maximally Stable Extremal Regions</td>
<td>130</td>
</tr>
<tr>
<td>NCC</td>
<td>Normalized Cross-Correlation</td>
<td>75</td>
</tr>
<tr>
<td>NDE</td>
<td>Non-Destructive Evaluation</td>
<td>24</td>
</tr>
<tr>
<td>NN</td>
<td>Nearest Neighbour</td>
<td>140</td>
</tr>
<tr>
<td>PCA</td>
<td>Principal Component Analysis</td>
<td>37</td>
</tr>
<tr>
<td>GeoPIV</td>
<td>Geo Particle Image Velocimetry</td>
<td>186</td>
</tr>
<tr>
<td>PMVS</td>
<td>Patch-Based Multiview Stereo</td>
<td>90</td>
</tr>
<tr>
<td>RANSAC</td>
<td>RANdom Sample Consensus</td>
<td>76</td>
</tr>
<tr>
<td>RBF</td>
<td>Radial Basis Function</td>
<td>113</td>
</tr>
<tr>
<td>RF</td>
<td>Random Forest</td>
<td>141</td>
</tr>
<tr>
<td>ROC</td>
<td>Receiver Operating Characteristic</td>
<td>113</td>
</tr>
<tr>
<td>SC</td>
<td>Shape Context</td>
<td>146</td>
</tr>
<tr>
<td>SHM</td>
<td>Structural Health Monitoring</td>
<td>19</td>
</tr>
<tr>
<td>SIFT</td>
<td>Scale Invariant Feature Transform</td>
<td>130</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Meaning</td>
<td>Page</td>
</tr>
<tr>
<td>--------------</td>
<td>----------------------------------------------</td>
<td>------</td>
</tr>
<tr>
<td>SFM</td>
<td>Structure from Motion</td>
<td>82</td>
</tr>
<tr>
<td>SURF</td>
<td>Speeded Up Robust Feature</td>
<td>131</td>
</tr>
<tr>
<td>SVM</td>
<td>Support Vector Machine</td>
<td>106</td>
</tr>
<tr>
<td>SVD</td>
<td>Singular Value Decomposition</td>
<td>86</td>
</tr>
<tr>
<td>TPR</td>
<td>True Positive Rate</td>
<td>182</td>
</tr>
<tr>
<td>WSN</td>
<td>Wireless Sensor Network</td>
<td>20</td>
</tr>
</tbody>
</table>
Notation

General

\( a \) \hspace{1cm} \text{Scalar}
\( \mathbf{a} \) \hspace{1cm} \text{Column vector}
\( |\mathbf{a}| \) \hspace{1cm} \text{Magnitude of} \ \mathbf{a}
\( \mathbf{a}^T \) \hspace{1cm} \text{Transpose of} \ \mathbf{a}
\( \mathbf{A} \) \hspace{1cm} \text{Matrix}
\( \mathbf{A}^{-1} \) \hspace{1cm} \text{Inverse of} \ \mathbf{A}

Geometry

\( \mathbf{x} \) \hspace{1cm} \text{Euclidean 2D point} \ \mathbf{x} = \begin{bmatrix} x & y \end{bmatrix}^T
\( \mathbf{X} \) \hspace{1cm} \text{Euclidean 3D point} \ \mathbf{X} = \begin{bmatrix} X & Y & Z \end{bmatrix}^T
\( \mathbf{\tilde{x}} \) \hspace{1cm} \text{Homogenous 2D point} \ \mathbf{\tilde{x}} \sim \begin{bmatrix} \tilde{x} & \tilde{y} & \tilde{w} \end{bmatrix}^T
\( \mathbf{\tilde{X}} \) \hspace{1cm} \text{Homogenous 3D point} \ \mathbf{\tilde{X}} \sim \begin{bmatrix} \tilde{X} & \tilde{Y} & \tilde{Z} & \tilde{W} \end{bmatrix}^T
\( d(\mathbf{x}, \mathbf{y}) \) \hspace{1cm} \text{Distance between the points} \ \mathbf{x} \ \text{and} \ \mathbf{y}
\( l(\mathbf{x}, \mathbf{y}) \) \hspace{1cm} \text{Line between the points} \ \mathbf{x} \ \text{and} \ \mathbf{y}
Computer vision

\( H \) Homography
\( P \) Projection matrix
\( K \) Camera Calibration matrix
\( R \) Camera orientation
\( T \) Camera translation
\( E \) Essential matrix
\( F \) Fundamental matrix
\( f \) focal length
\( u \) Euclidean pixel coordinate \( \mathbf{u} = \begin{bmatrix} u & v \end{bmatrix}^T \)
\( \tilde{u} \) Homogeneous pixel coordinate \( \tilde{\mathbf{u}} = \begin{bmatrix} \tilde{u} & \tilde{v} & \tilde{w} \end{bmatrix}^T \)
\( \hat{\mathbf{u}} \) Normalised homogenous pixel coordinates \( \hat{\mathbf{u}} = \mathbf{K}^{-1} \tilde{\mathbf{u}} \)

\( [\mathbf{T}]_x \) Cross product \( [\mathbf{T}]_x = \begin{bmatrix} 0 & -T_3 & T_2 \\ T_3 & 0 & -T_1 \\ -T_2 & T_1 & 0 \end{bmatrix} \)
Chapter I
Introduction

A large proportion of the underground infrastructure in the major cities around the world has shown evidence of deterioration due to ageing. The maintenance of this infrastructure remains a significant challenge for engineers in order to meet the public demand for safety, the uninterrupted operation of train services and budgetary constraints. Current maintenance procedures still rely largely on visual inspection, which has many limitations. The aim of this research is to develop a system based on state-of-the-art computer vision technology to improve the accuracy and efficiency of the inspection of underground infrastructure. The research presented in this thesis offers an improvement in two areas of inspection: (1) the visualisation of a large image database, and (2) the automatic detection of changes between images. This introductory chapter provides a background to the problems related to the current maintenance procedures and the motivations for this research. A brief overview and the contributions of the proposed system are presented. The structure of the rest of the thesis is outlined at the end of the chapter.

1.1 Background

Most underground infrastructure was constructed more than half a century ago and is now deteriorating (Stajano et al., 2010). Tunnels, in particular, are vulnerable to adjacent ground disturbance, caused by nearby activities, such as piling and deep excavation. Excessive stresses in tunnel linings, caused by the deformation of tunnels due to ground disturbance, may result in tunnel collapse. Therefore, effective maintenance strategies are urgently required in order to manage underground infrastructure properly to assure public safety while at the same time meeting usage demands. The most commonly used strategy is regular inspections, in which checks and assessments of the infrastructure conditions are performed periodically. This ensures that any defects are detected early and properly monitored in order to prevent failure, which may lead to costly repairs and
even catastrophes.

Infrastructure management is aimed at devising strategies to manage and maintain the safety and functionality of the infrastructure effectively, under challenging circumstances. Disruption due to closures for maintenance usually results in high economic costs and should be avoided. Infrastructure managers are, therefore, seeking tools to help them to manage the infrastructure with minimal disruption, ultimately increase efficiency and reduce inspection costs. The current techniques used to assess the condition of infrastructure can be categorised into network-wide visual inspection, destructive and non-destructive tests. The technologies involved range from those aiming to detect the characteristics of the flaws in the structural components, to localised surveying and the deployment of instruments for monitoring performance in critical sections. These technologies have many limitations, as will be discussed in more detail in Chapter 2. These limitations can generally be typified as follows (Soga, 2007):

- The high cost of monitoring equipment restricts its usage and, hence, the scope of any monitoring programmes can be hindered.

- Measurement techniques that interfere with vehicle movements in tunnels are seriously discouraged. Techniques that allow a quick inspection check are required as there is also a need to conduct checks when maintenance crews are working and moving along the tunnels.

- For specific classes of problem, e.g. monitoring joint rotations, the existing techniques may be unsuitable and new techniques are required.

To obtain a complete picture of the state of a structural system, a combination of monitoring sensors and regular inspection is required. For monitoring sensors, a number of state-of-the-art sensors are used to monitor displacement, and infer how structures deform. For example, fibre optics are able to monitor the strain profiles of a structural system over a long distance, inclinometers are used to see how a structure’s wall move and incline from the vertical plane, and crackmeters are used to monitor crack width openings. Deploying these sensors can be costly, and infrastructure owners generally have to prioritise the sites to be monitored due to cost considerations. Inspection can often help to provide an overview of the state of structural systems so that the owners can then make informed decisions about determining the sites at which the sensors should be deployed. Inspection and monitoring are, generally, closely linked.
The technologies employed in inspection as well as other surfacing technologies have their limitations, as will be reviewed in more detail in Chapter 2. One of the most widely used surfacing technologies is the LiDAR system, which is used to obtain 3D point clouds of structural systems. The density of the point clouds from the LiDAR system is high enough to be used to represent objects’ surfaces, which makes LiDAR technology very attractive for 3D modelling. However, as shown in Leberl et al. (2010), an image-based reconstruction can provide many advantages over a LiDAR system in terms of accuracy (e.g. greater point density, the ability to fuse data, thus providing seamless coverage), economy (e.g. single workflow, faster data collection, cheaper technology), interpretation (e.g. road sign classification, full automation and scene interpretation). The main attraction of an image-based system is not only the cost, but also the capability to process the image data further to recognise scenes similarly to how humans interpret their surroundings using vision. The ability to understand scenes has a large number of applications, which is a feature that is missing from the LiDAR system. Previously, a LiDAR system costs over £100k, although this has decreased over the last few years to around £30k. However, this system is still relatively expensive compared with a system of similar capability using just a standard digital camera costing less than £1000, as shown in this thesis.

This research aims to improve the current inspection techniques, especially visual inspection. Visual inspection is generally carried out by inspectors, who visually assess the condition of structural components based on their experience, so the problems arising from the use of this technique include inaccuracy, subjectivity and labour-intensiveness. This technique is expected to become difficult to implement in the future due to public demand for the continuous operation of metro systems as well as the limited human resources available. Hence, there is an urgent need to conduct inspections more rapidly and automatically, which will ultimately lead to savings in terms of inspection costs. This research proposes a damage detection system based on computer vision technologies, which has a number of advantages over other existing surface imaging technologies (e.g. line sensor cameras, LiDAR and infrared cameras). A major advantage is that the proposed system is cheaper than the other imaging technologies available, since has been developed for use with a standard digital camera.

The work presented in this thesis forms part of a collaborative project between the public and private agencies who are responsible for the maintenance and operation of the underground infrastructure. The proposed system is specifically designed to solve the
following problems that have been of great concern to the collaborators.

1.1.1 London Underground

Metronet alliance\(^1\) and Tubelines\(^2\) were commissioned by London Underground to manage and maintain the underground infrastructure. One project involved the use of a digital photographic technique to aid visual inspection. The technique is particularly useful for the inspection of shafts (e.g. ventilation shafts), which are usually not easily accessible. The technique starts by collecting a set of images for each ring of tunnel lining and then manually arranges these to form a mosaic-like diagram, as shown in Figure 1.1. The figure shows an example of an inspection report, in which one cell corresponds to one panel on the tunnel lining. Then, the manual sketching or marking of detected anomalies is done to produce a final inspection report. The report provides information indicating the type, location and size of any defects found on the tunnel surfaces.

The aim of inspection reporting is to provide a record of any anomalies found on a section of tunnel during inspection sessions so that subsequent analysis can be performed in out-of-inspection hours. Unfortunately, a current inspection report takes a long time to create due to such manual procedures. Commercial stitching software was used in an attempt to stitch images together automatically. The software can only create panoramas for each individual ring of tunnel lining, but it is impossible to stitch these panoramas together to form a bigger mosaic. This problem has led to one of the research focuses presented in this thesis.

1.1.2 Prague Metro

The trial site at Mustek Station, Prague Metro, has been studied and monitored by a collaborator from Czech Technical University. The site was affected by the great flood of 2001, and images of the tunnel linings have been recorded to monitor changes that are anomalies, such as changes in the colour of the water patches. As depicted in Figures

\(^1\) During 2003-2008, Metronet was responsible for the maintenance, renewal, and upgrading of the infrastructure on nine London Underground lines. However, in May 2008, the company responsibilities were transferred back to public ownership under the authority of Transport for London (TfL) (Transport for London, 2011).

\(^2\) Tubelines is responsible for the maintenance, renewal and upgrading of the infrastructure, including the track, trains, signals, civil work and stations, of three London Underground lines. The company has been a wholly owned subsidiary of TfL (Transport for London) since May 2010, although it remains a separate company (Tubelines Limited, 2011).
Figure 1.1: A diagram illustrating the production of an inspection report from tunnel images. A cell in the report corresponds to a single panel of the tunnel lining, and defects are manually labelled in the report using individual images. The report is provided by Metronet (2007).
Figure 1.2: Examples of images of a tunnel surface obtained from Mustek Station, Prague Metro. The images were taken at approximately the same location but at different times, (a) taken in 2003 and (b) taken in 2007. The red lines in both images illustrate the anomalies that need to be compared. Conducting a visual comparison of the images to detect changes in the anomalies is difficult because the images contain viewpoint and lighting variations.

1.2(a) and (b), the pictures were taken at approximately the same location but at different times, and so performing a visual comparison of these images is challenging. The difficulty arises because the images have different viewing angles and lighting. To compare these images, accurate geometrical and photometrical registration must be achieved, which leads to another focus in this research.

1.1.3 Barcelona Metro

The tunnels at Sagrada Familia Station, Barcelona Metro, are undergoing an extensive monitoring programme due to nearby tunnels being constructed. The tunnels at this site are shaped like a semi-circular arch. The main facility requested by the collaborator for this site is to be able to produce an inspection report similar to those created for the London Underground sites (Section 1.1.1).
1.2 **Motivation**

This research is motivated by two main reasons: (1) an urgent requirement to improve the current inspection techniques, and (2) the recent advancements in computer vision technology that offers possibilities for the improvement of inspection techniques. This section provides a summary of the current inspection techniques employed in civil engineering and also of the state-of-the-art computer vision technology and its potential to improve the current inspection techniques.

1.2.1 **Current approaches**

For visual inspection, a guideline for structural assessment suggests that the inspection must be carried out within the touching distance of an inspector (Federal Highway Administration and Federal Transit Administration, 2005). This is hard to achieve in many environments; for example, in ventilation shafts which are not easily accessible for close-up inspection. Other non-destructive tests, which are used for detailed inspections (e.g. impact acoustics, infrared thermography, test hammers, etc.) to check for degradation occurring inside the concrete (Yamaguchi et al., 2006), possess a number of limitations. For example, expert knowledge is required for the interpretation of the data obtained from these methods.

Photographic technologies, such as images and videos, are commonly employed to aid visual inspection. These technologies are able to provide rich information, such as texture, colour, and 3D cues, which are useful when assessing the conditions of structures. Some inspection systems employ photographic technologies to enable fast data acquisition, such as video systems used in underground pipe inspection. One example is the Tunnel Inspection Vehicle created by FATA automation (FATA, 2011), in which infrared cameras are mounted onto a vehicle to allow the acquisition of data via remote control. The main drawback of the system is that the data still need to be manually analysed by human inspectors.

1.2.2 **State of the art**

There have been attempts to develop automatic inspection systems based on image or video data to aid the inspection procedures. These attempts can be grouped into three main areas of development: detection, visualisation, and interpretation.
Detection  Several techniques for crack detection through visual inspection have been proposed (Yamaguchi and Hashimoto, 2008). Crack detection systems generally involve a pre-processing step and a crack identification step. The pre-processing step applies image processing techniques to extract potential crack features, such as edges, while the subsequent crack identification step involves crack modelling and pattern recognition techniques. There are a number of systems that develop algorithms for crack detection, although practical systems used for long-term monitoring are not commonly reported (Chen et al., 2006).

Visualisation  This area of development is concerned with improving the visualisation and organisation of a large number of images. As the number of images increases, it is essential to be able to organise them so that the image data can be visualised for a thorough examination. One solution is to use image mosaicing, which stitches images together to form a larger image. The resulting image or mosaic provides a larger field of view, which cannot be achieved with a single image. In the civil engineering literature, most mosaicing techniques are borrowed from the remote sensing community, which deals with the detection of landscape changes in satellite images. Their techniques usually require manually-identified control points to perform mosaicing and registration tasks. However, in the computer vision literature, the use of keypoint descriptors, such as Scale Invariant Feature Transform (SIFT) (Lowe, 2004), allows automatic image mosaicing without the need for manual control points. There exists a small number of software packages that can perform automatic mosaicing for home and industrial use (Brown and Lowe, 2003). Some programs have been applied to assist inspection (Jahanshahi et al., 2009, 2011), although they have some limitations, as shown later in Chapter 2.

Interpretation  This area of development involves the analysis of detected damage so that suitable actions for repair and remedy can be taken. The techniques are usually based on pattern recognition that relates the characteristics of the defects (e.g. type and size) to different levels of damage category. Another area of development is crack change monitoring, which aims to assess how cracks evolve continually over time. Information related to changes in defects provides important implications; for example, the deterioration rate of structures (Chen and Hutchinson, 2010). A number of change monitoring systems are implemented with cameras that remain in fixed positions (Lim et al., 2005). Although, the cost of digital cameras has been continually falling, deploying multiple fixed cameras
to monitor a large number of cracks in tunnels is impractical and not cost-effective. In the area of infrastructure inspection, there have been few systems that can deal with crack change monitoring when cameras are not fixed and controlled.

1.2.3 Potential of using Computer Vision for inspections

Advances in computer vision offer opportunities to improve the current inspection procedures. In the area of reconstruction, systems, such as Phototourism (Snavely et al., 2006), are able to recover the sparse 3D point clouds of scenes and camera poses from a large database of uncalibrated cameras with a high success rate. A denser 3D model, which can provide more realistic visualisation, can also be reconstructed. An example of a dense 3D system is the work by Hernandez et al. (2007), which is used for the reconstruction of museum collections. Recently, there have been systems that are capable of creating a 3D model of an entire city, although they still require considerable improvement before they will prove of any practical use. The accuracy of the 3D models produced by certain reconstruction systems is comparable to the results from LiDAR systems (Seitz et al., 2006). Hence, reconstruction systems offer much sought-after opportunities to create 3D models of structures in civil engineering, and raise the possibility of inspections being carried out in a virtual environment.

In the area of registration, algorithms to create panoramas are now commonly embedded in many modern digital cameras. There are commercial systems that can create panoramas automatically using smart keypoints, such as SIFT (Lowe, 2004), instead of traditional systems which rely on the user input of control points. Image mosaicing has many applications, such as in medical imaging, as shown in Can et al. (2002b,a), in which the stitched images of a curved human retina are used for medical examinations. In remote sensing, image registration is a common technique that is used to transform different satellite images into the same coordinate frame. Registration is used as a pre-processing step before change detection algorithms are applied to detect the changes in landscape in satellite images. Mosaicing offers possibilities to enhance the visualisation of a large image collection taken for inspection by providing an enlarged field of view.

1.3 Proposed approach

The aim of the research is to use state-of-the-art computer vision technologies to develop an automatic inspection system for underground infrastructure. The proposed system
improves the accuracy of the current inspection techniques as well as reducing the amount of time and money required for inspections.

Figure 1.3 shows an outline of the proposed system. The system starts with Image Acquisition, whereby images are collected from tunnel sites using the method described in Chapter 3. To mosaic images in the proposed system, tunnel images are first input into the Structure From Motion (SFM) module to recover the 3D point cloud of a tunnel model (an example of a 3D model is shown in Figure 1.4) and camera poses. SFM is the process of finding the 3D structure of an object from multiple uncalibrated images taken at various locations. The SFM program used in this thesis is called Bundler, implemented by Snavely (2008). Then, a surface proxy is fitted to the point cloud in the Surface Estimation module by non-linear least squares optimisation. Support Vector Machine (SVM) is applied to classify 3D points into those that either lie or do not lie on an actual tunnel surface, because non-surface points can corrupt the optimiser and result in convergence to an incorrect geometry. The SVM code from the Statistical Pattern Recognition Toolbox (SPRT) implemented by Vojtěch and Václav (2004) is used. The estimated surface is then used to warp input images in the Image Warping module, and finally the warped images are composited at the Stitching stage by a commercial program called ICE (Microsoft, 2011a).

Then, as shown in Figure 1.3, new images can be registered and warped in order to detect changes between new and existing images. This part of the system is called Change Detection, which involves geometric registration by Bundler and photometric registration, followed by applying a change detection algorithm (explained in Chapter 7).
Figure 1.4: An example of a 3D model reconstructed from images taken from Sagrada Familia Station, Barcelona Metro.
1.4 Themes and contributions

1.4.1 Improved mosaicing for inspection reporting

The research proposes a new method for creating an inspection report for tunnel images. A photographic inspection report as shown in Figure 1.1 is partially automated in this research by providing tools for creating a better quality mosaic of tunnel images. With the proposed system, a tunnel can be visualised and examined from a single, almost distortion-free mosaic image. The proposed mosaicing system makes specific contributions as follows.

Almost distortion-free mosaicing by robust surface estimation  The proposed system exploits the simplicity of tunnel geometries, which are usually developable surfaces. The developable surfaces have special properties such that they can be flattened onto a plane without distortion, such as stretching and compression. These properties allow an improved mosaic image to be created, as shown in Figure 1.6. The figure demonstrates that the mosaic image contains almost no distortion; it preserves line straightness, 90° angles between vertical and horizontal lines, and parallelism. Such desirable characteristics cannot be achieved through using commercial stitching software.

It is demonstrated from the experiments in this research that, to achieve a good quality mosaic, the geometry of a tunnel must be robustly and accurately estimated. Inaccurate estimation of the geometry leads to skewness of the mosaic. It is shown that the use of Support Vector Machine classification enables accurate surface estimation to be carried out automatically. SVM classifiers are used to segregate 3D points that do not belong to the true geometry of an estimated surface. This part of the system is explained in Chapter 5.

Spatially-consistent feature matching algorithm  The system proposed in this thesis extensively uses Bundler to build a 3D model and geometrical registration. The pipeline of the Bundler program is shown in Figure 1.5, which consists of four stages. The details of each stage are explained in Chapter 4. The performance in the SIFT matching stage can be improved by a new feature matching algorithm. The algorithm is based on applying a spatial constraint, which constrains feature matching based on the similarity of the configuration of neighbouring features. The spatial constraint is additional to a widely-used constraint, which matches features based on their similarity in the
appearance of local image patches only. The spatial constraint is based on the modified
Shape Context algorithm (Belongie et al., 2006), which is uniquely combined with the
Random Forest algorithm (Breiman, 2001). This combination of algorithms is explained
in detail in Chapter 6.

1.4.2 Change detection system

In previous studies related to crack change monitoring, there are two types of system.
The first type assumes that the position of the camera remains fixed, and the second
type assumes that there are tightly controlled camera positions (Lim et al., 2005). This
research proposes methods for improving both systems, to cope with cameras that are
neither fixed nor controlled. The methods perform geometrical registration based on
Bundler and photometrical registration before applying a change detection algorithm.
With the registration algorithms, images with different viewpoints are rectified to allow
direct comparison. The change detection system proposed in this thesis is explained in
Chapter 7.

1.5 Organisation

A summary of each chapter in the rest of the thesis is presented as follows.

Chapter 2 provides a review of the previous studies on the area of inspection. It
provides a background to infrastructure management, structural health monitoring and
inspection. The chapter provides a detailed review of the current techniques employed in
the field of inspection, including non-destructive evaluation and visual inspection. Fur-
ther, it reviews the previous research related to semi- and automatic computer systems
that have been developed to aid visual inspection.
Figure 1.6: An example of a mosaic of Mustek Station, Prague Metro.
Chapter 3 presents an overview of the proposed system developed in the research. It provides a background to the registration and reconstruction technologies. Preliminary experiments are conducted to discover the limitations of the current technologies. The chapter presents the results and a discussion related to the experiments.

Chapter 4 describes all of the datasets used in the research. The chapter explains how the datasets are obtained and used for evaluation.

Chapter 5 explains the proposed methods for improving image mosaicing. The chapter provides a review of the current mosaicing technologies. Then, each component in the pipeline of the proposed system is explained, including the background to Support Vector Machine classification and how it is used, non-linear least squares optimisation for cylindrical surfaces, and image warping and stitching. The results and a discussion of the experiments are presented at the end of the chapter.

Chapter 6 explains the proposed methods to improve feature matching. A review of the current feature matching algorithms is presented. The chapter then explains the proposed algorithms, which are based on a spatial consistency constraint. The results are shown and discussed at the end of the chapter.

Chapter 7 explains the pipeline of the proposed change detection system. The chapter begins with a review and background of change detection systems. Then, each component in the pipeline is explained, including geometrical registration by Bundler, photometrical registration, and change detection algorithms. The methodology for the quantitative evaluation of the system is explained. Finally, the results and a discussion are presented.

Chapter 8 provides the conclusions of the research. A discussion of these is presented and the chapter explains suggested future work.
Chapter II

Literature Review

The greatest challenge for research on infrastructure management is its breadth and depth. This area of research brings in knowledge and technology from across many disciplines. Integrating different technologies to build a system to manage infrastructure successfully is difficult and, often, the progress of an integrated system is hindered by the relatively slow technological developments. This chapter provides a review of the technologies used in infrastructure management, particularly the current techniques employed for inspection. The chapter starts with a review of the framework of infrastructure management, with an emphasis on two particular components of it, namely Structure Health Monitoring (SHM) and Inspection. This is then followed by a review of the inspection techniques, particularly those related to Non-Destructive Evaluation (NDE), which are used widely in the condition assessment of structures. The chapter then provides a detailed review of the inspection systems that are based on photographic technology, before the limitations and advantages of photographic-based inspection techniques are discussed.

2.1 Management of infrastructure in civil engineering

Structural infrastructure, such as buildings, bridges, dams and tunnels, must be properly managed to ensure that the intended service life spans are reached with satisfactory performance. Normally, infrastructure is designed to span decades or even centuries. The structural safety and condition of infrastructure gradually deteriorates due to normal wear and tear, exposure to aggressive environments (e.g. heating and cooling cycles, chloride ingress, and free-thaw effects), natural hazards (e.g. earthquake, flood, hurricane), and man-made disasters (e.g. terrorist attacks, fire). As a result, the capacity of infrastructure to carry loads safely is reduced and its functionality may also be impaired. As shown in Figure 2.1(left), the performance of infrastructure may not follow the initial predictions
due to the uncertainties of in-service loads and deterioration processes. This can result in either unacceptable levels of performance or the failure of infrastructure to fulfil its intended service life. However, as shown in Figure 2.1(right), effective maintenance and risk mitigation strategies can be deployed to ensure satisfactory performance over the intended life of infrastructure (Messervey, 2008).

Frangopol and Liu (2007) propose a framework for the management of infrastructure, based on a life-cycle cost (LCC) analysis. The goal of the analysis is to allocate resources effectively such that the conditions, safety and performance are optimised for individual structures as well as the network, within the budgetary constraints. Figure 2.2 presents a schematic diagram of the framework. The framework is designed to ensure that optimal decisions and action plans are created. The plans can be either preventive in nature (e.g. the application of sealer on a bridge deck) or corrective (e.g. the replacement of a structural member or system). Essentially, the framework can be grouped into three core components as follows.

1. System performance assessment and prediction—the performance indicator upon which decisions are made,

2. Structural Health Monitoring (SHM)—the monitoring and updating tools by which the optimum plan is validated throughout the lifespan of a structure or a network of inter-dependent structures,

3. Optimisation—by which the best future plans are determined.

The modelling, assessment and performance prediction of structures over time is complex and uncertain. Particularly, modelling is very sensitive to input parameters. The knowledge of the real conditions of structures obtained from assessment can help to reduce the uncertainty of the prediction models, and ultimately improve the structural assessment procedures (Frangopol, 2011). Visual-based inspections or non-destructive evaluation (NDE) have primarily performed these functions of assessment (Estes et al., 2004). SHM, as one of the NDE methods, provides continuous monitoring, while other NDE methods provide monitoring at a point in time (Frangopol, 2011). SHM and inspection methods complement each other, and therefore should always be integrated into the infrastructure assessment framework to obtain a complete picture of the structure conditions (Zaurin and Catbas, 2009). In the following sections, a brief introduction to SHM and inspection techniques is provided.
Figure 2.1: An example of deterioration curves: (left) if the structures are not maintained, serviceability is reduced; (right) proper maintenance schemes can extend the life span of structures, taken from Messervey (2008).

Figure 2.2: A life cycle framework of infrastructure maintenance, proposed by Frangopol (2011).
2.1.1 Structural Health Monitoring

*Structural Health Monitoring (SHM)* is an important component of an infrastructure management framework. In aerospace, civil and mechanical engineering, SHM is the process of implementing a damage identification strategy for infrastructure (Farrar et al., 2001, Farrar and Worden, 2007). In this context, only structural and mechanical systems are considered and, therefore, *damage* is limited to only *changes* to the material and/or geometric properties of the systems. These changes adversely affect the current and future performance of the systems. The objectives of SHM are to observe the infrastructure conditions, assess the in-service performance, detect deterioration and estimate the remaining service life. This is to ensure that the structural integrity and safety of the systems remain intact (Ahlborn et al., 2010). In this particular era of ageing infrastructure, the role of SHM has been given much greater importance in infrastructure management. This is evident from the rapid increase in the amount of research related to SHM that has been carried out during the past ten years. Farrar and Worden (2007) state that good SHM systems, implemented to monitor the condition of structures, have great potential to increase life-safety as well as offering considerable economic benefits. Aktan et al. (2001) provide a review of the state-of-the-art SHM systems and stress the importance of monitoring and managing the health of the civil infrastructure. Tominaga et al. (2002) propose an infrastructure management framework that places more importance on a monitoring system in SHM. Messervey (2008) conducted extensive studies and suggests that SHM should be integrated into the design and life-cycle analysis of infrastructure. These are some examples in recent studies related to SHM.

The studies related to SHM can be grouped into four areas, according to Sohn (2004). These are: (1) operational evaluation, (2) data acquisition, (3) feature extraction, and (4) statistical modelling for feature discrimination. *Operational evaluation* answers four questions regarding the implementation of SHM monitoring schemes.

1. How is damage defined for the system being monitored?
2. What are the conditions, both operational and environmental, under which the system to be monitored functions?
3. What are limitations related to acquiring data in the operational environment?
4. What are the economic and life safety motives for performing the monitoring?
Data acquisition deals with the design of a monitoring system and determines the types of sensor, the location of the installations, the number of sensors and other concerns. There are a large number of studies reporting the development of monitoring technologies, including Fibre optic sensors, (Micro-Electro-Mechanical System) MEMS sensors and wireless data acquisition systems (Brownjohn, 2007). This paper provides a review and case studies of the SHM systems employed in different fields of civil infrastructure, including dams, bridges, offshore installations, buildings and towers, nuclear installations, and tunnels and excavations (Hoult et al., 2010, Lynch, 2005, Mohamad et al., 2010). It states that one of the major focuses of SHM research has been to advance sensor technologies, and sensors developed within other engineering disciplines are now finding their way into civil applications. Feature extraction is the area of SHM that receives the most attention in the literature (Sohn, 2004). It is the process of identifying damage-sensitive properties, and also involves data compression due to the large amount of data with which SHM systems normally have to deal. Lastly, statistical modelling attempts to assess if the changes to selected features that are used to identify damage in a system are statistically significant (Farrar et al., 2001).

Smart sensing or wireless sensor networks (WSN) have become a new paradigm in SHM (Spencer and Nagayama, 2006). A wireless sensor network normally consists of spatially distributed autonomous sensors that monitor the physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants, and cooperatively pass data through a network to a main location (Sohraby et al., 2007). Some studies in this area (Hoult et al., 2010, Bennett et al., 2009, Spencer and Nagayama, 2006) address the practical problems related to deploying a WSN system at real sites. Much work is still needed to develop WSN systems to a stage whereby they can be adopted by industry.

Most research related to SHM has been focusing on the deployment of monitoring sensors (e.g. MEMS, Fibre optics). Only a small number of systems have utilised photographic technologies as monitoring sensors. Zaurin and Catbas (2007) integrate computer vision using video cameras as part of an SHM system, as shown in Figure 2.3. A computer vision module is used to track the motion of vehicles on a testing bridge, and the motion is verified against readings taken from a sensor array. Lee and Shinozuka (2006) apply video cameras, which are installed on bridge columns, to track displacements in their SHM system. Fraser et al. (2003) propose that computer vision should be integrated into a health monitoring framework because video signals can be used as archives for long-term
condition assessment, and in real-time monitoring, such as the motion of bridge decks, using a vision-based motion estimation algorithm.

Cameras have, in general, become cheaper and, therefore, it is economically feasible to utilise photographic technology in SHM systems. Another advantage of photographic technologies is that images give visual information, which is intuitive for humans, and can be interpreted by non-experienced users, unlike other sensors that require expertise in data interpretation. The systems based on photographic technologies are further explained in Section 2.3.

2.1.2 Roles of inspection for infrastructure management

A complete SHM system requires routine inspection, which is normally carried out by experienced engineers to assess the conditions of structures. For example, inspection forms an integral part of a monitoring system in a bridge management system, as shown in Figure 2.4 (Messervey, 2008). In Figure 2.5, optimisation is performed over all costs in the infrastructure management framework (see Section 2.1). The figure shows the importance of inspection as one of the main considered costs. The cost of manual visual inspection is normally high due to the expensive labour costs.

Bridge management systems rely heavily on manual inspection. As evidenced in 2006, when an interstate highway overpass collapsed during routine usage in Laval, Canada, killing 6 people (CTV News Online, 2011), the collapse was caused by the debonding of the internal steel reinforcement due to corrosion at the deck-column interface of the reinforced concrete structure. An important fact was that this collapse occurred after the bridge had recently passed its periodic visual inspection. This tragedy raised questions over the effectiveness of visual-based inspection and whether or not it can provide an adequate level of safety (Messervey, 2008).

So far, no SHM system exists that is able to determine the conditions of structural systems completely. The inspection technologies used widely in remote sensing offer the ability to integrate several methods to provide a better picture of the overall conditions of structural systems. Ahlborn et al. (2010) review the technologies in remote sensing that have the potential to be applied in infrastructure management systems. Remote sensing is attractive because it enables non-contact data collection at a great distance.
Health Monitoring can be defined then as measurement of the health of a structure by combining analytical methods that can be used to rapidly identify the onset of structural damage in an instrumented structural system [1], [2]. Structural health monitoring (SHM) paradigm offers an automated method for tracking structural responses under an operational environment [3].

The main contributions of the framework proposed in this study consists of five main components, integrated and closely interrelated as is described below: the vision module, the distributed sensors network array, the analytical model, the database, and the diagnostic module. The structure is monitored at every instant by using operational pre-recorded data, computer vision algorithms, and combining them with sensing technology however, only a few limited attempts have been tested and implemented. A framework for intelligent sensor network was proposed, combining a network sensors array and computer vision applications for detection and classification of traffic. Vision capabilities will be used to build the database. Input from video cameras will be used to detect and classify the traffic, or any other object.

In the beginning, vision capabilities will be used to build the database. Input from video cameras will be used to detect and classify the traffic, or any other object, unless testing was scheduled by closing the bridge. The use of video cameras was intended to be triggered when the activity monitored by sensors couldn't differentiate ambient or traffic readings. Very recently, some investigators have contemplated the possibility of implementing and converting pre-recorded data, computer vision algorithms, and artificial intelligence as a mean to classify and keep records of the main objectives was the use of video analysis of artificial intelligence as a mean to classify and keep records of traffic. Video cameras were intended to be triggered when the activity monitored by sensors couldn't differentiate ambient or traffic readings.

Figure 2.3: Computer vision in an integrated SHM framework, taken from Zaurin and Catbas (2007).
Figure 2.4: Bridge management system framework, taken from Messervey (2008).

Figure 2.5: The role of inspection in cost optimisation in infrastructure management, taken from Frangopol (2011). It can be seen that the cost of the inspection component is important as it is considered part of the main costs in the framework.
2.2 Inspection techniques

The purpose of inspection is to perform a condition assessment of structural systems—assessing whether the materials in structural systems have adequate capacity and durability. The materials commonly used in civil engineering structures are concrete and steel, such as reinforced concrete. Common defects in reinforced concrete include cracking, scaling, delamination, spalling, chloride contamination, efflorescence, ettringite formation, honeycombing, wear, collision damage, abrasion and steel corrosion in reinforcing steel. Common defects in the steel members of reinforced concrete are corrosion, fatigue cracking, overloading, heat damage, and paint failure (Mallett, 1994). One focus of inspection is to identify whether a crack is dormant or active, so that an optimal repair regime can be devised. For example, if cracks are dormant (i.e. not displaying any sign of growth), the remedy can be simply to seal them and so prevent the further ingress of water. On the other hand, if the cracks are changing, either by propagating or widening, the causes of the change must be located and primarily remedied before making a permanent repair.

Inspection can be classified as regular or detailed inspection (Yamaguchi et al., 2006). Table 2.1 provides a brief summary of regular and detailed inspection. From the table, regular inspection is periodically performed by looking and hammering to obtain an initial assessment of the structural components. Detailed inspection is carried out to assess problem areas further or when visual inspection cannot be performed, such as to detect cracks inside structural components.

The techniques of detailed inspection can be categorised as destructive and non-destructive evaluation. Destructive evaluation typically relates to material performance, which can affect the integrity of the structures under evaluation. The techniques used in destructive evaluation include concrete coring, the Brinell hardness test, the Charpy impact test and tensile tests. Destructive tests can affect the integrity of structural components; therefore, the amount of tests that can be done would typically be limited (Ahlborn et al., 2010).

Non-destructive evaluation (NDE) assesses the properties of materials, components or systems, without causing damage (Cartz, 1995). Techniques in NDE are classified as rapid screening and detailed investigation. In detailed investigation, techniques are used to reveal defects that sometimes do not appear on the surface of structures, such as discontinuities or cracks inside tunnel linings. However, these techniques require the
shutting down of normal operations, and hence cannot be carried out so regularly. An overview of some of the NDE techniques is presented in Table 2.2. The table provides a summary of the principles, usage, and limitations of these NDE techniques. The techniques summarised in the table aim to reveal discontinuities inside concrete structures using various types of signal, such as mechanical waves (e.g. manual acoustic, impact-echo, Spectral Analysis of Surface Waves), magnetic fields (e.g. magnetic method) and electrical fields (e.g. Ground Penetrating Radar). Imaging techniques aim to provide surface and subsurface imaging, such as infrared thermography, which produces a thermal image by detecting radiation from objects. The model testing shown in the table aims to obtain the parameters for the structural components by using a scaled version of the components. A detailed survey of the NDE techniques employed in the field of civil engineering can be found in Costello et al. (2007), Yamaguchi et al. (2006) and Delatte et al. (2003).

2.3 Vision-based inspection

From hereon, vision-based inspection will refer to inspection techniques that involve the use of photographic technology—i.e. videos or images. Visual information from videos or
Table 2.2: A summary of the non-destructive evaluation (NDE) techniques

<table>
<thead>
<tr>
<th>Method</th>
<th>Overview</th>
<th>Usage</th>
<th>Limitation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual Acoustic or hammering noise</td>
<td>Concrete surface struck with a hammer and a high-pitched or ringing sound indicates that it is free from defects or delamination. Conversely, a dull or hollow sound indicates potential defects. Now, a rotary percussion tool can be used to produce a uniform tapping.</td>
<td>Detect delamination or holes inside concrete</td>
<td>Effective for the top 4-6 inches of concrete.</td>
</tr>
<tr>
<td>Magnetic</td>
<td>Depth of concrete cover is measured from magnetic field distortion caused by a reinforcing bar.</td>
<td>Detect the concrete cover depth</td>
<td>Requires knowledge of the bar diameter; accuracy ±5 mm in 100 mm.</td>
</tr>
<tr>
<td>Impact-Echo</td>
<td>A mechanical impact generates a short duration stress pulse which will be reflected by discontinuities.</td>
<td>Used to locate flaws inside the concrete</td>
<td>Impact-echo investigation should include verification with other methods.</td>
</tr>
<tr>
<td>Spectral Analysis of Surface Waves (SASW) technology (2004)</td>
<td>Based on the dispersion effect that mechanical waves exhibit when travelling through an inhomogeneous media (e.g. concrete, asphalt, soil). Different wavelengths are affected by the conditions at various depth, and hence the shear stiffness can be found against depth.</td>
<td>Shear stiffness of concrete. Structure with freeze/thaw damages. Surface cracking</td>
<td>They must remain stationary while in use, hence road and bridge closures would be necessary.</td>
</tr>
<tr>
<td>Method</td>
<td>Overview</td>
<td>Usage</td>
<td>Limitation</td>
</tr>
<tr>
<td>-----------------------------</td>
<td>--------------------------------------------------------------------------</td>
<td>------------------------------------------------------------------------</td>
<td>-------------------------------------------------------------------------------------------------</td>
</tr>
<tr>
<td>Ground Penetrating Radar (GPR)</td>
<td>This is a geophysical method that uses radar pulses to image the subsurface. Electromagnetic radiation detects the reflected signals from the subsurface.</td>
<td>Locate reinforcing bars, determine thickness of concrete linings, detect cracks and delamination</td>
<td>Poor performance in high-conductivity materials such as clayey soils, high cost, requires experts to interpret the results.</td>
</tr>
<tr>
<td>Infrared Thermography</td>
<td>An infrared camera detects radiation in the infrared range of the electromagnetic spectrum and produces images of that radiation. The amount of radiation emitted by an object increases with temperature; therefore, thermography allows one to see variations in temperature.</td>
<td>Locate areas of near-surface delamination/cracking in concrete decks, facades and similar structural elements.</td>
<td>Requires expertise to interpret thermal images. Unable to measure depth and affected by the surrounding environment.</td>
</tr>
<tr>
<td>Model Testing</td>
<td>Spectrum analysis and multiple vibration pickups are a useful technique for identifying the modal parameters (e.g. mode shapes, frequencies, and damping ratio) of a structure. The parameters can be used to determine dynamic behaviour that identifies the overall stiffness reduction and seismic susceptibility</td>
<td>Overall stiffness. Behaviour and performance over time</td>
<td>Prediction may be unreliable; high cost of creating a model; a high degree of precision required.</td>
</tr>
</tbody>
</table>
images provides rich information, such as colour, texture and 3D cues, about the structures under inspection. As an example, leakage, which is identified as a major problem for tunnels (Delatte et al., 2003, Mallett, 1994), can only be detected via photographic techniques using colour information. In practice, photographs are commonly used in visual inspection as a means of recording any detected defects in the areas under inspection. In this section, a review of the technologies employed in vision-based inspection is presented.

2.3.1 Visual inspection

Visual inspection is a common technique used by inspectors. Structural components are examined based on the experience of the inspectors, who will assess the conditions of the structures based on their visual appearance. Inspection must be carried out within an arm’s length of an inspector, although this is very difficult in many inaccessible areas, such as ventilation shafts, which are usually narrow. An example guideline for the inspection of tunnels and underground infrastructure can be found in Federal Highway Administration and Federal Transit Administration (2005). This guideline presents the procedures that the inspectors must follow during inspections. In general, for reinforced concrete components, the commonly sought visible defects are cracks, scaling, spalling or staining, as exemplified in Figure 2.6. After inspection, an inspection report is produced to record any defects found on the tunnel linings. Usually, the report contains sketches of the defects found in problematic areas, as shown in Figure 2.7. In general, the most sought after defects are cracks, since they are the primary indicator of deterioration patterns, which are due to other severe causes (Soga, 2007) that need to be analysed. For example, repair regimes for cracks caused by the deformation of structures will differ from those caused by rusting due to water ingress. The failure to detect cracks can result in catastrophic accidents, such as that which occurred in the Rebunhama tunnel in Japan, which was caused by a failure to detect shear cracks (Asakura and Kojima, 2003).

Visual inspections must be carried out on a regular basis. As shown in Table 2.3 (Messervey, 2008), in the UK, the frequency of inspections ranges between 2-6 years, and they must be carried out by trained engineers. Similarly, for tunnel inspection, the guideline from the US Department of Transportation (Federal Highway Administration and Federal Transit Administration, 2005) suggests that tunnel owners should establish a frequency for up-close inspections based on the age and condition of the tunnels. For
new tunnels, this period could be as great as five years and, for older tunnels, a much more frequent inspection period may be required, possibly every two years. This up-close inspection is in addition to the daily, weekly, or monthly walk-through general inspections.

This poses a number of problems for visual inspections. Firstly, visual inspections cannot be performed more regularly due to the costs and time required. The second problem is accuracy, because visual inspection is subjective. Inspectors use their experience to assess the condition of structures based on their qualitative judgment. There may be discrepancies between the outcomes of inspections performed by different inspectors. Another problem is the inspection report itself, which usually contains long descriptions of the defects’ appearance and location. Referencing defect locations can cause considerable confusion for those reading the inspection reports. The current system of making references for defect locations is based on the reference numbers of tunnel linings. This system ensures that the exact location of each photograph is known and that the inspectors use the same system to avoid confusion.

In inspection reports, photography is a common method for providing records of defects. Photographs provide valuable information, such as the texture, colour and 3D cues about the surface of the assessed structures. For example, water ingress can cause changes to the colour of the tunnel surfaces. Other methods, such as the LiDAR system, cannot provide such colour-rich information.

At the time of writing, Metronet Alliance (Metronet, 2007) is currently implementing a new technique for creating inspection reports using photographs. The technique involves the manual stitching of images taken inside a shaft. The images are acquired at different heights along the shaft. An example of an inspection report produced by Metronet is shown in Figure 1.1, in which cracks and other defects are translated into animated formats. This report is time-consuming to create because manually labelling and locating defects from a large number of individual images is labour-intensive.

In recent years, visual inspection incorporates techniques to allow fast data acquisition, such as the use of a self-navigated robot to acquire videos of sites under inspection. A common example is found in sewer inspection systems, inside which CCTV is used (Makar, 1999, Costello et al., 2007). In this area of research, the aim is usually to improve the design of a robot navigation system and the remote control of a camera system. As shown in Lawson and Pretlove (1998), a stereo vision system and augmented reality are used as a vision system in a robot. This system can provide 3D information for the robot so that
it can be self-navigated.

The video stream obtained from the robot system is then processed by the inspectors, who are required to identify defects from watching video streams for hours (Sinha, 2001). In recent years, systems have been developed that can automatically detect defects from the video streams, although the performance of these systems, such as their accuracy, robustness and efficiency, still requires a great deal of improvement. Some of these systems are reviewed in Section 2.3. It is worth emphasising that, although these systems allow the fast acquisition of data, manual interpretation based on visual inspection is still required.

Idoux (2005) presents a system called ATLAS 70, which is a complete sensor and software package. The system mainly consists of two sensors: laser and infrared scanners. The laser scanner amplifies anomalies on the surface of tunnel walls to enable cracks as small as 0.2 mm to be viewed at a distance of 6 metres. The infrared scanner detects differences in surface temperature to allow the detection of tiny liquid or air infiltration into concrete. The acquired data are post-processed remotely by a trained inspector, who compares various images with reference digital drawings or images. Furthermore, images, taken at regular intervals, are used during visual comparisons to assess the evolution of cracks and the infiltration of water. This process of inspection is highly subjective, as it relies considerably on the experience of the inspector. The performance of this system in crack detection depends on a number of factors, such as the size of the crack, the distance of the scanner from the tunnel walls, and, most importantly, the experience of the inspector.

2.3.2 Automatic and interactive inspection systems

This section reviews the automated inspection systems which can be grouped into the following themes: detection, visualisation and interpretation. With a combination of different inspection methods, inspectors can be reassured that most defects are detected; no single method is sufficient to determine the state of deterioration. Nevertheless, visual inspection, which identifies defects from observation with the naked eye or photographs, can provide relatively accurate information for structural assessment.
Therefore, we decided to use the variance comparison:

\[ \text{The variance comparison is used to form a function of crack strength in an area.} \]

\[ \text{The resulting discriminate of pixels in region } R \]

\[ \text{is accurate, but } \]

\[ \text{the operator is not centered on an edge interface.} \]

\[ \text{Comparison is used to form a function of crack strength in an area.} \]

\[ \text{The mathematics for distribution parameter sets of points, each set being described by a normal distribution.} \]

\[ \text{Yakimovsky assume that cracks are interfaces between cracks, (e) Water marks caused by flood, (f) Spalling, and (g) Staining.} \]

Figure 2.6: Examples of images of defects found in tunnel linings (picture (a)-(d) taken from Sinha (2001). (a) Minor cracks, (b) Major crack, (c) Multiple cracks, (d) Mushroom cracks, (e) Water marks caused by flood, (f) Spalling, and (g) Staining.
Figure 2.7: An example inspection report, taken from Federal Highway Administration and Federal Transit Administration (2005).
Table 2.3: Selected bridge inspection data from Messervy (2008). For tunnel inspections, the guideline from the US Department of Transportation (Federal Highway Administration and Federal Transit Administration, 2005) suggests that up-close inspections should occur every 5 years for new tunnels, and every 2 years for older ones. This up-close inspection is in addition to the daily, weekly, or monthly walk-through general inspections.

<table>
<thead>
<tr>
<th>Country</th>
<th>Inspection</th>
<th>Interval</th>
<th>Inspector</th>
</tr>
</thead>
<tbody>
<tr>
<td>Denmark</td>
<td>Daily</td>
<td></td>
<td>Road Patrol</td>
</tr>
<tr>
<td></td>
<td>Semiannual Principal</td>
<td>6 years</td>
<td>Road Patrol</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Road Directorate trained inspector</td>
</tr>
<tr>
<td>France</td>
<td>Routine Annual IQOA Condition Evaluation</td>
<td>Frequent</td>
<td>Road Maintenance Crew</td>
</tr>
<tr>
<td></td>
<td></td>
<td>3 years</td>
<td>Local agencies</td>
</tr>
<tr>
<td></td>
<td>Detailed</td>
<td>3 to 9 years</td>
<td>Local agency certified inspectors</td>
</tr>
<tr>
<td></td>
<td></td>
<td>(6 on average)</td>
<td>National certified inspector</td>
</tr>
<tr>
<td>Norway</td>
<td>General</td>
<td>1 year</td>
<td>General knowledge of bridges</td>
</tr>
<tr>
<td></td>
<td>Major</td>
<td>5 years</td>
<td>Civil engineering degree and general knowledge of bridges</td>
</tr>
<tr>
<td>United Kingdom</td>
<td>General Principal</td>
<td>2 years</td>
<td>Training and quality control by engineering consultants</td>
</tr>
<tr>
<td></td>
<td></td>
<td>6 years</td>
<td>Training and quality control by engineering consultants</td>
</tr>
<tr>
<td>United States</td>
<td>Routine Underwater Fracture critical Damage, in depth, special</td>
<td>2 years as required</td>
<td>Team leader and certified inspection personnel</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Team leader and certified inspection personnel</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Team leader and certified inspection personnel</td>
</tr>
</tbody>
</table>
2.3.2.1 Detection

Crack detection systems are most commonly found in the literature related to inspections. Algorithms for crack detection generally involve a pre-processing step and a crack identification step (Guo et al., 2009). The pre-processing step applies image processing techniques to extract potential crack features, such as edges. The identification step usually applies crack modelling and/or pattern recognition techniques in order to determine if the extracted features belong to crack regions. One aim of the crack detection systems is to obtain a crack width, that is then monitored to see if the crack becomes larger over time. This indicates a change in the monitored structural components. The review below presents some examples of the work that has been done in this area.

Threshold-based algorithm Cracks are detected based on an assumption that pixels belonging to crack regions are darker than those in the neighbouring regions. Therefore, this method relies on thresholds, which can be applied either globally or locally, to extract the regions of the cracks. Miyamoto et al. (2007) compute the difference in intensity between each pixel and the average intensity of each row in an image matrix. A pixel that differs considerably from the average is deemed to belong to a crack. Fujita et al. (2006) apply a line filter using the Hessian matrix and a threshold is applied to extract the crack regions.

The main drawback of threshold-based algorithms is how to choose a suitable threshold for extracting crack features. The above algorithms select a threshold based on prior knowledge. However, such methods cannot be generalised and may be inapplicable to the imaging conditions found in real images. Also, threshold-based algorithms are prone to inaccuracy caused by shadows. The intensities of shadow pixels tend to have a similar brightness to pixels in crack regions, and therefore it is difficult to select a threshold that will distinguish between the two types of pixels. The above algorithms are tested on images that are unaffected by the shadow effect, and the results from these algorithms will not necessarily reflect images from a real environment.

Model-based algorithm Ukai (2000) develops a system for detecting cracks from the deformation of tunnel walls. The model of a crack in this system is characterised by eight
quantities, such as area and Feret’s occupancy rate.\textsuperscript{1} A filter is subsequently applied to remove noise. Yamaguchi and Hashimoto (2006) modelled cracks based on the percolation model, which is a physical model based on liquid permeation. The modelling process starts by initialising a seed region and then the neighbouring regions are labelled as crack regions based on the percolation process. Paar et al. (2006) propose a crack detection algorithm based on the line tracing algorithm. In this algorithm, a crack is assumed to be a series of short straight lines connected together. The algorithm starts from a seed point and then searches for a line within the neighbouring regions. Once a line has been detected, a connection is made to the previous line, and the algorithm continues.

Crack extraction by model-based methods relies on user input to initialise the seed pixels. Hairline cracks may go undetected because users may be unable to identify the seed pixels for these hairline cracks, which are usually invisible. Due to reliance on the user input, this method may not be scalable.

**Pattern recognition** In this theme of detection algorithms, pattern recognition is applied to obtain semantic information from image or video data. Zhu and Brilakis (2010) propose an algorithm for detecting concrete columns based on texture using artificial neural networks. Liu et al. (2002) applies a Support Vector Machine classifier to classify if crack features appear in an image patch, which is pre-processed to extract potential crack features based on intensity. Abdelqader et al. (2006) apply a Principal Component Principles (PCA) algorithm, which can be used to reduce the dimensions of feature vectors based on eigenvalues, to extract cracks from concrete bridge decks. The images are first pre-processed by line filters in three directions: vertical, horizontal and oblique; then further processed by the PCA algorithm and classified based on the nearest neighbour algorithm.

Algorithms based on pattern recognition rely on training data in order to build robust classifiers. Training and validation data are performed by manual labelling, which is a labour-intensive procedure. Also, there is no guarantee that the training and validation data are correct, which may lead to classifiers performing poorly.

\textsuperscript{1} The Feret diameter is a measure of an object size in a specified direction. It is equivalent to the measurement of an object size with a caliper. This measure is used for the analysis of particle sizes.
The purpose of this area of research is to enhance the visualisation of image or video data to enhance inspections. As the image or video data become large, it is essential to organise or, more importantly, be able to visualise the data.

Image stitching is a good method of visualising a collection of images. As shown earlier, Metronet (2007) tried this method, although the process of stitching was manual. Zhu et al. (2010) apply a commercial stitching software package to stitch the images of columns under bridges. The advantage of this method is that an entire column can be visualised in greater detail as individual images taken at a closer distance are stitched together. However, the fine detail of a column is lost if a single image is used to capture an entire column because the image has to be taken at a much greater distance. Song et al. (2005) create a stitched image of construction sites from a series of images taken by a robotic camera. The image is used as documentation to record the progress of the construction. Figure 2.8 shows an example of one of the test sites from this work.

Jahanshahi et al. (2011) create stitched images of structural systems from a specialised camera that can tilt and pan. The method detects missing parts such as bolts when comparing images taken at different times. The method applies a machine-vision algorithm to perform image registration to rectify images so that they are in the same coordinate frame. As a result, images can then be compared to find any missing parts. Figure 2.9 shows examples of the work.

Image stitching provides a good way of increasing the field of view that cannot be achieved by a single image. The methods reviewed above, however, can only handle a small number of images. This is because a mosaic is created based on a homography model in which a camera centre must be fixed at a single point. This method does not allow the translation of a camera. However, the images presented in this thesis are wide-baseline images and contain a significant degree of translation, so creating a mosaic based on the above model is impossible.

For up-close tunnel inspection, a wide-angle or stitched image may assist inspectors in detecting hairline cracks on the surfaces of structural components. Since the stitched image is constructed from individual images that can be taken at a closer distance, it is possible to provide a higher resolution image of cracks than if only a single image is used. However, a more important factor is the acquisition of close-up images. There are a large number of sites (e.g. shafts) that are not easily accessible, so obtaining close-up
images of these can prove extremely difficult. However, this problem can be solved by the use of robots or an image acquisition system that is able to access sites and obtain images remotely. This is a much more attractive method, as a human is not required and structural components can then be inspected off-line, although the further development of such hardware is required.

2.3.2.3 Interpretation

In this area of research, the aim is to obtain semantic information from detected defects so that a subsequent step, such as a repair regime, can be planned. Kaseko and Ritchie (1993) propose an artificial neural network system that classifies pavement surface cracking according to the type, severity and extent of the cracks detected in video images. Detected cracks are categorised into different types of crack, such as transverse, longitudinal, alligator and block. The severity is based on the extracted crack widths and the extent of the cracks. Sinha and Fieguth (2006) apply a new neuro-fuzzy classifier to classify the extracted features in buried pipe images into different types of cracks and other defects, such as holes on the pipe surface. Kim et al. (2007) propose a computer assisted crack diagnosis system for reinforced concrete structures to aid non-experts in diagnosing the cause of cracks at an expert level. The system diagnoses cracks based on crack symptoms and characteristics. The semantic information equips engineers with an in-depth knowledge of structural systems, which can be used to refine the prediction model of a system or design sensors for monitoring purposes.
Figure 5: An image database consisting of: (a) 24 images of a truss system, (b) 24 images of a full-scale structural model, (c) 24 images of a typical hospital ceiling support structure, and (d) 32 images of an MR damper with its attached structures.

Figure 11: (a) The reconstruction and the contribution of twelve selected images from the database (Figure 5), (b) current-view image of an MR damper (zoomed out), and (c) scene reconstruction using the Laplacian pyramid blending and exposure compensation. Note that two missing bolts in image (b) are shown by red circles in image (c). The lighting condition is different in images (a) and (b).

Figure 2.9: An example of image stitching from Jahanshahi et al. (2011), (a) original images and (b) a stitched image.

Figure 2.9: An example of image stitching from Jahanshahi et al. (2011), (a) original images and (b) a stitched image.
Crack change monitoring  Apart from establishing the types of cracks and associated sizes, it is also very useful to ask if the cracks have changed over time and how quickly they are changing. This information is extremely useful in helping to determine the **deterioration rate** of the structural components. In general, the information about the change can be applied to any type of defect. As shown in Figure 2.10, taken from Guo et al. (2009), the detection of change (i.e. change detection) is the main component in image interpretation for pipe inspection. Lim et al. (2005) propose a system for monitoring the changes in cracks from multi-temporal images. The system is based on a 2D projective transformation that can accurately extract the size of the cracks, which are then monitored in the images as the cracks propagate. Chen and Hutchinson (2010) propose a framework for concrete surface crack monitoring and quantification. The method is based on optical flow, which is used to track the movement of cracks. The regions, where the cracks become larger, are labelled as having changed.

The current systems related to monitoring cracks or anomalies rely greatly on some degree of user input. For example, in the system proposed by Guo et al. (2009), the final classification of anomalies is reviewed by human operators. In this work, the false alarm rate is 21%, which is still high. This system works with data obtained from CCTV, in which the camera is set in a fixed location. It may impractical to install many cameras to monitor cracks in tunnels. Lim et al. (2005) propose a system that can cope with images taken from different viewpoints. However, this system requires user input for the control points, which makes the system unscalable for a large number of images. The system proposed by Chen and Hutchinson (2010) is motion-invariant, which means that changes in the regions due to the motion of structural components are included in the model and, hence, actual change regions due to crack initiation and propagation will be detected. Unfortunately, this model only applies to a beam deforming within the image plane. Also, this method does not apply to images with different viewpoints and illuminations.

### 2.3.3 Applications of the vision-based inspection of underground infrastructure

Yu et al. (2007) present an integrated system, which performs automatic tunnel inspection. The system can be divided into two parts: image acquisition from a mobile robot and a crack detection algorithm. The robot system consists of a CCD line camera, the controlling apparatus for an auto-focus device, a vibration-reducing device, an illuminator, and encoders to measure the velocity and positions of the units, as shown in Figure
If corrosion has been recognized, a laser scanner could be further used to acquire data so that a quantitative measurement is conducted\[15\]. Hence, the municipality's need for identifying and quantitatively measuring extents of corrosion can be achieved automatically.

4.2. Change detection based approach for automated defect detection

As shown in Fig. 3, the first block Detector and the second block 1st Level Classifier compose the devised approach for automated defect detection for sewer pipeline inspection and condition assessment. It should be noted that there exists a variety of types of defects in sewer pipelines. For example, according to the standardized pipeline condition assessment grading system used in the Pipeline Assessment and Certification Program (PACP),\[18\], as shown in Fig. 1, there are four major families of defects, including Structural Defects, Operational and Maintenance Defects, Construction Defects, and Miscellaneous Defects; each family of defects is categorized further into several subclasses. Then, a critical question is how to detect those unknown multiple various targets automatically if each of them may be of unknown size, type, shape, color and so on. Hence, the challenge of this study is to find a way to automatically interpret a scene containing unpredictable objects (e.g. sewer defects and patterns) from a collection of several tens or hundreds of objects (e.g. various sewer defects and patterns which may also be of different sizes, poses, pipe materials, and shapes). Based on our knowledge and literature study, there is no existing work in sewer infrastructure systems that has yet adequately addressed this automated detection challenge.

The two most developed and used methods (known as template matching and feature matching) for detection in the image analysis and pattern recognition fields are difficult to apply to this pipe defect detection problem because: (1) no single definite template matching or matched filters can be implemented directly; and (2) no existing feature-based approach is feasible because features are unknown or of a large variety in sewer pipes. The traditional automated detection approaches in the pattern recognition field are not efficient when faced with the stated challenges in sewer infrastructure inspection and condition assessment. If for each defect or pattern we implemented a very accurate detector with very low false-positive and false-negative rates to determine whether it is a defect or false alarm, in principle, we could operate each such detector to achieve the detection of all the defects or patterns in an inspection image or video frame. However, this approach would be undoubtedly inefficient.

Therefore, we explored a different approach based on change detection in this research and report on it in the rest of this section.

Figure 2.10: A general block diagram of image processing and ROI detection through change detection from Guo et al. (2009).
2.11(a). The robot adjusts a camera focus based on its velocity and position without the need for an inspector. This study uses the main hardware and the path tracking algorithm to guide the robot. The crack detection algorithm is based on edge detection. The extraction of the crack regions is achieved through a graph search algorithm (i.e. Dijkstra algorithm), as shown in Figure 2.11(b).

This system claims to have an overall accuracy rate of 75-85% and the measurement error of recognised cracks is 10% or less. However, this high percentage of accuracy may have been achieved because only a small simple dataset is tested. This system removes non-crack features when the regions are small enough to be considered as noise, or when they are not long and narrow, and overly linear. These assumptions are not always true, as shown in Sinha and Fieguth (2006), which shows different types of crack, and some cracks may violate these assumptions.

2.3.4 Discussion

There are a number of studies related to crack detection. This area of research is still active and is aiming to create a crack detection system that is accurate and robust. Crack detection systems generally aim to determine the presence of cracks in images. Although such information is useful, it does not provide a complete picture of the health of structures. On the other hand, the systems related to crack monitoring provide a better picture of the overall conditions of structures. This information indicates the rate of structural deterioration, which is related to the rate of structural deformation, and is of more concern for engineers. Current crack monitoring systems, however, still require considerable improvement. The current systems do not account for all of the variations in the imaging conditions of real images. The main variations are due to the viewpoint, illumination and non-deformable objects. These variations are commonly found in real images. The work by Lim et al. (2005) can deal with the viewpoint variation problem, while Chen and Hutchinson (2010) can deal with non-deformable objects by a motion-invariant algorithm. Combining these algorithms can result in a better system that can deal with these variations. In Chapter 7, a new change detection system is proposed, which attempts to deal with these variations.

The systems for visualisation, shown in Section 2.3.2.2, are based on the homography transformation model. They cannot deal with images with translation and the resulting mosaic image usually contains a significant degree of distortion. Since the cameras cannot
Figure 2.11: System configuration by Yu et al. (2007): (a) the hardware components, and (b) the pipeline of the software system.
be translated in the homography model, a limited number of images can be stitched at one time. For tunnel inspections, a mosaicing system should be able to cope with images obtained from a general camera motion and the degree of distortion in a created mosaic should be minimised. Chapter 5 explains a proposed mosaic system, which can fulfil such requirements.

\subsection*{2.4 Summary}

The management of infrastructure requires the integration of many engineering disciplines in order to achieve the optimal strategies and decisions. These strategies and decisions must be optimised due to budgetary constraints. It can be observed that the framework, as proposed by Frangopol and Liu (2007), requires the constant feeding of information from the actual state of the structural system to rectify the prediction models as well as to adjust the action plans iteratively. This information is obtained from inspection and monitoring systems, which have seen an increase in the number of studies undertaken over the years because of the important roles that they play within infrastructure management.

Regarding monitoring, structural health monitoring is a paradigm describing the networks of sensors that are installed to monitor the state of a system. Much research has been focused on improving the capabilities of the sensors, such as precision and accuracy. However, methods for extracting useful information from these sensors’ measurements are underdeveloped. Many traditional SHM systems involve only sensors for monitoring purposes, although a better monitoring system should integrate an inspection system in order to assess the health of the structural systems more fully.

Inspection is one of the most important components of infrastructural management. Visual inspection and non-destructive evaluation are common practice for both rapid and detailed screening. Although visual inspection is an accepted practice for inspection, its intrinsic problems are known. Many systems have been created to allow the fast acquisition of data, although the manual analysis of the data is still required in these systems. As image or video data become large, it is impossible to perform a thorough analysis of these data and there is a real need for automatic vision systems. Vision-based systems have many advantages over visual inspection and many NDE techniques, especially the fact that costs have been declining in recent years. It is the author’s belief that vision-based systems are the future of inspection systems.

Regarding the technologies employed in vision-based inspection systems, a number of
studies have been carried out in the area of crack detection. Most of these studies show good results, although most fail to test their methods on images in a real site environment. The crack detection algorithms are not yet fully developed and new systems should show their results based on the testing of images obtained from a real environment. The areas for improvement in crack detection systems should focus on increasing accuracy, robustness and scalability.

There are a limited number of studies in the area of visualisation in vision-based systems, despite its importance. As image or video data become large, it is essential to be able to visualise them. Image stitching is one way to compress and summarise many images into a single wide-angle image. It can be used as documentation for inspection and it has an advantage over the current format of an inspection report, which often leads to confusion, especially in locating where the images were taken. There has not been a single system that attempts to produce a large stitched image of a tunnel wall. Therefore, this is one of the main objectives of this thesis.

With regards to a system for crack monitoring, most of the previous studies in the civil engineering literature fail to tackle a comparison of images taken from different viewpoints. Almost all systems assume a fixed camera position throughout an entire image sequence or otherwise require a precise camera calibration using control points. With the current technologies in computer vision, as shown in Jahanshahi et al. (2009), it is possible to relax all of these assumptions. Therefore, another research focus is to develop a system for comparing multi-temporal images taken from different viewpoints.
Chapter III

Evaluation Datasets

This chapter provides a summary of the datasets used in the experiments in this thesis. There are three sets of data: tunnel datasets, wide-baseline matching datasets and change detection datasets. In Section 3.1, the methods used to acquire datasets from tunnels and the details of the trial sites are described, together with a discussion of the practical issues. Section 3.2 describes the datasets that are used to evaluate the proposed feature matching algorithms in Chapter 6. In Section 3.3, the beam datasets, which are used to evaluate the proposed change detection system in Chapter 7, are described.

3.1 Tunnel datasets

The datasets are obtained from four sites, referred to as Bond Street, Aldwych, Mustek and Sagrada Familia. The datasets are named after the associated metro stations, and are obtained manually using a standard digital camera. Specialised equipment could have been used to speed up the acquisition time, such as a robotic system (Ahrary et al., 2007), but the aim of this project is to develop an inspection system that can readily be employed by inspectors without the need for any specialised equipment. The attractiveness of the proposed system is that it requires minimal equipment, which can easily be obtained from any camera shop.

3.1.1 Acquisition procedure

3.1.1.1 Equipment

There are three main pieces of equipment: a standard digital camera, a tripod and a portable spotlight. Various camera models are used for the different trial sites, as summarised in Table 3.1. The tripod can rotate 360° in any direction such that the full coverage of a tunnel ring can be obtained. The portable light is a diffuse light, designed
to ensure that the lighting is uniform on the tunnel linings. An external flash may be used, although it is not used in the datasets presented here due to the problems related to battery power, as the flash can drain the battery quickly.

3.1.1.2 Procedure

Figure 3.1 shows the schematic diagrams of the image acquisition procedure. The procedure ensures that the images or videos in the datasets possess the following attributes:

- a full coverage for each ring inside a tunnel,

- sufficient overlapping between images,

- sufficient baseline to avoid a degenerate configuration when performing 3D reconstruction.

As a rule of thumb, there should be some degree of overlapping for every three images; therefore, an overlap of more than 50% for every two images was chosen, as shown in Figures 3.1(a) and (b). The procedure is as follows. For each tunnel ring, a camera mounted on a tripod is rotated to cover approximately 270°, as shown in Figure 3.1(a), and the overlap between the images should be at least 50%. The camera is then moved along the tunnel by approximately half of the width of the ring, as shown in Figures 3.1(d) and (e). The amount of distance moved along the tunnel is flexible, depending on the width of the tunnel lining. In some datasets, extra sets of images are obtained, as shown in Figures 3.1(b) and (c), as redundancy datasets. The camera is rotated to cover the entire ring of a tunnel lining, but the rail tracks at the bottom section of the ring are omitted to allow a more accurate 3D model of the tunnel to be reconstructed.

For the image datasets, each image is taken using a self-timer function (e.g. 3-10s can be used) that releases the camera shutter to avoid blurring. The F-stop and shutter speed were set to the P mode when the flash was not used, and the Auto mode was employed when the flash was used. These modes are standard in modern digital cameras. The P mode lets the camera calculate both the shutter speed and aperture to obtain automatic image exposure, whereas the Auto mode allows the camera to alter all of the settings (e.g. flash, exposure compensation, etc.). An external light source is used when the flash is turned off (i.e. in the P mode). For image datasets, a camera is locked into position on a tripod when the images are taken to avoid vibration, which can cause image blurring.
Video datasets are also obtained at the trial sites, although they are not used in the experiments presented in this thesis.

3.1.2 Site descriptions

This section provides a brief description of the trial sites. The datasets are collected from four sites, Bond Street Station, Aldwych Station, Mustek Station, and Sagrada Familia Station. Table 3.1 provides a summary of the datasets.

3.1.2.1 Aldwych Station

Aldwych Station, originally opened as the Strand Station in 1907, is now closed. It was the subject of a number of proposals to extend its tunnel southwards, although these were never fulfilled. The station was the terminus of a short Piccadilly line back from Holborn, as shown in Figure 3.2(a). The Aldwych tunnel is 3.8m in diameter and 3.2m high, as measured from the track bed to the crown, as shown in Figure 3.2(b).

The images from the datasets are collected near the cross over area, as shown in Figure 3.2(c). The linings are made of cast iron, which has been painted, although there are many stains visible on the surface of the linings. The lighting at this site is poor and a camera flash was used to collect the images. The images were taken at this site using a similar procedure to that shown in Figure 3.1(b), although not with a tripod. The image dataset covers approximately 5-6 meters of the tunnel.

3.1.2.2 Bond Street Station

The images from this dataset were taken from the southbound tunnel between Bond Street Station and Baker Street Station of London Underground’s Jubilee line. The tunnel was constructed between 1973 and 1979 as part of the Stage 1 Fleet Line\(^1\) construction, as a solution to the overcrowded Bakerloo Line at that time (Cheung et al., 2010). Figure 3.2(a) shows the locations of these stations on the underground map. Currently, a new service, Crossrail line 1, is under construction, which will call at Bond Street Station. The service is due to commence in 2018. The Jubilee line tunnels between these stations

\(^{1}\) It was renamed as the Jubilee line in 1977 to mark Queen Elizabeth II’s Silver Jubilee celebrations.
Figure 3.1: Schematic diagram showing how the image and video datasets are obtained.
Table 3.1: A summary of the datasets presented in this thesis

<table>
<thead>
<tr>
<th>Site</th>
<th>Name</th>
<th>Date</th>
<th>Camera model</th>
<th>Calibration</th>
<th>Resolution (pixels)</th>
<th>Illumination</th>
<th>Capturing Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Prague</td>
<td>Prague</td>
<td>30/10/03</td>
<td>Unknown</td>
<td>Unknown</td>
<td>3.1M</td>
<td>Flash</td>
<td>Unknown (obtained by CTU)</td>
</tr>
<tr>
<td></td>
<td>2003 Prague 1</td>
<td>18/03/10</td>
<td>Canon 1000D</td>
<td>Unknown</td>
<td>2M</td>
<td>Flash</td>
<td>Figure 3.1d and e</td>
</tr>
<tr>
<td></td>
<td>2003 Prague 2</td>
<td>18/03/10</td>
<td>Canon 1000D</td>
<td>Unknown</td>
<td>3M</td>
<td>Flash</td>
<td>Figure 3.1d and e</td>
</tr>
<tr>
<td>Bond Street</td>
<td>Bond Street 1</td>
<td>18/03/09</td>
<td>Canon G9</td>
<td>Yes</td>
<td>2M</td>
<td>No Flash</td>
<td>Figure 3.1a and b</td>
</tr>
<tr>
<td></td>
<td>Bond Street 2</td>
<td>19/03/09</td>
<td>Canon G9</td>
<td>Yes</td>
<td>2M</td>
<td>No Flash</td>
<td>Figure 3.1a and b</td>
</tr>
<tr>
<td>Aldwych</td>
<td>Aldwych</td>
<td>06/01/08</td>
<td>Sanyo HD1A</td>
<td>Yes</td>
<td>2M</td>
<td>Flash</td>
<td>Free capture</td>
</tr>
<tr>
<td>Barcelona</td>
<td>Barcelona</td>
<td>27/08/10</td>
<td>Canon 1000D</td>
<td>Unknown</td>
<td>3M</td>
<td>No Flash</td>
<td>Figure 3.1d and e</td>
</tr>
</tbody>
</table>
Figure 3.2: (a) London Underground map showing Bond Street Station and Aldwych Station (the latter no longer appears on the underground map), (b) a cross-section of the Aldwych tunnels (Wu et al., 2009), (c) a picture of the site at the cross over area
are subjected to extensive inspection and monitoring due to the effect of the construction of the new lines and the fact that they themselves are beginning to deteriorate (Bennett et al., 2009). Figure 3.3 shows the locations from which Wu et al. (2009) collected the experimental data. The Bond Street datasets presented in this thesis were collected close to Bond T3, when Wu and the author visited the site together. The datasets cover the tunnel section from Ring no. 1680-1686, an area of approximately 7 metres. The tunnel has an internal diameter of about 3.8m. Another image dataset was obtained from Ring no. 1603-1607, where the tunnel is reinforced by extra steel structures. The images from this site are taken using an external diffuse spotlight.

3.1.2.3 Mustek Station

Mustek Station is a station on Prague Metro, Czech Republic. There are a total of three metro lines, as shown in Figure 3.5, which connect 57 stations across an area of nearly 60km. In August 2002, the metro suffered disastrous flooding that struck parts of Bohemia and other areas of Central Europe. Nineteen stations were flooded, causing the partial collapse of Prague’s transport system. The stations affected by the floods are shown in blue in Figure 3.5 (Jakoubek, 2007). Because of the flooding, Mustek has been subjected to extensive monitoring and inspection since then. The inspections are especially concerned with the ingress of surrounding water into the linings, in which many water patches are visible on the tunnel surfaces. The linings at this trial site are made of concrete and so prone to water ingress when cracks occur. 3D scanned data were obtained at this site, which are then compared with a 3D model obtained from the reconstruction algorithm, as discussed in Chapter 4. The images are collected at Ring no. 705-710 (approximately 5 meters), and Ring no. 740-750 (approximately 10 meters).

3.1.2.4 Sagrada Familia Station

Sagrada Familia Station forms part of the Barcelona Metro network and is named after the famous church of the same name designed by the architect, Antoni Gaudí. As shown in Figure 3.6(a), the station is served by lines L2 and L5. It was opened in 1970, when the Diagonal-Sagrada section was added to L5. The tunnel linings of this station are concrete, and in the shape of a semi-circular arch, as shown in Figure 3.6(b). Currently, a new tunnel is being constructed near this station, resulting in extensive monitoring at this site. A total of 190 images were collected, which covered approximately 10 meters.
Figure 3.3: Bond T1 is a straight 180m-long concrete tunnel from Rings no. 1984 to 2280, Bond T2 is a curved 120m-long cast iron tunnel from Rings no. 1782 to 1983, and Bond T3 is a curved 108m-long concrete tunnel from Rings no. 1423 to 1603, taken from Wu et al. (2009).

Figure 3.4: Inside of a tunnel at Mustek Station, Prague Metro.
Figure 3.5: Prague Metro map, the stations and tunnels affected by the great flood in 2001 are highlighted in blue.
along the tunnel. A portable light was used at this site.

3.1.3 Practical issues

This section describes some of the practical issues that arose when the image datasets were manually collected at the trial sites. These issues must be considered when designing an image collection system in the future.

3.1.3.1 Time

Apart from the data collected from Aldwych Station, all of the trial sites required the images to be collected during engineering hours in the underground, which are from the time after the last train stops at the terminal station and before the first train operates, which is approximately 00.30-4.30a.m. However, the actual working hours are approximately 3 hours maximum, since time must be spent conducting safety checks before access to the tunnels is allowed and also for preparation before the first train operates. This time constraint meant that only a small section of tunnel images could be collected per session. With the demand for train services to operate for even longer hours, this inspection time could become even shorter. An image acquisition system should be able to collect images as fast as possible.

3.1.3.2 Lighting

Insufficient lighting in tunnels is a major issue in this research. In practice, many sections of the tunnels are not illuminated, although additional lighting can be supplied using a flash or external spotlight. For the datasets obtained for this study, some images were collected using a camera flash, while others were taken using an external light. Using a flash can quickly drain the camera battery, and the camera will also take longer to charge up between each image if the flash is used continuously. A diffused light is employed when an external light source is used, which is sufficient for a small tunnel. However, the source is not powerful enough for a bigger tunnel, such as that at Aldwych Station. The advantage of using an external light over using a flash is that the images contain fewer shadows.
Figure 3.6: (a) a Barcelona Metro map, (b) a picture inside the tunnel at Sagrada Familia Station, Barcelona Metro.
3.1.3.3 **Site landscape**

A tripod can sometimes be difficult to adjust into position on the ground. In the Mustek Station tunnels, a tripod must be adjusted to stand on the sleepers because the rail tracks sit on top of small drain channels. As result, the degree of overlap between the images obtained from Mustek Station was relatively small compared with those obtained from other sites.

3.1.3.4 **Other issues**

In many London Underground tunnels, an electrical power supply is unavailable, so the equipment for acquiring images cannot rely on the provision of power supply at the site. Another issue is the use of vehicles on the rail tracks due to the strict health and safety regulations. Permission to use vehicles, such as small trolleys, is difficult to obtain so, when designing a system for acquiring images, the use of vehicles should avoided.

3.1.4 **Remarks**

Generally, man-made tunnels are circular, box-, horseshoe- or oval/egg-shaped (Federal Highway Administration and Federal Transit Administration, 2005). The datasets shown in this thesis are circular and horseshoe-shaped, which are the main types of tunnel found in the countries visited during the research period. Due to the time constraint for the PhD, and the site availability for the data collection, it was impossible to collect images for other types of tunnel, although this would have been beneficial for the validation of the research results.

The tunnel liners in all of the datasets presented in this thesis are segmental, made of either concrete or cast iron. Although these types may not represent all types of lining, these are the ones that require monitoring for ageing structural components. It would be beneficial to obtain other types of lining, as they have different surface textures, although it was impossible to obtain more datasets, as only access to the sites described in this thesis was permitted.

3.2 **Wide-baseline matching datasets**

This section describes the datasets which were created to evaluate the performance of the proposed algorithms in Chapter 6. There are three datasets: (i) tunnel datasets, (ii)
Mikolajczyk datasets (Mikolajczyk and Schmid, 2005) and (iii) beam datasets.

3.2.1 Tunnel datasets

To evaluate the performance of the proposed feature matching algorithms in Chapter 6, new datasets are created from the tunnel images created in Section 3.1. Three different types of imaging condition are found in the tunnel images, including illumination change, narrow-baseline viewpoint change, and wide-baseline viewpoint change. For each type of imaging condition, two sets are created. The difference between the sets is that one contains few or no 3D objects, while the other contains 3D objects. Table 3.2 summarises all of the created datasets. Note that the images for each dataset can be related by either a homography $H^2$ or a fundamental matrix $F^3$.

3.2.1.1 Illumination

There are two datasets in this category; the Barcelona dataset–Figure 3.7(a), and the Prague dataset–Figure 3.7(b). To obtain the images, a camera is fixed to a tripod and the images are taken with and without a flash in order to create a change in lighting conditions. The self-timer function is used to release the camera shutter to avoid camera vibration.

3.2.1.2 Narrow-baseline viewpoint change

This type of change in imaging condition is most commonly found in the tunnel datasets. The change in viewing angle is small in these datasets; hence the term narrow-baseline. There are two datasets in this category: the Barcelona dataset–Figure 3.8(a) and the Prague dataset–Figure 3.8(b). The difference between these two datasets is that the images from Barcelona appear more perspectively-distorted than those in the Prague dataset. This is due to the difference between the scene geometries in the two datasets; the former is a semi-circular arch, while the latter is a cylinder. The images are obtained from a fixed camera, which is rotated and translated about the axis of rotation of the tripod.

---

$^2$ Homography transformation is used to relate images of a planar scene or those with little translation (see Chapter 4 for further details).

$^3$ Transformation based on a fundamental matrix is used to relate images of general scene types (see Chapter 4 for further details)
Table 3.2: A summary of the wide-baseline datasets from the tunnel images.

<table>
<thead>
<tr>
<th>Image Condition</th>
<th>Figure–Dataset</th>
<th>Transformation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Illumination</strong></td>
<td>Fig. 3.7(a)–Barcelona</td>
<td>H</td>
<td>Containing 3D objects</td>
</tr>
<tr>
<td></td>
<td>Fig. 3.7(b)–Prague</td>
<td>H</td>
<td>No 3D objects</td>
</tr>
<tr>
<td><strong>Narrow-baseline</strong></td>
<td>Fig. 3.8(a)–Barcelona</td>
<td>H or F</td>
<td>Containing 3D objects</td>
</tr>
<tr>
<td></td>
<td>Fig. 3.8(b)–Prague</td>
<td>H or F</td>
<td>Few or no 3D objects</td>
</tr>
<tr>
<td><strong>Wide-baseline</strong></td>
<td>Fig. 3.9(a)–Bond Street</td>
<td>F</td>
<td>No 3D objects</td>
</tr>
<tr>
<td></td>
<td>Fig. 3.9(b)–Bond Street</td>
<td>F</td>
<td>Containing 3D objects</td>
</tr>
</tbody>
</table>

3.2.1.3 Wide-baseline viewpoint change

There are two datasets, both from the Bond Street datasets (see Figure 3.9). The change in viewing angle is much larger than the narrow-baseline due to the larger degree of translation.

3.2.2 Mikolajczyk datasets

Eight datasets were created by Mikolajczyk and Schmid (2005), downloadable from the Visual Geometry Group (2011). These datasets are commonly used to evaluate the performance of detectors, descriptors or matching algorithms. The images in these datasets are either planar scenes or the camera position is fixed during their acquisition. Therefore, these images can be related by homography. The homography matrices, provided with the datasets, are used in mapping related images so that ground truth data can be compared with the experimental results.

There are five different changes in imaging conditions, that are applied to different scene types to ensure that the effect of changing the image conditions is separated from that of changing the scene type. Examples of images from each dataset are shown in Figure 3.10. The changes in imaging conditions are viewpoint changes, scale and rotation, image blur, JPEG compression, and illumination. As for the scene type, one contains homogeneous regions with distinctive edge boundaries, Type A; and the other contains repeated textures of different forms, Type B. Table 3.3 summarises the imaging conditions and scene types of the datasets.
Figure 3.7: Illumination datasets

(a) Barcelona  (b) Prague

Figure 3.8: Narrow-baseline viewpoint datasets

(a) Barcelona  (b) Prague

Figure 3.9: Wide-baseline viewpoint datasets

(a) type A  (b) type B
Figure 3.10: Examples of the datasets from Mikolajczyk and Schmid (2005).
Table 3.3: A summary of the imaging conditions and scene types from Mikolajczyk and Schmid (2005).

<table>
<thead>
<tr>
<th>Scene Types</th>
<th>Bark</th>
<th>Bikes</th>
<th>Boat</th>
<th>Graff</th>
<th>Light</th>
<th>Trees</th>
<th>Ubc</th>
<th>Wall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Viewpoint Changes</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Scale and rotation</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Image Blur</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>JPEG compression</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Illumination</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
</tbody>
</table>

3.2.3  

Beam images

Since the images of the scenes from Sections 3.2.1 and 3.2.2 are static, a dataset from the beam experiments is created here to include deformability in the evaluation of the feature matching algorithms developed in Chapter 6. It is important to evaluate the matching algorithms against deformability, since deformability can significantly affect the performance of the matching algorithms. This beam dataset is created from the beam experiments, the camera and experiment setups for which are subsequently explained in Section 3.3. Examples of images from the datasets are shown later in Chapter 6 (section 6.4.4.1).

3.3  

Change detection datasets

This section describes the datasets which are used in the evaluation of the performance of the proposed change detection algorithm, described in Chapter 7. These datasets are obtained from simulating crack propagation experiments in concrete beams. The experiment setups are described below.

3.3.1  

Experiment setup

Typically, changes due to anomalies in real tunnels take a long time, and may be less apparent. Therefore, it is impossible to use datasets from a real environment to evaluate the performance of the change detection system developed in this study, given the limited time available for a PhD programme. Hence, the datasets are created by simulating
changes in anomalies by crack propagation in concrete beams, obtained under a controlled environment, so that the true changes are known.

The experiments are designed to enforce the following assumptions regarding image datasets: (i) the in-plane motion of the beam, (ii) fixed camera positions and (iii) constant illumination. A three-point bending test was performed on a reinforced concrete beam, which was loaded at regular intervals until failure. To ensure the in-plane motion of the beam, two LVDT sensors were placed at the back surface of the beam to detect twisting or out-of-plane motion. Additionally, four LVDT sensors were installed at different locations to measure the vertical displacement of the top surface of the beam. Figure 3.11(a) shows an image of the setup with the LVDT sensors placed at the top surface of the beam, and Figure 3.11(b) shows an image of the setup during loading.

Three cameras were set up, as shown in Figure 3.12. To ensure fixed camera positions, the cameras were fixed to tripods. As shown in Figure 3.12(a), the left and right cameras were located at approximately 30° from the middle camera. As shown in Figure 3.12(b), different focal lengths of 50mm, 24mm and 18mm were set for the left, middle and right camera, respectively. The camera triggers were remotely controlled via a PC to avoid any contact with the cameras to ensure that the camera positions were fixed at all times. Three cameras were set to take synchronized photos at 5s intervals during loading. The camera was a Cannon DSLR 1000D in all three cases. Software called DSLR Remote Pro Multi-Camera was used to control the three cameras remotely. To ensure constant illumination, the flash on each camera was turned off and only external diffuse lights were used to ensure constant lighting conditions. No object was allowed to move when taking the images as this might have obstructed the external light sources and caused variations in lighting in the datasets.

Before loading the beam, a set of images of a Mylar card placed in front of the beam were taken for the purpose of camera calibration. The card contains an accurate grid of the targets, in which the distances in terms of object-space between the targets are known; hence, the pixel distances can be converted to real world distances.

The beam was then loaded at the rate of 0.5mm/min to allow sufficient time for the pictures to be taken. The three cameras took pictures every 5s, synchronously. Figure 3.11(b) shows an example of the images being captured. The beam was loaded and stopped at intervals, during which another set of multiple images were taken by another camera; this image dataset was used to construct a reference 3D model, as shown in Figure 3.11(c). The beam was loaded until failure.
Further details of the experiments, such as the beam design, can be found in Zhou (2011).

3.3.2 Image datasets

The dataset from each of the three cameras shows a series of images of a beam under load, and how the cracks propagate. The middle camera produces the frontal view of the beam from the closest distance, and hence crack propagation is highly visible in this dataset. The propagation can also be seen in the images from the other two cameras, although it may be less apparent. Example images are shown in Figure 6.14.

3.3.3 Remarks

The time taken for cracks or anomalies to evolve in images of real tunnels is slow, and it is impossible to rely on the natural evolution of cracks to obtain sufficient data for validating the crack change detection system. It is reasonable to use images obtained from a laboratory environment for the validation. However, the images from the beam tests shown in this section may not entirely emulate images of crack evolution in real tunnels. The beam images have constant illumination, which is not the case in a real environment. Therefore, the change detection algorithm is expected to perform better in the beam images than in the real tunnel images.

3.4 Summary

This chapter provides details about the datasets used in evaluations of the performance of the proposed system. There are three datasets: the tunnel datasets, wide-baseline datasets and change detection datasets. The tunnel datasets are used in the work presented in Chapters 5 and 6, the wide-baseline datasets are used in the work described in Chapter 6 and the change detection datasets are used in Chapter 7.
Figure 3.11: (a) an image of the setup showing the beam and the LVDT sensors, (b) an image showing the beam under load, and (c) a 3D reconstruction model of the beam.
Figure 3.12: The setup of the cameras for obtaining image datasets of crack propagation on a three-point bending test of a concrete beam.
Chapter IV

System Background

In this chapter, an overview of the proposed inspection system is described and the results from the preliminary experiments are presented and discussed. The system is divided into three separate modules: registration, reconstruction and recognition. In section 4.1, an outline of the proposed system is explained. Section 4.2 provides the background of a registration system, and the results of the preliminary experiments from state-of-the-art registration software. Section 4.3 presents the background to a reconstruction system and shows the results from the experiments. The final section discusses the results and the subsequent research focus based on the obtained experimental results.

4.1 System outline

Figure 4.1 presents an outline of the proposed system. The first component is image acquisition (see Chapter 3, Section 3.1). Following the image acquisition step, the system can be subdivided into three modules as follows.

Registration system One application of registration is image mosaicing. The aim is to stitch images together to provide a wider view of a scene that a single image cannot capture. The registration module in the proposed system is composed of a number of components, including Structure From Motion (SFM), surface estimation with a support vector machine, and warping and mosaicing, the details of which are explained later in Chapter 5. However, in this chapter, commercial mosaicing software packages are used to process the tunnel datasets (see Chapter 3) in order to explore their pros and cons, and allow us to develop further improvements for the registration process in Chapter 5.

Reconstruction system Broadly, the term reconstruction here refers to using passive methods, in which the light sources are not controlled, unlike active methods (e.g. time-
of-flight, shape-from-shading, structure light, active stereo and photometric stereo (Moons et al., 2009)). The reconstruction module in the proposed system is based on *Bundler*, an SFM program by Snavely et al. (2006). This chapter explains the fundamentals of SFM systems and then discusses the preliminary results obtained from the SFM system used in this thesis. Feature matching, which is one of the main components in reconstruction systems, has been studied in this thesis and is presented later in Chapter 6.

**Recognition system**  This system involves *change detection* and accurate *registration*, as shown in Figure 4.1, and will be explained later in Chapter 7.

### 4.2 Registration system

Image registration is the process of transforming image sets from varying and unknown co-ordinate systems into a single system. A well-known application is panoramic stitching, which is a popular technology embedded in many modern digital cameras. It allows users to produce a wide-angle panoramic mosaic of a scene from a number of images that are overlapped. Image registration is also applied in other areas of study, such as medical imaging and remote sensing (Zitová and Flusser, 2003). To create a stitched image, *image alignment* and *image stitching* algorithms are applied. The former algorithm recovers
the correspondence relationships among images with varying degrees of overlap. Image stitching algorithms take the alignment estimates from registration algorithms and blend them to create a seamless mosaic, in which visible seams usually arise from blurring, ghosting caused by parallax and scene movement, and varying image exposures (Szeliski, 2006). The pipeline of most registration algorithms is shown in Algorithm 4.1 and Figure 4.2.

**Algorithm 4.1** The pipeline of a typical registration system based on the feature-based alignment approach

**Input** A set of images

1. Detecting features in all images
2. Matching features to find initial correspondences in all image pairs
3. Verification of correspondences and estimation of transformation for each image pair
   (a) (optional) Select m candidate pairs of images, which have the most number of correspondences
   (b) Find geometrically consistent correspondences using RANSAC to solve the homography between pairs of images
   (c) (optional) Maximum likelihood (ML) estimation of homography
   (d) (optional) Re-matching
4. Find connected components of correspondences across all image pairs
5. Global registration by bundle adjustment
6. Compositing images to a panorama

**Output** A panoramic image
arise. Physically corresponding features can be dissimilar due to the different imaging conditions and/or due to the different spectral sensitivity of the sensors. The choice of the feature description and similarity measure has to consider these factors. The feature descriptors should be invariant to the assumed degradations. Simultaneously, they have to be discriminable enough to be able to distinguish among different features as well as sufficiently stable so as not to be influenced by slight unexpected feature variations and noise. The matching algorithm in the space of invariants should be robust and efficient. Single features without corresponding counterparts in the other image should not affect its performance.

The type of the mapping functions should be chosen according to the a priori known information about the acquisition process and expected image degradations. If no a priori information is available, the model should be flexible and general enough to handle all possible degradations which might appear. The accuracy of the feature detection method, the reliability of feature correspondence estimation, and the acceptable approximation error need to be considered too. Moreover, the decision about which differences between images have to be removed by registration has to be done. It is desirable not to remove the differences we are searching for if the aim is a change detection. This issue is very important and extremely difficult.

Finally, the choice of the appropriate type of resampling technique depends on the trade-off between the demanded accuracy of the interpolation and the computational

Figure 4.2: Four steps of feature-based image registration and mosaicing: top row—feature detection (e.g. corners). Middle row—feature matching by invariant descriptors, such as the SIFT descriptors (the corresponding pairs are marked by numbers). Bottom left—the estimation of a transform model exploiting the established correspondence. Bottom right—image resampling and transformation using an appropriate interpolation technique. The figure and modified caption are taken from Zitová and Flusser (2003).
4.2.1 Image alignment

4.2.1.1 Motion models

Before images can be registered and aligned, the mathematical relationship that maps the pixel coordinates from one image to another must be chosen. A variety of parametric models exist for this purpose, from simple 2D transformation to planar perspective models, 3D camera rotations, lens distortions and mapping to non-planar (e.g. cylindrical coordinates). Two models, which are used in the experiments in this study, are explained.

Planar projective transformation A homography matrix $H$ is a direct mapping between points in the image planes. If all points in an image lie in a plane, the image can be rectified directly without needing to recover and manipulate the 3D coordinates. Hence, two images can be related by a homography matrix $H$ when they are viewing the same plane from a different angle, or when both are taken from the same camera but from a different angle (i.e. the camera is rotated about its centre of projection without any translation). A pixel coordinate $x_1$ from one image can be related to the pixel $x_2$ in another image by a homography matrix using the following equation.

$$x_2 = Hx_1$$ (4.1)

The above homography is simplified in the case when a camera undergoes pure rotation (i.e. no translation), which is equivalent to assuming that all points are very far from the camera (i.e. on the plane at infinity). In such cases, the homography matrix can be simplified to

$$H = K_2R_2R_1^{-1}K_1^{-1}$$ (4.2)

where $K_i$ and $R_i$ are a camera intrinsic matrix and a rotation matrix for camera $i$-th, respectively. This type of panorama can be referred to as a rotational panorama, which is a preferred method of commercial stitching software. The advantage of a rotational panorama is that it simplifies the general 8-parameter homography to 3-,4-,5- parameter motion models corresponding to the cases where the focal length $f$ is known, unknown and
fixed, or unknown and variable. For large-scale image stitching algorithms, estimating 3D rotation matrices associated with each image is intrinsically more stable than estimating full 8-degree-of-freedom homography (Szeliski, 2006).

**Cylindrical coordinates** Instead of computing homography matrices to relate images, the images can first be warped into cylindrical coordinates, and pure translational models are then applied to align images. This only works if the images are all taken with a level camera or with a known tilt angle. Assuming that the rotation matrix of a camera is the identity matrix, $R = I$, the image coordinates are projected onto a cylindrical surface of unit radius, in which they are related by the following equation

$$
x' = s\theta = s\tan^{-1} \frac{x}{f} \tag{4.3}
$$

$$
y' = sh = s\frac{y}{\sqrt{x^2 + y^2}} \tag{4.4}
$$

where $s$ is an arbitrary scaling factor (sometimes called the radius of the cylinder), $x, y$ are the image coordinates before warping, $x', y'$ are warped coordinates, $f$ is the focal length of a camera, and $h$ and $\theta$ are the height and theta in the cylindrical coordinates, as shown in Figure 4.3(left). Similarly, image coordinates can also be related using the spherical coordinates, which are parameterised by two angles, $\theta$ and $\varphi$, as shown in Figure 4.3(right) (Szeliski, 2006).
**4.2.1.2 Direct and feature-based alignment**

The approaches to estimating the parameters of a chosen motion model, as explained above, can be based on one of two methods: direct (pixel-based) or feature-based methods.

**Direct alignment** This approach performs pixel-to-pixel matching. The methods in direct alignment can be grouped into three themes: the Correlation-like, Fourier and Mutual information methods (Zitová and Flusser, 2003). Here, the correlation-like methods are explained, and the reader is referred to Zitová and Flusser (2003) for a survey of the other two methods. In correlation-like methods, a normalised cross-correlation (NCC) algorithm and its modifications are the classical methods employed. The formulation of the algorithm is based on choosing the alignment which maximises the normalised product of two images, i.e.

\[ E_{\text{NCC}}(u) = \frac{\sum_i [I_0(x_i) - \bar{I}_0][I_1(x_i + u) - \bar{I}_1]}{\sqrt{\sum_i [I_0(x_i) - \bar{I}_0]^2[I_1(x_i + u) - \bar{I}_1]^2}} \]  

(4.5)

where \( \bar{I}_0 = \frac{1}{N} \sum_i I_0(x_i) \) and \( \bar{I}_1 = \frac{1}{N} \sum_i I_1(x_i + u) \), which are the mean intensities of the images of patches in the two images and \( N \) is the number of pixels in the patch. If sub-pixel accuracy is demanded, interpolation techniques are applied. The NCC method can exactly align images with only translation, although images with more complicated deformations, such as rotation and scaling, can also be handled by modifying the NCC methods. Other variants of correlation-like methods exist, such as the sequential similarity detection algorithm (SSDA) (Barnea and Silverman, 1972), and correlation-ratio based methods (Roche et al., 1998). The correlation-like methods have two drawbacks, which are the flatness of the similarity measure maxima and high computational complexity (Zitová and Flusser, 2003). The flatness of the maxima occurs when images are too similar (e.g. images with repetitive texture), and a clear peak may be unobtainable.

**Feature-based alignment** Generally, there are four steps in feature-based alignment for finding the correct matches between images, which are feature detection, feature descriptor, feature matching and robust matching estimation. In feature detection, the interesting or unique features from an image are extracted. These unique features include corners, blobs, edges, or local image patches with particular properties (e.g. an area
of uniform intensity). A feature has an image pattern which differs from its immediate
eighbourhood; this allows the matching of similar features in other images. For exam-
ple, a corner corresponds to a point in the 2D image with high curvature, and can be
found at various types of junction. A blob is a region in an image that differs in terms
of properties, like brightness or colour, compared to its neighbourhood. Different types
of feature have different properties, and choosing which features to use depends on the
application. Some feature detection algorithms produce features that have a high degree
of invariance, but they may not be computationally efficient. Currently, choosing the type
of features for applications still remains an active area of research (see Tuytelaars and
Mikolajczyk (2008) for further discussions). Once features are detected, feature descrip-
tors are applied to create descriptor vectors to represent the appearances of the features.
The feature descriptors are matched by feature matching algorithms to establish putative
correspondences, which are then filtered by robust matching algorithms. The reader is
referred to Chapter 5 for more details on all of the steps involved in finding matches
between images.

The robust estimation module takes initial correspondences to estimate the motion
parameters that best register two images. The initial correspondences generally con-
tain outliers and a robust estimator algorithm, such as RANdom SAmple Consensus
(RANSAC) algorithms (Fischler and Bolles, 1981), is applied to remove the outliers. The
RANSAC algorithms are robust even in data that contain a large number of outliers,
and are used in practice. After RANSAC, the Maximum Likelihood Estimation (MLE)
of homography $\hat{H}$ can be applied using the filtered correspondences to obtain the best
set of parameters. The MLE solves a set of subsidiary points $\{\hat{x}_i\}$, which minimises the
following cost function

$$\sum_i d(x_i, \hat{x}_i)^2 + d(x'_i, \hat{x}'_i)^2$$

where $\hat{x}'_i = \hat{H}\hat{x}_i$. The above equation is called the Gold Standard algorithm (Hartley
and Zisserman, 2000), which minimises reprojection errors. Some motion models, such as
translation, similarity, and affine, have a linear relationship between the motion and the
unknown parameters, and linear algorithms, such as the Direct Linear Transformation
(DLT)\(^1\) algorithm, can be used (Szeliski, 2006).

### 4.2.1.3 Global registration

The aim of global registration is to find a *globally consistent* set of alignment parameters that minimises the mis-registration between all pairs of images. Mis-registration errors quickly accumulate in sequential registration when images are added to the mosaic one by one. A global registration algorithm, such as a bundle adjustment algorithm, simultaneously adjusts all image rotations and computes new estimated focal lengths (Shum and Szeliski, 2000).

Once global alignment is performed, *local adjustments*, such as *parallax removal*, often need to be applied to reduce blurring between images due to local mis-registration causing blur and ghosting in a stitched image (Szeliski, 2006). Local mis-registration is caused by a variety of factors, including unmodelled radial distortions and 3D parallax due to a failure to rotate the camera around its optical centre. The approaches to alleviating these problems differ. Radial distortion can be removed during a pre-processing stage. The parallax problem may be alleviated by estimating a full 3D structure of a scene, or by averaging the intensities from all associated pixels (Shum and Szeliski, 2000).

If an unordered set of images is given, *panorama recognition* can be performed to cluster similar images to speed up the stitching algorithms.

### 4.2.2 Image stitching

This stage determines how to produce a final stitched image. The techniques involved include *compositing surface* parameterisation, *pixel/seam selection*, *blending* and *exposure compensation*. Compositing surface selection is explained in detail in 4.2.2.1. Pixel/seam selection determines how to blend pixels to minimise visible seams (due to exposure differences), blurring (due to mis-registration), or ghosting (due to moving objects). This process involves both deciding which pixels to use and how to weight or blend them. The simplest technique is to take an *average* value at each pixel, i.e.

\[
C(x) = \frac{\sum_k w_k(x)I_k(x)}{\sum_k w_k(x)}
\]  

(4.7)

\(^1\) This algorithm transforms ordinary linear equations, \(y_k = a_m x_k\) into a matrix equation \(Y = AX\), where \(Y\) and \(X\) are known variables. Hence, matrix \(A\) containing unknown parameters can be solved by the Singular Value Decomposition (SVD) algorithm.
where $\tilde{I}_k(x)$ are the re-sampled images and $w_k(x)$ is the weight, 1 at valid pixels and 0 elsewhere (Szeliski, 2010). Blending and exposure compensation blend the images to compensate for exposure differences and other mis-alignments.

4.2.2.1 Choosing a compositing surface

This stage aims to determine how to represent the final image. For a small number of images, a simple approach is to warp all of the images into the coordinate system of a chosen reference image. However, for a large number of images, this approach results in a severely distorted panorama, especially near the border of the image, where the pixels are excessively stretched. The alternative approach is to warp images using a cylindrical or spherical coordinate projection, as explained in Section 4.2.1.1. The choice of the surface is application-dependent and involves a trade-off between keeping the local appearance undistorted and providing a uniform sampling of the environment. Once the surface is selected, other issues for consideration include, view selection (selecting which part of the scene will be centred), coordinate transformation (computing the mappings between the input and output pixel coordinates), and sampling (if the panorama has a lower resolution than the input images, pre-filtering can prevent aliasing) (Szeliski, 2010).

4.2.3 Experiments

4.2.3.1 Homography-based mosaic

In this experiment, images from the Prague datasets (see Chapter 3) are input into commercial panorama software called Microsoft Image Composite Editor (ICE) (Microsoft, 2011a). Figure 4.4(a) shows the result of a mosaic using a planar motion model. The mosaic is constructed from 79 out of 89 images. Some images are not used by the software, which may be removed during the global registration stage. It can be seen that there are many areas in the image with misalignment, where the mosaic appears to diverge more towards the edge, as labelled in the figure.

4.2.3.2 Cylindrical projection mosaic

In this experiment, the images from the Prague datasets are stitched using a cylindrical projection motion model. The mosaic image is created from 88 out of 89 images, and the
unused image may be removed during the global registration stage. The mosaic spans 134.1° radially and 63.0° longitudinally. Figure 4.4(b) shows the result of the mosaic. Similar to the mosaic shown in Figure 4.4(a), there are many areas with misalignment. Straight lines appear curved and are more distorted towards the edge of the mosaic, as labelled in the figure.

4.2.4 Discussion

At the time of writing, the prominent state-of-the-art commercial image stitching software available are Autopano (Kolor, 2011) and ICE. The experiments are only conducted on ICE, since it is freely available. In principle, the two software packages are very similar. Currently, both can operate based on the cylindrical and spherical projection models, and the planar projective model. They are automatic and also equipped with manual tools to allow the making of further refinements to a panorama. The automatic process is mainly made possible by a better feature matching algorithm, such as SIFT keypoint matching. This is the opposite of traditional image stitching software packages, in which the control points must be manually specified by the users. The global registration in these programs, as explained in Section 4.2.1.3, also helps to improve the quality of a mosaic by minimising registration errors. Additionally, these programs are good at compositing images to create a seamless mosaic by robust blending schemes.

However, one of the limitations in the software is the motion model. Images are related by a planar projective model. Therefore, the mosaic is best when the scene of interest is planar. However, this model is extended to mosaic a general scene by estimating the homographies using only camera rotations, as shown in Section 4.2.1.1. This restricts the movement of the camera to simply rotating at a fixed point. The result of this camera movement is that one panoramic strip is created from one sweep of the camera movement. Stitching multiple panoramic strips results in distortion in the final stitched image. The images of tunnels from the datasets are obtained by multiple sweeps of the camera as it is moved along the tunnels. The poor quality of the resulting mosaics is observed in Figure 4.4(a) and (b), because the software tries to stitch multiple strips together.

Special equipment can be used to obtain images with restricted camera movements to allow the stitching of multiple panoramic strips. An example of this is Panogear, as shown in Figure 4.5(a), which is a motorised head on a tripod that allows a camera to rotate in a full 360°. The head allows multiple sweeps of images to be taken to create a
Figure 4.4: Mosaic images obtained from Microsoft Image Composite Editor (ICE), (a) the result of a planar motion mosaic, and (b) the result of a rotating motion mosaic.
set of images, as shown in Figure 4.5(b). Because the camera centre is fixed at one point, all images can be related by homographies and a larger panorama can be created. This type of mosaicing is applied in Jahanshahi et al. (2011). However, this method does not allow image stitching when the centre of the camera has been translated to a different location, as in the datasets in this study, because the images can no longer be related by homographies.

Mosaicing by a cylindrical projection model for the tunnel datasets also poses some limitations because the model relies on a camera rotating at a fixed point and requires the camera to be levelled in order to create good results. This model cannot be applied to a set of images obtained from a camera with translations. These restrictions do not apply to the tunnel datasets and, therefore, a poor quality mosaic in Figure 4.4(b) is produced.

With the advancement of reconstruction systems, such as Structure from Motion algorithms, camera parameters and a 3D model can be recovered from photographs effectively. The SFM system offers the possibility of creating a mosaic image that allows unconstrained camera movement. This is similar to the work of Agarwala et al. (2006), where a street-view mosaic of architectural buildings is created using an SFM system. The background and preliminary experiments of an SFM system are explained in the following sections.
4.3 Reconstruction system

A reconstruction system aims to recover the 3D model of a scene from images or videos. The simplest reconstruction systems return a set of sparse point clouds. More sophisticated reconstruction systems are able to produce a watertight 3D surface model. Examples of applications of reconstruction systems are the reconstruction of archaeological sites (Moons et al., 2009) and museum collections (Vogiatzis et al., 2005).

4.3.1 Previous work

Structure from Motion (SFM) is a system that can simultaneously recover a 3D point cloud model and camera positions using only images (Hartley and Zisserman, 2000). Some SFM systems have recently been commercialised, such as Photosynth (Microsoft, 2011b), a software application based on SFM developed by Microsoft Live Labs and the University of Washington. The software enables users to create a 3D point cloud model from uploaded photographs and allows users to browse and navigate through them.

The pipelines of reconstruction systems are similar, and the stages involved in the pipelines are shown in Figure 4.6. From this figure, the systems can loosely be divided into two groups: those with fully calibrated cameras and those with uncalibrated ones.

4.3.1.1 Fully calibrated cameras

If the intrinsic parameters of a camera are known, it is possible to estimate directly the extrinsic parameters (e.g. rotation and translation) and a metric 3D model. Many systems prefer this approach because the process of camera calibration is simpler and, also, the initial estimation of a 3D model is reasonably accurate. Snavely et al. (2006) create an interactive system that allows the browsing and exploring of large unstructured collections of photographs using a novel 3D interface. The system consists of an image-based modelling front-end that automatically computes the viewpoint of each photograph and a sparse 3D model from images. The intrinsic parameters for each photograph are estimated based on initialisation from the Exchangeable Image File Format (EXIF) tags of photographs. This system relies on a Bundle Adjustment algorithm to refine a 3D model and camera parameters.
4.3.1.2 Uncalibrated cameras

The intrinsic parameters of the cameras are unavailable in this case. Without these parameters, one can only hope to recover up to a projective 3D model. The projective model may be sufficient for some applications, such as in a robotic system, but is insufficient for visualisation purposes. Nevertheless, it is possible to upgrade a projective model to a metric one by a series of transformations. Hartley and Zisserman (2000) show how to upgrade from a projective model to an affine model and then to a metric model. The most popular technique for removing projective ambiguity is by self-calibration. Pollefeys et al. (2008) present a self-calibration system using linear self-calibration. Moons et al. (2009) present a system called ARC3D, which reconstructs a sparse 3D model from images through a web service. Robertson (2004) presents a method for camera calibration using vanishing points. This method requires a moderate amount of user interaction to specify the geometrical primitives, such as lines, points and planes, in the scene.

4.3.2 Structure from motion

A typical pipeline of an SFM system is shown in Algorithm 4.2. Stages 1, 2 and 3 are similar to the registration systems shown in Algorithm 4.1, except that fundamental matrices are used instead of homographies. The subsequent sections explain the algorithms for initialising the camera parameters and a 3D-point-cloud model, and the Bundle Adjustment algorithm for refining the parameters and the model.

4.3.2.1 Two-view geometry

Epipolar geometry A pair of cameras viewing the same 3D point are related by epipolar geometry. The projection of a 3D point from one image is restricted to the corresponding epipolar line in the other. The epipolar constraint can be formulated algebraically using the essential matrix $E$ (Nister, 2004) which relates corresponding points between two views as follows

$$\tilde{x}^T E \tilde{x}' = 0$$  \hspace{1cm} (4.8)

where $\tilde{x}$ and $\tilde{x}'$ are normalised coordinates from the first and second images, respectively. The essential matrix $E$ depends on rotation $R$ and translation $T$ between the image
Figure 4.6: The pipeline of the stages involved in reconstruction algorithms.
**Algorithm 4.2** A typical pipeline of a Structure from Motion (SFM) system

**Input** A set of images

1. Detecting features in all images
2. Matching features to find initial correspondences in all image pairs
3. Verification of correspondences and estimation of transformation for each image pair
   - (a) (optional) Select m candidate pairs of images, which have the most number of correspondences
   - (b) Find geometrically consistent correspondences using RANSAC to solve for the fundamental matrices between pairs of images
   - (c) (optional) Maximum likelihood (ML) estimation of the fundamental matrices
   - (d) (optional) Re-matching
4. Find connected components of correspondences across all image pairs
5. Initialisation by multi-view triangulation
6. Global registration by bundle adjustment

**Output** 3D point cloud
pair as $E \sim [T]_x R$, and defined up to scale; note that $[T]_x = \begin{bmatrix} 0 & -T_3 & T_2 \\ T_3 & 0 & -T_1 \\ -T_2 & T_1 & 0 \end{bmatrix}$. Thus, only 5 parameters are unknown. For non-normalised image coordinates, $x$ and $x'$, the corresponding points are related by the fundamental matrix $F$ as

$$x^T F x' = 0 \quad (4.9)$$

The fundamental matrix has 8 unknown parameters, defined up to scale. When intrinsic camera matrices, $K_1$ and $K_2$, from an image pair are known, the essential matrix is related to the fundamental matrix by $E = K_2^T F K_1$. However, the essential matrix is more stable than the fundamental matrix because it requires only 5 parameters to be solved.

**Triangulation** Once the essential matrix has been estimated from the image correspondences between an image pair, it can be used to construct projection matrices to recover the poses of cameras and 3D coordinates by *triangulation*. The camera poses, which are the rotation and translation matrices, can be estimated up to scale by decomposing the essential matrix (Wang and Tsui, 2000). For a pair of cameras, the first camera $P_1$ is set to align with the world coordinates and the second camera $P_2$ is obtained from the SVD solution of the essential matrix.\(^2\) Given the camera projection matrices, it is possible to recover a 3D coordinate $X$ by the intersection of the rays that are back-projected from two cameras. There are errors in the measured points $x$ and $x'$, and there will not be a point $X$ which exactly satisfies $x = PX$ and $x' = P'X$. Therefore, the 3D coordinate is recovered using the *least-squares* method. A simple algorithm, called the Direct Linear Transform (DLT) algorithm, can be used to recover the 3D coordinates as

\(^2\)Singular Value Decomposition (SVD) is a factorization of a real or complex matrix, $A = U \Sigma V^*$, where $U$ is an $m \times m$ real or complex unitary matrix, $\Sigma$ is an $m \times n$ diagonal matrix with non-negative real numbers on the diagonal, and $V^*$ (the conjugate transpose of $V$) is an $n \times n$ real or complex unitary matrix.
\[
\begin{bmatrix}
  x_1p_1^3T - p_1^1T \\
  y_1p_1^3T - p_1^2T \\
  x_2p_2^3T - p_2^1T \\
  y_2p_2^3T - p_2^2T
\end{bmatrix} \quad X = AX = 0
\] (4.10)

where \( p_i^kT \) denotes the \( k \)-th row vector of a matrix \( P_i \). The unknown 3D point \( X \) can be solved from the SVD solution of the \( A \) matrix. The optimal triangulation can be achieved by the Gold Standard algorithm, as shown in Hartley and Zisserman (2000).

4.3.2.2 Multi-view structure from motion

Given a minimum of two camera views and camera parameters, 3D coordinates can be recovered using the epipolar geometry and triangulation method. The following section explains the methods used to solve the structure from motion problem for an arbitrary number of views. The final stage in the structure from motion system is bundle adjustment, which is used iteratively to refine the structure and motion parameters via the minimisation of an appropriate cost function. However, bundle adjustment depends critically on a suitable initialisation; otherwise, the algorithm may converge to a local rather than global minimum of the cost function. The initialisation schemes to extend from two-view geometry to multi-view geometry are grouped into sequential, factorisation and constraint-based reconstruction algorithms (Robertson, 2004). The former is explained here and the reader is referred to Robertson (2004) for a review of the other two algorithms.

Sequential algorithms

Figure 4.7 shows a schematic diagram of sequential reconstruction algorithms. As each view is registered, a partial reconstruction is extended by reconstructing all of the 3D points that are visible in two or more views using intersection. The bundle adjustment algorithm may be applied after the new view has been registered. Robertson (2004) groups the strategies for registering successive views as epipolar constraints, resection-intersection and merging partial reconstructions.

Epipolar constraints This strategy is performed by relating each view to its predecessor, using the essential matrix when the intrinsic parameters are known. The essential
matrix can be decomposed to obtain relative rotation and translation matrices, and triangulation is performed to obtain the 3D coordinates. The magnitude of the translation has to be fixed, which can be done based on known 3D points that have already been reconstructed from images in earlier views. The camera parameters $R$ and $T$ are related to global coordinates by

$$R_{i+1} = R_{m}^{i \rightarrow i+1} R_i$$

(4.11)

where the first camera is set to align with the world frame, as $R_1 = I_{3 \times 3}$, $R_{m}^{i \rightarrow i+1}$ is the rotation of the $m$-th model between the images $i$ and $i+1$. Similarly,

$$T_{i+1} = T_i + s_m T_{m}^{i \rightarrow i+1}$$

(4.12)

where $s_m$ and $T_m$ are the scale factor and the translation vector of the model between images $i$ and $i+1$ and $T_1 = [0 \ 0 \ 0]^T$ (see Shragai et al. (2005) for details).

**Resection-Intersection** This strategy determines the pose of a new view by using already-reconstructed 3D points. Given 6 or more 3D to 2D correspondences, a 12-parameter linear projection matrix can be found up to scale by resection (i.e. pose estimation algorithms).
Merging partial reconstruction Two- or three-view reconstructions are obtained using adjacent image pairs or triplets and are then merged using corresponding 3D points. For longer sequences of images, Gherardi et al. (2010) apply a tree structure to reconstruct a small number of images and then merge progressively up the hierarchy of the trees.

4.3.2.3 Bundle adjustment

Bundle adjustment is usually applied in the final stage in SFM systems to refine the camera parameters (i.e. $K_j$, $R_j$, $T_j$) and reconstructed 3D points (i.e. $X_i$). Assuming that a point $X_i$ is seen by a camera $j$, its re-projection is written as $K_j[R_j T_j]X_i$, which is compared with its image measurement coordinate, $x_{ij}$ to obtain an error between them. Bundle adjustment works by minimising the sum of the weighted distance errors between all reprojected points and associated image measurement points, which can be expressed as (Triggs et al., 1999)

$$\min \sum_{ij} w_{ij} |K_j[R_j T_j]X_i - x_{ij}|^2$$

(4.13)

The above equation is non-linear, and a numerical optimisation algorithm can be used to minimise it. An example of the optimisation algorithms used are the Newton’s method, and its variants such as the Gauss-Newton method and the Levenberg-Marquardt (LM) algorithm. Sparse bundle adjustment algorithms are commonly used to solve the above equation due to their computational efficiency. These algorithms exploit the sparsity of matrices in the computation involved (Snavely et al., 2006).

4.3.3 Experiments

4.3.3.1 Bundler

Bundler, which is a non-commercial state-of-the-art SFM software package developed by Snavely et al. (2006), was applied to the Prague dataset to obtain a 3D model. The software successfully reconstructed the model with the statistics shown in Table 4.1. All 173 images are registered in the model. Note that the computational time in the table does not include the feature matching and preparation stages. Figure 4.8(a) shows the result of a 3D model from the Prague dataset and Figure 4.8(b) shows a close-up view of the camera poses. It can be seen that most of the 3D points as well as the camera poses
are reasonably accurate. The final mean reprojection error of this model is 0.4799 pixels, which is low and corresponds well to the observed figures. However, another view of the model, as shown in Figure 4.9, reveals that many points lie outside the geometry of the model, due to either incorrect tracks or incorrectly recovered camera projection matrices. The experiment is performed on an Intel Core i3 3.06GHz with 4GB of memory.

4.3.3.2 Patch-based multi-view stereo

The camera parameters obtained from Bundler in the previous section are input into the Patch-based Multi-view Stereo (PMVS) program implemented by Furukawa and Ponce (2009) to obtain a dense 3D model. All 173 images are used for the reconstruction. Due to a memory problem that exists in this program, reconstruction is only possible at a 16-th of the original image resolution. The reconstruction is performed for every patch size of 2x2. This results in increasing the number of 3D points to 189052, as shown in Figure 4.10. Visually, the model is reasonable, although a higher resolution would improve it.

4.3.3.3 Comparison with LIDAR

In this experiment, the result of the 3D model from PMVS in Section 4.3.3.2 is compared with the LiDAR data. Figure 4.12 shows a point cloud of the LiDAR data, and there are 3112699 points. The LiDAR points have two resolutions, the top section of the tunnel is 10x20mm, and other sections of the tunnel are 20x30mm. The iterative closest point (ICP) algorithm is used to align the PMVS model to the LiDAR point cloud, as shown in Figure 4.11. To match the resolution of the two data sets, the LiDAR points are uniformly sampled at every 8 points. Therefore, there are 389084 points from LiDAR to compare with 189052 points from PMVS. The resolution of the LiDAR data after 3The LiDAR points are sampled in rectangular grids with a distance between the points in the x-direction of 10mm and in the y-directions of 20mm. Hence, the resolution in the x-direction is 10mm and that in the y-direction 20mm.
Figure 4.8: (a) A 3D point cloud reconstructed from the Prague dataset and camera poses are shown, (b) a close-up view showing the recovered camera poses.
Figure 4.9: A zoomed-out view of the 3D model of the Prague tunnel. It can be seen that a number of 3D points lie outside the true geometry of the tunnel.
The ICP algorithm is iteratively applied to minimise the RMS error until the errors stop decreasing. Figure 4.13 shows the plot of the RMS error against the number of iterations. The final RMS error is 0.0380 of the resolution of the LiDAR data, which is equivalent to \( \sim 0.0380 \times 120 = 4.56 \text{mm} \) on the top section and \( \sim 0.0380 \times 200 = 7.6 \text{mm} \) on the other sections. Hence, the resolution of the 3D model from PMVS is \( 120 \text{mm} \pm 3.8\% \) on the top section and \( 200 \text{mm} \pm 3.8\% \) for the other sections.

4.3.4 Discussion

SFM systems offer the possibility of creating a mosaic with unrestricted camera movement. The state-of-the-art system, Bundler, can create a 3D model from photographs successfully. However, inaccuracy can be observed in the reconstructed 3D point cloud in the Prague dataset, which arises due to incorrect tracks or bad camera projection matrices. This conclusion is based on that of Snavely (2008), who states that Bundler can fail into 4 distinct modes (as will be explained in Chapter 6).

The comparison between the LiDAR data and the 3D model obtained from the PMVS program in Section 4.3.3.3 shows that the PMVS system cannot produce a resolution comparable to the LiDAR data due to computational limitations. The PMVS system
Figure 4.11: The points in blue are the PMVS data and the points in grey are the LiDAR data. Note that the LiDAR points have two resolutions: the top section of the tunnel is 15mm and the other section is 25mm.
Figure 4.12: A point cloud of the LiDAR from the Prague tunnel

Figure 4.13: The graph shows the RMS error of the PMVS registered onto the LiDAR data using the iterative closest point algorithm.
can produce a dense model by using only $\frac{1}{16}$ of the original image resolution although, potentially, the PMVS system may be capable of producing a denser model. The work by Leberl et al. (2010) demonstrate that 3D reconstruction systems can produce a model of comparable resolution to the LiDAR systems, so the 3D systems should still be considered as an option for creating a 3D model.

4.4 Summary

In this chapter, an outline of the proposed system is introduced. The system consists of three main modules: a reconstruction system, a registration system and a recognition system. The backgrounds of the reconstruction and registration systems are given in this chapter together with some preliminary experiments. The experiments are conducted to discover the limitations of the state-of-the-art systems, including the Microsoft Image Composite Editor (ICE) for the registration system, and Bundler and PMVS for the reconstruction systems.

From the experiments using the ICE software, it is found that the software capability is limited by the motion models. These models require restricted camera motion, which do not apply in most of the datasets in this thesis. One way to deal with free camera movements for mosaicing is to recover a 3D model and camera poses using an SFM system. Therefore, image mosaicing for the tunnel datasets is studied and improved by using the SFM system, as explained in Chapter 5.

The experiments on the reconstruction system show that, currently, the accuracy of a reconstructed 3D model is not comparable to the LiDAR data. The reconstruction system may be preferred due to its lower cost. However, the experiments show that some incorrect 3D points can be observed in SFM systems due to incorrect tracks. Therefore, Chapter 6 focuses on the development of a novel feature matching algorithm, which will ultimately improve the accuracy of the tracks for the SFM system.
Chapter V

Robust Surface Estimation for Mosaicing

5.1 Introduction

Panoramic images provide wide-angle visualisation of a scene, which cannot be achieved with a single image. The field of view of the human eye spans almost 180-degrees in the horizontal direction, and it is natural for humans to see wide-angle images of large scenes. For tunnel inspection, a single image does not give inspectors a spatial sense; hence, it is difficult to indicate the locations of any observed anomalies based on a single image. Typically, this problem is solved by a non-intuitive referencing system used in an inspection report to relate images to their location inside a tunnel. A more elegant solution is to stitch images together to increase the field of view, which then facilitates the localisation of the anomalies. There are, however, problems associated with the current commercial stitching software. Most software can only stitch images in one direction, which may prove less useful for tunnel inspection because the increase in the field of view is only marginal. Another problem is the severe perspective distortion caused by the limitations of the image alignment models employed in the commercial software. Usually, mosaics from commercial software are distorted and less intuitive for humans to visualise, as shown in Figure 5.1(top). In the tunnel datasets described in Chapter 3, commercial stitching software cannot produce a high quality mosaic because the datasets do not comply with the assumptions made for the software to work.

In this chapter, a system which constructs a mosaic image of the tunnel surface with little distortion is presented. The system can deal with general camera motion, unlike the existing mosaicing algorithms that have to constrain camera motion (i.e. a camera rotating around its optical axis) to achieve a high quality mosaic. The system is based on a Structure from Motion (SFM) system, which allows images to be stitched in two directions (i.e. radial and longitudinal), which increases the field of view in a mosaic significantly. A low level of distortion in a mosaic is achieved by exploiting the properties
of the developable surfaces that are assumed to be the geometry of tunnels. The mosaic results, as shown in Figure 5.1(bottom), are intuitive for human, as they preserve the physical attributes, such as line parallelism, collinearity, line straightness and angles.

The system relies on the accurate estimation of tunnel geometry to achieve a high quality mosaic. As shown in Figure 5.1(middle), the mosaic looks incorrect when an inaccurately estimated surface proxy is used. In the proposed system, a Support Vector Machine (SVM) classifier is used in order to estimate the surface proxy automatically.

5.1.1 Previous works

The visual presentation and quality of a panorama depend largely on the accuracy of the image alignment step (Chapter 4, Section 4.2.1), which in turn relies on an accurate motion model. There are many motion models, though they can be broadly grouped into single-viewpoint and multi-viewpoint. A single-viewpoint panorama is the most common type in the commercial stitching software. This type of panorama is created with the assumption that all images share the same centre of projection (Haenselmann et al., 2009). This can be achieved by rotating a camera around its optical centre (Agarwala et al., 2006). One model is based on planar projective transformation, in which all images are aligned using homographies. The final panorama is created by warping other images based on a homography to a chosen reference image. One drawback of this method is that the panorama appears to diverge towards the edge and only a small number of images can be combined without causing severe distortion (Peleg et al., 2000). To avoid this problem, one can first warp each image to cylindrical or spherical coordinates and then relate each image using a translational model (Chen and Klette, 1999). This model, however, distorts a panorama by making all of the straight lines appear curved. This model also requires a level camera for the cylindrical projection or a level camera with known tilt angles for the spherical projection. The drawback of the single-viewpoint panorama is that it does not work well with images with general camera motion.

A multi-viewpoint panoramic image consists of patches that do not have a common projection centre, but are taken from changing viewpoints (Haenselmann et al., 2009). Strip panoramas, which are created from a translating camera, are one of the methods used for creating a multi-viewpoint panorama. There are many variants of strip panoramas (Agarwala et al., 2006), such as pushbroom panoramas (Gupta and Hartley, 1997), adaptive manifolds (Peleg et al., 2000) and x-slits images (Zomet et al., 2003). The
strip panoramas are suitable for visualising images or videos with long sequences, such as long street scenes. A common method used with strip panoramas is to sample vertical strips from each image or video frame and then warp the strips according to the chosen transformation model in order to stitch the strips onto a final image plane. The strip panoramas exhibit some degree of distortion. Only objects at a certain depth from the camera plane are shown with a correct aspect ratio; objects that are further away may appear horizontally stretched while closer objects appear squashed. These panoramas are usually created from videos, which are generally of lower quality than images. In addition, acquiring suitable video data can prove challenging (Agarwala et al., 2006).

Carroll and Seitz (2009) create multi-viewpoint panoramas using developable surfaces, which can be unwrapped onto a plane without distortion. The result is a mosaic without any perspective distortion. The system estimates a camera pose for each frame and then, with known camera poses, an inverse projection of each frame can be performed onto a developable surface. This system, however, requires tracking pixels in the pose estimation algorithm; hence, the camera path has to be smoothed. Therefore, it may only be suitable for cameras with forward motion, as shown in Figure 5.2. Forward motion is unsuitable for scenes taken inside tunnels due to the large size of the tunnels, since only a small portion of the tunnel surface for each panel is visible in the images, as shown in Figure 5.2. This is inadequate for inspection purposes.

Agarwala et al. (2006) develop a system that creates multi-viewpoint panoramas from digital images taken from a standard SLR or a fish-eye lens camera. The system uses Structure From Motion, which explicitly recovers a sparse 3D point cloud of a scene and camera poses. The texture from each camera is projected onto a dominant plane, which corresponds to the facade of the buildings along the streets, to create the final panoramas. In this work, a dominant plane is chosen automatically by Principal Component Analysis (PCA). PCA is able to find a direction with the largest variance of data. In this work, PCA is applied to a 3D point cloud and the direction of the largest variance in the point cloud is used to form the parameters for a plane. The PCA method works when a street scene is not curved. If it is slightly curved, a dominant plane is selected manually. For a scene with a large curvature, such as a tunnel, a geometry proxy, such as a cylindrical surface, can be used as a dominant plane, as demonstrated later in this chapter. In Agarwala et al. (2006)”s work, when a distance between a scene and a camera is large, only a slight distortion in the mosaic is observed for objects that do not lie on a dominant plane, such as cars. Therefore, standard stitching and blending algorithms are sufficient
to cope with these objects. Figure 5.3 shows an example, in which a panorama appears as an orthographic projection perpendicular to the scene as a person walking along the street would see it in the real world. The author claims that the algorithm should work for long and tall scenes, although only the panoramas of the long scenes are demonstrated in the paper.

The method developed in this study is similar to Agarwala et al. (2006), which relies on an SFM system to create multi-viewpoint panoramas. The SFM system allows stitching for a general camera motion, which eases the process of image acquisition, and also enables images to be stitched in both the radial and longitudinal directions for tunnels. Since the tunnel surface is a developable surface, the panoramas can be distortion-free. The main challenge of the current work, also faced by Agarwala et al. (2006), is how to extract a geometry proxy for mosaicing. Agarwala et al. (2006) identify a dominant plane automatically by the use of PCA. In this thesis, the automatic extraction of the tunnel proxy is achieved by Support Vector Machine classification, as explained in Section 5.3, and the manual method is used when the automatic method fails, as explained in Section 5.2.1.

5.1.2 The proposed mosaicing system

Figure 1.3 presents an outline of the proposed system. The first component is image acquisition, as explained in Chapter 3 (Section 3.1). Then, the images are input to Bundler, implemented by Snavely et al. (2008), to obtain a sparse 3D point cloud and camera poses. The point cloud is used to estimate the geometry proxy of a tunnel surface. The point cloud contains 3D points that lie on and off the tunnel surface. Automatic and manual methods for estimating the surface are proposed in this study to ensure that an estimated surface is close enough to the true proxy of the tunnel surface to obtain a high quality mosaic. The automatic method is explained in Section 5.3, which is based on a Support Vector Machine (SVM) classifier. The manual method requires user input to estimate the surface, as explained in Section 5.2.1. In the final step, image warping and mosaicing, each image is warped based on the estimated surface and camera matrices, and then the warped images can be stitched together using standard stitching software.
Figure 5.1: An example of the results obtained from the Aldwych dataset: (Top) a typical result from commercial stitching software (e.g. Kolor (2011)), (Middle) the result obtained from the experiment when an incorrect surface proxy is used for mosaicing, (Bottom) the final result obtained from an accurate surface proxy.
Figure 5.2: An example of pictures of tunnels from the Aldwych site obtained by forward motion.
Figure 5.3: An example of a strip panorama of a street scene from Agarwala et al. (2006)
The final step is explained in Section 5.4.

5.2 Robust surface estimation

5.2.1 Circular tunnel

Let $C$ be a cylinder defined by a centre $(x_0, y_0, z_0)^T$, a unit directional vector $(a, b, c)^T$ and a radius $r$. A point $p \in \mathcal{R}^3$ that lies on the cylinder is then defined as

$$\|p - \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix}\|^2 = \left(\begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \cdot \begin{bmatrix} a \\ b \\ c \end{bmatrix}\right)^2 + r^2 \quad (5.1)$$

Let $x_i = (x_i, y_i, z_i)^T$ be a 3D coordinate, and the minimum distance (i.e. the shortest distance between a point to a surface) between the point $x$ and the cylinder $C$ is then defined as $d_i = \min ||x - p||$, where $p \in C$. To fit a cylinder to a point cloud $X = \{x_i, ..., x_n\}$, the following cost function is minimised

$$\min \sum_{i=1}^{n} w_i d_i^2 \quad (5.2)$$

where $w_i \in [0, 1]$ is the weight of a point $x_i$. The above cost function is minimised using the non-linear least squares method by the Gauss-Newton algorithm, in which the implementation from Eurometros (2011) is used.

It is assumed that the geometry of a tunnel can be modelled by a cylinder. This assumption is valid because the deformation of a tunnel is negligible in comparison with the size of the tunnel.

5.2.1.1 Automatic

Images obtained from tunnels usually contain protruding objects, such as power cables. The keypoints lying on these objects can distort the accuracy of a tunnel’s geometry. It is desirable to use only keypoints that lie on a tunnel surface so that an estimated surface is as close to the true tunnel geometry as possible.

In this thesis, the automatic classification of keypoints is achieved by applying a Support Vector Machine (SVM) classifier. This classifier is used to determine keypoints that
lie on the tunnel surface and are subsequently assigned non-zero weights in Equation 5.2. This approach is explained in Section 5.3.

5.2.1.2 Manual

Sometimes, SVM may fail to classify keypoints and subsequently lead to incorrect estimation of a tunnel surface geometry. User input is used to obtain the initialisation of the surface estimation in Equation 5.2. A user is required to specify an initial estimate of the cylinder parameters including a radius $r_0$, a directional vector $\mathbf{a}_0$ and a cylinder centre $\mathbf{x}_0$, and the algorithm then optimises over the initial estimates. A summary of the algorithm for the manual method is shown in Algorithm 5.1.

**Algorithm 5.1 A manual estimation of a cylindrical surface**

**Input** Three users inputs: radius $r_0$, centre $\mathbf{x}_0$, and direction of the cylinder $\mathbf{a}_0$

- While mean(distances) < threshold
  - Compute Gauss Newton algorithm to refine $r_0$, $\mathbf{x}_0$, and $\mathbf{a}_0$
  - Compute distances for each 3D point using the current $r_0$, $\mathbf{x}_0$, and $\mathbf{a}_0$

**Output** Optimised cylindrical surface parameters: $r_1$, $\mathbf{x}_1$ and $\mathbf{a}_1$

5.3 Automatic surface estimation with a Support Vector Machine

The 3D coordinates reconstructed from tunnel images are composed of two regions: points lying on the tunnel panels (i.e. surface points), and points not lying on the panels (i.e. off-surface points). Figure 5.8(left) shows an example of a typical image in which the points in red are surface points and the points in black are off-surface points. The image patches of the keypoints from these regions have a distinctive appearance, as shown in Figure 5.4. In this figure, the patches of the non-surface points or class 1 contain apparent edges, unlike those of class 2. Hence, these two classes can be separated based on the
Figure 5.4: An example of the two classes: Class 1–non-surface class and Class 2–surface class.

distinctive appearance of image patches, and the 3D point cloud can be classified into the surface region and the non-surface region, based on their 2D associated keypoints. The classified point cloud is then input into the algorithm in Section 5.2.1 to estimate the geometry of a tunnel surface.

5.3.1 Background to the Support Vector Machine

A Support Vector Machine (SVM) is a supervised learning method whose objective is to obtain the optimal decision boundary to separate data in a vector space. For binary classification, the optimal decision boundary is a boundary which is most distant from the vectors nearest to the boundary between the two sets of data. SVM can be applied to data that is linearly separable as well as data that is non-linearly separable by kernel methods (Bishop, 2006). The kernel methods convert a non-linear classification problem in a low-dimensional vector space to a linear problem in a high-dimensional feature space, as shown in Figure 5.5.

For binary classification, the training data is a set of pairs of training samples $T_{XY} = \{(x_1, y_1), ..., (x_l, y_l)\}$, where $x_i \in \mathbb{R}^n$ is the observation for the $i^{th}$ sample and $y_i \in \mathcal{Y} = \{-1, 1\}$ is the class label for the $i^{th}$ sample. The SVM classifier is the discriminant function mapping an input vector space $x_i$ into a class label $y_i$, $f : \mathcal{X} \subseteq \mathbb{R}^n \rightarrow \mathbb{R}$. The function can be written as

$$f(x) = \langle w \cdot x \rangle + b \quad (5.3)$$
where \( \mathbf{w} \) is the weight or the direction of the linear decision boundary, and \( b \) is an added bias. Note that the dot product is written as \( \langle \mathbf{w} \cdot \mathbf{x} \rangle \). Once the SVM classifier \( f(\mathbf{x}) \) is trained, it can then be used to assign a class label to a new input vector based on the following decision rule

\[
\hat{y} = \left\{ \begin{array}{ll}
1, & f(\mathbf{x}) \geq 0 \\
-1, & f(\mathbf{x}) \leq 0
\end{array} \right.
\]

The training objective of the SVM classifier is to place an optimal decision boundary, distinguished by the maximum margin of separation between any points on the decision boundary. The points that are closest to the decision boundary or \textit{support vectors} satisfy the following condition

\[
y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) = 1
\]

Subsequently, all data points that are correctly classified (i.e. the training samples) will satisfy the following constraint

\[
y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1
\]

for all of the training samples. As shown in Figure 5.6, the margin \( \gamma' \) is the distance from the decision boundary to the support vectors and the size of the margin is found as \( 1/\|\mathbf{w}\| \). The goal of training the SVM classifier is to maximise this margin. Hence, the following function is minimised

\[
\min_{\mathbf{w}, b} \tau(\mathbf{w}) = \frac{1}{2} \|\mathbf{w}\|^2
\]

subject to

\[
y_i(\langle \mathbf{w} \cdot \mathbf{x}_i \rangle + b) \geq 1
\]

for all inputs in the training samples \( i = 1, \ldots, m \). To optimise the above equation, a
positive Lagrange multiplier, $\alpha_i$, is applied to each of the $m$ constraints. Hence the above optimisation equation becomes

$$L(w, b, \alpha) = \frac{1}{2} ||w||^2 - \sum_{i=1}^{m} \alpha_i (y_i (\langle w \cdot x \rangle + b) - 1)$$  \hspace{1cm} (5.8)$$

The above equation leads to

$$\sum_{i=1}^{m} \alpha_i y_i = 0$$  \hspace{1cm} (5.9)$$

$$w = \sum_{i=1}^{m} \alpha_i y_i x_i$$  \hspace{1cm} (5.10)$$

Once the support vector classifier is trained to obtain the weight $w$ and the bias $b$, a new input vector $x$ can be classified using Equation 5.3. The function from Equation 5.3 can be rewritten by replacing the weight $w$ using Equation 5.10 as follows

$$f(x) = \sum_{i=1}^{m} y_i \alpha_i \langle x \cdot x_i \rangle + b$$  \hspace{1cm} (5.11)$$

The above equation indicates that classification is only a function of the support vectors. Furthermore, only dot products between the support vector $x_i$ and a new input vector $x$ are required for classification. These dot products can be replaced by a kernel function in order to map non-linear input data to a linear feature space (as shown in Figure 5.5) so that SVM can also be applied to non-linear data. Hence, Equation 5.11 becomes

$$f(x) = \sum_{i=1}^{m} y_i \alpha_i K(x, x_i) + b$$  \hspace{1cm} (5.12)$$

where $K(x, x_i)$ is a kernel function. The performance of SVM also depends on the choice of kernel function. So far, it is assumed the training data points are linearly separable in the feature space, i.e. once they are transformed by a kernel function. This is not always the case in practice, since there may be overlapping between the classes.
Therefore, SVM is modified in order to allow some training points to be misclassified. *Slack variables* $\xi$ are used to penalise misclassified training points. These slack variables can be 0 for points that are on or inside the correct margin boundary and greater than 0 for all other points (i.e. points that lie on the wrong side of the boundary). A positive constant $C$ is used to control the trade-off between the slack variables penalty and the margin. In other words, misclassification is allowed to the extent that the decision boundary to be found is still considered optimal.

### 5.3.2 The proposed approach

An SVM classifier is applied to discriminate the tunnel surface points from the non-surface points. Interest points from SIFT on or near the protruding regions are collected as the non-surface class and others as the surface class, as shown in Figure 5.8. The training dataset is composed of $\{x_i, y_i\}$, where $x_i$ is the 128-dimension SIFT descriptor vector and $y_i \in \{-1, +1\}$ the class label for an interest point $i$-th. The SVM classifier is trained and used for classification based on kernel methods. The kernel function used in this work is a Gaussian kernel as follows

$$K(x_i, x_j) = \exp\left(-\frac{\|x_i - x_j\|^2}{2\sigma^2}\right)$$

where $\sigma$ is the kernel variance. To select the best performance of the SVM classifier, choosing the kernel variance $\sigma$ and the penalty constant $C$ is important and is done using a validation set, as explained in Section 5.3.3.

Each SIFT interest point $x_{ji}$ from an image coordinate is related to a reconstructed 3D point by a projection matrix $P_j$ based on the following equation

$$X_i = P_j^{-1}G(x_{ji})$$

where $G$ represent a SIFT function, $i$ indicates an index of a 3D point, and $j$ indicates an index of an image. From Equation 5.14, the classification of 2D points automatically classifies the associated 3D points. Since a single 3D point is reconstructed from at least two cameras, the classification of a 3D point should come from an average result from all
Since (Cristianini and Taylor, 2005; Sapankevych and Sankar, 2009), the regression problem in low-dimensional space can be transformed to a linear problem in high-dimensional space.

Figure 5.5: An illustration of a Support Vector Machine, in which non-linear data can be transformed into linear data by a kernel function, taken from Dong et al. (2011).

Figure 5.6: A decision boundary is maximised by maximising the separation between a set of support vector data and a boundary, taken from Dong et al. (2011).
associated 2D points.

From Equation 5.12, the value \( f(x) \) can be converted to a probability of \( x \) belonging to a class given by the classifier. The probability \( p(x) \) is defined as

\[
p(x) = \frac{1}{1 + \exp(-f(x))}
\]  

(5.15)

From Equation 5.2, the weight \( w_i \) for each 3D point \( X_i \) is assigned as the sum of the probabilities of all associated 2D points \( x_{ji} \) as

\[
w_i = \sum_j p(x_{ji})
\]  

(5.16)

where \( w_i \in [0, 1] \). The sum of the probabilities is the likelihood of a 3D point \( X_i \) belonging to the tunnel surface. If the weight is high, then a 3D point is likely to belong to the surface. The conceptual diagram of the proposed algorithm is shown in Figure 5.7.

Note that the implementation of a Support Vector Machine in the experiments is obtained from Vojtěch and Václav (2004).

5.3.3 Experiments

5.3.3.1 Training and validating datasets

An experiment is carried out using the Aldwych dataset (see Chapter 3). There are 23 images in this dataset, 14 of which are labelled to separate the non-surface region from the surface region, as shown in Figure 5.8(a). Eight labelled images are used for training an SVM classifier, in which 1369 SIFT keypoints are collected as training samples, and the remaining 6 labelled images with 635 keypoints are used as validation samples. Samples of the keypoints are shown in Figure 5.8(b).

As shown in Figure 5.8(a), a labelling process is followed by colouring the regions that are not the surface panel; for example, tunnel ridges, cables, bolts, and joints. Then, the SIFT keypoint detection algorithm is run on the labelled images, in which the detected keypoints that lie on the coloured regions are flagged as the non-surface class keypoints, and all other keypoints are flagged as the surface class.
Figure 5.7: A conceptual diagram of the proposed algorithm using a Support Vector Machine
5.3.3.2 SVM parameter selection

There are two free parameters for an SVM classifier that have to be selected: the regularisation constant $C$ and the argument of the selected kernel function $k(x_a, x_b)$. In this work, the Radial Basis Function kernel (RBF), which is a 2D Gaussian kernel, is used; the argument of the kernel for the selection process is the variance $\sigma$. The constant $C$ is set with different values for classes 1 and 2, i.e. $C = [C_1, C_2]$. A common method for tuning the free parameters is cross-validation, in which the validation samples are used. The receiver operating characteristic or ROC curves of the classifier performance is plotted to select the best parameters from a pre-selected set $\Theta = \{(C_1, \sigma_1), \ldots, (C_k, \sigma_k)\}$, as shown in Figure 5.9(a). From the graph, the best performance is achieved when $\sigma = 0.5$ and $[C_1, C_2] = [2, 1]$. Hence, these values are used in this experiment for the Aldwych dataset.

5.3.3.3 Results

Figure 5.10(a) and (b) show the sparse 3D reconstruction of a tunnel with and without applying the Bundle adjustment (BA) algorithm, respectively. The tunnel linings are clearly seen after the BA algorithm is applied. The convergence graph from the BA algorithm as seen in Figure 5.9(b) quantitatively shows a significant improvement in global registration as the cost function converges.

Figure 5.10(c) shows the result of the 3D reconstructed points that are classified by SVM as marked in red for the surface point, and the non-surface points are marked in black. From Figure 5.10(d), the estimated surface, shown in yellow, is without the use of SVM (i.e. all points are used for the estimation), whereas the blue surface is estimated using the SVM classifier. From Figure 5.1, in the middle figure, the mosaic is created from the yellow surface (i.e. the surface without SVM), whereas the bottom figure is constructed from the blue surface (i.e. an accurate surface). It can be seen that the quality of the final mosaics depends largely on the accuracy of the estimated surfaces. From Figure 5.1(middle), the mosaic results in curvature, while the mosaic from the bottom figure preserves all physical geometries, such as line straightness, parallelism and a 90° angle between horizontal and vertical lines. The curvature is caused by the skewness of the estimated surface, which is induced by the non-surface points. In contrast
Figure 5.8: (a) an example of a labelled image, where the regions in red are protuberant regions or the non-surface class, (b) an example of the SIFT interest points, where the red points are surface points, and the black points are the non-surface points.

Figure 5.9: (a) The ROC curves of the SVM classifier used in the Aldwych dataset, (b) the curve showing decreasing reprojection error when the bundle adjustment algorithm is applied.
to the mosaic produced from the commercial software shown in Figure 5.1(top), which exhibits strong perspective distortion, the mosaic generated from this study is almost distortion free. This improvement in the mosaic quality is desirable and suitable for a tunnel inspection report.

5.4 Final composite

From Sections 5.2 and 5.3, a surface proxy is estimated to fit onto a 3D reconstructed point cloud. This proxy is a developable surface whose properties allow the surface to be flattened onto a plane without distortion, such as stretching or compressing. These special properties enable a mosaic from tunnel images to be created with little or no distortion.

In this section, the process of obtaining the final mosaic image is explained. Section 5.4.1 explains how each image is warped using the estimated surface from Section 5.2. The warped images can then be stitched together using standard panoramic stitching software, as explained in Section 5.4.2. The results and discussion are presented in Sections 5.4.3 and 5.4.4.

5.4.1 Image warping on developable surfaces

A developable surface is considered a special case of a ruled surface. A ruled surface is a surface that can be swept out by moving a line in space. Therefore, the ruled surface $R$ carries a one parameter family of straight lines $L$ (Peternell, 2004), known as generators or generating lines. Some familiar examples of ruled surfaces are a plane, a cylinder and a cone. Other examples include a hyperboloid and a helicoid. If all points on each generator, $x \in L$, have the same tangent plane, this surface is said to be a developable surface. In Euclidean space $\mathbb{R}^3$, there are three different basic classes of developable surfaces: cylinders, cones and tangent developable surfaces.

Given the constraints on the image collection process, cameras are located inside a developable surface and each ray intersects the surface at a single visible point only. The intersection defines, for each image, a one-to-one mapping between the image samples and the points on the surface. In other words, each image is projected onto the surface. This allows us to define a warping that produces flattened versions of the input images, as shown in Figure 5.11.
Figure 5.10: (a) a reconstructed model before applying the Bundle Adjustment algorithm, (b) the model after applying the BA algorithm, (c) classification of 3D points by SVM, (d) the estimation of the surface; the yellow surface represents the estimation, with all 3D points having equal weight $w_i$, and the blue surface is a more accurate estimation using the weights obtained from the SVM.
Figure 5.11: An example of input and warped images from the Bond Street dataset, (a) input images, (b) warped images
5.4.2 Compositing

In the final stage of mosaicing, it is necessary to decide how to produce the final stitched image. This involves different processes, including selecting a compositing surface, selecting pixel contributions to the final composite, and blending these pixels to minimise any visible seams, blur and ghosting.

Most commercial stitching software packages already contain the above algorithms for producing a final composite. However, these packages only work well with images related by planar projective transformation, and do not work well with the images as shown in Figure 5.12(top). However, in the proposed system, the input images are transformed using actual camera calibrations and a real surface, so they can be warped accurately. Once warped, the images can be modelled using a translational model and then stitched using a standard stitching algorithm. The warped images are mosaiced using ICE to obtain the final mosaics from the experiments.

5.4.3 Results

Figure 5.12 shows an example of the result from the Bond Street dataset. The top figure shows the results from ICE, in which a strong curvature can be observed. Quantifying distortion is a relatively new topic for a multi-viewpoint image, as discussed in Swaminathan et al. (2003), who introduced methods for evaluating image distortion. The degree of distortion is, therefore, not quantified in this research and only a qualitative evaluation is performed. Visually, the result from the proposed system does not contain perspective distortion, and a mosaic from the proposed system preserves all of the physical entities, such as parallelism, line straightness and a 90° angle between vertical and horizontal lines.

Figure 5.13 shows another example from the Bond Street dataset. This result demonstrates that the proposed system allows images to be stitched in the longitudinal direction. This means that the mosaic can be created to cover a much larger section of a tunnel. The ability to stitch in the longitudinal direction is impossible in standard stitching software. Misalignment can be observed near the rail tracks in this result. This is because the rail track lies far away from the true geometry of the tunnel, which violates the assumption made in the proposed system. The proposed system works well for the areas that are close to the tunnel geometry. The further the scene is away from the surface, the more violated the assumption becomes. As a result, greater inaccuracy can be observed in the
places that lie further away from the tunnel surface.

An example from the Prague dataset is shown in Figure 5.14. This result covers almost 8 metres of the tunnel surface in the longitudinal direction and approximately 270° in the radial direction. A small amount of misalignment can be observed in this result, as labelled in Figure 5.14, for the same reason explained above for the Bond Street dataset. Nevertheless, this result preserves all of the physical geometries and provides a larger field of view for the tunnel surface, which is desirable for tunnel inspection.

### 5.4.4 Discussion

The proposed system relies on the accuracy of an SFM system, especially for the estimations of the camera poses. For each image, an estimated camera pose and an estimated tunnel proxy are used to compute the warping of the input images. Hence, warping inaccuracy can arise from the incorrect estimation of the camera poses. The SFM system used in the proposed system is sequential, and as discussed by Snavely (2008) and Nister et al. (2004), error accumulation and propagation are common in the sequential methods. If a camera pose is inaccurate, it will then cause an incorrect estimation of a 3D point cloud. Such inaccuracy in the camera pose will be propagated to other camera poses, and the reconstruction system may fail altogether. The error in one camera will propagate through the sequence. This problem may be alleviated by loop-closing, which is used in Borrmann et al. (2008) to bound the registration error of the camera poses. However, loop-closing is not implemented during image acquisition for the tunnel datasets in this thesis, and reducing errors in the reconstructions is impossible using this solution. Another solution for improving an SFM system is by employing a better feature matching algorithm. Better algorithms can provide more accurate tracks and, hence, errors caused by incorrect matches can be alleviated, which will improve the overall accuracy of an SFM system. Novel feature matching algorithms are proposed in this study, as explained in Chapter 6.

In Section 5.3, the estimation of the geometry of a tunnel is improved by SVM. An SVM classifier classifies 3D points based on the appearance of associated 2D image patches. Generally, the performance of an SVM classifier can be measured by ROC
Figure 5.12: The results from the Bond Street dataset: (top) the result from the homography-based mosaic using ICE, (bottom) the result from the proposed system. While the parallel lines (tunnel ridges) curve along the horizontal axis of the image in the top image, the result from the proposed system (bottom) preserves all physical senses, e.g. line parallelism and straightness, which is important for tunnel inspection.

Figure 5.13: The mosaic, obtained from Bond Street Station, is created from 123 images and covers 7 metres in the longitudinal direction.
Figure 5.14: The mosaic is created from 173 images from Mustek Station, and covers 8 metres in the longitudinal direction.
curves (Veropoulos et al., 1999). The ROC curves in the experiments are shown in Figure 5.9, which are constructed from a validation dataset using different parameters. The parameters are chosen such that the False Positive rate and the False Negative rate are minimised. As can be seen from the curves, even in the best performance curve, the classifier will be unable to achieve an ideal curve, and the SVM classifier is fine-tuned to the Aldwych dataset. The classifier is trained using only images from the Aldwych dataset, and these images have a different appearance to the other tunnel datasets. Therefore, the classifier may not work well when applied to other datasets. This can be improved by using more datasets to train the classifier, although this lies beyond the scope of this thesis.

The results shown in this chapter are derived from the tunnel datasets, whose geometries are cylindrical. For other non-cylindrical datasets, such as the Barcelona dataset, a different geometry is required for the surface estimation process. This can be done by modifying Equation 5.1, such that other types of geometries can be estimated. For example, the Barcelona datasets, which are semi-circular in shape, can be estimated as a combination of planes and cylinders. These datasets require multiple types of geometry, which is a non-trivial problem, and further research is required. The work by Schnabel et al. (2007) is a good example of an algorithm that can be used to extract geometries from a point cloud automatically.

For the datasets with a cylindrical shape, the geometry is assumed to be a perfect cylinder. This assumption is not entirely correct as real tunnels are not perfectly cylindrical in shape due to the deformation of tunnels. However, the deformation can be assumed to be negligible in relation to the tunnel sizes in the mosaicing system presented in this thesis. The amount of deformation is insufficient to cause the effect of 3D parallax in mosaicing. Parallax can occur if the deformation is of the same magnitude as the distance from the tunnel wall to other objects, such as cables. Since the deformation is generally many times smaller than such a distance, the effect of parallax due to tunnel deformation can be ignored.

A cylindrical tunnel is assumed to be parameterised using only a uni-directional vector. This assumption may be true for a small section of a tunnel. However, for a longer section, a tunnel will be bent, and the assumption is no longer valid. Subsequently, it is necessary to obtain an actual directional vector for an entire length of a tunnel, which can be done by concatenating the shorter directional vectors, but this lies beyond the scope of the study.
Some degree of variation in lighting can be observed in the mosaic images presented in this study. The input images are obtained using a single directional external light source whose intensity is strongest at the point on which the light focuses. Obtaining images with the same degree of illumination is somewhat difficult in the tunnel environment. One solution to this problem is to use multiple light sources to illuminate an entire section of tunnel surface.

The mosaicing system presented in this chapter is validated for the tunnel datasets with a cylindrical shape. Cylindrical tunnels are commonly found in man-made tunnels although additional datasets are required to verify the system with other types of tunnel. The types of tunnel surfaces validated in this chapter are cast iron and concrete. These types of surfaces are commonly found, although additional datasets for other types of surface would provide a more complete validation. Note that more complete datasets for all types of tunnel shape and surface were not obtained since the datasets can only be obtained from the sites provided by participating research partners.

Another source of inaccuracy may be the commercial software used in the final stage of the proposed system. The image alignments computed by the software may be inaccurate due to the incorrect matching of images. Nevertheless, since the algorithms in the software are unknown, it is difficult to determine which stages in the software cause the inaccuracy.

5.5 Summary

A system for stitching images of a tunnel surface with little distortion is presented. The mosaic images produced from the proposed system enable a wider field of view for visualising tunnel surfaces, which eases the process of the localisation of observed anomalies on the surfaces, and allows an inspection report to be easily created.

The quality of the mosaics depends largely on the accuracy of the surface estimation of a tunnel. In this work, a Support Vector Machine is used for the automatic estimation of a tunnel proxy, while a manual method is also included. With the proposed mosaicing system, it is now possible to stitch images obtained from tunnel surfaces in both the radial and longitudinal directions with little distortion.

The mosaic images from the presented results have a number of advantages. One direct impact is the ease of creating an inspection report. The mosaics allow the inspectors to have a wider field of view of a tunnel surface, which makes it easier for them to carry
out inspections and analyses from the mosaics. The inspectors can use mosaic images to
guide the localisation of anomalies when creating an inspection report. The automatic
localisation of cracks or anomalies can also be performed, but this lies beyond the scope
of our study. Lastly, the mosaics can be compared with query images taken later. Any
changes between the mosaics and the query images can be identified and highlighted. This
aspect of the work is called \textit{change detection}, and is discussed in detail in Chapter 7.
Chapter VI
Spatially Consistent Feature Matching

6.1 Introduction

Feature matching is one of the most important and fundamental problems in computer vision. Its purpose is to match landmark features (e.g. corners, blobs and edges) in one image with the same features viewed in other images. It is usually the first module of many systems and it can determine the success or failure of the system. It is commonly applied in many areas of computer vision, including automatic 3D reconstruction, object recognition and registration. Considerable progress has been made towards improving the algorithms of feature matching so they can achieve desirable properties, such as scalability, accuracy, robustness and efficiency. However, the feature matching problem still remains an active area of research.

As illustrated in Figure 6.1, failure during the feature matching stage can lead to an unsuccessfully reconstructed model. In this figure, the 3D model of a tunnel, reconstructed from the Bond Street datasets, fails due to the ambiguity of the generated tracks—tracks are features that are matched and linked across multiple images and are formed by matching features in pairs of images. Snavely (2008) concludes that his reconstruction system (Bundler) can fail in 4 distinct modes: (1) insufficient overlap or texture; (2) ambiguous and repeating texture; (3) poor initialisation; and (4) cascading errors. An example of failure due to repeating textures is shown in Figure 6.2, in which Bundler is unable to distinguish between two similar sections of the building. Inevitably, texture repetition is common in buildings in the urban environment. An example of cascading error is shown in Figure 6.3(a). Cascading error refers to the propagation of reconstruction errors due to inaccurate camera positions or the uncertainty of recovered 3D coordinates due to the narrow baselines between cameras. A similar error is found in the 3D model obtained from the tunnel datasets in this thesis, as shown in Figure 6.3(b). Essentially, however, these different modes of failure are largely due to a failure during the feature matching
stage, which can be solved by improving the feature matching algorithm.

This chapter describes the methods intended to improve the feature matching stage, which will lead to the improved performance of the overall reconstruction system. This is achieved by applying a spatial consistency constraint in addition to matching based on an appearance similarity constraint. The spatial consistency constraint helps to reduce ambiguity in the matching features. The proposed methods are evaluated using widely-used datasets by Mikolajczyk and Schmid (2005) as well as the tunnel datasets created in this thesis (see Chapter 3, Section 3.2). The tunnel datasets are unique because they have repetitive textures in low brightness, which is common in the tunnel environment, whereas the Mikolajczyk datasets are obtained in an environment where lighting is not an issue.

This chapter is focused on the wide-baseline feature matching problem, which is applied only to images. Feature matching in video data requires a different approach to the method required for image data. This is because the viewpoint changes between the video frames are significantly smaller. Image data are chosen in preference to video data for the following reasons.

**Deployability** Nowadays, a high quality digital camera is inexpensive; hence, acquiring high quality image data is cost-effective. It does not require specialised equipment as the reconstruction algorithms that use images do not require the continuous tracking of features, unlike the algorithms using the video data. Hence, an image-based system can easily be deployed for tunnel inspection.

**Compatibility** An image-based system can be readily applied to existing images of tunnels, which have been collected as part of tunnel inspections over many years. The newly acquired images can be readily compared with the existing images for change detection (Chapter 7).

**Quality** In low lighting conditions, such as the underground environment, the quality of video is usually poor due to noise and motion blur. This can be mitigated more easily by using still images and flash bulbs.

In this chapter, it is demonstrated that the spatial consistency constraint can improve the performance of the feature matching algorithms. Two algorithms are proposed. The first is presented in Section 6.3, where any incorrect matches between images with small
viewpoint variations are removed using simple filtering schemes. The second algorithm, presented in Section 6.4, combines the Random Forest (RF) and Shape Context (SC) algorithms, which show promising results.

A review of the feature matching algorithms is included in Section 6.2. Sections 6.3 and 6.4 explain the methods of the proposed algorithms together with the results and discussion. Section 6.5 presents the conclusion based on the experiments.

6.2 Background and review of feature matching algorithms

Figure 6.4 shows the typical stages involved in a feature matching system. The input is a pair of images, each of which is processed separately. For each image, the Detector is first applied to identify local features that are distinctive, such as corners and blobs. The Descriptor then computes multi-dimensional descriptor vectors to describe the local appearance of the regions around the features. The descriptor vectors from the image pair are passed to the Feature Matching algorithm to select putative correspondences. These correspondences are further processed by a RANdom SAmping Consensus (RANSAC) algorithm to verify whether they are geometrically consistent according to a chosen geometrical transformation. The output from the image matching system is a set of features that are geometrically consistent. Advances have been made in each of the components in the feature matching algorithms, as reviewed below.

Feature Detectors A good feature detector should produce a set of features that have the following properties: repeatability, distinctiveness, locality, quantity, accuracy and efficiency (Tuytelaars and Mikolajczyk, 2008). Repeatability, which is arguably the most important, is the capability of a detector to detect a high percentage of the same features given images of the same object or scene under different viewing conditions. This property can be achieved in two ways: invariance to nuisance variations in images, such as changes in scale, viewing angles and lighting, or robustness. Choosing a suitable detector will depend on the type of applications and settings. Table 6.1 provides an overview of the properties of the state-of-the-art feature detectors, which can be used as a guideline when
Figure 6.1: An example of the failure of a 3D model from Bundler due to incorrect tracks

![Image A](image1.png) ![Image B](image2.png) ![Image C](image3.png)

Figure 3.13: Three neighboring images from the nskulla data set. The structure from motion algorithm fails on this data set due to too little overlap between views. Images A and B have sufficient matches, as do B and C, but A and C have too few. Having matches between triplets of views is a requirement for the SfM algorithm.

Figure 3.14: Incorrect interpretation of the Florence Duomo. Left: an overhead view of the Florence Duomo. Right: a reconstruction of the Florence Duomo from a collection of 460 images. The Duomo is mistakenly "unwrapped," with the north and south sides of the cathedral joined together on the south side. That is, the north wall is erroneously reconstructed as an extension of the south wall, and the west wall appears again on the east side of the reconstruction, as highlighted in red. This problem occurs because both sides of the dome area appear very similar, and are indistinguishable to Bundler. The figure is taken from Snavely (2008).

Figure 6.2: (left) an overhead view of the Florence Duomo, (right) a reconstruction of the Florence Duomo from a collection of 460 images. The model of the Duomo is wrongly reconstructed, as the north and south sides of the cathedral are joined together on the south side. In other words, the north wall is erroneously reconstructed as an extension of the south wall, and the west wall appears again on the east side of the reconstruction, as highlighted in red. This problem occurs because both sides of the dome area appear very similar, and are indistinguishable to Bundler. The figure is taken from Snavely (2008).
Figure 3.17: Cascading errors in the Colosseum.

This overhead view of a reconstruction of the Colosseum shows evidence of a large problem. The interior is well-reconstructed, but the outer wall (the spray of points curving towards the top of the image) is jutting out from the interior at nearly right angles. This problem occurred because the inside and outside of the Colosseum are very weakly connected. The reconstruction began from the inside, and proceeded correctly until images of the outside began being added; at that point a few mistakes occurred and were compounded as the reconstruction of the outside grew. Interestingly, when the reconstruction is started from images of the outside of the Colosseum, this problem does not occur (see the reconstruction in Figure 3.11), suggesting that the order in which the images are added can make a difference.

Figure 6.3: An example of the cascading error: (a) the Colosseum, taken from Snavely (2008), (b) an example of a 3D model from the tunnel dataset. Many reconstructed points have drifted away from the true geometry of the scene. This error occurs due to the large uncertainty of the camera positions, which results in the propagation of errors. It may also occur due to the narrow baseline between certain camera pairs, which results in highly uncertain depth estimations.
selecting a suitable feature detector for a required application (Tuytelaars and Mikolajczyk, 2008).

As shown in Table 6.1, feature detectors can be broadly grouped into three types: \textit{corner detectors}, \textit{blob detectors} and \textit{region detectors}. The \textit{Harris} corner detector, proposed by Harris and Stephens (1988), is based on the computation of image derivatives to locate intensity changes in two directions, which indicate the presence of a corner. However, this detector is not scale- or affine-invariant, as shown in Table 6.1. Mikolajczyk et al. (2005) extend the detector to make it scale invariant (Harris-Laplace detector), and both scale and affine invariant (Harris-Affine detector). In the Harris-Laplace detector, the Laplacian operator is used for scale selection. The scale is chosen to be the one with the maximum value of similarity between the feature detection operator and the local image structures. The \textit{Harris-Affine} detector detects elliptical affine regions to obtain affine invariant corners based on the initial regions estimated from the \textit{Harris-Laplace} corners.

An example of a good region detector is the Maximally Stable Extremal Regions (MSERs) detector (Matas et al., 2004). This detector picks up the connected components of an appropriately thresholded image. The pixels inside an MSER have either higher or lower intensities than all of the pixels on its outer boundary. The MSER detector when combined with certain descriptors can perform very well on flat surfaces and can work well when changes in illumination are present (Forssén, 2007).

Blob detectors detect areas of uniform intensity in the image, and the blobs are localised at the centre of the areas. The well-known Scale Invariant Feature Transform (SIFT) algorithm (Lowe, 2004) uses the Difference of Gaussians detector (DOG), which
is an approximation of the Laplacian for blob detection. This detector is invariant to image scaling and rotation, partially invariant to changes in illumination and also partially affine-invariant.

As discussed in Tuytelaars and Mikolajczyk (2008), the choice about which features to use depends on the applications, and this still remains an active area of research. The features for the tunnel datasets should be efficient as the number of images is large; and should also be invariant to rotation and scale. The features may not be required to be affine invariant, as the images do not contain a large amount of deformation. According to Table 6.1, features that are inefficient, such as Harris-Laplace and Hessian-Laplace, may be eliminated. In addition, features that are not scale-invariant, such as Harris corner or SUSAN, can also be dismissed. Therefore, feature detectors, such as DOG, SURF, Harris-Affine and MSER, may be selected for the tunnel images as they meet the requirements for the tunnel datasets.

Descriptors The feature descriptor describes the local appearance around a feature so that it can be matched uniquely with other features in different images. Mikolajczyk and Schmid (2005) provide an extensive evaluation of the performance of different descriptors in a wide range of images with changing imaging conditions. Improvement in the descriptor algorithms is focused on improving speed, accuracy and robustness. Two examples of recent descriptor algorithms are Binary Robust Independent Elementary Features (BRIEF) (Calonder et al., 2010) and Speeded-up Robust Features (SURF) (Bay et al., 2008). These two descriptors are fast and perform well in the Mikolajczyk datasets. The SURF descriptor represents a distribution of Haar-wavelet responses within the interest point neighbourhood and makes efficient use of integral images. The BRIEF descriptor classifies image patches on the basis of a relatively small number of pairwise intensity comparisons. It is fast to compute and match, memory efficient, and performs comparably with the SURF descriptor.

The SIFT feature descriptor (Lowe, 2004) remains a popular choice for many applications and its performance is used as a benchmark for evaluation. SIFT descriptor vectors are computed as a set of orientation histograms on a 4×4 grid neighbourhood around a feature. Each grid is 4x4 pixels, and an intensity gradient at each pixel is binned into one of 8-orientation histograms. Therefore, each grid has 8 dimensions and the total dimension of the SIFT descriptor vectors is 4×4×8 = 128. The orientation for each image pixel is computed from the intensity gradients within a grid. The size of the grids is scaled and
rotated according to the scale and the orientation of the extracted SIFT feature. This descriptor performs relatively well in the Mikolajczyk datasets. It is invariant to scale and rotation, and partial invariant to affine transformation and illumination. The SIFT descriptors have properties that are sufficient for many applications, including the tunnel images presented in this thesis. However, for more time-critical applications, SURF and BRIEF may be better choices.

Table 6.1: An overview of the feature detectors with their associated properties, taken from Tuytelaars and Mikolajczyk (2008), that can be used as a guideline when choosing a suitable detector for a required application

<table>
<thead>
<tr>
<th>Feature Detector</th>
<th>Corner</th>
<th>Blob</th>
<th>Region</th>
<th>Rotation invariant</th>
<th>Scale invariant</th>
<th>Affine invariant</th>
<th>Repeatability</th>
<th>Localization accuracy</th>
<th>Robustness</th>
<th>Efficiency</th>
</tr>
</thead>
<tbody>
<tr>
<td>Harris</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>++</td>
<td>+++</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Hessian</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>++</td>
<td>+++</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>SUSAN</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>++</td>
<td>+++</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Harris-Laplace</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>+++</td>
<td>+++</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Hessian-Laplace</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>+</td>
</tr>
<tr>
<td>DoG</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>++</td>
<td>+++</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>SURF</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>++</td>
<td>+++</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Harris-Affine</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>+++</td>
<td>+++</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Hessian-Affine</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>+</td>
</tr>
<tr>
<td>Salient Regions</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>+++</td>
<td>+++</td>
<td>+++</td>
<td>+</td>
</tr>
<tr>
<td>Edge-based</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
<tr>
<td>MSER</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>+++</td>
<td>+++</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Intensity-based</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>++</td>
<td>++</td>
<td>++</td>
<td>+</td>
</tr>
<tr>
<td>Superpixels</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>+</td>
<td>+</td>
<td>+</td>
<td>+</td>
</tr>
</tbody>
</table>

Matching In the final step of the feature matching, the descriptor vectors of each feature from one image are matched with those from other images to select putative correspondences. These correspondences are further verified for geometric consistency based on a global rigid transformation model, such as a fundamental matrix or homography (Brown and Lowe, 2005). The model is coupled with a RANSAC algorithm so that the best model can be found from the subset of putative correspondences, and any outliers can be removed.

Descriptors are usually matched in a high dimensional space based on a distance measure (i.e. Euclidean distance and Mahalanobis distance) to find their nearest neighbours. The search in the nearest neighbour algorithm can be achieved through an exhaustive search or by using a k-d tree (Friedman et al., 1977). A k-dimensional tree (i.e. k-d tree) algorithm partitions data points in a multidimensional space into k partitions. When searching for a data point’s nearest neighbour, the data points within the same parti-
tion are searched first. If the first search proves unsuccessful, the data points in other neighbouring partitions are then tested. As shown in Lowe (2004), however, using the k-d algorithm is no faster than conducting an exhaustive search of datasets of more than 10 dimensions.

Lepetit and Fua (2006) formulate keypoint matching as a classification problem. This approach allows keypoints to be matched in real-time, though a classifier must be trained, and the computation time required to be spent on training may be high. The algorithm works by selecting a set of prominent keypoints in an image which are then given class labels. The patches centred on the keypoints are used as descriptors to train a classifier and, at run time, the classifier can give a label to the input patches with the likelihood of the patches belonging to particular classes. The Randomised Forest (RF) classifier is used in this work as it is very fast to train and classify. The main drawback of this approach is that, although descriptors may not be required, choosing which keypoints to use for training the classifiers is somewhat ad hoc.

So far, all of the literature discussed largely involves matching descriptors or patches solely based on the similarity of the local appearance of features. The descriptors, however, can fail to account for the global context in an image, which can help to reduce the degree of ambiguity in the matching task. Mortensen et al. (2005) add a global context descriptor to a standard SIFT descriptor and this method shows improvements in the matching result. This approach works especially well in the case of images containing deformable objects as well as those with many local regions with similar appearances, such as a chessboard pattern. Similarly, Carneiro and Jepson (2007) and Deng et al. (2006) apply Shape Context descriptors (Belongie et al., 2002) to create global context descriptors based on the distribution and spatial arrangement of the neighbourhood keypoints. Torresani et al. (2008) match keypoints based on the spatial arrangement of the features and formulate the matching problem as a graph matching problem. This method works particularly well for matching images with deformable objects.

6.2.1 Discussion

The pipeline in feature matching algorithms has been inevitably unchanged for many applications. Much work has been done to improve the performance of each component in the pipeline to develop either better detectors or descriptors in order to increase the invariance and efficiency. To match features, the focus has been on comparing the simil-
arity of local appearance alone and much improvement has been made to the matching speed. A high degree of accuracy can be achieved by these matching algorithms as they usually incorporate a prior knowledge of the scenes and global transformation models, which makes the algorithms application-specific. Many variants of RANSAC algorithms use prior knowledge to improve the speed and accuracy of the matching algorithms.

The work based on Caetano et al. (2009) and Torresani et al. (2008) suggested that, for many feature matching problems, appearance-based models perform similarly or better than state-of-the-art spatial-constraint models. However, ideal matching algorithms should produce correspondences that are both similar in feature appearance and spatially coherent. Some studies have attempted to enforce the spatial consistency constraint to reduce ambiguity for matching, such as Mortensen et al. (2005). Therefore, in the subsequent section, the proposed matching algorithms have utilised the spatial consistency constraint, and their performances are evaluated against the standard datasets from Mikolajczyk and Schmid (2005) (see Chapter 3, Section 3.2.1), and the beam and tunnel datasets.
6.3 Distance and angle filtering algorithm

6.3.1 Method summary

6.3.1.1 Length and Angle assumptions

Figure 6.5(a) shows a typical example of the matches from a correspondence set for a pair of affine images. Let \( \mathbf{x}_i = (x_i, y_i)^T \) be the \( i \)-th image coordinate from the left image, \( \mathbf{x}_j' = (x_j', y_j')^T \) be the \( j \)-th image coordinate from the right image, and let \( (\mathbf{x}_i, \mathbf{x}_j')^k \) be the \( k \)-th correspondence from the correspondence set. Hence a vector for the correspondence \( (\mathbf{x}_i, \mathbf{x}_j')^k \), when the left and right images are placed next to each other, is defined as

\[
\mathbf{x}_j = \mathbf{x}_i + d^k \mathbf{n}^k
\]

where \( d^k = \sqrt{(x_i - x_j')^2 + (y_i - y_j')^2} \) is the length between \( \mathbf{x}_i \) and \( \mathbf{x}_j' \), and \( \mathbf{n}^k = \frac{\mathbf{x}_j' - \mathbf{x}_i}{d^k} \) is the unit directional vector. From Figure 6.5(a), it can be seen that the lengths and orientations of most of the vectors are similar. From this observation, a simple filter can be applied to remove some of the incorrect matches based on the average lengths and directions of all vectors from the correspondence set. Two filters can be used for this purpose:

**Distance Filter** A correspondence \( (\mathbf{x}_i, \mathbf{x}_j')^k \) from the correspondence set will be removed if \( \left| d^k - \frac{1}{M} \sum_{k=1}^{M} d^k \right| > \tau_d \), where \( \tau_d \) is a specified distance threshold.

**Angle Filter** A correspondence \( (\mathbf{x}_i, \mathbf{x}_j')^k \) from the correspondence set will be removed if \( \left| \tan^{-1} \mathbf{n}^k - \frac{1}{M} \sum_{k=1}^{M} \tan^{-1} \mathbf{n}^k \right| > \tau_\theta \), where \( \tau_\theta \) is a pre-defined angle threshold.

6.3.2 Experiments

6.3.2.1 Evaluation protocol

To evaluate the performance of the proposed algorithm, the protocol based on Sattler et al. (2010) is followed. The RANSAC algorithm is applied to an original putative correspondence set. The performance of the proposed algorithm is measured by the ratio of the inliers to the number of total matches. In the RANSAC algorithm, the normalised
Figure 6.5: (a) matching of SIFT features without filtering (b) matching of SIFT features with the Length and Angle filters. Not every match is shown for viewing clarity.
8-point algorithm with a fundamental matrix is implemented, and the normalised DLT algorithm\(^1\) is applied when computing homographies. Inliers are determined as those points with a Sampson error of less than 1. The termination probability of the RANSAC algorithm is set at 5%. The experiments are carried out using the tunnel datasets (see section 3.2.1). For each image pair, a putative correspondence set is obtained from the original SIFT matching algorithm from Vedaldi’s implementation (Vedaldi, 2007) using default SIFT parameters. The RANSAC algorithm is applied to the putative correspondence set to obtain the inlier ratios for correspondences with and without the proposed filtering algorithm, for comparison. Unless specified otherwise, every experiment is repeated 1000 times to obtain statistically meaningful results. The average values with the standard deviation denoted by error bars are plotted.

6.3.2.2 Choosing thresholds for the angle and distance filters

The images from the wide-baseline viewpoint change datasets, as shown in Figure 3.9 (Chapter 3), are used in this experiment. For each pair of images, the angles and distances for each pair of features in a bipartite set are computed, and their means subtracted, as explained in Section 6.3.1. The distances and angles from each pair of images are identified by different markers and colours, and plotted in Figure 6.6. The distance threshold \(\tau_d\) and the angle threshold \(\tau_\theta\) are set at 0.5 and 20°, respectively, as shown by the red dotted lines in both Figure 6.6(a) and (b). The thresholds are set to remove approximately 5% of the total correspondences. Choosing the thresholds should be conservative such that not too many correspondences are removed. From Figure 6.6, it can be observed that most distances and angles lie about their means—this corresponds well with the assumptions made in this approach. Note that this approach does not work with images with large deformations, such as those with rotation and scale variation.

6.3.2.3 Effects of the application of the filters

Figure 6.7 shows the plots of inlier ratios in all image pairs in each set of data from the tunnel datasets. The results in blue and red are derived from the SIFT matching algorithm (labelled as SIFT) and the proposed filters (labelled as SIFT+Filter), respectively. The

---

\(^1\) This algorithm transforms ordinary linear equations, \(y_k = a_n x_k\) into a matrix equation \(Y = AX\), where \(Y\) and \(X\) are known variables. Hence, matrix \(A\), containing unknown parameters, can be solved.
standard deviations for all results are shown as error bars and it can be seen that the standard deviations from the proposed method for all results are similar to those achieved when using the SIFT matching alone. The proposed method improves the inlier ratios in all results with increases in the inlier ratios ranging from approximately 0.05 to 0.4. The proposed method works particularly well in the narrow-baseline viewpoint change dataset, shown in Figure 6.7(b). The standard SIFT algorithm struggles to obtain consistent values for the inlier ratios over all of the images in this set, while the proposed algorithm finds consistent values. In Figure 6.7(a), the inlier ratio of image pair number 4 appears low because this image pair has very little overlap and the algorithm struggles to find consistent matches between them.

6.3.2.4 Discussion

It is important to set thresholds that do not remove too many matches from the original set of correspondence. If the thresholds are set too close to the means, some correct matches may also be removed.

The SIFT matching algorithm searches for correspondences based on the local appearances of features alone, whereas the proposed method utilises the structural information in images. Structural information or spatial neighbourhood considers that matched features should lie within a close neighbourhood under a rigid global transformation. This assumption is true in most cases for the tunnel datasets where there is not a significant amount of rotation and scaling between a pair of images.

The proposed method is similar to Sattler et al. (2010). This work demonstrates that the matches whose feature locations are not within their neighbourhood should be discarded, and the method results in speeding up the RANSAC algorithm. In scenes with repetitive textures, such as the tunnel datasets, many features have a similar local appearance, and the structural information should be used to help to reduce the degree of ambiguity in the matches. The filtered correspondence set also improves the accuracy of the transformation models and the speed of the RANSAC algorithm, as demonstrated in the experiments.

One drawback of the proposed approach is that it may not work well with an image pair with a wide-baseline or significant rotation and zoom. It is suggested in Sattler et al. (2010) that the performance of the matching algorithm diminishes when the viewing angle is more than $56.6^\circ$, which roughly corresponds to the reliability range of the SIFT
matching algorithm (Mikolajczyk and Schmid, 2005).

Figure 6.6: Angle and distance thresholds of the proposed filtering algorithm for the wide-baseline image datasets (see Chapter 3, Section 3.2.1).
Figure 6.7: The effects of using the angle and distance filters; the inlier ratios for all image pairs are improved.
6.4 Spatially consistent randomised clustering forest

6.4.1 Method summary

In this section, a new algorithm is created to replace the feature matching stage and improve its performance. As shown in Figure 6.8, it combines the Random Forest (RF) and Shape Context (SC) algorithms.

In the SIFT matching algorithm, features descriptors are matched by the Nearest Neighbour (NN) algorithm to obtain a similarity score, \( s \), which lies between 0 and 1 (0 means no similarity and 1 means a perfect match). The similarity score of the new algorithm is a sum of the scores from the RF and SC algorithms, expressed as

\[
s_{total} = w_{RF}s_{RF}(f_a, f_b) + w_{SC}s_{SC}(f_a, f_b)
\]  

(6.2)

where \( w_{RF} \) and \( s_{RF}(f_a, f_b) \) are the weight and the score from the RF algorithm, and \( w_{SC} \) and \( s_{SC}(f_a, f_b) \) are the weight and the score from the SC algorithm. Sections 6.4.2 and 6.4.3 explain how the scores of the RF and SC algorithms are obtained, respectively. The summary of the new algorithm is shown in Algorithm 6.1.

![Diagram](https://via.placeholder.com/150)

Figure 6.8: A new proposed approach to feature matching, using the Random Forest and Shape Context algorithms.
6.4.2 Random Forest

The Random Forest algorithm, developed by Breiman (2001), is simply an ensemble of the many decision trees working together to provide more accurate classification results than a single tree can achieve. Section 6.4.2.1 explains the algorithm for a single tree, and Section 6.4.2.2 explains how these individual trees are combined to form the Random Forest algorithm.

6.4.2.1 Randomised tree

This algorithm clusters a set of features into different groups such that the features within the same groups are thought to have similar characteristics to qualify to be in the same cluster. If features are not in the same cluster, they are considered to have a different appearance, and to be unmatched. If a feature from one image resides in the same cluster as a feature in the other image, these features are defined as a matching pair. An example is shown in Figure 6.9; all of the features shown in blue (i.e. they have the same label) on the left image are matched with all of those features shown in blue in the right image. It can be seen that each feature can have multiple potential matches since only one tree is used for the matching; hence multiple trees are applied in order to improve the accuracy of the matching, as explained in Section 6.4.2.2.

To match the features in a pair of images, a randomised clustering tree works as follows, and as illustrated in Figure 6.10. Let \( f = (f_1, f_2, f_3, \ldots, f_D) \in \mathbb{R}^D \) be a feature descriptor vector, and let us start from the root node with 32 features from both images. The descriptor vectors are then mapped with a randomly generated vector \( b \in \mathbb{R}^D \) to form a split function, \( f^T b \). Features with \( f^T b \geq \tau \) go to the left intermediate node, and features with \( f^T b < \tau \) go to the right intermediate node, where \( \tau \) is a threshold, which is currently taken as the median of \( f^T b \) from all features. Then, each intermediate node becomes a root node and a set of features on each node keep splitting in the same manner until the stop criterion is met. In this case, the stop criteria is active when the number of features at a node is less than or equal to 4. When a node stops splitting, that node becomes a leaf node, and the algorithm stops. Each leaf node is given an index or a label, i.e. \( n_1, n_2, \ldots, n_M \). As shown in Figure 6.10, 32 features at the root node are now clustered into 8 groups of 4 features at the leaf nodes, and features in the same clusters are taken
as matching. Formally, this can be written as

\[ T(f) = L \]  \hspace{1cm} (6.3)

where \( T \) is a randomised clustering tree function that maps a feature \( f \) into a one of the leaf node indices, \( L \in \{n_1, ..., n_M\} \).

### 6.4.2.2 Randomised Forest

A single randomised tree, as explained in Section 6.4.2.1, cannot provide sufficient discriminative power. The Randomised Forest algorithm uses many trees to increase discriminative power for matching. As shown in Figure 6.11, a feature is mapped into different leaf node indices by corresponding trees. Since each tree is randomised, an identical input \( f \) will not necessarily fall into the same leaf nodes in different trees. This means that there is no consistency between the leaf node indices in different trees. As illustrated in Figure 6.11, a feature is mapped into Leaf ID 2 for Tree #1, 5 for Tree #2, and 6 for Tree #F, etc. By concatenating all of the indices of \( f \) from all of the trees, a leaf node index vector can then be constructed as

\[ L(f) = (L_1, ..., L_{|T_j|})^T \]  \hspace{1cm} (6.4)

where \( L_j \) represents a leaf index obtained from \( T_j \), as shown in Figure 6.11.

### 6.4.2.3 Similarity score

To measure the similarity of a pair of features, leaf node vectors, \( L(f_a) \) and \( L(f_b) \), are compared, which can be expressed as

\[ s_{RF}(L(f_a), L(f_b)) = \frac{1}{|T_j|} \sum_{j=1}^{|T_j|} \begin{cases} 
1, & \text{if } L_{T_j}^{f_a} = L_{T_j}^{f_b} \\
0, & \text{otherwise}
\end{cases} \]  \hspace{1cm} (6.5)

where \( L_{T_j}^{f_a} \) is a leaf node index of the tree \( T_j \) for the feature \( f_a \) (and similarly for \( f_b \)). Equation 6.5 compares leaf node indices of the same trees between a pair of features, and then a similarity score is added if the indices by the same trees are identical. A similarity
score of a pair of features is the total number of identical indices and, the higher the score, the more similar the features are.

6.4.3 Shape Context

Spatial relations between features can help with reducing ambiguity to improve matching accuracy. This is similar to the human visual system. Humans are able to recognise and distinguish an object by using the spatial relationships between multiple objects in a scene to help to disambiguate regions accurately (Mortensen et al., 2005). In Mortensen et al.’s work, the global context descriptors are added to the SIFT descriptors to provide an additional constraint, using the spatial relationship between features to improve the matching accuracy over the original SIFT matching algorithm.

The spatial relationship constraint can be achieved by applying the Shape Context algorithm. Introduced by Belongie et al. (2002), the algorithm was originally used in shape recognition. It is constructed by computing the distribution of the neighbourhood keypoints in a log-polar histogram. As shown in Figure 6.12, the neighbourhood keypoints
Algorithm 6.1 Algorithm explaining the combined algorithm

Input A descriptors vector \( f_a \) from image 1 and \( f_b \) from image 2

- Obtain a similarity score from the Random Forest algorithm, \( s_{RF}(f_a, f_b) \), explained in Section 6.4.2.

- Obtain a similarity score from the Shape Context algorithm, \( s_{SC}(f_a, f_b) \), explained in Section 6.4.3.

- Obtain a combined score, \( s_{total} = w_{RF}s_{RF}(f_a, f_b) + w_{SC}s_{SC}(f_a, f_b) \), where \( w_{RF} \in [0, 1] \) and \( w_{SC} \in [0, 1] \), and \( \sum w_{RF} + w_{SC} = 1 \), are the weights.

Output A combined similarity score \( s_{total} \)

Figure 6.10: An illustration of how a single randomised tree clusters a set of features. Section 6.4.2.1 explains how this operation is performed.
Figure 6.11: Mapping of a feature by the Random Forest algorithm. The diagram is taken from Perbet et al. (2009). A feature is labelled with different node indices by each randomised tree.

are binned according to their distance and angle relative to an interest keypoint to produce a 2D histogram of log r and theta. Figure 6.12(a) shows the points distributed along the edge of a letter 'A', and Figure 6.12(b) shows the points along a differently shaped letter 'A'. Figure 6.12(c) shows an example of the Shape Context algorithm computed at the keypoint ♦ in Figure 6.12(a), and its corresponding histogram is shown in Figure 6.12(d). Figure 6.12(e) and (d) are the histograms of the features □ and △, respectively from Figure 6.12(b). It can be seen that the histogram of the feature □ looks more similar to the histogram of the feature ♦ than the histogram of the feature △. This is expected since the histograms of the features ♦ and □ are computed at similar locations of the letter A, and the distributions of their neighbouring features should be similar.

The Shape Context algorithm is applied to improve the performance of the Random Forest algorithm, as explained in Section 6.4.2. The modified Shape Context by Carneiro and Jepson (2007) is used in this thesis as explained below.

6.4.3.1 Modified Shape Context

The original Shape Context algorithm by Belongie et al. (2002) is modified in Carneiro and Jepson (2007) to include a scaled distance and the weight of neighbouring features. As shown in Figure 6.13, a feature $f_1$ is a feature of interest, and $f_2$ and $f_3$ are its neighbouring features. There are 5 bins for distance and 8 bins for angle, making a
total of 40 bins. For distance bins, the distances are divided into \(2r, r, \frac{r}{2}, \frac{r}{4}, \frac{r}{8}, \frac{r}{16}\), where \(r\) is the mean distance of all pairs of features in an image. The angle bins are divided into \(45^\circ, 90^\circ, 135^\circ, 180^\circ, 225^\circ, 270^\circ, 315^\circ, 360^\circ\), as shown in Figure 6.13. The feature \(f_2\) is added to the bins \(2r\) and \(360^\circ\) according to its position in relation to the feature \(f_1\), and similarly for the feature \(f_3\). Note that \(f_2\) is given less weight than \(f_3\) since it lies further away, and this is one modification from the original Shape Context descriptor. The method of computing the distance, angle and weight of the neighbourhood features to the feature of interest is explained below.

**Scaled Distance** To increase the robustness to scale changes, the distance relation between a pair of features, \(f_a\) and \(f_b\), is transformed by their image scales. This makes the shape context histogram scale-invariant. The scaled distance is defined as

\[
D(f_a, f_b) = \frac{||x_a - x_b||}{\sqrt{\sigma_a^2 + \sigma_b^2}} \tag{6.6}
\]

where ||.|| is the Euclidean distance, \(x_k\) is the image position of the image feature \(f_k\) and \(\sigma_k\) is the scale of the image feature. Note that scale here refers to the scale of the SIFT detector.

**Heading** The angle relation or heading between \(f_a\) and \(f_b\) is defined as

\[
\mathcal{H}(f_a, f_b) = \Delta_\theta(\theta_a - \theta_{ab}) \tag{6.7}
\]

where \(\Delta_\theta(.) \in (0, 2\pi]\) denotes the principal angle, \(\theta_a\) is the main orientation of the feature \(f_a\) and \(\theta_{ab}\) is the relative angle between the feature \(f_a\) and \(f_b\), \(\theta_{ab} = \tan^{-1}(x_a - x_b)\). This definition means that the shape context histogram is rotated according to the main orientation of the feature \(f_a\) and, therefore, the shape context is rotation-invariant.

**Weighting** The weight is applied to the neighbourhood keypoints of \(f_a\) based on its scaled distance with respect to \(f_a\). This is believed to improve robustness to non-rigid deformation, as closer neighbourhood features are less constrained by the global rigid transformation than those located further away. The weight is formulated as

\[
w(f_a, f_b) = e^{-\frac{0.5D^2(f_a, f_b)}{\tau^2}} \tag{6.8}
\]
where $D^2(f_a, f_b)$ is the scaled distance between the features and $L = \frac{D_M}{L_{SC}}$, $L_{SC}$ is a tuning variable, and $D_M$ is the maximum model diameter in pixels, which is equal to $D_M = 2 \times \bar{D}^2(f_a, f_b)$ and $\bar{D}$ is the mean scaled distance between all pairs of features across the entire image.

### 6.4.3.2 Similarity score

To measure the similarity between a pair of features from the SC algorithm, the SC histograms are compared as follows. The SC histograms of a pair of features, $H(f_a)$ and $H(f_b)$, are compared based on the chi-square test statistics $\chi^2(H(f_a), H(f_b))$ (Belongie et al., 2002) as follows

$$s_{SC}(H(f_a), H(f_b)) = 1 - \chi^2(H(f_a), H(f_b))$$

(6.9)

$$\chi^2(H(f_a), H(f_b)) = \frac{1}{2} \sum_{k=1}^{K} \frac{[h_k(f_a) - h_k(f_b)]^2}{h_k(f_a) + h_k(f_b)}$$

(6.10)

where $h_k(f_a)$ and $h_k(f_b)$ are the $K$-bin normalised histograms of features $f_a$ and $f_b$, respectively. Note that each histogram bin is normalised by the total number of features in an image.
Figure 6.12: Shape context, taken from Belongie et al. (2002)

Figure 6.13: A modified shape context descriptor. The feature \( f_3 \) is further away than \( f_2 \), and hence \( f_3 \) is weighted less than \( f_2 \)
6.4.4 Experiments

6.4.4.1 Evaluation method

The evaluation of the proposed matching algorithm is conducted on two datasets. The first dataset is provided by Mikolajczyk and Schmid (2005). This dataset is commonly used in evaluating the performance of feature matching algorithms, as described in Section 3.2.3. The performance of the proposed algorithm is evaluated using a toolbox provided with the dataset, downloadable from the Visual Geometry Group (2011). An explanation of the evaluation method found in the toolbox is briefly given in Appendix A.1, and the reader is referred to Mikolajczyk and Schmid (2005) for full details.

Two matching schemes are considered in the evaluation, the Nearest Neighbour or NN matching scheme and the Similarity or Sim scheme. For the Sim scheme, a pair of features is matched if their similarity score exceeds a specified threshold. Therefore, for the Sim scheme, it is always possible for a feature to have a 1-to-many matching relationship. For the NN scheme, only the pairs with the highest similarity scores are considered as matched. Therefore, the NN scheme results in a 1-to-1 matching relationship. A full explanation of the different types of matching schemes can be found in Appendix A.1.

The second dataset is constructed from the beam datasets, as described in Section 3.3. Six pairs of images are used for the evaluation, as shown in Table 6.2 and Figure 6.14. Various viewing angles and focal lengths are covered by these image pairs. Furthermore, various stages of crack propagation in the beam are also present in these images in order to evaluate the algorithm performance when an object is non-rigid. The method of evaluation is identical to that shown in Section 6.3.2.1.

Types of feature detectors and descriptors The Random Forest algorithm matches features based on their descriptors. There are many types of descriptor, as discussed in Section 6.2. The SIFT descriptors are used in the experiments in this section and their implementation by Mikolajczyk and Schmid (2005) is used.

The Shape Context algorithm is constructed based on the distribution of the features in an image. In the experiments, the SIFT detector is used since it generally provides a reasonable number of features in an image. If the number of features in an image is too low, the Shape Context algorithm may not perform well. Furthermore, the scale parameters from the SIFT detector can be conveniently used in the Shape Context algorithm, unlike other types of detector that do not have the scale parameter.
6.4.4.2 Parametric study

Number of trees

The performance of the Random Forest algorithm depends greatly on the number of trees in the forest. In this experiment, matching features are based solely on matching by the Random Forest algorithm. This is achieved by setting $w_{SC} = 0$ and $w_{RF} = 1$ in Equation 6.2, i.e. there is no contribution from the Shape Context algorithm to the total similarity score, i.e. $s_{total}(f_a, f_b) = s_{RF}(f_a, f_b)$. Figure 6.15 shows the matching performances with varying numbers of trees from 2 to 200 for the Sim matching strategy in different types of image conditions from the Mikolajczyk datasets. Note that the number of trees is varied at an approximately logarithmic scale interval.

It can be seen that all plots show the same trend whereby performance is improved as the number of trees increases. In all plots, two distinctive regions can be seen: the region of high precision (i.e. $1 - precision < 0.1$), and the region of low precision ($1 - precision > 0.1$). In the former, the plots do not change much as the number of tree increases. In contrast, the performances for all datasets improve as the number of trees increases in the low precision region. There is, however, a trade-off between the number of trees and the computation time. Hence, a low number of trees might be used to reduce the computation time, while its performance may also be improved by using the SC algorithm.

Effect on a tuning parameter

The tuning parameter of the modified shape context $L_{SC}$ has a direct impact on the performance of the proposed algorithm. A higher $L_{SC}$ means that the neighbourhood points far away from a keypoint of interest become less important. The $L_{SC}$ value may be considered as the degree to which an object is allowed to deform in an image.

This experiment is aimed at discovering the optimal value of $L_{SC}$. Hence only the score from the Shape Context is activated, i.e. $s_{total}(f_a, f_b) = s_{sc}(f_a, f_b)$, $w_{RF} = 0$ and $w_{SC} = 1$. In Figures 6.16(a) and (b), the plots of the performances obtained from the Graff dataset with varying number of tuning parameters $L_{SC}$ are shown. In both the NN and Sim matching strategies, $L_{SC} = 10$ performs best, and $L_{SC} = 0$ produces the
Table 6.2: A summary of the images used in the evaluation of the beam dataset. All of the figures are shown in Figure 6.14

<table>
<thead>
<tr>
<th>Pair No.</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>Figure 6.14(a)-left</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Figure 6.14(a)-middle</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Figure 6.14(a)-right</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Figure 6.14(b)-left</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
</tr>
<tr>
<td>Figure 6.14(b)-middle</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Figure 6.14(b)-right</td>
<td></td>
<td></td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Figure 6.14(c)-left</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
</tr>
<tr>
<td>Figure 6.14(c)-middle</td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Figure 6.14(c)-right</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

(a) 0°, f = 55 mm

(b) 30°, f = 24 mm

(c) -30°, f = 18 mm

Figure 6.14: Non-rigidly deforming bodies imaged in the beam dataset. Chapter 3 explains how the beam dataset was created.
worst performance. When $L_{SC} = 0$, this is essentially the shape context descriptors without modification and its poor performance is expected. The original Shape Context descriptors place an equal importance on all neighbouring features, whereas the modified descriptor places more weight on closer neighbouring features. The latter should be more important because they are likely to provide a more accurate topology for a feature. The value of $L_{SC}$ is set to 10 from now on due to its better performance over other values.

In both plots 6.16 (a) and (b), the performances are worse than those obtained via the SIFT matching algorithm, which is in line with the results of Mikolajczyk and Schmid (2005). The Shape Context descriptor was originally intended to be used with shape recognition, and poor performance can be anticipated if used alone.

6.4.4.3 Experiment Results

Comparison with the Mikolajczyk datasets

In these experiments, the results from the proposed method are shown against the SIFT matching on the Mikolajczyk datasets based on the NN matching strategy. Equal weights from both the RF and SC algorithms are applied, i.e. $w_{RF} = w_{SC} = 0.5$.

Figure 6.17 shows the results for the viewpoint change condition from the Graff dataset (the images are shown in Chapter 3, Figure 3.10). Figure 6.17(a) is for the image pair 1 and 2, which has a smaller viewing angle difference than Figure 6.17(b) for the image pair 1 and 5. For the legend, 50NN represents the plot of the combined algorithm (i.e. the RF and SC algorithms combined); the number of trees is set at 50 and the tuning parameter at 10. The legend 50treeNN refers to when only the RF algorithm is applied and 50 trees are used. The legend 10TunSTR refers to when only the SC algorithm is used, and $L_{SC}$ is set at 10. Finally, NNSIFT is the plot of SIFT matching by the NN scheme. In Figure 6.17(a), the combined algorithm’s performance is shown by the solid green line. It improves upon the performance of the NNSIFT matching in the high precision region and dominates the individual algorithms, as shown by the green dotted line for the RF algorithm and the solid orange line for the SC algorithm. However, the performance of the combined algorithm for the image pair 1 and 5, shown in Figure 6.17(b), is not as good as the SIFT matching algorithm.

In the illumination change dataset (i.e. the Light dataset), the combined algorithm
Figure 6.15: Examples of the effect of increasing the number of trees in the Random Forest. In the legend, the numbers indicate the number of trees and Sim means the benchmark performance by the similarity matching scheme.
outperforms the SIFT algorithm in the high precision region, as shown in Figure 6.18(a) and (b). The performance by the shape context alone (the orange line) is particularly good in Figure 6.18(a). This is to be expected, as all of the images in this dataset are obtained from a fixed camera position, and the SC algorithm performs well when the camera viewpoint changes are small.

In the blur change (Bikes) dataset, the combined algorithm performs similarly to the results discussed above, as shown in Figure 6.19. However, it can be observed that the performance of the SC algorithm (the orange line) is poor compared to that of the other datasets. As the blurring increases, fewer features will be detected, which directly affects the SC algorithm performance. The performance of the SC algorithm depends on the number of features in the images, as it computes the histograms of the neighbourhood features. With fewer features present, the histograms become less distinctive.

For the rotation and zoom (Boat) dataset, as shown in Figure 6.20, the combined algorithm performs poorly compared to the other results presented earlier. The SC algorithm, however, helps to improve the performance of RF dramatically.

Figure 6.16: An effect of different tuning parameters obtained with the Graff dataset.
Figure 6.17: The results from the Graff dataset (images subjected to viewpoint change): (a) image 1 and 2; (b) image 1 and 5. 50NN is the combined algorithm with \#Tree = 50, \( L_{SC} = 10 \); 50treeNN is the RF algorithm alone with \#Tree = 50; 10TunSTR is the SC algorithm alone with \( L_{SC} = 10 \); and NNSIFT is the SIFT matching algorithm.

Figure 6.18: The results from the Light dataset (images subjected to illumination change): Left–image 1 and 2; Right–image 1 and 5, see the legend description in Figure 6.17.
Figure 6.19: The results from the Bikes dataset (images subjected to blurring): Left—image 1 and 2; Right—image 1 and 5. The legend description is shown in Figure 6.17.

Figure 6.20: The results from the Boat dataset (images subjected to zoom and rotation changes): Left—image 1 and 3; Right—image 1 and 6. The legend description is shown in Figure 6.17.
Comparison with the beam dataset

In this experiment, the proposed algorithm is compared against the SIFT matching algorithm using the beam dataset (Section 6.4.4.1). The images in this dataset are wide-baseline images and contain a deformable object (i.e. the beam). The performances of the algorithms are evaluated based on visual comparison. In addition, the RANSAC-homography algorithm is applied in order to evaluate quantitatively the performance of the proposed and SIFT matching algorithms. The RANSAC-homography algorithm iteratively chooses a model from a subset of input matches to obtain the best transformation model for a pair of input images, and the set of inliers or correct matches is obtained based on the chosen transformation model. Note that homography transformation is not the true transformation model of the images in the beam datasets, because the beam is deformable and cannot be accurately modelled by a rigid body transformation such as a homography. However, the homography transformation is a good approximation for the beam datasets. The RANSAC-homography algorithm is repeated 1000 times to obtain statistically meaningful results; the mean and standard deviation are plotted as shown in Figure 6.21. Table 6.2 summarises the image pairs used in this experiment.

Figure 6.21(a) shows the plots of the means of the inlier ratios in each image pair. The proposed algorithm performs better than SIFT matching for all image pairs. The trial counts, as shown in Figure 6.21(b), are also significantly improved by the proposed algorithm, with up to a 50% improvement in pairs 3 and 6, when comparing with the SIFT matching algorithm. As mentioned earlier, homographies are not a true transformation of the images in this dataset; hence, only small improvements can be observed in the inlier ratios. However, based on visual comparison, the proposed algorithm (shown in Figure 6.22(b)) contains many more accurate matches than the SIFT matching algorithm (Figure 6.22(a)). The proposed algorithm works better than SIFT matching because the SC algorithm can cope with some degree of deformation. The SIFT matching algorithm, on the other hand, can only make matches based on the local appearances of features, which clearly does not work well in this dataset.

6.4.4.4 Discussion

From the results, it can be concluded that the performance of the proposed algorithm is best when the RF algorithm is combined with the SC algorithm. This can be observed
in all of the results from the Mikolajczyk datasets shown in Figures 6.17-6.20, where the plots of the combined algorithm show better performance than the individual algorithm. The results are in line with previous work by Deng et al. (2006) and Carneiro and Jepson (2007), who apply the SC algorithm in their feature matching algorithms. Essentially, the SC algorithm makes use of the constraint that considers the spatial relationship between a feature and its neighbourhood. When this constraint is enforced, two features can only be matched if they both have a similar local appearance and topology.

In the beam dataset, which contains a deformable beam, the proposed algorithm performs better than the SIFT algorithm, as shown in Figure 6.21. The results agree with previous work, such as Torresani et al. (2008), which attempts to solve the feature matching problem for scenes containing deformable objects. This problem is usually formulated as a graph matching problem, which essentially considers the spatial relationship among features. Torresani et al. (2008) formulate the problem as the optimisation of a single cost function, whose terms are composed of the constraints from the spatial relationship, the local appearance similarity and a global transformation. This, however, has the disadvantage of requiring a high degree of computational complexity. While the proposed algorithm does not explicitly involve optimisation; it similarly makes use of each of these three constraints. The spatial relationship reduces any ambiguity that may arise from the matching based solely on appearance similarity. When possible, the spatial relation-
ship constraint should be incorporated when designing a feature matching algorithm for wide-baseline images.

6.5 Conclusions

Based on the results of the experiments, the following conclusions have been drawn:

- The shape context algorithm, which exploits the spatial neighbourhood constraint, enhances the performance of the random forest algorithm by reducing ambiguity in the matching.

- The proposed algorithm in Section 6.4, which utilises the spatial neighbourhood constraint, performs better than the SIFT matching algorithm in the beam datasets containing deformable objects.
Figure 6.22: An example of matching from the beam dataset, where the beam is deforming in the scene.
Chapter VII
Change Detection

7.1 Introduction

The goal of change detection is to identify the regions of changes between multiple images of the same scene taken at different times. Figure 7.1 illustrates typical images taken from a real site. Inspectors visually compare the images to determine whether any anomalies have arisen between inspections. In tunnel inspection, being aware of any changes occurring on the tunnel surfaces is useful as it allows an appropriate repair regime to be devised. For example, crack changes may suggest tunnel deformation, and additional reinforcement may be required to strengthen the tunnel linings and so prevent further deformation. Another example, illustrated in Figure 7.1, is the change in staining due to an ingress of saline through cracks in the tunnel linings. If the change is occurring rapidly, an urgent repair regime must be undertaken to prevent further ingress and harmful structural damage but, if the change is slow, the regime may involve merely a cosmetic repair.

Detecting changes between images via visual comparison poses a number of difficulties, mainly due to the variations in image viewpoints and lighting. Visual comparisons are subjective, and the task becomes more difficult with large sets of image data. This chapter aims to address such problems. Given a set of images taken at time $t_0$, denoted as $\mathbb{I}^{t_0} = \{\mathbb{I}_i^t | i = 1, \ldots, n and t = t_0\}$, and an image taken at time $t_1 > t_0$, denoted as $\mathbb{I}_1^{t_1}$, the aim is to produce a binary change mask $\mathcal{B}_1^{t_1}$, where $\mathcal{B}_1^{t_1} = \mathcal{B}(\mathbb{I}_1^{t_1}, \mathbb{I}_0^{t_0})$. The binary change mask indicates the change regions, but does not attempt to quantify the amount of change. Figure 7.2(d) shows an example of a query image with a change mask overlaying the image. In this figure, the change regions are labelled as red patches.

Figures 7.2(a) and (b) show a query image and a chosen reference image, respectively. Figure 7.2(c) shows a ground truth change mask, in red, overlaying the query image. The change mask indicates that the cracks have widened between figures (a) and (b), in
Figure 7.1: An example of images taken from Mustek Station, Prague Metro: (a) images taken in 2003 and (b) 2007. The regions inside the red boundaries in both (a) and (b) are manually labelled to illustrate the change regions of these anomalies, and they have changed over time.

this case as a result of the beam deforming under load. Figure 7.2(d) shows a change mask produced from the system developed in this study. The mask can detect some change regions correctly, although many false positives are observed in Figure 7.2(d). False detection occurs due to the limitations of the methods used in the proposed change detection system and, as will be discussed in this chapter, overcoming these limitations is a challenge for future research.
Figure 7.2: (a) original query image, (b) original chosen reference image, (c) original rectified query image overlaid by a ground truth change mask, and (d) query image with a change mask from the proposed change detection system (the false detection regions are labelled).
7.1.1 The change detection system pipeline

Figure 7.3 shows an outline of the proposed change detection system. The system starts with the pre-processing module. A query image is input into Bundler to register it onto the existing 3D reference model. Then, the reference images are synthesised such that they have the same viewpoints as the query image, based on Surface Estimation. The synthesised reference images and the query image are then processed by a photometric adjustment algorithm so that any variations due to lighting are removed. These images are then input to the Change Detection module to produce a binary change mask. Suitable algorithms in each step are chosen, as discussed below.

7.1.2 Contribution of change detection to civil engineering

The proposed system tackles the problem of change detection by accounting for variations in viewpoint and illumination changes occurring in real images. Not only can the system be used to detect changes due to crack growth, it can also be applied to detect changes due to other anomalies, such as stains due to water ingress. Additionally, the proposed change detection system accounts for variations in the non-rigid deformation of structural components. The proposed system attempts to create an algorithm that is invariant to changes in viewpoint, illumination and non-rigid deformation.

The chapter is organised as follows. Section 7.2 provides a background and reviews the previous work on change detection. Section 7.3 explains the pre-processing techniques for rectifying the variations in images caused by different viewpoints and illumination. Section 7.4 provides details about the change detection algorithms applied after pre-processing. The experiment setups, together with the results, are shown in Section 7.6, and the conclusions are drawn in Section 7.7.

7.2 Previous work

Change detection is an important preliminary task for visual interpretation. It involves the identification and detection of change regions of the same scene taken at different times. Change detection has a large number of applications in various fields of study, such as video surveillance (Collins et al., 2000), remote sensing (Bruzzone and Prieto, 2002), medical diagnosis and treatment (Bosc et al., 2003), civil engineering (Nagy et al., 2001, Landis et al., 1999) and underwater sensing (Edgington et al., 2003). The goal of
change detection is to identify a set of pixels that are significantly different between the last image of the sequence and the previous images, highlighting them for further attention by outputting a change mask. A typical change mask may result from a combination of underlying factors, including the appearance or disappearance of objects, the motion of objects relative to the background, shape changes in the objects, or stationary objects that undergo changes in brightness or colour. It is important to detect significant changes while rejecting unimportant ones, such as those induced by camera motion, sensor noise, illumination variation, non-uniform attenuation, or atmospheric absorption (Radke et al., 2005).

In general, geometric and radiometric (i.e. intensity) adjustments are applied in the pre-processing steps to suppress or filter out common types of unimportant changes before making change decisions. In terms of geometric adjustment, accurate image registration is a pre-requisite, allowing several images to be automatically aligned, warped and stitched into a common coordinate frame. Pixel changes due to camera motion should never be detected as real changes. If the scenes of interest are rigid and the degree of camera motion is light, image registration can often be performed using spatial transformations, such as similarity, affine, or projective transformation. However, for wide-baseline images or those containing deformable objects, a non-global transformation (e.g. optical flow, tracking, or structure from motion algorithms) may be required. Radiometric adjustments remove intensity variations in images caused by changes in the strength or position of the light sources. Various techniques are used in radiometric adjustments, such as intensity normalisation, homomorphic filtering and illumination modelling (Radke et al., 2005).

After the pre-processing steps, the decision rules are applied to determine if the change is significant or not. Examples of decision rules are the significance tests and the likelihood ratio tests, both of which are based on statistical hypothesis testing. Then a binary change mask $M$ is produced where, for a given pixel location $x$, $M(x) = 1$ if there is a significant change at that pixel location, and $M(x) = 0$ otherwise (Radke et al., 2005).

For geometric adjustments, choosing an appropriate spatial transformation is critical for a good change detection system. A good example is the registration of curved human retinal images (Can et al., 2002b), in which a 12-parameter quadratic model is applied. This model is used as an approximation to the retinal surface because a simple planar transformation, which is commonly used in commercial panoramic software, is insufficient. However, this model cannot be applied to a general scene in wide-baseline images.

For a more complex scene captured by wide-baseline images, geometric adjustments
performed by a Structure From Motion (SFM) system are more suitable, as shown by Delaunoy et al. (2008), who apply a change detection system to coral reef images because registration based on 2D image transformation introduces local misregistration due to parallax of the 3D relief. To solve this problem, an SFM system is applied to the images to recover a 3D model and camera poses, and then a synthetic image is constructed from the 3D model so that the synthetic image has the same viewpoint as the query image. Images viewed from the same camera position do not suffer from any parallax. The 3D model is textured on a Delaunay triangulation mesh created from a 3D point cloud. The synthetic image is photometrically corrected using histogram matching in the subregions. A final change mask is created by averaging out the three change masks with the most overlap. An example of their results is shown in Figure 7.4, with the real image on the left and the synthetic image with a change mask on the right. The work concludes that inaccuracies in the results are caused by the quality of the 3D model, and also by non-rigid objects, such as moving fish, in the scene. Nevertheless, this system can be used to mitigate the 3D parallax problem and can cope with complex geometries. This system is fully automatic in the registration stages. Similar to Delaunoy et al. (2008), Buchanan (2009) also applies an SFM system to synthesise new views by using geometric adjustment for change detection in surveillance footage. In this work, it is concluded that an SFM system provides an accurate method for synthesising new views for change detection.

7.2.1 Change detection in civil engineering

For civil engineering applications, change detection can be used to identify changes in anomalies in order to monitor structural changes over time. Lim et al. (2005) solve camera parameters based on the pose estimation algorithms so that images taken from different viewpoints can be compared. The parameters are used to convert the image of cracks from the first image of the sequence, using control points, into other images of cracks at real-world coordinates. The transformation parameters for subsequent images are then estimated using image coordinates and the crack image in object-space found previously. The method for extracting crack features in Lim et al. (2005) is based on image thresholding. However, extracting cracks by image thresholding only works well if an image is clean, i.e. if visible cracks lie against clean plain walls. The results of this work are still impractical, as they are derived from the datasets that are unrepresentative.
Figure 7.3: An outline of the proposed change detection system

Figure 7.4: The result of change detection from coral reef images from Delaunoy et al. (2008). The red and green regions indicate changes that are detected with high and low confidence, respectively.
of a real outdoor environment.

Chen et al. (2008) propose a method that attempts to tackle the non-rigid deformation in change detection. The crack features in this method are motion-invariant, as explained in detail in Section 7.4.2. This method fails to incorporate changes in viewpoints and illumination as these variations are assumed to be constant in the experiments due to the use of fixed cameras and controlled lighting. The installation of stationary cameras for monitoring a large number of cracks is impractical and likely to be costly.

7.2.2 Discussion

Before the decision rules are applied to obtain a change mask, the images must be pre-processed to remove any unimportant variations. Such variations, which should be taken into account in real images in civil engineering applications, are viewpoint, illumination and non-rigidity.

Viewpoints Unless the camera is fixed, multi-temporal images are not usually taken from the same position. The images must be pre-processed by geometric adjustment. Most change detection systems in civil engineering, such as Chen and Hutchinson (2010), use fixed camera systems to avoid the geometric adjustment step. However, fixed camera systems may be less practical for sites requiring a large number of monitoring points.

Illumination Variations in pixel intensity are caused by different strengths or positions of light sources. For images obtained in a controlled environment, the illumination variation (if any) can be modelled, although this does not hold true for general images. Images must therefore be pre-processed by radiometric adjustment algorithms to reduce the number of incidences of false change detection.

Non-rigidity It is not uncommon to observe the deformation of objects when they are monitored over a period of time. Registration by global transformation models alone may prove insufficient if the deformation is considerable, e.g. deformation occurs in the beam datasets. However, for tunnel images, it may be possible to assume that there is no deformation in the tunnel datasets because the deformation of tunnel linings is too small to cause errors in image registration.
Most change detection systems do not consider all of the above parameters. Hence, in the work presented in this chapter, the proposed change detection system consists of different components aiming to cope with all of the above variations commonly found in real tunnel images.

### 7.3 Pre-processing module

From Figure 7.3, the input images are pre-processed before the change detection module is applied. Algorithm 7.1 summarises the pre-processing steps applied in the proposed system. For geometric adjustments, the input images are rectified using Bundler. For photometric adjustments, two techniques are applied: intensity normalisation and homomorphic filtering. These steps are described in detail below.

**Algorithm 7.1 Pre-processing steps**

**Input** existing images with a 3D point cloud and camera poses, and a query image

- **Geometric adjustment**
  - Query image registration (Section 7.3.1.1)
  - Camera selection (Section 7.3.1.2)
  - Synthetic image generation by model-based image rendering (Section 7.3.1.3)

- **Photometric adjustment**
  - Photometric normalisation by intensity normalisation (Section 7.3.2.1) or homomorphic filtering (Section 7.3.2.2)

**Output** photometrically-adjusted synthesised image

#### 7.3.1 Geometric adjustment methods

In this section, geometric adjustment is performed on input images to remove image variation due to camera motion. This method creates a synthetic image from the reference
images such that they have the same viewpoints as a query image. In this thesis, a synthetic image is created based on a 3D model, which has already been reconstructed based on Bundler in Chapter 4.

The results of a synthetic image depend on the type of 3D models used. The 3D model can be represented based on the following techniques: point-based, surface-based, volumetric-based, and model-based (Szeliski, 2010). The point-based representation can be seen in Furukawa and Ponce (2009), Goesele et al. (2007), and Habbecke and Kobbelt (2007). This representation makes use of techniques such as correspondence estimation, local region growing and filtering to build up a final dense surface by increasing the density of a point cloud model (Campbell et al., 2008). As illustrated in Figure 7.5(b), a synthetic image is created based on a dense point cloud using the implementation of Furukawa and Ponce (2009). However, comparing this figure to its original image in Figure 7.5(a), the synthetic image contains many untextured pixels. Producing a hole-free synthetic image from the point-based 3D model is, so far, impossible due to occlusion and incorrect point correspondences. Therefore, in this chapter, synthetic images are created based on the model-based technique.

The model-based method is used in this thesis due to its simplicity and the availability of the surface model of tunnels, already estimated in Chapter 4. This method fits a geometrical proxy to a 3D point cloud. The texture of the proxy is obtained by projecting colour from each image to the proxy. Once the 3D model is textured, it can be re-projected onto a required camera pose to create a synthetic image. The steps involved in creating the synthetic image are explained as follows, and in Algorithm 7.1.

7.3.1.1 Query image registration

A query image is registered on the current reference 3D coordinate frame using Bundler. The registration steps are as follows. Firstly, the SIFT features from the query image are extracted and their descriptor vectors matched to all descriptors in the existing reference images. The camera pose of the query frame is then estimated using the RANSAC pose estimation algorithm and then optimised by the least squares method. The camera pose is then refined via Bundle Adjustment. Figure 7.6 shows an example of a query image registered to the reference 3D point cloud in the beam dataset. The existing camera poses are shown in black and the query camera in blue.
7.3.1.2 Camera selection

In order to make a synthetic image as similar as possible to the query image, the closest reference camera pose to the query camera pose must be chosen. Therefore, a reference camera with its location closest to the query image and with a similar viewing angle to it is chosen automatically. The closest camera is found by comparing the Euclidean distance of a camera centre between all reference cameras and the query camera; the top five closest are chosen. These five cameras are then ranked according to the angles between their viewing direction, \( \theta_{rq} = \cos^{-1} \mathbf{n}_r \cdot \mathbf{n}_q \), where \( \mathbf{n}_r \) and \( \mathbf{n}_q \) are the unit vectors of the viewing directions of the reference camera and the query camera, respectively. The reference camera with the smallest angle \( \theta_{rq} \) is chosen.

An alternative to automatic camera selection is to specify manually a reference camera to be synthesised.

7.3.1.3 Synthetic view generation by model-based image rendering

In this method, the synthetic image is synthesised from a chosen reference image based on the estimated geometry of a scene. This method is suitable for a scene that does not contain complex geometries. However, estimating even simple geometry from a 3D point cloud is non-trivial by itself. In this work, scene geometry is performed semi-automatically,
Figure 7.6: An example of the registration of a query camera (shown in blue) to the reference 3D point cloud model; all reference cameras are shown in black.

requiring input from the user at various stages.

For the beam datasets, the user is required to specify a region of interest in one of the reference images. The 3D points cloud associated with this region is used to estimate a 3D plane using the least squares method.

For the tunnel datasets, a cylindrical surface is estimated using initialisation from the user. The initialisation parameters required are the radius, the directional vector and the centre of the cylinder. These parameters are then optimised by the least squares method.

The textures from a chosen reference image are mapped onto the 3D proxy and then these textured 3D coordinates are re-projected using the parameters of a query camera. The 3D rays from the chosen reference $X_i = [R_r, T_r]^{-1}K_r^{-1}x_i$ (the subscript $r$ denotes a reference camera) intersect the 3D proxy at $P_i$, and then $P_i$ is re-projected into the query camera frame as $p_i = K_q[R_q, T_q]P_i$ (a subscript $q$ denotes a query camera). Figures 7.7(a) and (b) are the original and warped reference images, respectively. Figure 7.8(b) shows the warped reference image of the tunnel scene; note that only overlapping region will be warped.
7.3.2 Photometric adjustments

As shown in Algorithm 7.1, two techniques used in photometric adjustments are explored: intensity normalisation and homomorphic filtering. The homomorphic filtering method is applied due to its simplicity as well as its relatively high performance based on a comparison of different photometric normalisation techniques described in Short et al. (2004).

7.3.2.1 Intensity Normalisation

Intensity normalisation is a simple, global technique whereby the pixel intensity values in one image are normalised to have the same mean and variance as another image by using the following equation (Radke et al., 2005)

$$\tilde{I}_2(x) = \frac{\sigma_1}{\sigma_2} \{I_2(x) - \mu_2\} + \mu_1$$  \hspace{1cm} (7.1)

where $\tilde{I}_2$ is the normalised image and $\mu_i, \sigma_i$ are the mean and standard deviation of the intensity values of $I_i$. This algorithm applies to grey scale intensity values.

Figure 7.9 shows an example of a warped image that has been normalised to match
the range of the intensity values of a query image. Visually, the normalised image, as shown in Figure 7.9(a), looks similar to the unnormalised one, as shown in Figure 7.9(b). However, Figure 7.9(c), which shows the intensity differences between Figures 7.9(a) and 7.9(b), reveals that the intensity of the pixels outside the concrete surface area has been modified more than that of those belonging to the concrete surface.

Figure 7.8: An example of a synthetic image based on the model-based method from the Prague dataset: (a) a chosen reference image, (b) a warped reference image.

Figure 7.9: (a) an original warped image, (b) a normalised warped image, (c) the difference in intensity values between image (a) and (b).
7.3.2.2 Homomorphic filtering

An image is modelled as a combination of light sources in the scene and the reflectance (Radke et al., 2005, Toth et al., 2000)

\[ I(x) = I_l(x)I_o(x) \]  \hspace{1cm} (7.2)

where \( I \) is a modelled image, \( I_l \) is an illumination component and \( I_o \) models the reflectance. A high pass filter \( F(X) \) is applied to the log image so that the components of the intensity signal can be separated as

\[ \ln I(x) = \ln I_l(x) + \ln I_o(x) \]  \hspace{1cm} (7.3)

The reflectance component can be found as

\[ \tilde{I}_o(x) = \exp \{ F(\ln I(x)) \} \]  \hspace{1cm} (7.4)

This method is applied to both query and reference images, as shown in Figure 7.10, in order to obtain the reflectance images. The implementation of this technique by Štruc and Pavešić (2010) is used. This implementation is a MATLAB toolbox containing various photometric normalisation techniques, which are commonly used in face recognition applications.

7.4 Change detection module

Once the input images have been pre-processed from Section 7.3, the change detection module is applied to determine the pixels or regions that contain significant changes. In this chapter, two change detection algorithms are applied: pixel differencing (Section 7.4.1) and motion-invariant change detection (Section 7.4.2).
Figure 7.10: An example of images processed by homomorphic filtering
7.4.1 Pixel differencing

This pixel differencing algorithm is based on a signed difference image \( D(x) = I_2(x) - I_1(x) \). The change mask \( B(x) \) is generated according to the following decision rules

\[
B(x) = \begin{cases} 
1 & \text{if } |D(x)| > \tau \\
0 & \text{otherwise} 
\end{cases}
\]  

where \( \tau \) is a specified threshold. The threshold is usually chosen empirically in order to produce high quality change masks for a given dataset.

7.4.2 Motion-invariant change detection

This work is based on Chen et al. (2008), and Zhou (2011) implemented this work in collaboration with the author. The pixel intensities of a sequence of images across time can be denoted as \( u(x, t_0), u(x, t_1), \ldots, u(x, t_i) \). The current intensity value \( u(x, t_i) \) can be obtained by evolving \( u(x, t_{i-j}), j \geq 1 \), through a spatial transformation of the spatial coordinates. Using \( u^t(x) \) to replace \( u(x, t_i) \) and \( u(x) \) to replace \( u(x, t_{i-j}) \), for the corresponding images \( u^t \) and \( u^t \), respectively, the transformation can then be expressed as

\[
u^t(x) = u[\phi(x, \theta)]
\]  

where \( \phi(x, \theta) \) defines a transformation model, \( x \) is the pixel 2D coordinate and \( \theta \) are the motion parameters. The above equation is known as the image constancy assumption. An initialisation is usually adopted for \( \theta = 0 \), which is \( u[\phi(x, 0)] = u(x) \). If the assumed transformation holds globally in the spatial domain for arbitrary images, then any image can be fully determined by another image and the spatial transformation model \( \phi(x, \theta) \) between them. When a crack develops or propagates, new pixels belonging to the crack region are usually much darker than the other pixels. These new pixels are not modelled by the aforementioned transformation model; hence, there is a residue at \( x \), which is expressed as \( |u[\phi(x, \theta^*)] - u^t(x)|^2 \), and the total residue over the spatial domain \( \mathcal{X} \) by \( r(\theta^*, t) = (\int_{\mathcal{X}} dx |u[\phi(x, \theta^*)] - u^t(x)|^2)^{1/2} \), where \( \theta^* \) are the ground truth motion parameters that fully account for the spatial transformation in the image frames from \( u \) to \( u^t \). If there are no cracks, \( r \) will yield a value of zero; whereas if there are cracks, \( r \) will be larger than zero. Therefore, \( r(\theta^*, t) \) is theoretically an ideal feature of crack occurrence, that is
invariant regarding the underlying motion of the structural component.

Replacing $\theta^*$ with an estimated motion parameter vector $\theta$ in the expression above, the total residue is given by $r(\theta, t) = (\int_X dx | u(x) - u'(x)|^2)^{1/2}$, and is the Euclidean distance (ED). In the manifold space, given that some potential crack occurrence may be left unmodelled and a certain amount of motion determined by $\theta^*$, the quantity $r(\theta^*, t)$ is the minimum distance between $u(\theta)$ and $u'$. This is called the Manifold Distance (MD), which can be formulated in an optimisation equation as

\[
\text{minimise} \quad r^2(\theta, t) = \int_X dx | u[\phi(x, \theta)] - u'(x) |^2
\]

subject to $u[\phi(x, 0)] = u(x)$ \hfill (7.7)

The above optimisation equation is invariant to the underlying motion of the structural component. As shown in Chen et al. (2008), the ED measurements display fluctuations due to the motion variation across time, while the MD measurements are narrow-banded due to their inherent invariance to motion.

For simplicity, $u[\phi(x, \theta)]$ is an approximation based on the transformation matrix $\phi(x, \theta)$ that assumes rigid-body motion only. The motion parameter vector is parameterised as $\theta = (\theta_1, \theta_3, \theta_3)^T$, where $(\theta_1, \theta_2)^T$ is the translation vector and $\theta_3$ is the rotation angle. The motion function $\phi$ is therefore expressed as

\[
\phi(x, \theta) = \begin{pmatrix}
\cos \theta_3 & \sin \theta_3 \\
-\sin \theta_3 & \cos \theta_3
\end{pmatrix} \begin{pmatrix}
x \\
y
\end{pmatrix} + \begin{pmatrix}
\theta_1 \\
\theta_2
\end{pmatrix}
\] \hfill (7.9)

Other transformation models are possible, although this one is sufficient for our current purpose. As equation 7.7 is non-linear, it is optimised numerically using Newton’s method. The details of this method can be found in Chen et al. (2008) and Zhou (2011).

7.4.2.1 Localisation by Outlier Analysis

The MD measurements computed at an arbitrary time are, in fact, a lump sum indication of the occurrence of concrete cracks in the image domain. They do not provide an indication of the location of the cracks (i.e. crack localisation). For the tunnel in-
pection problem, the localisation of the surface damage or the change in damage is of great interest to inspectors. Based on Chen et al. (2008), an image is divided into $K \times L$ subregions, upon which the MD measurements are estimated. The MD measurements for each subregion are then used in the outlier analysis in order to localise the changes in the concrete cracks in an image domain. It is suggested in Chen et al. (2008) that one needs a prior knowledge of the magnitude of the potential motion change in order to select a suitable size for a subregion. In Section 7.6.2.1, the effect of the size of the subregion on the performance of the change detection algorithm is studied.

An outlier analysis has been implemented, which assumes that the underlying features follow a normal distribution (Hodge and Austin, 2004). This means that, for a no-crack region, its MD values should lie about the mean values while, for a crack region, an MD value would lie at the tail of the normal distribution. Two input images that do not contain any changes between them, e.g. images at $t_0$ and $t_1$, are used to compute the sample mean $\mu_m$ and the sample standard deviation $\sigma_m$ of the MD measurements. $\mu_m$ and $\sigma_m$ are used to provide a null hypothesis against which other images (e.g. images at $t_k, k > 1$) may be tested. For each subregion, this null hypothesis takes the form

$$H_0 : |MD[i,j] - \mu_m| < \lambda \sigma_m$$

(7.10)

where $\lambda$ is a chosen fixed threshold and $1 \leq i \leq K$ and $1 \leq j \leq L$. Any subregion, whose MD values fail the test $H_0$, is classified as an outlier and labelled 0 in the binary mask. Other subregions are treated as normal MD values and are labelled 1 in the binary mask. The normal MD values can be used to update $\mu_m$ and $\sigma_m$. The output from the outlier analysis is a binary map, localising the regions where crack changes occur.

### 7.5 Evaluation method

#### 7.5.1 Evaluation data

#### 7.5.1.1 Datasets

**Beam I** This dataset is constructed from the beam images shown in Section 3.3. It contains a series of images of crack propagation from the middle camera (i.e. the focal length is 55mm). An example of the images in the series can be seen in Figure 6.14(a). Note that, in this dataset, the camera viewpoint does not change and the lighting is
assumed to be constant.

**Beam II** This dataset is also constructed from the beam images shown in Section 3.3. The images in this dataset consist of two parts: query images and reference images. The reference images are obtained from free camera movement at various positions. These images are used to obtain a 3D model and the camera positions of the beam images. The query images are chosen from the images taken from the middle camera (i.e. the one with a focal length of 55mm), as shown in Figure 6.14(a). In this dataset, the viewpoints and lighting in the reference images and query images vary.

**Prague** The reference images used in this dataset are those used for the mosaicing in Chapter 5, as shown in Figure 5.15. The query images are 4 images of the Prague tunnel, taken in 2007. In this dataset, the viewpoint and lighting variations are considerable, which is challenging for the proposed change detection system.

### 7.5.1.2 Ground truth change masks

The ground truth change masks in the beam datasets are obtained by manually labelling any pixels that belong to the crack regions, as shown in Figure 7.11(a). Firstly, the pixels are manually labelled for a pair of images. The labelled images are then transformed into binary images, as shown in Figure 7.11(b). Finally, a change mask is obtained by subtracting the binary images from the image pair.

**Region of Interest (ROI)** It is important to evaluate the performance of a change detection system only within a Region Of Interest (ROI), as shown in Figure 7.12. Misregistration or random signals outside the ROI could reduce the overall performance of a system, and it may prove impossible to design a system that operates well across an entire image.

Creating an ROI mask requires users to specify a region within an image that they wish to monitor. The mask is then applied to the image to mask out the other regions.
Figure 7.11: (a) the crack pixels are shown in red, (b) a binary image of the cracks and (c) the ground truth change mask.

Figure 7.12: An example of an image masked by the ROI.
7.5.2 Receiver Operating Characteristic curve

The performance of a binary change mask is quantitatively evaluated using the Receiver-Operating Characteristic (ROC) curve. This method is an addition to the evaluation method demonstrated in Chen et al. (2008), which only compares results visually. Visual comparison is qualitative and, therefore, should not be used as the only performance indicator for change detection systems.

The ROC curve is a plot between the True Positive Rate (TPR) and the False Positive Rate (FPR), computed by comparing two different change masks. The TPR and FPR are computed from the frequency of the occurrences of different outcomes when comparing a predicted mask against a ground truth mask, as shown in Figure 7.13. This comparison is a two-class prediction problem, in which there are four possible outcomes: True Positive (TP), False Positive (FP), False Negative (FN) and True Negative (TN). Subsequently, the values of the TPR and FPR can be estimated as:

\[
\text{TPR} = \frac{TP}{TP + FN} \tag{7.11}
\]

\[
\text{FPR} = \frac{FP}{FP + TN} \tag{7.12}
\]

Let \(u_g(x)\) and \(u_p(x)\) represent a binary value of the ground truth mask and the predicted mask, respectively. The binary value is either 0 (i.e. a change pixel) or 1 (i.e. an unchanged pixel). The resulting values of TP, FP, FN and TN are summarised in Table 7.1.

- A pixel is labelled as TP if \(u_g(x) = 0\) and \(u_p(x) = 0\), i.e. the predicted mask correctly detects a change pixel as changed.

- A pixel is labelled as FP if \(u_g(x) = 1\) and \(u_p(x) = 0\), i.e. the predicted mask incorrectly detects an unchanged pixel as changed. This is also known as a false alarm.

- A pixel is labelled as FN if \(u_g(x) = 0\) and \(u_p(x) = 1\), i.e. the predicted mask incorrectly detects a changed pixel as unchanged. This is also known as a miss.
Figure 7.13: (a) a change mask obtained from our experiments and (b) a manual ground truth mask.

<table>
<thead>
<tr>
<th>Predicted mask $(u_p)$</th>
<th>Manual mask $(u_g)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>TP</td>
</tr>
<tr>
<td>1</td>
<td>FN</td>
</tr>
</tbody>
</table>

Table 7.1: A summary of the classifications of a change mask.

- A pixel is labelled as TN if $u_g(x) = 1$ and $u_p(x) = 1$, i.e. the predicted mask correctly detects an unchanged pixel as unchanged.

**Ideal ROC curve**

Figure 7.14 shows an ideal ROC curve, which is a step function. This suggests that the system has a classification accuracy of 100%. In other words, with 0% FPR, 100% TPR is achieved. Unfortunately, the ideal curve is impossible to achieve even with a very high quality system, due to noise signals. A typical ROC curve has a rounded corner and tends towards the top left corner, as shown in Figure 7.14.
7.5.3 Remarks on image warping

In practice, query images should be warped onto a mosaic of reference images, which is more intuitive, for comparison. However, in the experiments presented here, reference images are warped onto query images for the following reasons.

Ground truth change masks are created from images taken by a fixed position camera, which are used as query images in the experiments. Images taken by the fixed camera are assumed to contain no geometric or illumination variation, and hence the ground truth masks are assumed to contain minimal inaccuracy. If query images are warped onto reference images, the ground truth masks also need to be distorted so that the change masks can be compared. Warping a ground truth mask involves pixel-interpolation, which will result in inaccuracy in the ground-truth data because this process generally introduces unwanted noise. Therefore, reference images are warped onto query images to avoid interpolation noise.

The change detection algorithm of Chen et al. (2008) requires the estimation of sample mean and variance. The sample mean and variance, however, can only be computed from multi-temporal images (i.e. a series of images taken over a period of time). Unfortunately, in the Beam II and Prague datasets, the reference images are not multi-temporal, and
the sample mean and variance cannot be estimated. Therefore, the reference images must be distorted onto query images instead, because the sample mean and variance can be estimated from the multi-temporal query images.

Ideally, the sample mean and variance should be estimated from reference images, which should be created as a multi-temporal image set, and the query images should be warped onto the reference images for comparison. However, due to the limitation of the datasets for the reference images, it is impossible to warp the query images onto the reference images.

7.6 Experiments

7.6.1 Comparison with vertical displacements

In this experiment, the algorithm of Chen et al. (2008) is evaluated by comparing the motion parameters obtained from the algorithm with the actual measurements obtained from LVDT sensors. As explained in Chapter 3 (Section 3.3), LVDT sensors are placed on top of the beam to obtain the vertical displacements, which are compared with those obtained from Chen’s algorithm.

7.6.1.1 Conversion

The vertical displacements from Chen’s algorithm in image space can be converted to object-space based on estimated motion parameters by a conversion factor or by calibration methods. A simple conversion factor, which can be applied globally (i.e. using the same conversion factor for an entire image), is estimated for this experiment as

\[ D(t_i) = \sum_{j=1}^{j=i} \theta_1(t_j) \times 0.091 \text{ mm/pixel} \]  

(7.13)

where \( D(t_i) \) is a converted vertical displacement and \( \theta_1 \) is an estimated motion parameter at time \( t_j \). The details of the conversion using other calibration methods can be found in Zhou (2011).
7.6.1.2 Results

The vertical displacements computed from an algorithm called GeoPIV White and Take (2002) are also compared with the results from Chen’s algorithm in this experiment. Figure 7.15 shows the plots of the LVDT readings against the vertical displacements from Chen’s and the GeoPIV algorithms.

It can be seen that the results from Chen’s algorithm are close to the LVDT readings, as shown in the slope as 0.9782, which is close to 1 in an ideal case. GeoPIV, on the other hand, performs slightly worse than Chen’s algorithm, as shown by the slope value of 0.8887.

The image patches used to estimate the motion parameters in Chen’s algorithm are located closer to the LVDT sensors than the patches used for GeoPIV. This may explain why Chen’s algorithm performs better than GeoPIV. However, it is worth noting that the results from GeoPIV do not include inaccuracy from radial distortion. If the radial distortion is corrected, GeoPIV may perform as well as Chen’s algorithm, and this requires further study.
7.6.2 Evaluation of Chen’s algorithm

In this experiment, Chen’s algorithm is evaluated using the Beam I dataset. The objective is to assess the performance of the algorithm based on Chen’s work in a quantitative manner, which was missing from the original evaluation methods in the paper (Chen et al., 2008). In collaboration with Zhou (2011), the evaluation methodology is improved through the use of ROC curves, which are obtained by comparing predicted change masks with ground truth masks.

Figure 7.16 shows a plot of the MD distances for all subregions. There are three sets of MD distance. The MDs of the training set are obtained from two images at the start of the sequence (i.e. at $t_0$ and $t_1$) when the changes between images are minimal. This can be seen as the blue line, where most MDs are of similar value. For the updating set, the MDs are obtained from images $t_i$ and $t_j$, where $j > i + 1$. Between these images, there may be small changes, as shown by the pink line, in which more subregions exhibit higher values of MDs. The new parameters (i.e. the mean and standard deviation of MDs) can then be used to update subsequent change masks. The testing MDs are shown in cyan. Based on the plot, it can be seen that some MDs are higher than average and, if they are above a trained threshold, a subregion will be defined as a change.

One conclusion from Zhou’s work is that the change masks produced by Chen’s work are able to detect changes in cracks from the beam dataset. The finest change in cracks that the algorithm could detect for the beam datasets is $0.5\text{mm}$. This is concluded using input images at time $t_1 = 0s$ and $t_2 = 40s$, as shown in Figures 7.17(a) and (b). In other words, the algorithm cannot detect any change between images that are taken before 40s ($t_2 < 40s$). It is impossible to spot any change via making a visual comparison of these images, whereas the algorithm is able to detect the changes as shown in the produced change mask in Figure 7.17(c). The smallest patch size used to detect these changes is $8 \times 8$, which corresponds to approximately $0.5\text{mm}$. This conclusion corresponds well to the actual vertical displacement, which is roughly $0.3\text{mm}$, based on $0.5\text{mm/min}$ for 40s, i.e. $0.5 \frac{\text{mm}}{\text{min}} \times 40s \simeq 0.3\text{mm}$. However, it was observed that the quality of the results are highly dependent on two parameters, the size of a patch $R$ and the threshold $\lambda$ (Section 7.4.2.1). These parameters are further studied in the following subsection.
Figure 7.16: The MD distances of all image patches, not all MDs are plotted for viewing clarity. The blue lines are estimated from images in the training set $t_0$ and $t_1$, the pink line is from images in the updating set $t_0$ and $t_2$, and the cyan line is from images in the testing set $t_0$ and $t_j, j > 2$
Figure 7.17: (a) input image at $t = 0$ s, (b) input image at $t = 40$ s, and (c) a change mask from Chen’s algorithm.
7.6.2.1 Patch size and threshold

The accuracy of the change masks depends on two parameters: the patch size $R$, and the threshold $\lambda$. As shown in Figure 7.18(a), the patch size may vary, e.g. 1x1, 2x2,..., NxN pixels. As suggested by Chen et al. (2008), the patch size should be at least larger than the expected movement in an image, although such expected movement can be difficult to predict in practice. The threshold $\lambda$ determines the cutoff from the mean $\mu_m$ for MDs to be considered outliers. As shown in Figure 7.18(b), if $\lambda$ is low, the cutoff is closer to the mean $\mu_m$; hence, many regions will be detected as changes. This may result in a large proportion of false detection. If $\lambda$ is too large, the cutoff is further from the mean, and fewer changes will then be detected.

To investigate the correlation between the parameters $R$ and $\lambda$, and the accuracy of a change mask, $R$ and $\lambda$ are varied to create different ROC curves. Since the accuracy is a function of $R$ and $\lambda$, the contour of the Accuracy Rate (AR) is plotted against varying $R$ and $\lambda$ instead. The Accuracy Rate (AR) is defined as

$$\text{AR} = \frac{\text{TPR}}{1 - \text{FPR}}$$ (7.14)

where, TPR and FPR are the True Positive Rate and False Positive Rate, respectively (see Section 7.5). Figure 7.19 shows a contour plot of the input images shown in Figure 7.17. In this plot, on the x-axis, $R$ varies from 10x10 to 100x100 and, on the y-axis, $\lambda$ varies from 1 to 6. The values of AR vary from a minimum 0 (in blue) to a maximum 0.6 (in red). From the plot, it can be seen that AR is highest when $\lambda \approx 2, R \approx 30$ and $\lambda \approx 3.5, R \approx 60$. This result suggests that certain combinations of parameters will work well for a given dataset. Currently, choosing a good combination of parameters is performed heuristically. Further study is required in order to select the parameters automatically.

It is noted that the aim of this experiment is to demonstrate that Chen’s algorithm is sensitive to patch size and threshold. The aim is not for it to be used as a guide for selecting values when creating a change mask. It is impossible to obtain sub-pixel accuracy using this algorithm. A change mask presented in this thesis may be used to identify the change regions, but its capability cannot be extended for change pixels. As suggested in Chen et al. (2008), the patch size should be larger than the expected movement of objects. In practice, predicting the expected movement is difficult, especially in tunnels, where it is
not uniform. It is impossible to select values for the patch size and the threshold without a loss of information in the real images. However, as shown in Figure 7.19, a reasonable change mask can still be obtained for a wide range of patch sizes and thresholds, e.g. the range of patch sizes between 20-100 with the threshold range of 1.5-3.5. Therefore, it may be concluded that a reasonable change mask can be obtained even when predicting the expected movement of objects is impossible, provided that the selected patch size and threshold values are loosely chosen.

The method of change detection presented in this thesis and Chen et al. (2008) can be used to provide preliminary information for change regions in images so that these regions can be examined more closely by other methods. The framework shown in Figure 7.20 demonstrates that a change detection algorithm can be used as an initial indication of damage in images so that they can be further analysed (e.g. quantifying the amount of damage), after which other unimportant unchanged images can be discarded.

7.6.3 Evaluation of the change detection systems

In this section, an evaluation of the proposed change detection system is conducted on the Beam II dataset. The images in this dataset contain viewpoint and illumination variation,
Figure 7.19: A contour plot of Accuracy Rate (AR) as a function of the threshold $\lambda$ and the grid size $R$.

Figure 7.20: An image-based structural damage monitoring framework from Chen and Hutchinson (2010).
and therefore the input images require pre-processing both geometrically and photometrically. Figure 7.21 shows the original input images, and the same input images after pre-processing. All of the results are shown in Figure 7.22, with the legends summarised in Table 7.2.

7.6.3.1 Image differencing

Figure 7.22(a) displays the ROC curves when the image differencing algorithm is applied during the change detection stage. The threshold $\tau$ in Equation 7.5 varies between 0.1 to 0.9 to produce the ROC curves. Only the area of change within the ROI is evaluated (Section 7.5.1.2).

As shown in Figure 7.22(a), Real Set has the best performance, followed by Real Set Homomorphic. As expected, these two sets are better than the others because the input images do not contain any geometric or lighting variations.

For the input images with viewpoint variation, the ROC curves (i.e. Homomorphic, No Radiometric, Intensity Normalisation), as shown in Figure 7.22(a), are similar at the low FPR region, i.e. $\text{FPR} < 0.05$. However, at the high FPR region, $\text{FPR} > 0.05$, the ROC curves from No Radiometric have the worst performance. This is as expected, since the variation in lighting between the two images is large, and without any radiometric pre-processing, the change detection algorithm will perform poorly on the images. It can be seen that, in the high FPR region, Homomorphic performs best.

7.6.3.2 Chen’s algorithm

Figure 7.22(b) shows the ROC curves when Chen’s algorithm is applied during the change detection stage. In this figure, the threshold $\lambda$ in Equation 7.10 varies between 1 and 30 to produce the ROC curves. A comparison is made within the ROI, and the patch size is fixed for all input image pairs at 20x20. For TPR up to 0.7, all of the curves are similar. However, when TPR > 0.7, it can be seen that No Radiometric performs the worst when compared with the other sets of results. The performance of Homomorphic is close to the benchmark results from Real Set. This result reinforces the view that, for Chen’s algorithm to work well, the images must be pre-processed by radiometric adjustment, because the algorithm relies on the assumption that the intensity values remain constant in multi-temporal images.
Image differencing vs. Chen’s algorithm  Figure 7.22(c) shows a comparison of different algorithms applied during the change detection stage, i.e. image differencing (denoted as Imdiff) and Chen’s algorithm (denoted as Chen). A summary of the input images for each plot is shown in Table 7.2(b). For the input images without viewpoint and radiometric variation, Real Set by Imdiff performs slightly better than Real Set by Chen. However, for the input images with pre-processing, Homomorphic by Chen works better than Homomorphic by Imdiff. Considering the results from Chen’s algorithm alone (i.e. Homomorphic by Chen and Real Set by Chen), although the input images contain radiometric and geometric adjustment, as shown in Figures 7.21(a) and (c), the algorithm still performs similarly to the benchmark input images (i.e. no viewpoint and lighting variation) from Figures 7.21(a) and (b). This suggests that detecting changes between images can be achieved even in images that have lighting and viewpoint variations, provided that appropriate pre-processing methods are applied.

Threshold and patch size effect  Figure 7.22(d) shows the ROC curves with different patch sizes from 10x10 to 300x300 pixels for Chen’s algorithm. As discussed in Section 7.6.2.1, the size of the patch and the threshold are two important parameters affecting the accuracy of Chen’s algorithm. This figure illustrates different ROC curves for different patch sizes. To determine the optimal patch size and threshold, a contour plot is produced, as shown in Figure 7.23. From this figure, it can be seen that the accuracy is highest with a patch size of $R \approx 40$ and a threshold of $\lambda \approx 15$. Patch size relates directly to movement in images and, as a rule of thumb, the patch size should be bigger than the expected movement. For example, if the expected movement is 10 pixels, the patch size should be at least 10x10 pixels. The threshold is empirically chosen.
Table 7.2: A summary of the legends

<table>
<thead>
<tr>
<th></th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
<th>(e)</th>
<th>(f)</th>
<th>(g)</th>
<th>(i)</th>
<th>Geometric Pre-processing</th>
<th>Radiometric Pre-processing</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homomorphic</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes, Homomorphic used</td>
</tr>
<tr>
<td>No Radiometric</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>No</td>
</tr>
<tr>
<td>Intensity Normalisation</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>Yes</td>
<td>Yes, Intensity Normalisation used</td>
</tr>
<tr>
<td>Real Set</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
</tr>
<tr>
<td>Real Set with Homomorphic</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>Yes, Homomorphic used</td>
</tr>
</tbody>
</table>

(a) A table summarising the legend in the input images from Figure 7.22(a) and (b)

<table>
<thead>
<tr>
<th></th>
<th>(a)</th>
<th>(b)</th>
<th>(c)</th>
<th>(d)</th>
<th>(e)</th>
<th>(f)</th>
<th>Geometric Pre-processing</th>
<th>Radiometric Pre-processing</th>
<th>Change detection algorithm</th>
</tr>
</thead>
<tbody>
<tr>
<td>Homomorphic by Chen</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>Yes</td>
<td>Yes, Homomorphic</td>
<td>Yes, Chen’s algorithm</td>
</tr>
<tr>
<td>Real Set by Chen</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
<td>Yes, Chen’s algorithm</td>
</tr>
<tr>
<td>Homomorphic by Imdiff</td>
<td></td>
<td></td>
<td></td>
<td>✓</td>
<td>✓</td>
<td></td>
<td>Yes</td>
<td>Yes, Homomorphic</td>
<td>Yes, Image differencing</td>
</tr>
<tr>
<td>Real Set by Imdiff</td>
<td>✓</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>No</td>
<td>No</td>
<td>Yes, Image differencing</td>
</tr>
</tbody>
</table>

(b) A table summarising the legend in the input images from Figure 7.22(c)
Figure 7.21: Input images, (a) $I_q$ is the original query image taken at $t = 60s$, (b) $I_o$ is the original reference image taken from the same viewpoint as $I_q$ at $t = 0s$ and (c) $I'_o$ is the original chosen reference image taken from a different viewpoint to $I_q$ at $t = 0s$, (d) $I_q$ with homomorphic filtering, (e) $I_o$ with homomorphic filtering, (f) $I'_o$ with homomorphic filtering, (g) a ground truth change mask, (h) $I_o$ with intensity normalisation, (i) $I'_o$ with intensity normalisation.
Figure 7.22: The ROC curves of change masks using two different methods of change detection, (a) image differencing, (b) Chen’s algorithm, (c) comparison between image differencing vs. Chen’s algorithm, (d) the effect of varying patch sizes and thresholds.
7.6.3.3 Evaluation with other viewing angles

The experiments are conducted on the images from the left and right cameras in the beam experiment (Section 3.3). Cracks are less apparent on these images due to the viewing angles and the distance of the cameras from the beam, as shown in Figures 7.24(a) and 7.25(a).

Figure 7.24(a)-left shows an input query image and Figure 7.24(a)-right shows a reference image. The latter is chosen from the reference set and is the most similar to the query image. Figure 7.24(b) shows the query and reference images that have been geometrically and photometrically registered. The reference image, as shown in Figure 7.24(b)-right, has been rectified using homography based on the front surface of the beam. Note that the other parts of the image are not expected to be aligned because they do not necessarily lie on the same plane as the beam’s surface. Chen’s algorithm is applied to Figure 7.24(b) and the result is shown in Figure 7.24(c). The patch size used in these images is 40x40 and the threshold is 10. These values show a reasonably good change mask when assessed visually. If the optimal values from Section 7.6.3.2 are used (i.e. $R = 40 \times 40$, $\lambda = 15$), the change mask is shown in Figure 7.24(d), which does not seem as accurate as Figure 7.24(c).
These optimal values do not work well because the images used here are of different scales (i.e. a different amount of zoom is used) to those evaluated in Section 7.6.3.2. For the optimal values to work well, the images should have a similar degree of zoom. The size of the image patches largely depends on the degree of zoom, as the patches have a different appearance at different scales. Mikolajczyk et al. (2005) show that, at different scales, the rate of recognition accuracy of the features or image patches varies. In the SIFT detector (Lowe, 2004), only local image patches that are scale-invariant are retained, and different stages of the image processing are required in order to select these scale-invariant patches automatically.

The work in this thesis follows the method described by Chen et al. (2008), who apply a fixed patch size throughout their experiments. The results of Chen’s work are valuable because the images used in his experiments have a fixed focal length (i.e. there is no change in the zoom). The images presented in this thesis, however, contain a varying degree of zoom so the patch sizes should be changed accordingly. The optimal values found in Section 7.6.3.2 may be used as a guideline for estimating a suitable patch size in the input images in this section, although these values will not be optimal for them. Since the patch size applied to this view is not optimal, the value of the threshold \( \lambda \) must then also be modified in order to obtain a good change mask. The automatic selection of a patch size based on the degree of image scales would be possible if a scale-selection algorithm is used, similar to that used in the SIFT algorithm, but this lies beyond the scope of this thesis.

A quantitative comparison cannot be made to obtain the true accuracy of the change masks since there is no ground truth available for the images shown in Figure 7.24. The regions coloured in red in Figure 7.24(c) are change masks that indicate change regions. In Figure 7.24(c), the change regions are mostly centred on the big diagonal cracks, which correspond well with the mask obtained for the input images with a frontal view, as shown in Figure 7.2(c) and the ground truth 7.2(d). The change regions around the loading plate are mostly false alarms, caused by shadow from the loading plate. More false alarm change regions can be seen around the edges of the beam. The hairline cracks near the bottom of the beam are not detected in the change mask in Figure 7.24(c), and these are miss regions. Hairline cracks are difficult to detect and further improvement to the proposed system is required. However, some of the large cracks in these images are detected, which demonstrates the usefulness of the proposed change detection system.

Similarly, Figure 7.25 shows all of the results from the camera on the right. Figure
7.25(a) shows the input images, 7.25(b) shows rectified images, and 7.25(c) shows the change mask. Again, the change mask corresponds well with the diagonal crack regions, although more false negative regions are labelled around the loading plate. This change mask is similar to the mask from the left camera in Figure 7.24(c). Note that the patch size and threshold are 40x40 and 10 in Figure 7.25(c) while, in Figure 7.25(d), the patch size and threshold are 40x40 and 15 (i.e. this value is obtained from the result shown in Section 7.6.3.2). Again, the patch size and threshold are empirically chosen. If the threshold value is low, more regions will be identified as changed, increasing the false positive rate. If the patch size is smaller, the change mask will contain a larger amount of noise; hence, the false positive rate will also be increased.

7.6.4 Evaluation with underground images

The proposed change detection system is applied to the tunnel datasets (Mustek Station, Prague Metro). The query image is shown in Figure 7.26(a), taken in 2003. Figure 7.26(b) shows four chosen reference images that have been pre-processed and overlaid on top of each other. All of the reference images were obtained in 2008. All four images are used in comparison with the query image so that a change mask will cover an entire area of the query image. By visually comparing Figure 7.26(a) and (b), several changes can be seen, as labelled in Figure 7.26(a). These changes include, a change in the colour of the water patches, the appearance of new holes, and the addition of a new cable.

The query and reference images are pre-processed using Bundler and homomorphic filtering. Then change masks shown in Figure 7.27 are produced by applying the image differencing algorithm at the change detection stage to each reference image, after which a final mask is combined from the individual masks. Let \( M_i \) be a change mask between a single reference image and a query image; hence, the masks from each reference image can be denoted as \( M_1, M_2, M_3 \) and \( M_4 \). A final change mask is then combined as \( M_f = M_1 \cup M_2 \cup M_3 \cup M_4 \). The method for producing a final mask is conservative because a region is labelled as changed if only one region from one of the change masks is labelled as changed. Hence, the over-estimation of a number of change regions can be anticipated. Over-estimating the change regions may be considered a conservative approach, because it is better to allow more false alarms than to miss the detection of change regions.
Figure 7.24: An example of the results from the left camera: (a) an example of input images, (b) input images after pre-processing, (c) the change mask result for $R = 40 \times 40$ and $\lambda = 10$, (d) the change mask result for $R = 40 \times 40$ and $\lambda = 15$. 

199
Figure 7.25: An example of the results from the right camera: (a) an example of input images, (b) input images after pre-processing, (c) the change mask result for $R = 40 \times 40$ and $\lambda = 10$, (d) the change mask result for $R = 40 \times 40$ and $\lambda = 15$. 

200
Missing the detection of some critical regions may lead to more detrimental effects in the structural components.

Figure 7.27 shows change masks at different levels of threshold $\tau$, ranging from 0.1-0.6. The pixels labelled as change are shown in red. Note that, for the image differencing algorithm, change regions means change pixels. As shown in Figure 7.27(a), when $\tau = 0.1$, many pixels are labelled as changed, although most of them are false alarms. It can be seen that, as the threshold increases from 0.1 to 0.6, the number of changed pixels decreases. In all of the figures, most of the changed pixels are false alarms, occurring in the area of strong shadow. This is due to the photometric correction being unable to eliminate the shadow effect. The results of the change masks rely substantially on the thresholds, and, in practice, a pre-selected threshold is difficult to choose. Based on these results, it is clear that the proposed change detection system requires substantial improvement.

A better change mask can be obtained by improving the geometric adjustment module, since the current proposed algorithm still produces a small amount of misalignment between images. When pixels are not perfectly aligned due to inaccuracy in Bundler, they may be labelled as changed. A change due to misalignment is undesirable, and misalignment should be treated at the geometrical adjustment stage. Another issue is shadow, which mostly occurs near the boundaries of protruding objects, such as on cables. Shadow effects cannot be easily eliminated by the homomorphic filtering method.

7.6.5 Discussion

A large amount of false alarm pixels can be observed in all masks (see Figure 7.27). These false alarm pixels are due to inaccuracy in both the geometric and radiometric adjustment steps. For geometric inaccuracy, pixel misalignment can occur when the camera pose of a query image is inaccurately determined. Inaccurate camera poses cause a slight pixel shift when synthesising a query image, which leads to a change mask becoming noisy, as observed in all masks (see Figure 7.27).

Shadow cannot be corrected by radiometric adjustment, especially in the tunnel datasets. As observed in Figure 7.27, the shadow behind the cables is labelled as a change in all change masks. Another example can be observed in the Beam dataset shown in Figures 7.2(c), 7.24 and 7.25, where the areas around the loading plate are classified as changed because of a strong shadow effect. The algorithm used to correct light variation in images, such as the homomorphic filtering algorithm applied in this thesis, is unable to remove the
shadow. Therefore, the change detection algorithm of Chen cannot perform well, since too much variation in pixel intensities violates its assumption. Removing the shadow effect in images is non-trivial, as suggested in some face recognition literature (e.g. Short et al. (2004)).

Chen’s algorithm is sensitive to the size of a subregion $R$ and a threshold $\lambda$. Different values of $R$ and $\lambda$ give different change mask results. Without prior knowledge of the expected movement in a scene, choosing suitable values for $R$ and $\lambda$ is challenging. As discussed in Section 7.6.2.1, choosing suitable values for $R$ and $\lambda$ without the loss of information is very difficult and impractical, especially in real images from tunnels, where the movement of the tunnels is difficult to predict. The effects of $R$ and $\lambda$ on the accuracy of the change masks and how best to select them require extensive further study.

### 7.7 Conclusion

- The performance of the proposed change detection system is insufficient for practical use, and much further study is required.

- Inaccuracy in the geometric adjustment step by Bundler causes false alarms in the change masks for all datasets.

- Strong variations in lighting due to shadow cannot be corrected by the intensity normalisation and homomorphic algorithms in the radiometric adjustment step, and inaccuracy in the change masks in all datasets is observed.

- The accuracy of the change detection algorithm of Chen depends on the threshold $\lambda$ and the patch size $R$. 
Figure 7.26: (a) a query image, (b) four reference images after pre-processing and being overlayed on top of each other, (c) individual reference images after the geometric adjustment step.
Figure 7.27: The result of change masks from the Prague dataset with various thresholds \( \tau \), ranging from \( \tau=0.1-0.6 \).
Chapter VIII

Conclusion

8.1 Summary

Inspection plays an important role in the maintenance of infrastructure. It is an integral part of structural health monitoring systems, designed to assess the condition of structural components and determine whether or not they are fit-for-purpose. One of the most frequently adopted inspection techniques is visual inspection, which is carried out by inspectors. It has a number of limitations, such as the high labour costs associated with carrying out tasks, and inaccuracy due to subjectivity. A significant amount of research has been carried out to develop automated inspection systems to improve on manual visual inspection. In the field of civil engineering, a number of inspection systems have been developed to utilise inexpensive technologies based on digital videos and cameras. Examples of these systems include crack detection systems (Yamaguchi and Hashimoto, 2006, Abdelqader et al., 2006), crack monitoring systems (Chen and Hutchinson, 2010), and systems to allow the fast acquisition of video data (Yu et al., 2007). Technologies in image processing and pattern recognition have been applied to detect, localise and quantify damage in monitored structural components from video or image data. However, none of these systems offer complete solutions and a significant amount of work is still required to create an automatic inspection system that can be adopted by the industry.

The advancement of computer vision technologies offers the possibility of creating an automatic system for inspection. This thesis describes a system, which is based on state-of-the-art computer vision technologies, to aid the visual inspection of tunnels. The system uses a standard digital camera, which is cheap compared to other technologies, such as LiDAR or infrared cameras. The system aims to aid inspection based on the two following themes and contributions.
Novel method for inspection and reporting  This thesis proposes a framework for creating a large mosaic from images of tunnel linings. A traditional inspection report is usually a collection of individual images, which is difficult to visualise and does not provide a sense of where the images are taken. However, novel inspection reporting combines these individual images into a large mosaic image, making it far easier to visualise a large section of the tunnel. The research provides a system that enables inspectors to create a mosaic for inclusion within an inspection report in an automatic or interactive approach.

Interactive multi-view change detection  The main concern with regard to change detection systems is accurate registration, both geometrically and photometrically. This thesis proposes a framework for change detection so that multi-temporal images can be compared even though they have different viewpoints and illumination. A query image can be registered to existing image data automatically via Structure From Motion and any changes between the images can be identified.

Based on the above themes, this thesis makes specific contributions as follows.

8.1.1  Mosaicing via robust surface estimation

This contribution is mainly based on the work described in Chapter 5. A framework is proposed to create a mosaic image from tunnel images. It is based on Structure From Motion, which recovers a sparse 3D point cloud and camera parameters from uncalibrated images. Each image is warped using an estimated tunnel surface geometry and then input to stitching software to create a final mosaic image. The framework exploits the simple geometry of a tunnel, which is a cylinder and a developable surface, to produce an almost distortion-free mosaic.

The accurate estimation of the surface parameters is an important factor for obtaining a mosaic containing a low degree of distortion. This thesis provides two approaches to surface estimation. The first is Support Vector Machine (SVM) classification, which is used to classify the surface points lying on and off the surface. Points classified as lying on the surface are used in the estimation of the tunnel surface, which is computed by the non-linear least squares method. Another approach relies on users interactively providing an initial estimation of the surface parameters.
It is shown that inaccurate surface parameters result in distortion in a final mosaic, such as curvature. However, with accurate parameters, the mosaic preserves all of the physical entities, such as line straightness, $90^\circ$ between horizontal and vertical lines, and parallelism. This type of mosaic image with little distortion can only be achieved if the geometry of a scene contains developable surfaces.

An SFM system in the proposed framework allows the mosaicing of images with general camera motion. Commercial stitching software only provides the mosaicing of images based on planar and cylindrical projection models. This can only be achieved if the images are taken with a camera fixed at one point. The SFM provides greater flexibility with regard to camera motion when mosaicing.

8.1.2 Spatially consistent matching

Obtaining accurate feature tracks is a crucial step in SFM systems. Incorrect tracks can result in inaccurate or unsuccessful reconstruction. Chapter 6 shows one of the contributions of this thesis that is devoted to improving algorithms for feature matching. A number of feature matching algorithms rely on matching features based on the local appearance of image patches. This thesis proposes improved feature matching algorithms that utilise a spatial constraint.

Two algorithms are proposed in this thesis. Both rely on a spatial consistency constraint. This constraint uses the fact that features can be matched if the spatial arrangement among their neighbourhood features is similar. The details of the algorithms can be found in Chapter 6. The first algorithm can only work with small-baseline images. The first method increases the inlier ratios in all of the testing datasets. However, this algorithm still relies on choosing a suitable threshold for the algorithm to work well. The threshold must be carefully chosen to prevent correct matches being removed if the threshold is too tightly bound.

The second algorithm applies the Random Forest algorithm to match features by appearance, and the modified Shape Context algorithm to match features using a spatial constraint. It is found that this algorithm performs comparatively well with the state-of-the-art SIFT matching algorithm on the Mikolajczyk dataset. However, the algorithm performs better than the SIFT matching algorithm on the Beam dataset, which contains a deformable object in the scene. This is expected because the Shape Context algorithm exploits the spatial consistency constraint to reduce ambiguity in the matches. The al-
algorithm, however, relies on two important parameters, namely the number of trees in the RF algorithm and the tuning parameter in the Shape Context algorithm.

8.1.3 Multi-view change detection

The final contribution of this research is a change detection system that applies to images with variations in terms of their viewpoint and illumination. Many change detection systems in the civil engineering literature are designed to monitor crack changes under simple imaging conditions which do not contain viewpoint or illumination variations. This type of system is impractical for real world images. This research proposes a framework that aims to cope with variations in real world images.

The framework corrects variations in viewpoint by geometrical registration using Bundler, which registers a query image to the existing database previously reconstructed to obtain a 3D model and camera parameters. The query image is then registered and compared with a chosen image from the existing ones using the geometry proxy of a scene. The query and reference images are photometrically registered to eliminate the effect of illumination variation. Two algorithms are used in photometric registration: intensity normalisation and homomorphic filtering. It is found that the latter algorithm provides more accurate results. Finally, after the query and reference images are pre-processed, a change detection algorithm is applied to determine the change regions between the images.

Two change detection algorithms are applied for comparison: image differencing and Chen’s algorithm. In most testing datasets, Chen’s algorithm produces better results than the image differencing algorithm. However, the accuracy of Chen’s method depends on two important parameters: the size of patch $R$ and the threshold $\lambda$. Suitable parameters must be chosen for a given movement of crack change in images in order to obtain optimal accuracy in a change mask. Choosing these parameters is difficult for real tunnel images as the tunnel movement is not uniform. The results from the experiments demonstrate that the change masks created for the tunnel datasets are poor and inadequate for practical use. It is concluded that further study is required to find better methods for the pre-processing steps.

8.2 Discussion

The proposed system for mosaicing is practical for tunnel inspection. It only requires a standard digital camera for the hardware, and the software can be run on a standard
home computer. With further development of the user interface for the proposed system, inspectors can adopt this system in their actual inspection work. With the novel method of inspection reporting, inspectors can now easily produce a mosaic with little distortion with minimal user interaction. The mosaic helps us easily to translate all of the concerned anomalies on the tunnel surface into animated diagrams in an inspection report. This system can be extended to reconstruct mosaic images for a larger number of images using more powerful computers.

The system presented in this thesis is based on an SFM system (i.e. Bundler), which can offer the potential for further extension. One application is to integrate a point cloud obtained from an SFM system to CAD models or a LiDAR point cloud so that the image texture provided by the SFM system can enhance the visualisation of the CAD models or the LiDAR data. At the time of writing, SFM systems have become far more advanced, and can now be used to reconstruct a 3D model of an entire city. Recent examples of 3D urban modelling are Agarwal (2009), Irschara et al. (2007) and Pollefeys et al. (2008). It is therefore possible to reconstruct mosaic images of tunnels for hundreds of kilometres.

The proposed change detection system is based on an SFM system and new images can be compared with those in the existing database. SFM systems provide the possibility of easily creating an archive of images taken earlier and then new images can be compared with the ones in the archive to monitor any changes in terms of damage. However, the proposed change detection system requires further study for real tunnel images.

8.3 Suggestions for future work

8.3.1 Image acquisition system

For the inspection of a small section of a tunnel, a handheld digital camera and tripod are sufficient to obtain images within the time limit set for engineering hours. However, to acquire a large number of images within the time available, special hardware is required. The hardware should obtain images in the fashion outlined in Chapter 3, Section 3.1, to ensure that the images achieve those properties required for SFM systems. Currently, working in partnership with Cambridge Toshiba Research, two hardware designs are being investigated. These designs serve as prototypes for future commercialisation. The design in Figure 8.1(a) contains only one camera, which is rotated by a motor to obtain pictures for each ring of tunnel lining. The design in Figure 8.1(b) contains multiple cameras that aim to take multiple pictures synchronously to cover an entire ring by a single trigger.
8.3.2 Automatic primitive extraction

To obtain a distortion-free mosaic, accurate estimation of the geometry of surfaces must be achieved as shown in Chapter 5. The geometry of all tunnel datasets in this research is mainly cylindrical, except the Barcelona dataset which is a semi-circular arch. When a scene only contains one geometry primitive, automatic extraction of primitives is achievable as shown in this thesis using Support Vector Machine classification. However, if a scene has more than one primitive, such as in the Barcelona datasets, which is composed of two planes and a cylinder, automatic extraction becomes a non-trivial problem. Currently, the commercial software Autocad can automatically extract only one primitive, which is a plane. Therefore, automatic extraction of multiple primitives is required. Schnabel et al. (2007) proposes a framework to extract primitives from laser scanned data based on RANSAC. This method, however, works with a dense point cloud, such as LiDAR data. This is not practical for the data in this thesis because only a sparse point
cloud is available. Further development is required to develop algorithms for automatic primitive extraction so that it works with a sparse point cloud as well as a dense point cloud. Also, an engine to provide users with tools to manually extract primitives should also be developed.

8.3.3 Integration with CAD models

Many modern tunnels are designed based on CAD software. There are a number of problems in CAD models. They have synthetic texture mapped onto the models. Visualisation of CAD models does not provide a sense of realism due to the lack of real texture. The models, however, can be integrated with a sparse point cloud obtained from an SFM system. The image texture from an SFM system can then be texture-mapped onto CAD models to enhance visualisation. However, automatic 3D registration to align CAD models and SFM models is still an active research area. Nevertheless, interactive tools should be developed to provide users with the ability to manually align 3D models. Such further development requires a robust way of image rendering so that integrated models can be navigated in a seamless manner.

8.3.4 Damage quantification

The change detection system proposed in this work can approximately localise the region of changes between query and reference images. However, there is no information to indicate the amount of change in images. Such information is useful for sensor installation so that areas with more severe damage are prioritised for monitoring. This helps in determining the allocation of sensors to achieve the best scheme for sensor installation. Chen and Hutchinson (2010) propose a framework to quantity crack damage in change regions using level sets. This approach is attractive for crack monitoring. However, further study is required to extend quantification of damage to general anomalies, such as water patches.
Appendix

A.1 Evaluation method by Mikolajczyk

The evaluation procedure follows Mikolajczyk and Schmid (2004) using the Mikolajczyk datasets (see Chapter 3, Section 3.2.3), which studies the relative performance of different types of descriptor. The performance is evaluated from the plot of recall against $1 - precision$ with varying distance thresholds. Let us define these parameters as $\text{recall} = \frac{\#\text{correct matches}}{\#\text{correspondences}}$ and $1 - \text{precision} = \frac{\#\text{all matches} - \#\text{correct matches}}{\#\text{all matches}}$, where the terms correct matches, correspondences and all matches are defined as follows.

**Correct matches** In Mikolajczyk and Schmid (2004), an affine-invariant detector is represented by an elliptical region, that is used to determine the match correctness by the amount of overlap between two regions. Let us define a region or a feature $A$ with its centre located at $x_A$ and its descriptors as $f_A$ in image $I$ and define the set of all regions in this image as $V_I$. The same definition applies to a region $\tilde{A}$ located $x_{\tilde{A}}$ with the descriptors $f_{\tilde{A}}$ in an image $\tilde{I}$ and the set of all regions $V_{\tilde{I}}$.

The match correctness is determined by the overlap error (Mikolajczyk et al., 2005). The overlap error measures how well a region $A$ from an image $I$ corresponds to a region $\tilde{A}$ from an image $\tilde{I}$ under a known homography $H$. The overlap error is defined from the ratio of the intersection and union of the regions as

$$\varepsilon_S = 1 - \frac{A \cap H^T \tilde{A} H}{A \cup H^T \tilde{A} H} \quad (1)$$

Two regions are assumed to be a correct match if $\varepsilon_S < 0.5$. A set of matching features with $\varepsilon_S < 0.5$ can be defined as $E = \{(A, \tilde{A}) | \varepsilon_S < 0.5\}$

A match between two descriptors $f_A$ and $f_{\tilde{A}}$ will differ depending on the matching strategy. There are three strategies: Similarity matching (Sim), Nearest Neighbour-based matching (NN), and Ratio Matching (RN).

1. In Similarity matching, the descriptors $f_A$ and $f_{\tilde{A}}$ are matched if the Euclidean dis-
tance between them is below a threshold. This strategy generally leads to many-to-
many matching between a set of regions from the images \( I \) and \( \tilde{I} \). A set of matching features based on the Similarity matching strategy is defined as \( S \). Therefore, a set of correct matches based on the Sim strategy is given by \( \{(A, \tilde{A}) | (A, \tilde{A}) \in E \cap S\} \).

2. In *Nearest Neighbour-based matching*, the descriptors \( f_A \) and \( f_{\tilde{A}} \) are matched if \( f_{\tilde{A}} \) is nearest to \( f_A \) based on the Euclidean distance. This approach, however, can sometimes result in many-to-many matching. To obtain one-to-one matching, the approach is modified such that the matching of the bipartite graph between \( V_I \) and \( V_{\tilde{I}} \) minimises the total distance cost, i.e. optimal matching. With this approach, one-to-one matching is guaranteed. Let us define a set of matching features based on the NN strategy as \( N \); hence, a set of correct matches based on NN is given by \( \{(A, \tilde{A}) | (A, \tilde{A}) \in E \cap N\} \).

3. In *Ratio Matching*, the matches from the NN strategy are accepted if the ratio of the distance of the first nearest neighbour matching \( ||f_A - f_{\tilde{A}}|| \) to the second nearest neighbour \( ||f_A - f_{\tilde{B}}|| \) is less than a specified threshold, i.e. \( \frac{||f_A - f_{\tilde{A}}||}{||f_A - f_{\tilde{B}}||} < \tau \). The reason for this is that, when the distances of the first and the second nearest neighbours are too short, it is unlikely that the first nearest neighbour is a correct match. In other words, the descriptors are not far apart enough from their neighbours in the feature space. Let us define a set of matching features based on the RN strategy as \( R \); hence, a set of correct matches based on RN is given by \( \{(A, \tilde{A}) | (A, \tilde{A}) \in E \cap R\} \).

The number of correct matches \( \# \text{correct matches} \) is the number of members in a set of correct matches as defined above.

**All matches**  The number of all matches, \( \# \text{all matches} \), is the total number of matches obtained from matching the descriptors by one of the matching strategies. Therefore, \( \# \text{all matches} \) is given by \( |S|, |N| \) and \( |R| \) for the Sim, NN and RN matching strategies, respectively. This parameter is varied by changing the parameters of the matching thresholds to create a curve of recall from high precision to low precision.

**Correspondence**  The number of correspondences (possible correct matches) is the maximum matching size of the correct matches in a bi-partite graph. Let \( U \) and \( \tilde{U} \)
be a set of all of the features from image $\mathcal{I}$ and $\tilde{\mathcal{I}}$, respectively. Let $V \subset U$ and $\tilde{V} \subset \tilde{U}$ be a set of matching features belonging to the set $E$. In a bi-partite graph, a feature in $V$ can match multiple features in $\tilde{V}$, but there must be at least one link for each feature. Therefore, the minimum number of matches between $V$ and $\tilde{V}$ is $\max(|V|, |\tilde{V}|)$. For the NN and RN strategies, only 1-to-1 matching is allowed; hence, the number of correspondences $\#\text{correspondences}$ is given by $\min(|V|, |\tilde{V}|)$. For the Sim strategy, $\#\text{correspondences}$ is given by the total number of links between $V$ and $\tilde{V}$. 
Bibliography


Soga, K. (2007). “Micro-measurement and monitoring system for ageing underground infrastructures (underground m3).” *World Forum on Smart Materials and Smart Structures Technology: Proceedings of the World Forum on Smart Materials and Smart Structures Technology (SMSST’07)*, Taylor & Francis, Chongqing & Nanjing, China (May 22-27,).


Research Report CTU–CMP–2004–08, Center for Machine Perception, K13133 FEE Czech Technical University, Prague, Czech Republic (June).


229


