Self localisation in indoor environments using machine vision

Nabeel Younus Khan

a thesis submitted for the degree of

Doctor of Philosophy
at the University of Otago, Dunedin,
New Zealand.

17 March 2014
Abstract

The performance of outdoor positioning has become excellent with the emergence of Global Positioning System (GPS), but GPS is not reliable indoors. The ability of a system to perform indoor positioning without GPS is still challenging and has gained a lot of attention in recent years.

Indoor positioning has become a focus of research during the past decade. Despite a lot of research efforts, existing indoor positioning systems based on different technologies are still limited because most of them either require expensive infrastructure (ultrasound), offer limited coverage (Wi-Fi, Bluetooth) or provide low accuracy (audible sound). On the other hand, machine vision offers the potential for a cheap and effective solution for robust indoor positioning.

This thesis describes the research, experiments and analysis conducted to develop a machine vision based system, known as “Indoor Positioning System (iPoS)”, which can provide reliable positioning in indoor environments. iPoS is based on a client server model where the client is a smartphone application and the server uses the proposed “BoWLocator” algorithm to match the incoming query image from the application. The key approach to the system is the use of minimum information i.e. a single image of a location from the phone camera for localisation. The main purpose of iPoS is to use it as a navigation aid for blind people and guide them while they move in unfamiliar indoor environments because they often feel lost in the absence of current location information.

To create a reliable indoor positioning, iPoS uses three proposed components (1) voting module, (2) homography verification method, and (3) post-verification method. iPoS has been demonstrated to localise on four realistic datasets covering a total of 50 indoor self-similar locations with a correct acceptance rate of 72-93% with few wrong matches depending on the test set and queries typically require 5-14 seconds on average to return a result. iPoS gives a very low localisation error with an average wrong match rate of 5.5%.
Acknowledgements

There are many people that I have to thank because without them, I would not be the student that I am today.

First of all, I would like to thank Associate Professor Brendan McCane for accepting me as a Ph.D. student and supervising my work. The current work has only been possible due to his valuable guidance throughout my PhD. I am also grateful to Professor Geoff Wyvill who agreed to be my second supervisor. His useful advice and remarks has played an important role during the early stages of my PhD. I am especially grateful to Dr. Steven Mills who has served as my third supervisor. Steven’s expertise in 3D modeling has helped in achieving the milestones with ease during the last year of my PhD. I have learned a great deal from my supervisors about research, computer vision and I deeply cherish the time I have spent working with them. They always had time to answer my silly questions and always gave me a constructive feedback.

I would like to thank all members of my “Graphics and Vision” lab who were fundamental in order to have a good and fun time. I am glad of having them as colleagues, especially to Nurazlin Zainal Azmi, Umair Mateen Khan and Xiping. I am also grateful to Jie Fu with whom I used to have creative discussions regarding my work. I have also great regard for the staff of Computer Science department for their support throughout my studies.

I am also grateful to Otago University for awarding me a PhD scholarship and providing me an opportunity to spend a quality time in New Zealand, the “Youngest Country” of the world.

Finally and most importantly, I would like to express my deepest gratitude to my parents for their continuous and selfless support. They always have been the source of inspiration for me throughout my life and without them I would not be in the current position. I will also like to thank my wife, Amna who has been always very understanding and provided her faithful support in writing this work.
# Contents

1 Introduction ........................................................................................................... 1
   1.1 Motivation ........................................................................................................ 2
   1.2 Challenges and contributions ........................................................................ 4
   1.3 Application ..................................................................................................... 5
   1.4 Limit of scope ................................................................................................. 6
   1.5 Thesis layout ................................................................................................. 7
   1.6 Publications .................................................................................................. 8
      1.6.1 APPSTAR Competition ........................................................................... 8

2 Navigation tools .................................................................................................... 10
   2.1 Navigation system requirements ...................................................................... 10
   2.2 Research goals ............................................................................................... 11
   2.3 Assistive technology ...................................................................................... 12
      2.3.1 White cane .............................................................................................. 13
      2.3.2 Guide dogs .............................................................................................. 13
      2.3.3 Miniguide ............................................................................................... 14
      2.3.4 Soundpost .............................................................................................. 15
      2.3.5 Loadstone GPS ...................................................................................... 16
      2.3.6 Research projects .................................................................................... 16
   2.4 Conclusion ..................................................................................................... 19

3 Indoor localisation .................................................................................................. 20
   3.1 Visual navigation ............................................................................................ 20
      3.1.1 Map-based systems .................................................................................. 20
      3.1.2 Mapless systems ...................................................................................... 23
   3.2 Localisation systems ...................................................................................... 24
      3.2.1 GPS-based ............................................................................................... 25
      3.2.2 Infrared (IR) ............................................................................................ 26
      3.2.3 Ultrasonic ............................................................................................... 27
      3.2.4 Radio Frequency (RF) ............................................................................ 28
      3.2.5 Wireless Local Area Network (WLAN) .................................................. 29
      3.2.6 Audible sound ......................................................................................... 30
      3.2.7 Inertial sensors ....................................................................................... 30
      3.2.8 Machine vision ....................................................................................... 31
   3.3 Computer vision methods for localisation ....................................................... 32
      3.3.1 Feature based matching ........................................................................... 32
3.3.2 Visual Bag of Words ........................................ 34
3.3.3 Pose based matching ......................................... 37
3.3.4 Mobile phone based matching ............................... 38
3.4 Summary .......................................................... 40

4 Datasets and Performance metrics .................................. 42
4.1 Datasets ............................................................ 42
4.1.1 Offline evaluation ............................................. 42
4.1.2 Realistic evaluation ............................................ 44
4.1.3 Challenges with existing indoor datasets ..................... 48
4.2 Performance metrics .............................................. 49
4.2.1 Definitions ..................................................... 49
4.2.2 Metrics ........................................................ 50
4.2.3 Selection of metrics .......................................... 52
4.3 Conclusion ........................................................ 53

5 Feature evaluation .................................................. 54
5.1 Image descriptions ............................................... 54
5.2 Features ........................................................... 55
5.2.1 Scale Invariant Feature Transform (SIFT) .................. 56
5.2.2 SIFT variants .................................................. 58
5.2.3 Speeded up Robust features (SURF) ......................... 60
5.2.4 Histogram of Gradients (HoG) ............................... 61
5.2.5 GIST .......................................................... 61
5.2.6 Oriented FAST and Rotated BRIEF (ORB) ................. 62
5.2.7 Fast Retina Keypoint (FREAK) ............................. 63
5.3 Evaluation of features ............................................. 64
5.4 Datasets and performance metrics ............................... 65
5.5 Results ............................................................. 66
5.5.1 General matching .............................................. 66
5.5.2 Image transformations ........................................ 69
5.6 Conclusion ........................................................ 76

6 Visual Bag of Words for localisation .............................. 78
6.1 Related work ....................................................... 78
6.2 Proposed System .................................................. 82
6.2.1 Extraction of features ......................................... 84
6.2.2 Vocabulary building ........................................... 84
6.2.3 Inverted index .................................................. 86
6.2.4 Query image classification .................................... 86
6.2.5 Weighting scheme ............................................. 87
6.3 Datasets and performance metrics ............................... 89
6.4 Results ............................................................. 90
6.4.1 Comparison of weighting schemes ......................... 90
6.4.2 Voting module analysis ...................................... 90
6.4.3 Weighting scheme analysis .................................. 92
6.4.4 Soft versus hard assignment ........................................ 92
6.5 Conclusion ................................................................. 95

7 Verification methods for Visual Bag of Words ........................ 96
7.1 Verification methods ....................................................... 96
7.1.1 SIFT distance based (SD) ............................................. 97
7.1.2 SIFT+ global hue based (SH) ........................................ 98
7.1.3 SIFT+ local hue based (SSH) ........................................ 100
7.1.4 Local binary pattern based (LBP) ................................... 101
7.1.5 Homography based (HM) ............................................. 102
7.1.6 Fundamental matrix based (FM) .................................... 105
7.2 Datasets and performance metrics ..................................... 108
7.3 Results ......................................................................... 109
7.3.1 Fundamental matrix based variants analysis ...................... 109
7.3.2 Verification methods analysis ...................................... 111
7.3.3 Computational Time .................................................. 114
7.3.4 Image matching with multiple query images ..................... 115
7.3.5 Scene Confusion Matrix ($S_{cm}$) .................................... 117
7.4 Conclusion .................................................................. 118

8 Reduced feature sets ......................................................... 121
8.1 Large scale image matching ............................................ 122
8.2 Track based feature reduction ......................................... 124
8.2.1 Track generation ..................................................... 124
8.2.2 Reduced feature set .................................................. 124
8.2.3 Scene representation ................................................ 125
8.3 Datasets and performance metrics .................................... 127
8.4 Results ......................................................................... 129
8.4.1 Feature reduction statistics ........................................ 129
8.4.2 Naive matching ........................................................ 130
8.4.3 Visual Bag of Words ................................................. 133
8.4.4 Scene clustering ($S_c$) ............................................... 133
8.5 Conclusion .................................................................. 136

9 3D based scene localisation ................................................. 137
9.1 3D models .................................................................. 137
9.1.1 Tools ................................................................. 138
9.1.2 3D localisation ...................................................... 139
9.2 Building 3D models ...................................................... 141
9.2.1 Compact model .................................................... 142
9.2.2 Points reduction ..................................................... 143
9.3 Scene localisation ....................................................... 143
9.3.1 2D naive matching (NM-2D) .................................... 143
9.3.2 3D naive matching (NM-3D) .................................... 144
9.3.3 Pose matching (PM) ................................................ 144
9.3.4 Hybrid matching (HM) ............................................. 145

vi
9.4 Datasets and performance metrics ........................................ 147
9.4.1 Mobile devices ............................................................ 149
9.4.2 Performance metrics ...................................................... 149
9.5 Results .............................................................................. 150
9.5.1 Naive matching (NM-3D) analysis .................................... 150
9.5.2 Pose matching (PM) analysis .......................................... 151
9.5.3 Hybrid matching (HM) analysis ...................................... 153
9.5.4 Best pose (B_p) .............................................................. 155
9.6 Conclusion ......................................................................... 156

10 Indoor Positioning System (iPoS) ......................................... 158
10.1 Android application .......................................................... 158
10.1.1 Development ............................................................... 159
10.2 BoWLocator ................................................................. 159
10.3 Datasets and performance metrics ...................................... 162
10.4 Results .............................................................................. 163
10.4.1 Combined versus Single Collections .............................. 164
10.4.2 Voting module analysis ................................................ 166
10.4.3 False acceptance rate .................................................... 166
10.4.4 BoWLocator versus Hybrid matching ............................ 167
10.4.5 iPoS runtime performance .......................................... 169
10.4.6 iPoS robustness ............................................................ 170
10.5 Conclusion ......................................................................... 172

11 Conclusion .......................................................................... 173
11.1 Future work ....................................................................... 175

Bibliography ............................................................................ 177

A Scale Invariant Feature Transform (SIFT) ............................... 192
A.1 Scale space extrema .......................................................... 192
A.2 Keypoint localisation ......................................................... 194
A.3 Orientation Assignment ..................................................... 195
A.4 Keypoint descriptor ........................................................... 196

B Demonstration video ............................................................ 198
List of Tables

3.1 Comparison of technologies for positioning in large indoor environments. 40
4.1 Statistics of offline and realistic datasets ............................. 48
5.1 Statistics indicating number of training features for datasets, M means million ......................................................... 66
5.2 Average correct acceptance rate ($C_a$) of features across all datasets. 68
5.3 $p$-values for SIFT variants versus other features ............................................ 69
5.4 Average correct acceptance rate ($C_a$) across rotated images. .................. 71
5.5 SURF versus SIFT variants $p$-values for rotation experiments. ................ 71
5.6 Average correct acceptance rate ($C_a$) across illuminated images. .......... 72
5.7 SURF versus SIFT variants $p$-values for illumination experiments. ............. 72
5.8 Average correct acceptance rate ($C_a$) across blurred images. ................. 74
5.9 Average correct acceptance rate ($C_a$) across noisy images .................... 75
5.10 SURF versus SIFT variants $p$-values for noise experiments. ...................... 75
5.11 Average correct acceptance rate ($C_a$) across scaled images. .................. 76
7.1 Average correct and wrong feature matches between query and retrieved images .................................................................................. 98
7.2 Performance analysis of SH with different values of histogram threshold across the query images of the CS dataset ................................................. 100
7.3 Proposed visual BoW wrong match rates ($W_m$) and corresponding standard deviations against different vocabulary sizes. The lowest $W_m$ are highlighted in red for all methods .................................................. 113
7.4 Average time to match one query image (in seconds). .......................... 114
7.5 Average running time in seconds for verification methods ........................ 115
7.6 Performance with multiple query images. The bold values indicate a large increase in $C_a$ or large decrease in $W_m$ compared to HM and FM. Note that HM and FM results are different to Figure 7.9 because of different datasets ................................................................. 116
7.7 Average time to match a query image with one and multiple images (in seconds). Note that HM and FM results are different to Table 7.4 because of different datasets ................................................................. 117
7.8 Scene Confusion Matrix for HM. LEGEND: L, LABS; CR, CONFERENCE ROOM; CoR, COFFEE ROOM; C, CORRIDORS; H, HALLS; W, WASHROOM; O, OFFICES; NL, No-Location; $W_m$, wrong match rate ................................................................. 118
7.9 Scene Confusion Matrix for \( FM \). LEGEND: L, LABS; CR, CONFERENCE ROOM; CoR, COFFEE ROOM; C, CORRIDORS; H, HALLS; W, WASHROOM; O, OFFICES; NL, No-Location; \( W_m \), wrong match rate. .................................................. 118

8.1 The feature reduction via track based approach on all datasets. LEGEND: THR, Similarity threshold; FS, Feature sets; M, Million. .................. 130

8.2 Average correct acceptance rate (\( C_a \)) for normal SIFT and reduced features across indoor and outdoor datasets. .......................... 132

8.3 Scene clustering and error rates for the CS dataset with \( ST \) reduced features. ................................................................. 134

9.1 Image registration, wrong match and no image match rates with different top 2D to 3D correspondences using \( T_p =8 \) with \( HM \). ............. 147

9.2 Image registration, wrong match and no image match rates with varying values of \( T_p \) using top 50 2D to 3D correspondences with \( HM \). ......... 147

9.3 Statistics for our indoor 3D Models ............................................. 148

9.4 Mobile devices specifications ......................................................... 149

9.5 Average time to register a query image for all approaches in seconds. 151

9.6 \( p \)-values for PM versus PMv. .................................................. 152

9.7 \( p \)-values for NM-2D versus HM and HMc. ................................. 155

10.1 \( p \)-values for \( BoW(H+pv) \) versus \( HM \). ................................. 168

10.2 Wrong match rates (\( W_m \)) for \( BoW(H+pv) \) versus \( HM \). ............. 169
# List of Figures

1.1 Images of two locations within an office building which appear similar to one another. .................................................. 3
1.2 Images of the same location taken on different occasions. .......... 6
1.3 Query image with a single person in it. .................................. 7

2.1 Blind person on the pedestrian crossing carrying a white cane. ...... 14
2.2 Blind person accompanied by a guide dog. ............................ 14
2.3 Person holding a miniguide. ................................................... 15
2.4 The Electro-Neural Vision System. ........................................ 18

4.1 Sample images from the David Nister dataset. ....................... 43
4.2 Sample images from the Computer Science dataset. ................ 43
4.3 Sample images from the Hongwen dataset. ............................ 44
4.4 Sample images from the Caltech dataset. ............................... 44
4.5 Sample images from the Pasadena Buildings dataset. ............... 44
4.6 Sample images from the Owheo dataset. ................................. 45
4.7 Sample images from the Commerce dataset. ......................... 45
4.8 Sample images from the Museum dataset. ............................. 46
4.9 Sample images from the Stadium dataset. ............................. 46
4.10 Sample images from the Indoor dataset. .............................. 47
4.11 Sample images from the Benchmark dataset. ....................... 47

5.1 4 x 4 orientation histogram each with 8 bins. ....................... 59
5.2 Customised 4 x 4 orientation histogram array configurations used to generate shorter SIFT descriptors. .......................... 60
5.3 Features matching performance on Benchmark Datasets. ............. 67
5.4 The actual and corresponding deformed training images. .......... 70
5.5 SIFT versus SURF performance on rotated images across the CS dataset. 70
5.6 SIFT versus SURF performance on rotated images across the HW dataset. 71
5.7 SIFT versus SURF performance on illuminated images across CS dataset. 72
5.8 SIFT versus SURF performance on illuminated images across HW dataset. 72
5.9 SIFT versus SURF performance on blurred images across the CS dataset. 73
5.10 SIFT versus SURF performance on blurred images across the HW dataset. 73
5.11 SIFT versus SURF performance on noisy images across the CS dataset. 74
5.12 SIFT versus SURF performance on noisy images across the HW dataset. 74
5.13 SIFT versus SURF performance on scaled images across the CS dataset. 75
5.14 SIFT versus SURF performance on scaled images across the HW dataset. 76
6.1 Scene localisation by the system. Colored circles indicate the identified indoor places against the corresponding input images.

6.2 Vocabulary building from training image features. C1-C7 are the visual words obtained after clustering training features.

6.3 A sample Inverted Index.

6.4 Histograms generation for trained images.

6.5 Correct acceptance rates $C_a$ for ntf, ntfidf and BM25 schemes on all datasets using standard visual BoW.

6.6 Voting module performance analysis with different number of images across the CS dataset.

6.7 Weighting scheme consistency and accuracy for ntf, ntfidf and BM25 schemes.

6.8 Correct matches by the voting module (ntf scheme) on CS dataset.

6.9 Wrong matches by the voting module (ntf scheme) on CS dataset.

6.10 Soft vs Hard assignment in visual BoW with the ntf scheme on CS dataset.

7.1 An example showing the top ranked image retrieved against a query image by standard visual BoW.

7.2 An example showing the ranked image selected as the best match against the query photo by the $SD$ on the left. Hue histograms of the images are shown on the right.

7.3 Wrong image match found by the verification method $SH$.

7.4 Generating a 8-bit number from a 3x3 window.

7.5 10 best SIFT correspondences between two images.

7.6 Homography estimation and candidate feature transformation to new locations in a RANSAC iteration. The blue dots are used for homography estimation.

7.7 fm-BoW variants performance comparison. The curves that are higher and further to the right indicate a better performance.

7.8 fm-BoW variants no-decision rates.

7.9 Visual BoW performance with different verification methods on CS dataset. The curves that are higher and further to the right indicate a better performance.

7.10 Wrong image matches by $FM$ and $HM$.

8.1 An example of a track showing one similar feature traced across three images of an indoor scene.

8.2 The correct acceptance rate ($C_a$) for normal unreduced and compact features on all datasets.

8.3 The correct acceptance rate ($C_a$) for unreduced normal SIFT and reduced features on all datasets. Similarity thresholds (T) of 8%, 20%, 40%, 80% are used for images clustering in our experiments and the best results are reported.

8.4 Correct grouping example for some scenes of CS dataset.

8.5 Wrong grouping example for some scenes of CS dataset.

xi
9.1 Point cloud reconstruction of Trafalgar Square (London) from several thousand photos. The reconstructed cameras are shown as pyramids and one of the input photos taken from approximately the same viewpoint is shown on the top. ................................................. 138
9.2 3D model of one room showing the view from the top. The green dots indicate the recovered cameras. ................................. 142
9.3 3D models for one corridor in two directions. The green dots indicate the recovered cameras. ........................................... 142
9.4 Sample registered and query images. Noticeable changes can be observed. 149
9.5 The $R_c$ for the NM-3D versus the NM-2D with all, less-visible and high-visible included points. ........................................ 151
9.6 The $R_c$ for the PM versus the NM-2D. .................................. 152
9.7 The $R_c$ for the PMv for all included (A) points with three pose estimation algorithms versus PM. .................................... 153
9.8 The $R_c$ for the HM Vs. the NM-2D. The curves that are higher and are further to the right are better. .......................... 154
9.9 The $R_c$ for the HM versus thresholded NM-2D. ....................... 155
9.10 The $B_p$ for the HM less-visible included (L) points. ................ 156
10.1 The interface of the Android application. ................................. 159
10.2 The results for BoWLocator variants used with iPoS during realistic testing. The curves that are higher and further to the right indicate a better performance. ............................................. 165
10.3 Sample query images captured at night and corresponding training images of the same location captured during day time in Owheo building. 165
10.4 BoWLocator analysis using single datasets with and without voting module. .......................................................... 166
10.5 $F_a$ for negative samples using combined and single datasets configuration. 167
10.6 $BoW(H+pv)$ versus $HM$ (with less visible; L, high visible; H and all visible; A points). ........................................... 168
10.7 The time taken by iPoS to match one query image for “Location found” and “No Location found” scenarios. ....................... 170
10.8 Challenging query photos captured to test the robustness of the iPoS. 171
10.9 Sample images where a localisation decision was not made by iPoS. 171
A.1 Gaussian blurred images in octaves are on the left side. Images are downsized by factor of 2 in every octave and adjacent Gaussian images in octaves are subtracted to produce DoG images as shown on right. 193
A.2 Blob detection by pixels comparision within same and at different scales. 193
A.3 Keypoint detection after scale space extrema (656 keypoints detected). Green “+” signs represent the detected keypoints, which are drawn by keeping the detected keypoint as a centre. 194
A.4 Keypoints after localisation phase (298 keypoints are retained i.e more than 50% unstable points are rejected). 196
A.5 Building the SIFT descriptor. ............................................ 197
Chapter 1

Introduction

Self localisation is the process of determining the current location in an unknown environment by collecting and analysing the data from sensors such as ultrasound, camera, infrared or audible sound. Such systems are often used indoors for navigation of mobile robots who need to identify their current location followed by positioning themselves for autonomous movements. However, applications of indoor self localisation systems are not only limited to navigation but also include location based services, coordinating joint activities between people moving dynamically indoors, augmented reality games which use location information to blend the virtual and real world, or monitoring people by receiving notices when individual leaves the designated area within a building.

Basic localisation can be achieved without any special instruments e.g., sailors have been using astronomical objects for localisation for more than a thousand years during sea travel, and Jacob’s staff was a surveying instrument that had been used in making nautical and astronomical measurements since 1300’s. Many specialised tools were developed for more accurate localisation during the early centuries such as astrolabe, compass, sextant, marine chronometer etc. The major advancements in this area occurred in the 20th century and different electronic tools were invented. In the 1920s, localisation based on radio signals from shore based transmitters was proposed which improved sea based navigation. The first practical radar system was developed in 1935 which was capable of determining the presence and range of an object, its position in space, and its size and shape. It was used for marine applications, controlling air traffic, or detecting weather patterns.

In 1978, the United States Department of Defense developed the Global Positioning System (GPS) (Pace et al., 1995), which is a satellite based system capable of providing accurate positioning to within about 30 feet in any weather conditions. Modern GPS systems such as Differential GPS (DGPS) can provide accuracy up to 10 cm. It receives signals from multiple satellites and determines physical location through a triangulation process. GPS is the most popular system to find the location of objects in outdoors but GPS satellites are often occluded by buildings or trees in urban environments, affected by bad weather conditions which reduces signals strength or obstructed if GPS receiver cannot receive signals from satellites. Therefore GPS information is often combined with the data of other sensors such as cameras (Agrawal and Konolige, 2006; Leung et al., 2008), geographical information system (Bonnifait et al., 2007),
GPS is an outdoor positioning system but it does not work well in indoor because its signals cannot penetrate into buildings. High sensitivity GPS can provide indoor positioning but signals are heavily attenuated and reflected by the building materials making it highly unreliable. It’s also hard to extract reliable elevation details, such as floor level information from GPS signals. Self localisation in an indoor environment without GPS is still an open and ongoing research problem. Some technologies such as infrared, audible sound, or ultrasonic provide very precise solutions for self localisation in indoor but the associated cost of such systems is high. Cameras have proven to be good sensors for indoor localisation in terms of cost and reliability (Filliat, 2007; Kang et al., 2009; Kawaji et al., 2010; Mulloni et al., 2009; Werner et al., 2011), where machine vision extracts different pieces of information such as landmarks, color cues or image geometry from images via image analysis to recognise the image and hence the corresponding location. The basic idea is to extract a meaningful information from the captured image and match it with an existing labeled database of images to identify the current location.

This thesis focuses on the problem of self localisation indoors using machine vision. There are two main problems with most of the existing indoor self localisation solutions based on machine vision: (1) few indoor places are used in experiments, and (2) focus is kept on non-office buildings. Self localisation indoor based on machine vision becomes quite challenging in office buildings where different places look similar as shown in Figure 1.1. In order to address these problems, we propose an “Indoor Positioning System (iPoS)” which is currently based on a client-server model. The smartphone application (client) allows its user to take a picture of the indoor scene, sends it to the server to find the best match from a database of annotated images and then generates a voice message on the smartphone to indicate the current location. The algorithms which enable iPoS to operate robustly include, (1) a visual Bag of Words (BoW) scheme to retrieve images similar to the query image, and (2) a proposed voting module along with a verification method to find the match from retrieved images followed by post verification if required. We have tested the system in large indoor environments having 50 indoor locations and obtained 72-93% correct image matching accuracy with an average wrong match rate of 5.5%. Our results show that a single camera is a feasible sensor for indoor positioning in challenging indoor environments.

1.1 Motivation

The indoor localisation system based on any technology should at least meet some requirements: (1) cost should be low, which includes the costs of infrastructure components, positioning device, and installation, (2) positioning accuracy should be high, which means that the average error distance between true and estimated location should be less such as in meters, and (3) it should be easy to use. The target of our work is to develop a smartphone based indoor localisation system and use it as a navigation aid for blind people to guide them while they move inside buildings. Therefore, a cost-effective solution offering good indoor positioning accuracy is required in order to be used in many buildings.
Indoor positioning systems based on infrared light (Want et al., 1992), ultrasound (Ko et al., 2008), WLAN (Chintalapudi et al., 2010) and Active-RFID (Ni et al., 2004) provide good positioning accuracy but the cost is high. Technologies other than vision, inertial sensors and WLAN are quite expensive (more details follow in Chapter 3). WLAN solutions are cheaper if the existing infrastructure is re-used which is often the case these days. On the other hand, inertial solutions are cheap if no extra infrastructure is used such as positioning based solely on a smartphone. However, some problems associated with WLAN and inertial sensors (discussed in Chapter 3) make machine vision based solutions preferable. Machine vision not only offers a cost-effective solution but also a reasonable positioning accuracy. Therefore, the main motivations of our research work is to:

produce an indoor localisation system based solely on machine vision which uses few possible images for location recognition and is effective in any indoor environment particularly in office buildings.

The other motivations for our work are:-

- Most machine vision based works focus on self localisation in outdoor environments.

- The literature shows that not much work is done on localisation in large scale indoor environments using machine vision. Most works limit experiments to few indoor places.

- Indoor localisation is often performed in the context of non office buildings. It becomes a challenge in office buildings where places are visually similar. Therefore, the performance of a machine vision based indoor localisation system needs to be analysed in office environments.

- Video streams rather than single images are mostly used as an input for localisation in most machine vision based works (Gemeiner et al., 2008). The use of video stream drains the battery of smartphone which motivates the use of still images in our work hence making indoor localisation more challenging.
1.2 Challenges and contributions

Most of the successful machine vision based recognition work has been done in outdoor environments (Gavves et al., 2012; Irschara et al., 2009; Nister and Stewenius, 2006; Philbin et al., 2007; Robertson and Cipolla, 2004; Sattler et al., 2011). Comparatively, less work has been done for inside office buildings. It is quite hard to find a framework in which smartphone uses a single image of the current scene and uses machine vision for indoor localisation (Hile and Borriello, 2008; Kawaji et al., 2010; Werner et al., 2011). Amongst the challenges we faced were:-

- Identifying features which offer robust image matching in any environment under different image transformations. Captured images of the same place may be taken from different viewpoints and may also suffer from rotation, blur, noise etc. Therefore, suitable features are required which can handle such image deformations during image matching.

- Visual Bag of Words (BoW) retrieve relevant images against a query image. People have proposed different techniques to verify retrieved images to find the correct match for a query image on different datasets (Filliat, 2007; Kang et al., 2009; Philbin et al., 2008; Sattler et al., 2011; Zhang and Mayo, 2010). There is no standard method to verify indoor images retrieved from visual BoW against a query image.

- Unavailability of office indoor datasets covering various places such as rooms, corridors, toilets or halls for experimental evaluation.

We followed a bottom up approach. We first identified the most reliable features to be used for image matching in the work. We then performed image matching in outdoor environments followed by the experiments on our developed indoor dataset. After a careful analysis, we proposed a robust image matching system which can localise indoor scenes with minimum wrong matches. Finally, I developed a smartphone application to send the indoor query pictures to the proposed system (running on a server) and evaluated the localisation performance on realistic indoor datasets. The main contributions of this thesis are:-

- We proposed shorter versions of scale invariant feature transform (SIFT) features (Lowe, 2004) and compared their matching performance with other well known feature descriptors (Khan et al., 2011b). Shorter SIFT features perform comparably to unreduced SIFT features.

- We evaluated different ranking functions namely normalised term frequency ($ntf$), normalised term frequency inverse document frequency ($ntfidf$) and Okapi $BM25$ with visual BoW to analyse image retrieval performance (Khan and McCane, 2013). To our knowledge, $BM25$ has not been experimented before with visual BoW.

- We proposed the use of a voting module and a verification method with visual BoW. The voting module makes most of the image matching decisions against
query images, which results in decreased use of the expensive verification method. The correct matching accuracy of the voting module is more than 95% across four indoor datasets which not only improves the efficiency but also the accuracy of the system.

- We proposed the use of planar homography in verification method and found that it was almost as accurate but more efficient than a fundamental matrix based verification method (Khan and McCane, 2013; Khan et al., 2011a).

- We proposed a track based approach to effectively reduce the number of features from large scale datasets by as much as 50% (Khan et al., 2012a). Reduced features perform comparable to unreduced features for image matching on indoor datasets.

- We compared localisation performance based on pose estimation with 2D image matching on images captured from different mobile devices and proposed a hybrid approach which uses both 2D and 3D information for indoor localisation (Khan et al., 2013). In experiments, hybrid approach performs better than 2D image matching with mobile devices having 5MP cameras.

- I developed five indoor datasets which can be used as a standard to test the localisation performance of any algorithm in indoor environments.

1.3 Application

Our smartphone based indoor localisation system is intended to assist blind people while moving indoors. However, it can also be extended to other applications:

1. **Prospective students and other campus visitors**: It is hard to figure out the different places in the University campus in the beginning. The application can offer a natural way of learning information about the campus by taking pictures and receiving informative messages. New students can use this application to become familiar with the campus environment.

2. **Large malls**: It is easy to get lost in large shopping malls. This application could help shoppers to find where they are and receive directions.

3. **Tourists**: Each city has historic landmarks (e.g. buildings, statues, historic trees etc). The application can serve as a guide for tourists, giving information about the images of landmarks captured from a smartphone therefore promoting tourism.

4. **Location based marketing**: Mobile phone based augmented reality applications require the current indoor location to overlay it with graphical information. This is used for location based marketing where shops, facilities, or offers are displayed and highlighted to mobile phone users. Our system can be used to recognise the current location for such applications.
1.4 Limit of scope

As a whole, the system presented in this thesis takes a single picture of the current scene (via smartphone camera) and uses machine vision to identify the current location. However, there are different factors which can affect the performance of our system. It is not possible to address all challenges in a single PhD project. The limitations of this work are:

1. The system is intended to assist the blind people while they move indoors. This PhD project has progressed to the point of testing our system with non-blind users. We did not get the time to test it with blind subjects.

2. We have used a non-incremental approach and the mapped images of the building are not updated while the application performs localisation. In incremental approaches, the existing database is updated with the new captured images in order to keep the training database up-to-date.

The system does well with small changes within indoor places as shown in Figure 1.2 where it correctly recognised the query image even though chairs, bottles etc., are displaced from their actual positions as shown in the training image. The system also works fine in the presence of some extra stuff in an indoor place such as newspapers on tables, an extra chair etc. Therefore, short term variations in indoor places do not affect the system. However, the system will not work with major structural changes such as when indoor locations are re-arranged. The maps will need updating to handle such scenarios. This motivates the use of an incremental approach to update the training images during navigation (Angeli et al., 2008) but is not the focus of this work.

![Training Image](image1.png) ![Query Image](image2.png)

Figure 1.2: Images of the same location taken on different occasions.

3. The system works well with a few people in the scene as shown in Figure 1.3 because it leads to a small change. The system’s capability to handle crowds has not been tested but this is not the focus of this work.

4. The indoor localisation system presented in this thesis currently runs on a server. The smartphone application is an interface for sending pictures and getting location information. We have used this configuration for simplicity but the whole system, could in principle, be deployed on a smartphone.
5. The mapped images of the building intended for localisation are labelled manually with location IDs in the work. However, automatic labelling can be done by using scene semantics or by image classification (Ranganathan, 2012) and is not the focus of the work.

1.5 Thesis layout

This thesis describes the experiments, results and analysis carried out to evaluate algorithms on different datasets for indoor image matching. The thesis consists of eleven chapters and details are as follows:

- **Chapter 2** discusses research goal, navigation system requirements in context of blind users and presents some navigation tools which are currently in use by blind people.
- **Chapter 3** reviews research works which use different vision techniques for indoor navigation. Vision technology is also compared with other technologies for indoor positioning in detail. Localisation is then discussed in context of indoor and outdoor scene recognition.
- **Chapter 4** presents datasets and performance metrics used in the evaluation.
- **Chapter 5** proposes shorter versions of SIFT feature descriptor and compares them with different feature descriptors to evaluate its image matching performance on different datasets.
- **Chapter 6** presents a visual Bag of Words (BoW) approach based on the shorter features proposed in Chapter 5. The system uses several ranking functions and is compared with a standard visual BoW.
- **Chapter 7** offers a comparison of different verification methods which can be used with the visual BoW presented in Chapter 6.
- **Chapter 8** presents ways to reduce the number of features proposed in Chapter 5 from a large collection of training data. The matching performance of reduced feature set is evaluated on different datasets.
• **Chapter 9** evaluates image localisation performance using pose estimation and simple 2D image matching with different mobile devices.

• **Chapter 10** presents the BoWLocator algorithm used by the iPoS based on the best techniques presented in the previous chapters. The system’s performance is evaluated on four new indoor datasets and results are reported in this chapter.

• **Chapter 11** contains the final remarks and suggestions for possible future research work.

### 1.6 Publications

1. “Homography Based Visual Bag Of Word Model For Scene Matching In Indoor Environments”, (Khan et al., 2011a). This paper presents a visual BoW based on voting scheme and homography verification method for robust image matching.

2. “SIFT And SURF Performance Evaluation Against Various Image Deformations On Benchmark Dataset”, (Khan et al., 2011b). This paper discusses the proposed shorter version of SIFT features and compare it with SURF features (Bay et al., 2008).

3. “Vision based indoor scene localisation via smart phone”, (Khan et al., 2012b). We discuss the first prototype of our smart phone application and reported the preliminary results.

4. “Smart phone application for indoor scene localisation”, (Khan and McCane, 2012). We present the final prototype of our system and test it on larger images.

5. “Feature set reduction for image matching in large scale environments”, (Khan et al., 2012a). We propose the method to reduce the number of features from training images by more than 50% and compare our reduced features against well known descriptors.

6. “Analysis of verification methods for indoor image matching”, (Khan and McCane, 2013). We use fundamental matrix and homography based verification methods with visual BoW to compare and analyse their performance for indoor image matching.

7. “3D versus 2D based indoor image matching analysis on images of low cost mobile devices”, (Khan et al., 2013). We compare the performance of 2D and 3D indoor localisation methods with query images captured from low cost mobile devices.

### 1.6.1 APPSTAR Competition

APPSTAR was a competition run by Otago Innovation Ltd, Otago University in 2012. The competition asked the University of Otago’s academic and research staff and students all over New Zealand to put forward an idea for a mobile or web application. The competition received 108 entries and 5 ideas were selected by a panel of judges.
The proposed application in collaboration with William Levack “Living it up/iPos” was one of the top 5 finalists in the competition. The theme of “Living it up/iPos” was to help people with memory loss to know exactly where they are in their house and then guide them to perform daily life time activities. For example if someone is in the kitchen then they can be prompted with a list of things to do in the kitchen such as make breakfast.
Chapter 2

Navigation tools

This chapter discusses the requirements of a navigation system for a blind person in the context of indoor and outdoor navigation. The goal of our research work along with a desired set of specifications is then presented. Current navigation tools in use by blind people are finally examined followed by some research projects which assist them in their daily life. The contribution of this chapter is the explicit statement of our research goal and review of well known navigation tools, either commercial or research-based, used by visually impaired or blind people all over the world. Such a review is important to understand the benefits and limitations of existing tools or works before developing our system.

2.1 Navigation system requirements

Navigation systems should provide a number of functionalities such as obstacle detection, pedestrian crossing, location recognition or curbs detection to blind users to ensure safe movement in an environment. Navigation tools are designed to address these requirements in different ways. Blind people make choices when it comes to travel and use different navigation tools depending upon their preferences. It is important to have a good understanding about these requirements before developing our indoor localisation system because we want it to be used for assistance while moving indoors. Therefore, I began the research by conducting an informal interview with a blind person which took place in Disability Information and Support Center, Otago University\(^1\) at our request. The main purpose of the interview was the requirements analysis for a navigation system which are summarised as follows:

- **Pedestrian crossing**: The navigation system should guide the blind person to remain in the centre and to move in the right direction while on the pedestrian crossing. The crossing distance, intersection street name, traffic rules etc should also be communicated to increase perception of the surroundings.

- **Curb detection**: Possible curbs or holes in the path should be detected and the user should be alerted so that precautionary steps can be taken.

\(^1\)http://www.otago.ac.nz/disabilities/
• **Sign boards:** The system should recognise places or landmarks from sign boards. Location details along with the distance information to that landmark should be communicated to the user.

• **Temporary hazards:** Temporary hazard signs such as wet floor signs, warning signs on a construction site etc., must be detected and corresponding information should be communicated to the user.

• **Key destinations:** The user should be guided to important city destinations such as hospitals, supermarkets, bus stops or airport.

• **Speech accent:** Navigation systems often use voice to communicate with the user but accent of the speech is sometimes not easy to understand. Therefore, a Braille display should also be available along with the system. Braille displays are electro-mechanical devices for displaying braille characters usually by means of raising dots through holes in a flat surface enabling blind users use to read text output.

   It should be possible to use a language and accent to suit each other after some trials, if a blind person is not familiar with a Braille display.

• **Indoor localisation:** Blind people want to know about their current indoor location in unfamiliar buildings and feel lost in the absence of such information.

• **Obstacle avoidance:** The system should detect and alert users about possible obstacles in the path to avoid collisions.

• **Toilets:** The system should be able to identify and classify the male and female toilets in buildings.

• **Scene description:** Some sort of room description such as power socket location, exit information (opening inside/outside), wireless access points etc., should be conveyed to the user. The location of power sockets is especially important because blind people often carry several portable electronic devices.

• **Stairs:** Blind people prefer lifts and the system should guide blind users towards the lift. However, the use of stairs should also be incorporated to deal with emergencies.

• **Guidance in the corridors:** The system should indicate the number of steps to the next turn in a corridor.

   Up until now, there has been no research to incorporate all of the above requirements but these problems have been addressed separately in different research works.

### 2.2 Research goals

We realised from the interview that a blind person feels lost in the absence of current location information. Blind people have to rely on nearby people to determine their
current location and would like to have a system to assist them while they move indoors. Moreover blind people often carry smartphones for making calls, reading emails, messages and prefer to have a smartphone application for assistance in daily life.

The requirements analysis from the blind user helped us to formulate the research goal of this thesis, which is to: “develop a smartphone based indoor localisation system which can give effective performance in an indoor environment of any type”. The blind user can use the smartphone application when he/she needs guidance. The application will take a picture of the current location and will forward it to the server. The server will use machine vision to recognise the image and the corresponding location information will be sent back to the application where it will be conveyed to its user. We also identified a certain set of specifications for our smartphone based indoor localisation system which are as follows:

- The developed system will use minimum information, such as as few as possible images for location recognition. Because use of multiple images or videos will drain the smartphone battery and add to the worries of a blind user.

- The developed system should produce very few localisation errors because it can be a disaster for a blind person such as if stairs are recognised as corridor. It’s better not to find a match rather than finding a wrong match.

- The smartphone application will use voice messages to communicate with the blind user such as welcome message on the start of application or message indicating the current location.

- The smartphone application will use WiFi to transmit and receive data (images and location information) from the server. For each location query, the response time of the localisation system should not exceed half a minute because it can further confuse a blind person who is already feeling lost. The ideal response time will be few seconds.

- The interface of the smartphone application will be highly accessible to make it easy to use for blind people.

### 2.3 Assistive technology

Assistive technology includes assistive, adaptive and rehabilitative devices for people with disabilities. Assistive technology is provided by navigation tools and is used by individuals with disabilities to perform functions that might otherwise be difficult or impossible. Assistive technology has made it possible for blind people to get an education and pursue a career because of the use of computers and other devices. Assistive technologies for blind people include, (1) programs that speak the text on a computer’s screen, and (2) stand alone products designed to serve as a navigation or mobility tool including applications designed for smartphones and personal digital assistants (PDAs). There are two types of navigation tools:
• **Primary tools:** These tools aim to provide safe navigation and are needed by blind people while navigating in an environment.

• **Secondary tools:** These tools only increase the perception of the user’s surroundings and must be used with some primary tool.

285 million people in the world are visually impaired and 90% of these people live in developing countries. The last few years have seen the developments of different navigation tools in developing countries to help blind people. At any given time, a blind person can travel using a human guide, which involves holding onto someone’s arm or they use some navigation tools. This section describes some of the popular navigation tools used in daily life by blind people.

2.3.1 **White cane**

A white cane serves as a primary tool and is commonly used by blind people. It is an inexpensive solution. Blind people have been using a stick, cane or shepherd’s staff as an assistant tool for independent travel for centuries (Kelley, 1999). It was not until last century that the importance of the white cane as a symbol was realised. The white cane was used as a symbol for the first time by James Briggs in 1921 in order to alert passing motorists that he was a blind traveler. The significance of white canes further increased because of the large number of returning blind veterans after World War II. Different types of white canes have been developed to address different requirements. With time, the awareness of the white cane has increased and now it serves a dual role both as a travel tool and a symbol identifying the user as a blind traveler in our society (Strong, 2012).

The white cane is very useful for blind people to centre themselves while walking in corridors or detecting obstacles on the way as shown in Figure 2.1. Only tactile information within the reach of the cane is available and the blind user may be unaware of his or her wider surroundings in the navigating environment while using the white cane. Therefore, the situations that require route planning in unfamiliar environments will be difficult with only a white cane and blind people also use some wider awareness such as sounds, smell etc.

2.3.2 **Guide dogs**

Guide dogs are useful primary navigation tool for blind people. A guide dog is trained to guide their users around hazards, negotiate traffic, locate common destinations such as the supermarket, post shop, travel on buses etc. However, a guide dog is not capable of navigating (can still avoid hazards) in unknown environments. The blind user therefore does the directing, based upon skills acquired through previous mobility training.

The first guide dog training school was established in Germany during World War I, to enhance the mobility of returning blind veterans. Later years have seen the establishment of such schools all over the world. There are about 240 working guide dogs in New Zealand (RNZFB, 2012). The main problem with the guide dog is the
associated cost which is over US$ 42,000 (Guide Dog of America, 2012). This cost includes training the dog and providing instructions to the guide dog user. Blind organisations rely on donations to prepare guide dogs and provide them free of cost to blind people which limits their availability. So white canes are used as a cheaper alternative and by people who are allergic to dogs. Some people use both as shown in 2.2.

2.3.3 Miniguide

The Miniguide is a secondary tool to detect obstacles. It is light, hand held and pocket sized as shown in Figure 2.3. The device uses ultrasonic echo-location to detect obstacles or objects while the blind person navigates. The tool vibrates to indicate the distance to objects i.e. a faster vibration means that an object is nearby. The device can be adjusted to detect obstacles within different distance ranges e.g. 1 meter, 2
meters etc. However, for a distance range of 4 meter and above, only large objects such as trees and walls can be detected.

Figure 2.3: Person holding a miniguide.
(Adopted from American Printing House for the Blind website (http://shop.aph.org) with permission.)

The miniguide has assisted blind people in many ways such as avoiding obstacles, detecting overhanging obstacles (tree branches) and locating door ways. With all these merits, it cannot detect drop offs and the cost of the miniguide is up to US$ 399 which makes it hard for blind people to afford (SenderoGroup, 2012).

2.3.4 Soundpost

The Soundpost is a primary navigation tool. It is basically an orientation device and uses infrared technology to guide the blind person to move accurately from one landmark to another (Povidi, 2008). It allows a blind person to cross up to 30 meters of an open space. The soundpost uses multiple base stations for generating signals and a blind person carries a hand held controller to receive the signals. The controller starts vibrating once it comes within the range of a base station and keeps on vibrating while a blind user moves towards a landmark. The vibration stops when a blind user reaches a landmark. The controller can also generate a voice message indicating the landmark information.

The company website reports a small successful independent trial of the device at the University of Canterbury, New Zealand in October, 2008 to assess its utility for a blind citizen (Povidi, 2008). It was reported to be preferred by both guide dog and white cane users as it can assist the navigation in any environment. The blind person who I interviewed also used this device and he mentioned the following problems with the Soundpost:

- Special hardware such as base stations are needed in any environment.
- The infrared signal is often broken which poses problem during navigation.
- It is hard to estimate the movement direction towards a landmark once the vibration starts.
- Voice messages are limited.
- It costs about US$500 which includes the transmitter and two base stations.
2.3.5 Loadstone GPS

The Loadstone GPS is free open source software and offers navigation based on satellite for blind users in outdoor environments (LoadstoneGPS, 2010). The software currently runs on different Nokia devices and requires a GPS receiver. This project was initiated by Monty Lilburn and Shawn Kirkpatrick in 2004 who are themselves blind. The whole program was released under the General Public License (GPL) in 2006 and was financed entirely by private developers and by donations.

The software uses maps which are imported from common map databases for navigation based on GPS. In some cities or rural regions, no exact map data is available in common map databases. For these cases, the software provides users with an option to create and store their own waypoints for navigation and share them with others. There is also a website which allows users to share their custom points with each other and these points can then be easily added to the map.

This application is useful to search for specific locations in a given area. However, it lacks some of the basic features such as automatic download of maps and a route planner.

2.3.6 Research projects

We have discussed some commercial products which provide mobility aids to blind people. A lot of research has been carried out in the last few years to propose reliable navigation tools. Some of the well known research projects which use computer vision and other technologies to aid blind user mobility are discussed in the following subsections.

Holes and curbs detection tool

Coughlan and Shen (2007) developed a navigation tool based on a vision algorithm to detect curbs and holes for blind wheelchair users. A disparity map is retrieved from stereo images and then an edge map is generated. On the edge map, each pixel is checked for an appropriate depth leading to detection of possible curbs or holes and the blind user is alerted.

Objects and obstacles detection system

A visual saliency based assistance system is proposed to point out areas of interest in a scene that present either particular interest or potential threat (Deville et al., 2010). The areas of interest refer to those objects that attract the visual attention of a human being.

The system makes uses of a stereoscopic camera, a laptop and standard headphones. The user first defines an objective such as to find an object, to find a door etc. Depending upon the objective, the system computes the specific feature maps (colours, orientations and edges) followed by the conspicuity maps. The conspicuity maps contain information about regions of an image that differ from their neighborhood and are used to compute focus of attention (FoA). When the same FoA is seen over a number
of frames, the user is informed of the global position of an object or obstacle through a voice message.

**See Color**

It is a navigation tool designed to provide environment information to its blind users by transforming coloured pixels into musical sounds (Deville *et al.*, 2009). *See Color* encodes coloured pixels from frontal images by spatialised musical sounds in order to represent the colour and location of visual entities in the environment. The basic idea is to represent a pixel as a directional sound source with depth estimated by stereovision. Finally, each emitted sound is assigned to a musical instrument, depending on the colour of the pixel. The experiments demonstrated that blind users could perform simple tasks like following a coloured line, finding a coloured object etc with this approach.

**Text detection system from natural scenes**

Ezaki *et al.* (2004) presented a system capable of detecting the text from natural scenes to assist visually impaired people. The system uses a personal digital assistant (PDA), CCD-camera (placed on the user’s shoulder) and voice synthesiser. The image is captured and the system automatically searches for text areas. In case of text area detection, the camera zooms to obtain a more detailed image. After zooming in, methods based on either edge or colour are used to extract characters in the form of connected components. Finally, certain rules such as size and spacing between the characters are used to filter out incorrectly identified text areas. Extracted characters can then be read to visually impaired people as voice messages.

**Drishti**

Drishti is a navigation system which guides and helps blind people to travel safely both in indoor and outdoor environments (Ran *et al.*, 2004). It integrates several technologies such as a precise position measurement system, Geographic Information System (GIS), a wireless connection, a wearable computer and a vocal communication interface to provide navigation. For outdoors, it uses differential GPS as its localisation system to keep users as close as possible to the central line of sidewalks of campus and downtown areas. The user can switch the system mode from an outdoor to an indoor environment with a simple vocal command and a RFID ultrasound positioning system is then used to provide precise indoor location measurements. The system performs dynamic routing to provide its blind users with an optimal route and offers a step-by-step walking guidance for navigation in unfamiliar environments. Different indoor locations were chosen during experiments and location error was within 22 cm. Drishti may be considered as the first assistive technology system which can provide reliable navigation to blind people in dynamically changing environments. However, the system suffers from two limitations, (1) the prototype is heavy and weighs about 3.6 kgs, and (2) the degradation of radio frequency signals inside buildings degrades the accuracy of localisation.
Crosswatch

Crosswatch is a real time mobile application to detect pedestrian crossings in outdoor environments (Ivanchenko et al., 2008). At the time, this was the first portable system capable of providing real time orientation information at urban traffic intersections. The system first extracts straight line segments as features from the mobile image. A factor graph is then used to group the features into figure and ground, representing crossing and background respectively. Factor graph is a graphical model (Markov random field) that expresses interactions among sets of two variables. The detection of enough features having sufficient length indicates the detection of a pedestrian crossing and a brief audio tone is sounded for frames in which a pedestrian crossing is detected. A Nokia N95 mobile phone was used in experiments which could process approximately three frames per second. The application was tested with blind subjects to test usability and the subjects answered correctly whether a pedestrian crossing was present or absent at all used intersections.

Crosswatch was further extended to incorporate the detection of walk lights on pedestrian crossings (Ivanchenko et al., 2010). Crosswatch is now capable of alerting the user once the green walk light is illuminated at controlled intersections. The application is robust, easy to use and preliminary experiments were conducted with blind volunteers at traffic intersections in the United States to demonstrate its feasibility.

Electro Neural Vision System (ENVS)

ENVS is a system capable of avoiding obstacles, perceive landmarks and assist in navigation via visual sensors, GPS and electro-tactile stimulation (Meers and Ward, 2005). ENVS first extracts depth information from the environment via a disparity map obtained from head mounted stereo cameras as shown in Figure 2.4. This range data is then delivered to the fingers via electro-neural stimulation to indicate the range of objects being viewed by the cameras to the user. To perceive the location of obstacles and the 3D structure of the environment, the user imagines that the hands are held in the direction viewed by the cameras, with fingers extended and the amount of stimulation felt by each finger indicates the range of objects in the pointed direction.

Figure 2.4: The Electro-Neural Vision System.
(Adopted from Meers and Ward (2005) with permission.)
In order to perceive landmarks, it uses a GPS, a digital compass and a database of landmarks. The relative location of significant landmarks is determined (using GPS and stored GPS coordinates) and delivered to the fingers via encoded pulses when the landmarks are in the field of view. The system does quite well in conveying the 3D structure of the environment to blind people. The experimental results indicate that ENVS enables the user to achieve localisation, obstacle avoidance and navigation without using the eyes. However, it is not easy to use because blind users need to carry a laptop and wear gloves.

2.4 Conclusion

The purpose of this chapter was to determine a number of functionalities required by a blind person from a navigation system in order to move in unknown environments. We then presented some popular navigation tools in every day use by blind people along with research projects which aid the mobility of blind users. Navigation tools address few requirements of blind people and they need to carry more than one tool to ensure safe navigation in an unknown environment.

The review in the current chapter highlighted the limitations of different navigation tools. In the next chapter, we investigate indoor localisation systems based on various technologies to understand the limitations of each technology including machine vision followed by a review of well known research works which perform image matching in indoor and outdoor environments, and can be used for our smartphone based indoor localisation system.
Chapter 3

Indoor localisation

The previous chapter mentioned some navigation tools which aim to assist blind people while they walk in an environment. This chapter first reviews works which offer navigation based on machine vision for robots because suggested localisation methods can also be used for our work. Indoor localisation systems based on different technologies are then compared with machine vision based localisation systems. Finally, we present some well known research work that uses image analysis techniques for robust scene matching or localisation in indoor and outdoor environments.

3.1 Visual navigation

Visual navigation is based on applying machine vision to images or videos captured from the current scene by sensors such as cameras and is referred to as the “walk through” problem, where the robot has to move safely through an unknown environment (Maciel and Shirley, 1995). A vision sensor is a good choice for navigation because it has low cost, is ubiquitous and acquired images or videos offer high potential information content. However, it is hard to extract useful and reliable information from image or video feeds for navigation. Visual navigation focuses on guidance with the goal of automatically reproducing tasks such as detecting the current location (referred as localisation) followed by positioning within that location, plan the route and guide the robot safely while avoiding possible obstacles on the way. Bonin-Font et al. (2008) provide a very good survey on visual navigation for robots and key research contributions up until 2008. Visual navigation systems are divided into two main categories, map-based systems and mapless systems.

3.1.1 Map-based systems

Map based systems need some representation of the environment i.e. some sort of map before the start of the navigation process. There are two kinds of map based systems, systems which use an existing map of the environment usually referred as map-using, and systems that build the map of the environment themselves usually referred as map-building. Navigation starts once a representation of the environment is available.
For truly autonomous visual navigation, the system needs to do automatic exploration, localisation and mapping of the environment via techniques known as Simultaneous localisation and mapping (SLAM). SLAM basically involves the concurrent estimation of a map and the robot pose. Recently, cameras have been used as the sole sensor to yield SLAM based on vision i.e. visual SLAM (Chen et al., 2007; Clipp et al., 2010; Davison et al., 2007; Kundu et al., 2011; Silveira et al., 2008; Zou and Tan, 2013). A standard scheme for visual SLAM consists of first extracting a sufficiently large set of features and robustly matching them between successive frames. These corresponding features are then used for estimating the camera pose and scene structure. In this section, we discuss some well known visual SLAM systems with a focus on indoor buildings.

Davison (2003) presented MonoSLAM which is based on monocular vision and is considered a seminal work. The landmark positions and camera pose are both estimated and refined directly from observations in the image using an Extended Kalman Filter (EKF). The camera can be positioned accurately for extended periods provided that sufficient landmarks are tracked. The system compares the detected features of landmarks from the input video stream with the map of features developed during exploration of the environment for localisation purposes. The positioning of MonoSlam was limited to small scale environments with 100 landmarks in total.

Clemente et al. (2007) extended MonoSLAM to deal with larger environments by using the submaps framework presented by (Estrada et al., 2005). MonoSlam is used on a small scale and a new map is started once the number of landmarks grows beyond some limit. The local submaps are combined into an accurate global map by optimising transformations between submaps. The authors demonstrated extended MonoSLAM over long walks outdoors and the system successfully positioned itself in a 250m area with an accurate global map. The main contribution of this work is large scale outdoor mapping in near real time with a single hand held camera.

In another study, a system was developed which could map a large and complex environment in real time using a pair of stereo cameras (Sim and Little, 2006). The main contribution of the work is a fully automatic mapping system which operates online and consistently produces accurate maps of large scale environments. The system uses a hybrid approach consisting of 3D landmark extraction based on SIFT features for localisation and an occupancy grid to obtain a map for safe navigation. During map building, system uses stereo images to compute the epipolar geometry to find out the 3D positions of landmark features and stores them. The system performs localisation by comparing the detected features of landmarks from query images with the stored landmark features and then uses occupancy grid for safe movements as occupancy grid provides a spatial representation of the world indicating the presence of obstacles. The system was deployed on a robot and a laboratory environment having two rooms was explored in experiments. The system was reported to generate accurate occupancy grids during laboratory exploration which helped in maintaining a good map. However the system was not tested in large indoor environments.

Topological maps are simple graph based representations of the environment and are easy to scale. Winters and Santos-Victor (1999) present a system for robots which uses omnidirectional camera images to generate a topological map of the indoor structured environment during a training phase. The links in the graph are sequence of images
belonging to places, such as corridors and rooms; and nodes represent specific places
associated with actions such as turning in a corridor or passing through a door. The
image set is represented by a low dimensional eigenspace and the robot determines its
position by projecting the current image into the eigenspace during movement. After
position estimation, the robot uses ground plane images to extract corridor guidelines
to control its trajectory during movement. The system was deployed on a robot that
traveled about 21m along the corridors of Institute for Systems and Robotics\(^1\) in
experiments. The system was reported to correctly determine the topological position
100\% of the time by the authors but experiments were conducted in a simple indoor
environment. The topological representation will need to be extended to handle large
indoor environments with rooms, corridors, halls etc.

Kidono \textit{et al.} (2002) built a system which needs training from a human to build a
map. As the user guides a mobile robot from a starting point to a goal point through
an environment, the robot captures images with a stereo camera and constructs a
3D map on-line incrementally frame by frame. For autonomous movement, the robot
utilises the learned map and past observations to safely navigate from the starting
point to the goal point. To localise the robot, the system finds a set of object points
which are nearest to the robot followed by computation of distance to each object to
infer the position of the robot. The robot was demonstrated to successfully repeat the
same route learned during the training in an indoor room in experiments. The authors
claimed that their method provides accurate localisation and better estimate of robot
position compared to a method which directs the camera head forward without the
use of past observations. The system capability to handle large environments was not
discussed by the authors.

One of the map-building applications is museum guiding robots for visitors. These
robots need to be autonomous in their missions, recognise people and guide them
through the museum while avoiding obstacles such as chairs, tables or other people.
Shen and Hu (2006) developed ATLAS, a more advanced museum guiding robot com-
pared to RHINO (Burgard \textit{et al.}, 1998), a first generation and MINERVA (Thrun \textit{et al.},
1999), a second generation museum guiding robot. Laser range data and images are
used to match map data of the environment for localisation. A visual appearance based
algorithm is then used for normal topological navigation. In such algorithms, different
regions in the current observations are compared with existing models or templates to
determine the location. The authors reported the successful demonstration of ATLAS
in the foyer of London County Hall for three months.

Some navigation systems support an online construction of a local occupancy grid
which is the portion of the environment surrounding the robot. This local information
is then used for subsequent map construction frame by frame for online safe navigation.
Gartshore \textit{et al.} (2002) developed a framework for map building and a feature position
detector algorithm that processes online images captured by a single camera. The
main contribution of this work is the combination of an occupancy grid and visual data
from a single camera. The system first uses features to identify the object boundaries
in occupancy grids from the current frame and back projects the detected features
from the 2D image plane considering all locations at any depth. The positioning

\(^1\)http://welcome.isr.ist.utl.pt/home/
module uses odometry data to compute the position of the robot combined with image feature extraction. The colour or gradient from past frames are used to increase the confidence of the objects’ presence at certain locations. The system therefore computes probabilities of finding objects at every location and then uses this information for navigation while avoiding possible obstacles. The experiments were conducted in indoor environment and the robot started with no knowledge of obstacles in the environment. The robot moved 100 mm between consecutive images and was able to navigate safely while avoiding obstacles. The authors did not mention details regarding the scale of test environment such as number of rooms.

In another work, a BoWSLAM scheme was presented which enabled real-time navigation of robots in dynamic environments using a single camera (Botterill et al., 2010). The robot can position itself in real-time while exploring a previously unknown environment. The system works by representing every frame as a Bag of Words (BoW). BoW representation is then used to select multiple nearby frames against the current frame to find correspondences followed by computation of relative camera positions and the latest frame is positioned relative to each of these frames. A subset of these multiple position hypotheses with minimum gross errors is selected to accurately position the robot in a global map. The authors demonstrated the working of their system on suburban, indoor and outdoor datasets. The contribution of this work is a workable system which allows robot navigation in dynamic and self-similar environments.

3.1.2 Mapless systems

These systems refer to those techniques which do not need any knowledge of the environment but navigate as they perceive the environment. These techniques often grab video frames to produce enough information about the unknown environment for safe navigation. Such systems do not use a map of the environment and do not deal with localisation.

Optical flow is the apparent motion of features in a sequence of images and is commonly used for mapless systems (Green et al., 2003; Talukder and Matthies, 2003). In one important study, a novel solution based on optical flow and dense stereo for robots to detect the presence of dynamic objects in the camera field of view during navigation was presented (Talukder and Matthies, 2004). The system assumes that moving objects cause discontinuities in optical flow orientation and changes its magnitude with respect to the background pixels orientation and magnitude. The algorithm robustly estimates 6 Degree of Freedom (DoF) robot egomotion in the presence of moving objects using dense flow and dense stereo along with measurements collected from solid-state inertial measurement units. The dense stereo and egomotion estimates are then used to identify other moving objects while the robot itself is moving. The system was claimed by the authors to successfully detect moving people and vehicles in their experiments. The main contribution of this work is general unconstrained dynamic scene analysis for mobile robots and help them to remain in center while moving in corridors.

Visual appearance features are also used for mobile robot localisation (Zhou et al., 2003). Such strategies have two phases: first, in a pre-training phase, the images or prominent features from the environment are observed and stored as model templates; and then in the navigation stage, the robot matches the current image with the stored
templates to recognise the environment and self-localise. Zhou et al. (2003) used colour, gradient, edge density and texture histograms to describe the appearance of pre-recorded indoor images in the form of multidimensional histograms. Navigation is performed by matching the histogram of the current image with the stored histograms for localisation. The system correctly determined the location of 82.9% of the input scene images in an indoor environment with 14 locations such as rest rooms, laboratories etc. The main problems with such techniques are to identify the ways to represent the environment and perform on-line matching.

The techniques for tracking moving elements in a video sequence have become robust enough to be useful for navigation. But such techniques do not handle obstacle avoidance. Se et al. (2001, 2005) used features for robot navigation which were observed from different viewpoints, angles, distances and with different illumination changes. The detected features from frames serve as appropriate landmarks to be traced over time for navigation, global localisation and a real time vision based SLAM performance. The authors evaluated the localisation performance of their system by placing the robot at 8 different positions in an area of about 10m by 10m and allowed the robot to localise itself globally. In their experiments the average error in translation was 7cm and in rotation was one degree. However, a richer database map needs to be developed by observing landmarks all over the environment from multiple views during map building to avoid localisation failures.

In one important work, Saeedi et al. (2006) addressed the problem of 3D localisation of a mobile robot in unstructured indoor/outdoor environments. This work focuses on the estimation of robot motion independently from any prior scene or landmark knowledge. The system uses a new, faster and more robust corner detector. Distinctive features are identified and are positioned in 3D with high accuracy by a stereo algorithm. The 3D positions of scene features and the robot are refined by Kalman filtering over time. The authors evaluated their system with two sets of experiments. In one experiment, the robot was moved on an outdoor path of 6 km where it produced a translational error of 0.4%. In another experiment, the robot was moved on a circular path and reported errors were 0.9% in rotation and 0.6% in translation. The authors further reported improvement in tracking accuracy with their scheme compared to one-stage tracking schemes already in use.

### 3.2 Localisation systems

In the previous section, we reviewed some key research works in the area of visual SLAM for robotics. However, suggested localisation approaches can also be utilized for our smartphone based indoor localisation system.

As mentioned before, the focus of my research work is indoor localisation and not complete navigation. Indoor localisation based on machine vision is challenging and is one of the key components of map-based visual SLAM. Different technologies other than machine vision have been used for indoor positioning. It is important to briefly review well known indoor positioning systems based on these technologies to understand the limitations of each technology.

It is not enough to measure the performance of positioning systems only with ac-
accuracy. In this section, we compare indoor positioning systems based on different technologies for comparison using the following performance metrics (Al Nuaimi and Kamel, 2011; Gu et al., 2009; Liu et al., 2010; Song et al., 2011):

1. **Cost:** This metric includes the cost of infrastructure components, positioning device cost for each user, installation and maintenance costs. Some positioning systems need a large infrastructure to support the location measurement in large environments which makes them expensive. While some systems reuse existing infrastructure and hence save cost.

2. **Performance:** This metric refers to accuracy and success rate. Accuracy (or location error) is defined as the average error distance between the estimated location and true location while success rate is the estimation of correct locations regardless of distance error. Therefore, accuracy is basically the finer resolution, such as I am standing 20 m from the wall in the coffee room while success rate is the room level localisation, such as I am in the coffee room. Usually, there is a trade-off between the cost and the performance of an indoor positioning system. A positioning system which has high performance may have a high cost. For example, the accuracy of an Infrared positioning system can be improved by adding filters to reduce the influence of sunlight. However, the cost of the whole system is increased because of extra filters.

3. **Complexity:** This metric refers to human effort required during the deployment and maintenance of positioning systems. For indoor positioning system deployment, a rapid set-up of a system is desirable with a low number of fixed infrastructure components and easily used software for users. Another aspect of the complexity is that the positioning system should be scalable. For example, a positioning system deployed in a large building with many floors should not contain a large number of infrastructure devices and should not need time-consuming installation or maintenance. Otherwise, the complexity of the system will increase which may also increase the associated cost.

3.2.1 **GPS-based**

GPS is one of the most successful positioning systems in outdoors. It receives signals from multiple satellites and performs triangulation to determine physical locations, but its accuracy decreases indoors because GPS signals are unreliable inside buildings.

Qualcomm Company\(^2\) developed a wireless assisted GPS (A-GPS) to provide indoor positioning with an average of 5-50m positioning accuracy. A-GPS uses a location server with a reference GPS receiver that can simultaneously detect the same satellites as the wireless handset does with a partial GPS receiver. The wireless handset forwards measurements to location server after gathering them from the wireless mobile network and the GPS constellation. The location server carries out expensive computations to estimate the indoor position for the person carrying a wireless handset. This technology suits mobile devices, but it uses mobile network resources which may cost some money.

\(^2\)http://www.qualcomm.com/
Pseudolite is a device that performs a specific task that would otherwise require a satellite. A new positioning technology based on pseudolite transceiver called a LocataLite was developed to provide reliable positioning indoors and outdoors (Barnes et al., 2003). A network of LocataLites, called a Locata network, transmits GPS-like signals that allow a single point positioning for a mobile device with sub-centimeter level success rate. Recently, Locata network has been used to get positioning accuracy in centimeters in some challenging areas, such as in urban canyons (Montillet et al., 2009).

The limitations of this technology are:

- High sensitivity GPS such as A-GPS can provide indoor positioning, but signals are heavily attenuated indoors which decreases the positioning accuracy.

- LocataLites offer good indoor positioning solutions but pseudolite transceivers are quite expensive.

### 3.2.2 Infrared (IR)

IR positioning technology uses infra-red emitters and measures the position of the object according to the receiving time of infrared signals. Harter et al. (2002); Want et al. (1992) developed Active Badge, one of the first successful IR based indoor positioning system. The person needs to carry an active badge and the system provides room level indoor localisation with these badges. An active badge transmits a globally unique IR signal every 15 seconds. In each located place, one or more sensors are fixed and detect the IR signal sent by an active badge. The position of an active badge can be determined by the information from these sensors and location information of the tracked active badges is forwarded to a central server for further processing. More than one networked sensor is needed to provide positioning service in large rooms. The prices of active badges and networked sensors are low, but cables are used to connect the sensors which significantly raises the overall cost of the system.

OPTOTRAK system is designed for marker-based positioning in congested shops and workspaces (NorthernDigital, 2008). OPTOTRAK uses a system of three cameras as a linear array to track 3D positions of numerous markers on an object. The markers mounted on different parts of a tracked object emit IR light which is detected by cameras and location is estimated by triangulation. The system offers a positioning accuracy of 0.1mm to 0.5mm with 95% success rate. However, a disadvantage of OPTOTRAK is the line-of-sight requirement between the objects and the tracking system. This can be partially solved with a large number of IR markers at higher cost.

IR based systems provide very good localisation accuracy, often in millimeters. However, IR based systems have the following limitations:

- IR signals are affected by interference from sunlight and fluorescent light hence such systems may not work well in many buildings. This problem can be solved with optical and electronic filters but the overall cost of the system will be high.

- IR emitters are cheap but the whole positioning system with camera array, transmitters, sensors and wire connectivity for each isolated place is complex which increases the system cost in large indoor environments having many places.
The IR device taken by a person is often covered by his/her clothes causing
the system to perform unreliably because IR signals cannot penetrate opaque
materials.

3.2.3 Ultrasonic

Ultrasound signals are another good alternative to locate position. These signals are
used by bats to navigate at night which inspired people to design a similar navigating
system. This technology uses mainly a reflective distance method to determine the
location of objects.

The Active Bat system is a well known ultrasonic indoor positioning system created
by AT&T Labs (Addlesee et al., 2001; Ward et al., 1997). The system uses multiple
ultrasonic receivers embedded in the ceiling and finds position by the measurement of
distances i.e. trilateration process. The person can carry a transmitter called a “Bat”
which emits short pulses of ultrasound. The system works by finding the distance to a
minimum of three reference receivers followed by a multilateration technique to find the
exact position of the “Bat”. The system is reported to provide localisation accuracy as
good as 3cm with 95% success rate. However, the scalability of this system is affected
due to the deployment of a large number of sensors on the ceiling. Moreover, receivers
also need to be accurately placed which increases the complexity of this system and
hence the cost.

Cricket is another positioning system which offers good performance and has low
cost (Priyantha, 2005; Priyantha et al., 2000). Cricket uses time of arrival measurement
and triangulation to locate a target. The system uses ultrasound emitters as infrastructure
attached on the walls or ceilings at known positions and a receiver mounted on
each object to be located. Unlike the Active Bat system, the Cricket system has low
cost and is scalable for large area deployment because it requires fewer emitters at fixed
positions. Cricket provides a localisation accuracy of 10cm, which is lower than the
Active Bat system.

Sonitor ultrasound IPS is an indoor tracking and positioning solution provided
by Sonitor Technologies Inc (SonitorSystemWebsite, 2008). The Sonitor system uses
several detectors to provide proximity location information with room level accuracy
and can track people in real-time. In Sonitor, tags attached to people are tracked by
wireless detectors fixed in various places in rooms. A tracked tag transmits ultrasound
signals which are received by wireless detectors which forward this information to the
central module to compute the position. Sonitor system requires numerous detectors
in a room to be tracked and cost of this system is high for large indoor environments
with many rooms.

Ultrasound technology offers an inexpensive solution compared to Infrared position-
ing systems. However, the associated problems are as follows:

- Ultrasound signals are often combined with radio frequency signals at the cost
  of expense to increase the coverage area of an indoor positioning system. Such
  systems can locate people or objects in large places such as halls.

- These systems have lower localisation accuracy (several centimeters) than IR
  based positioning systems which can provide measurement accuracy in milli-
meters. However, a high success rate is important and enough to locate a person indoors.

- The positioning results are influenced by the environment. The presence of obstacles between tags and receivers degrades system accuracy.

### 3.2.4 Radio Frequency (RF)

This technology uses radio frequency (RF) signals which can travel through walls and human bodies more easily. These systems can cover the same area with fewer sensors compared with Infrared and Ultrasound positioning systems.

LANDMARC is a RF based indoor positioning system developed by Michigan State University and the Hong Kong University of Science and Technology (Ni et al., 2004). The system uses reference tags which are placed at fixed positions and RF readers. The signal strength from the nearest reference and target tags is used to estimate the location by the system. With four RF readers in the lab and with one reference tag per square meter, the system located the objects within error distance such that the largest error is 2m and the average is about 1m. Jin et al. (2006) further improved LANDMARC with respect to the system’s energy consumption and costs. The problems with LANDMARC are high latency and variation in the behavior of used tags. It also gives unreliable results if signals from some transmitters become unavailable or get blocked.

The WhereNet system offers indoor and outdoor real-time positioning (WhereNet, 2008). The RFID technology is used to identify tags mounted on the target objects, such as a device or a person. The system uses a differential time of arrival algorithm to calculate the locations of these tags. WhereNet offers an error range around 2-3m, which is not very accurate for indoors. The system is complex and requires numerous infrastructure components required to be fixed in different locations which increases its cost.

The advantage with RF based positioning system is that it can uniquely identify and track objects or people, moreover used tags in these systems are lighter, smaller and more portable. The associated problems with such systems are:

- The proximity and absolute positioning techniques used by these positioning systems require numerous infrastructure components installed and maintained in the working area. This increases the complexity of most of these systems and makes them less scalable to large area deployments.

- RF uses electromagnetic waves for information exchange between tags and readers. During travel, these waves pass through different materials, encounter interference from their own reflection and may be blocked by various objects, which may make positioning results unreliable.

- These systems have a lower positioning accuracy than the IR based positioning systems.
3.2.5 Wireless Local Area Network (WLAN)

WLAN technology is very popular and is widely used in business districts, universities, airports etc for positioning. These systems reuse the existing WLAN infrastructures in indoor environments which leads to a low marginal cost. However, the cost may be relatively high if WLAN infrastructure is not available. This technology normally uses the WLAN client such as laptop, PDA, phone etc., to get the received signal strength (RSS) or signal to noise ratio (SNR) from its wireless network interface card.

RADAR is a WLAN based indoor positioning system developed by Microsoft Research which offers localisation accuracy up to 4m with 50% success rate (Bahl and Padmanabhan, 2000). It uses signal strength and signal to noise ratio with the triangulation location technique to provide 2-D absolute position information. The major advantages of the RADAR system are that the use of existing indoor WLAN infrastructures and requirement of few base stations results in easy set up of the system. However, the located object needs to be equipped with WLAN technology and RADAR system also suffers from the methodology used to compute the received signal strength. Issues regarding system installation and maintenance are not addressed by the authors.

The COMPASS system uses WLAN infrastructure and digital compasses to provide low cost and high accuracy positioning services to locating a single user carrying a WLAN-enabled device (King et al., 2006). COMPASS uses a fingerprinting location technique and a probabilistic positioning algorithm to determine the location of a user. A fingerprinting location technique builds a location map during an offline phase and uses it to estimate position during the online phase. The COMPASS system achieved localisation accuracy of about 1.65m compared to the RADAR system which has localisation accuracy of 2.26 m in the same indoor environment. The problem with COMPASS system is that it does not provide real time tracking services.

WLAN based indoor positioning systems provide low cost solutions compared to the above mentioned technologies because they do not require additional infrastructure. The associated problems with this technology are:

- Complex indoor environments consist of various sources which can interfere with signals. Therefore, the performance of the positioning systems are not very accurate indoors with a positioning error of several meters.

- Most WLAN based systems use fingerprinting location techniques which first build a location map during the offline phase which is referred as the calibration step. Useful location related data is measured and collected with respect to different places in the position estimation area during calibration. During the online phase, the location map is used to locate the position of the object.

The calibration step is costly because it often requires infrastructure deployment (access points) in specialised ways to build a good location map. Moreover, the calibration step is also time consuming because it requires prior knowledge about existing WiFi access points, active user participation etc. Lately, some solutions are presented which offer non-complex and fast calibration with good performance (Chintalapudi et al., 2010).
3.2.6 Audible sound

Audible sound is a possible technology for indoor positioning because almost every mobile device is capable of emitting sounds. BEEP is designed as a cheap positioning solution based on audible sound technology (Mandal et al., 2005). The area is equipped with microphones at fixed positions which receive an audible sound from the target device. This data is forwarded to a central server via wireless technology from the mobile device. The central server uses time of arrival and triangulation to estimate the position of the device and the location information is then sent to the mobile device. BEEP provides a positioning accuracy of 0.4m with a 90% success rate. However, it has been reported by Gu et al. (2009) that the positioning accuracy of BEEP decreases from 0.9 to 0.8 in the presence of noise and obstacles.

The limitations of such systems are:

- Other sounds in the environment can interfere with the sound from the device, which leads to unreliable indoor positioning.
- Audible sound does not have high penetration ability, so the scope of an infrastructure component is within a single room. More infrastructure is required for a large indoor environment which will increase the cost of these systems.
- Transmitting audible sound is a kind of noise indoors, where people would not like to hear audible sounds made by positioning systems.
- The positioning accuracy of these systems is low, on the order of meters.

3.2.7 Inertial sensors

Inertial sensor is an electronic device that measures velocity, orientation and gravitational forces of a device using a combination of accelerometers, gyroscopes and a compass. Inertial based navigation systems rely on measurements from inertial sensors to track a user by continuously estimating the displacement from a known location. The position displacement is determined by aggregating individual steps. A step detector algorithm is used to detect steps of pedestrian and heading directions are estimated for each step during navigation. The drift error introduced during stepping is a common problem in such systems which increases with a walk and positioning results become unreliable. Drift errors are small errors in the measurements of sensors, which are integrated into progressively larger errors leading to great errors in overall positioning.

The FootSLAM system tracks pedestrians indoors by using foot-mounted inertial sensors and does not need any prior indoor layout (Robertson et al., 2009). The system uses an accelerometer and a digital compass in the user’s pocket to measure the heading and moving distance of the pedestrian accurately during stepping. The system is reported to provide accuracy of 1-2m, but the system is limited in practical application because the sensor placement is constrained to a certain position in order to get correct sensor readings and avoid drift errors.

The SparseTrack system tracks pedestrian location by using a digital compass and an accelerometer of a smartphone in sparse indoor environments (Jin et al., 2010). This
scheme uses an additional ultrasonic sensor, which is sparsely distributed, to adjust the current location directed by the smartphone. The system is reported to give a tracking error ranging from 0.27-0.50m. The problem with this approach is that an additional sensor is required to correct errors caused during movement.

Inertial based positioning systems offer cheap solutions for small walks because they can be used as a stand-alone method without any infrastructure requirement, but the accuracy of such systems becomes poor over time due to accumulation of drift errors. The limitations of this technology are:

- Drift error increases with walking distance which leads to unreliable positioning results. The solution is to use more sensors to update the location of pedestrians repeatedly, but it will increase the cost of system for large indoor environments.

- Smartphones give noisy inertial sensor readings. Therefore, smartphone based inertial systems have to use extra sensors at the expense of cost to tackle error-drifts.

3.2.8 Machine vision

Machine vision based positioning systems acquire images or videos from cameras and apply computer vision techniques to locate the position of an object or a person in an environment. These systems offer a cheap solution to track locations in large indoor environments.

A research group from Microsoft designed the EasyLiving positioning system (Brummitt et al., 2000). EasyLiving uses a computer vision based location technique with two stereo cameras covering the area which needs to be tracked. The colour and depth information from two cameras is used to provide location estimation. The EasyLiving system is less complex, convenient to use and has low cost because a single room can be covered by only two cameras. But its performance can be affected by changes in the environment that alter the video background. EasyLiving also needs substantial processing power to process images captured by the stereo cameras to perform indoor positioning.

MoVIPS was an indoor positioning system provided as a mobile application by (Werner et al., 2011). A database of labelled and oriented images is first captured and then stored on a server during training. Once a user starts the mobile application, it takes a picture of the current scene and sends it to the server. The server uses computer vision algorithms to compare it with stored images to locate the user and location is reported back to its user. The reported localisation accuracy of the system is 1.32m, but experiments are limited to a long corridor in a building. The system has low cost but its installation is time consuming which is basically the human effort to develop a database of images. Moreover, the accuracy of this system cannot be guaranteed because incorrect image matching can happen.

The main merit of a vision-based system is the use of low price cameras for indoor positioning. Cameras can cover a large area compared to other technologies which may need a large number of sensors. Therefore, the cost of vision based positioning systems is comparatively lower than systems based on other technologies. However, these systems have the following drawbacks:
• Position estimations are based on saved vision information in a database which needs to be updated to account for changes in environments, such as changing the place of your desk in your room.

• Such positioning is influenced by the lighting conditions, such as turning on and off a light.

• Positioning accuracy of these systems is low, on the order of meters.

• The installation or maintenance cost is high if system uses a database of images.

3.3 Computer vision methods for localisation

Image matching can be used for localisation indoors and outdoors. A database of images is first created by taking pictures of an environment and is stored with corresponding location information. This process is done offline as the environment gets mapped. During localisation, the query image of a location is compared with stored database images. Images similar to a query image are first retrieved and are then ranked on the basis of similarity against the query image. Ranked images are then verified to find the best match with the query image and corresponding location of the best matched image conveys the location information. Image matching methods provide either room level (success rate) or finer resolution (accuracy) localisation. 2D image matching methods offer room level localisation. On the other hand, some methods use 3D models to compute the pose of the camera and offer finer resolution localisation. Image based indoor location recognition is an active research problem and has been widely studied by computer vision researchers. In this section, We categorise image recognition works in four categories with respect to the used methodologies and discuss related research works for each category:

3.3.1 Feature based matching

This is a common method for image matching based on features. Features represent information from unique parts of an image to describe an image uniquely. These descriptions are used to match images during image classification. The images of the same place may be taken from different viewpoints and the captured images may also suffer from transformations such as noise, blur or scale, which makes it highly likely that images of the same place will tend to be different. Therefore, suitable features are required to describe images which can handle most transformations and can offer reliable image matching. Different algorithms have been proposed up until now to extract reliable features from images such as SIFT (Lowe, 2004), HoG (Dalal and Triggs, 2005), SURF (Bay et al., 2008), CENTRIST (Wu and M. Rehg, 2009), ORB (Rublee et al., 2011) or FREAK (Alahi et al., 2012).

Features are extracted from every trained image and are stored in memory during the training phase. During the testing phase, feature based matching system extracts features from a query image and compares them with features of trained images. The trained image giving the maximum feature similarities is considered a right match, but
this often gives incorrect image matches. Because it is not possible to avoid incorrect feature matches between images. Therefore, some systems compute the matching features between images and use those feature correspondences to estimate the image geometry between images. Images are only considered to match if a sufficient geometrical relationship is established between images. Image geometry has been applied in different works during image matching (Li et al., 2007, 2008; Vincent and Laganiere, 2001; Yukhzu and Hwang, 2009).

Robertson and Cipolla (2004) presented one of the successful prototype systems, which allows a user to navigate in an outdoor environment using a mobile phone equipped with a camera. The work uses homography and rectification to perform scene recognition in an outdoor environment of buildings. Image rectification is a transformation process which projects two-or-more images onto a common image plane. In their work, cameras are assumed to be calibrated (or at least approximately so), and database images are assumed to be rectified. Readily available map data is used to associate buildings in the database with meaningful 3D coordinate system. For a given query image, the system identifies features using the Harris corner detector (Harris and Stephens, 1988) and applies a Random Sample Consensus (RANSAC) (Fischler and Bolles, 1982) based algorithm for image registration to obtain a nearby database view. Then the relative pose of the query view is obtained from the plane-to-plane transformation that relates it to the building of the database view. RANSAC is an iterative method to estimate parameters of a mathematical model from a set of data, which contains outliers and inliers. Outliers are observations that are numerically distant from the rest of the data compared to inliers. During feature matching, some features are wrongly matched which results in outliers along with inliers i.e. the correct feature matches. The purpose of RANSAC is to select a set of matches from all feature matches randomly in different iterations to estimate solutions, which is an attempt to pick correct feature matches to find a reliable solution and the best solution is selected at the end. The developed system is tested by the authors on a database of 200 images covering an area of 2 km in Cambridge’s city centre and successfully registers 93 query images out of 97 images.

Viewpoint change may be large between images and different images may have repetitive structures. In such cases, it is hard to extract reliable feature correspondences between images. The wrong correspondences have two problems, firstly: pose of the camera cannot be estimated between images of the same place, and secondly: pose is successfully computed between images belonging to different locations. This affects the recognition performance of a feature based image matching system. In one important work, Zhang and Kosecka (2006) addressed this problem during image localisation in urban environments. The system computes the GPS location of a novel query image from a database of city street scenes tagged with GPS locations. Five closest views relative to the query image are first retrieved from the database based on SIFT feature matching. A large number of matched feature correspondences are not correct due to a change of viewpoint and presence of repetitive structures. Therefore, the system uses a RANSAC algorithm based on either fundamental matrix or homography model to remove outliers and uses determined inliers for motion estimation. During final localisation, top five reference views are re-ranked based on the number of correctly identified matches and two closest views are selected. The location of the query image
is finally estimated by triangulation of translation directions of the closest views. The system is reported to give higher accuracy on the International Conference on Computer Vision (ICCV) 2005 Computer Vision Contest dataset than previously reported results. The details regarding previous results and used methods are not discussed in the article.

### 3.3.2 Visual Bag of Words

The main drawback with feature based matching approaches is that the efficiency of the system decreases with an increase in the number of training images due to more time needed to search for feature correspondences between query and each training image. Visual Bag of Words (BoW) is an alternative approach, which offers efficient image matching and was introduced for the first time in vision by Sivic and Zisserman (2003). The recognition accuracy of standard visual BoW is high despite geometric information being ignored, and the results are robust to small changes in the scene.

The main idea of visual BoW is to reduce the number of features of training images and to generate a concise vector representation of each training image. Such representations are efficiently compared during image matching in visual BoW. During the training stage, visual BoW generates vector representations for training images, which are compared with the query image vector representation during the testing stage to find the best match. Visual BoW performs following steps during training:

- **Visual vocabulary**: Features are extracted from training images by using an algorithm such as SIFT, SURF or ORB. Clustering is then performed on training features via a suitable method such as approximate k-means or k-medoids. This step reduces the training feature space, which plays a key part in making the system efficient during image matching.

  The resulting cluster centres are called visual words and a group of cluster centres is called a visual vocabulary. Visual words are discrete set of representative features and may represent “trees”, “doors” etc. The idea behind the use of a visual vocabulary is that the same feature viewed in multiple images should map to the same visual word each time and different looking features should map to different visual words.

- **Visual word distribution**: Features of training images are then mapped to nearest visual words in the visual vocabulary to obtain distributions of visual words for every training image.

- **Vector representation**: In this step, a weighting scheme uses distribution of visual words to generate unique vector representations for each training image. The commonly used weighting scheme in computer vision is the term frequency-inverse document frequency ($\text{tfidf}$). The idea of this scheme is to down-weight visual words which appear in many images and give more weights to visual words, which appear in fewer images.

- **Inverted index**: Finally, a data structure which is called an inverted index, is developed to store visual word distributions for every training image. An inverted
index helps in quick retrieval of training images which are similar in appearance to query images during image matching.

During testing, visual BoW performs the following steps to match a query image:

- **Retrieving relevant images:** Inverted index is used to retrieve training images quickly, which have almost same visual words as our query image has.

- **Ranking of images:** A similar weighting scheme, which was used during the training phase, is used to obtain vector representation for a query image. Vectors of a query image and retrieved images are compared to generate rankings. Retrieved images are then arranged in order of increasing ranks, such as the top ranked image is the closest to query image and so on.

In standard visual BoW, the top ranked image is the best match. But this often leads to wrong matches in complex environments, where the top ranked image may refer to a wrong place. Therefore, standard visual BoW has been used in different works with varying configurations for reliable image matching, such as use of colour or texture information with visual BoW (Botterill et al., 2008, 2010; Filliat, 2007; Zhang and Mayo, 2010), use of spatial information (Lazebnik et al., 2006; Philbin et al., 2008), use of techniques to reduce errors introduced during mapping of features to visual words (Bai et al., 2012; Kang et al., 2012).

A large image database with thousands of images may have training features in millions. A visual vocabulary with a large number of visual words is required in such cases because a large number of cluster centres generate refined grouping of features, which improves image retrieval performance (Aly et al., 2011; Jiang et al., 2007; Philbin et al., 2007). The drawback with a large visual vocabulary is that clustering time increases and at the same time, visual words lookup time also increase, which makes the system inefficient for large image databases. Nister and Stewenius (2006) addressed this problem and developed a hierarchical clustering scheme to maintain the visual vocabulary, which allowed efficient real time searching from large image databases. The authors report that their system matches a query image in about 0.027 seconds on average from the self-developed dataset of 10200 images and retrieves 3.29 correct matches out of 4 on average. The presented scalable tree model has been successfully used with visual BoW in many works (Botterill et al., 2008; Irschara et al., 2009; Ji et al., 2012; Jiang et al., 2007; Philbin et al., 2007).

Visual BoW offers an orderless representation of an image and spatial information is ignored during the ranking of relevant images against the query image, which often results in poor rankings and top ranked images do not correspond to a query location. Philbin et al. (2007) incorporated spatial information in visual BoW to verify the consistency of retrieved images. The initially returned result list from normal BoW is re-ranked using spatial constraints by estimating affine homographies between a query image and each of the top-ranking results. The score used in re-ranking is computed from the number of verified inliers for each result, and the best match is identified for the query image. The method is reported to improve the average precision from 0.393 to 0.465 on a dataset of over 1 million images of outdoor environment compared to
standard visual BoW. Precision is the ratio of retrieved positive images to the total number retrieved.

In visual BoW, two image features are only considered identical if they are assigned to the same visual word. The mapping of features to visual words provides a very coarse approximation to the actual distance between the two features that is zero if assigned to the same visual word and infinite otherwise. In practice, such mapping leads to errors known as quantisation errors, which arise due to variability in the feature descriptors. This variability arises from image deformations and results in same features of images assigned to different visual words, which decreases image retrieval performance. Many researchers have addressed the problem of reducing quantisation errors on outdoor datasets to improve image retrieval performance (Bai et al., 2012; Cai et al., 2012; Philbin et al., 2008). In one work, Cai et al. (2012) mapped features to a visual word only if its distance to that visual word was not greater than a threshold; otherwise this feature was removed. This approach removes a large number of features with large quantisation errors, which contributes to improvement in image retrieval performance. The system improves the average precision from 0.574 to 0.577 and from 0.524 to 0.553 on two outdoor datasets compared to standard visual BoW with quantisation errors. The contribution of this work is to improve the correct image retrieval performance for datasets with million of images.

Visual BoW systems can be made more efficient by reducing the size of its visual vocabulary i.e. the number of visual words. An effective reduced visual vocabulary can offer efficient image retrieval performance not only for desktop systems but also for mobile devices, which have limited processing power and memory. In one work, a system is presented to reduce the size of visual vocabulary by linking semantically similar words in visual BoW (Gavves et al., 2012). The work focuses on identifying pairs of independent visual words that are likely to host image patches of similar visual appearance. The system uses images of landmarks because their geometry is unchanged. This allows for mapping between different images with different recording conditions, which opens the door for linking words as synonyms. The authors evaluate their system on a dataset which contains five thousand Flickr images from 11 landmark scenes in Oxford. The system is reported to give a average precision of 0.36 which is slightly lower than 0.38, a performance given by the state of the art visual BoW approach presented by Philbin et al. (2007) but with a vocabulary that is compressed by 89%. This technique has been used on landmark images and needs to be investigated for use on non-landmark images.

Most works have focused image matching on outdoor datasets. Indoor image matching is challenging because indoor locations look alike. Filliat (2007) investigated the indoor image matching problem and developed a two stage voting scheme. SIFT, hue and texture features are used in visual BoW. During indoor image classification, features from the query image are used to find corresponding visual words. These visual words vote at the first stage for the rooms in which they have been perceived at least once. In the second stage, the winning category votes at the second level only if the quality and the number of visual words are above some threshold. The best match is identified once the quality of the second level vote reaches a given threshold. The work is first tested for localisation on a small indoor environment with four rooms where it gave a localisation success rate of 83%. The experiments are then performed on a data-
base having images from 10 different rooms such as laboratory, kitchen or bathroom and the system reports 70% localisation success rate and no-location error that is a no-location decision 30% of the times. This work is a good contribution towards indoor image matching but this methodology has not been shown to work in office buildings, which have similar colour/texture schemes in many places.

In an office building, a similar colour or texture scheme is often used in many places, such as floor tiles have the same colour or corridor interior walls have the same pattern. Therefore, indoor image matching becomes more difficult in office environments especially large ones. In one important study, Kang et al. (2009) proposed a two-pass novel approach for image matching in a large indoor office environment. In the first pass, a smaller number of candidate images similar to the query image are retrieved via Bag of Words (BoW). In the second pass, these candidate images are used to discover more subtle details. These local statistics are then used to re-rank the candidate images. During indoor image matching, the system retrieves 8 of the most similar pre-captured images and suggests a potential localisation if there is a cluster of pre-recorded images less than 3m from each other among the re-ranked images. The system is tested on an indoor database of 8.8 thousand images with two different sets of test data and database images were pre-annotated with location information. The system improves the image matching precision from 0.97 to 0.98 and from 0.70 to 0.85 on two test sets, compared to image matching schemes which use only one stage. The system is a good contribution towards large scale indoor image matching. The training and test data cover a single floor of a building but authors have not provided the information about the number and types of indoor places used to build the image database.

3.3.3 Pose based matching

Recent advancements in structure from motion (SfM) has made it possible to construct 3D models of environments of any scale effectively. The 3D model of an environment is a dense cloud of 3D points obtained from reconstructed cameras. Each 3D point has a list of features, which are seen in different images or cameras and are used to triangulate that point. Using a 3D model to represent the scene offers the additional advantage that the full camera pose can be determined, which can be used to provide exact location information within a place.

During the offline phase, a pose based matching system extracts points of all 3D models and stores them. During localisation, it extracts features from a 2D query image and compares them with 3D points of every 3D model. This results in 2D to 3D correspondences, which are used for pose estimation. During pose estimation, a transformation is applied to 3D points to generate re-projected 2D points which are then compared with corresponding 2D features of query images. Numbers of re-projected 2D points which overlap with corresponding 2D query features are called inliers. The system selects the 3D model which gives a reasonable number of inliers against a query image and registered 3D model indicates the corresponding location. Lately, 3D localisation has been used for image matching in urban environments in different works (Bieliciki and Sitnik, 2013; Garg et al., 2011; Lim et al., 2012; Skrypnyk and Lowe, 2004; Xiao et al., 2011).

Reconstructed 3D models of large urban environments contain a large number of
3D points, often in millions. In such cases, the search time for 2D to 3D correspondences takes too much time during localisation, which decreases the efficiency of the system. Irschara et al. (2009) addressed this problem by developing a fast location recognition technique based on the structure from motion for image retrieval from large urban environments having million of images. A minimal set of “synthetic” images which effectively represents the 3D models of places is derived first to compress 3D data. A reduction of about 61% is reported in the number of SIFT descriptors from 3D models. The system computes features from a query image and then uses the vocabulary tree to retrieve similar images from compressed data via pose estimation. The system accepts a pose with at least 10 inliers. The authors report their system to give image registration performance (correct image matching) of 0.95 and 0.42 in experiments on two outdoor video sequences compared to a pure image based method which gives a performance of 0.92 and 0.31. The main contribution of this work is a system which can match a query image on average with a dataset having a million images in seconds. This problem has been studied in many works and researchers have also proposed other schemes to improve the efficiency of image matching with large urban environments (Li et al., 2010; Sattler et al., 2011, 2012).

Most research works compare 2D features of a query image with 3D points during localisation. On the other hand, matching a subset of 3D points with query 2D features is desirable because a significant amount of information is available such as what is the visibility of 3D points or which 3D points are likely to appear together. This information can be used to select a small subset of 3D points for comparison with query image features efficiently. In one important study, Li et al. (2010) explored this idea and presented an approach based on 3D to 2D matching for localisation in large urban environments. The approach uses a novel priority scheme and matches 3D features with 2D query features to compute 3D to 2D correspondences efficiently. These correspondences are then used for pose estimation. A query image is registered with a 3D model only if at least 12 inliers are detected during the pose computation. The scheme is tested on three datasets having 3D points in millions and registers more than 90 correct images on average compared to the work of Irschara et al. (2009) with no false positives. The authors also claim that their system is 40% times efficient than the system of Irschara et al. (2009).

### 3.3.4 Mobile phone based matching

Mobile phone camera has been used to capture an image of the current scene, which is then processed either on the phone or on the server during localisation (Arth et al., 2009; Hile and Borriello, 2008; Ji et al., 2012; Kawaji et al., 2010; Ruf et al., 2008; Werner et al., 2011). Image matching is often carried out at a server because mobile phones have limited processing power and memory. In those cases, mobile phones forward captured images to a server for image matching. In order to process images entirely on a mobile phone, efficient techniques are needed.

One common way is to use installed indoor markers for localisation (Mulloni et al., 2009). During localisation, system detects markers from images captured from a mobile camera. These markers contain 2D barcodes that provide a unique ID and can be used to estimate position. A better approach is to perform indoor image matching
without any marker. Werner et al. (2011) presented a markless indoor positioning system. During the training phase, a database of images captured by phone is created and stored at the server. To retrieve position, the mobile application takes an image of the surroundings and uploads them to a server, which compares the query image with the database of correctly located and oriented images based on feature matching to find the corresponding database image. The actual position is then computed by comparison of the object scale in both images using geometric position scheme. The system is evaluated at a university building by modeling a long corridor with 68 images. The authors use 16 test images and compare their results with the positioning accuracy of the RADAR system. The system results in position error of 1.32m which is better than 2.94m given by the RADAR system. However, the system applicability is not tested in an environment with rooms, central halls, corridors etc.

Some other information can also be forwarded to a server along with query image to assist localisation. In one work, Hile and Borriello (2007) developed a smart phone application to send the query image along with Wi-Fi fingerprints to a server for localisation. The server processes the query image to compute the features. Next, a relevant portion of floor plan is chosen on the basis of Wi-Fi fingerprints. Matching is performed to find correspondences followed by pose estimation to figure out the location and orientation of the camera. The location of objects and other information are finally translated back into image space and displayed on the phone for users. The system is shown to do well in hallway images of an indoor environment. The system takes about 10 seconds to display information on the phone, and a localisation accuracy of 30cm is reported in experiments. The authors have reported preliminary results, which illustrate that it is possible to provide the capability of information overlay on camera phone for a variety of indoor environments. However, authors mentioned some problems with the developed framework, such as slow speed, reflective floors, and unusual structures including curved walls and catwalks can pose problems and must be handled. Additionally, people or clutter in the hallways will cause problems in feature detection and also if the contrast difference is low between between floors and walls in environment. Therefore, several improvements are required in order to achieve a workable system.

In one important work, Arth et al. (2009) performed image matching directly on a mobile phone, which eliminated the requirement of a server. The authors developed an efficient method to localise the query image on a mobile phone based on reconstructed 3D models. The whole database of 3D points cannot not fit into a mobile phone memory. So the system splits the 3D model into chunks of points that are organised by visibility constraints, which allows to load a small portion of points in phone memory at a time. During online localisation, the authors use WiFi or Bluetooth to determine coarse location, which is then used to load the relevant chunk of 3D model in the phone memory to compute the pose based on 2D to 3D determined correspondences. The system accepts a pose only if at least 20 inliers are detected. The experiments are conducted on reconstructions of several adjacent rooms in a university campus, and the system is reported to provide 89% localisation success rate across images having 640x480 resolution. The main contribution of the work is that the developed method is resource efficient and requires only a few megabytes of memory, which makes it possible to run on a mobile phone. However, the experiments are limited to rooms within one
building in this work.

3.4 Summary

We compared different positioning systems to identify the advantages and disadvantages of their underlying technologies as summarised in Table 3.1. Technologies such as Infrared, Ultrasonic and Radio frequency offer good positioning accuracy, but such systems require large infrastructure in large indoor environments which increases the overall cost of positioning system. The following technologies seem to offer cheap solutions:

- **WLAN**: It offers a cheap positioning solution if wireless infrastructure is available, which is generally the case these days. However, the fingerprinting step in these systems is costly, time consuming and requires technical expertise.

- **Inertial sensor**: Systems based on smartphone inertial sensors offer cheap solutions. The problem is that smartphone sensors give noisy measurements and extra sensors are needed to handle drift errors in large environments to reduce positioning errors.

- **Machine vision**: On the other hand, machine vision based positioning systems are low in cost due to cheap cameras. The problem is that the image database needs to be maintained to handle structural changes over time, which increases the maintenance cost.

Table 3.1: Comparison of technologies for positioning in large indoor environments.

<table>
<thead>
<tr>
<th>Technology</th>
<th>Accuracy</th>
<th>Cost</th>
<th>Issues in large indoor environments</th>
</tr>
</thead>
<tbody>
<tr>
<td>GPS-based</td>
<td>&lt;1 cm</td>
<td>High</td>
<td>Expensive pseudolite transceivers are required for better accuracy indoors.</td>
</tr>
<tr>
<td>Infrared</td>
<td>0.1mm-10m</td>
<td>High</td>
<td>More sensors, optical filters and electronic filters are needed to tackle light interferences and signal blocking.</td>
</tr>
<tr>
<td>Ultrasonic</td>
<td>1-10cm</td>
<td>High</td>
<td>Infrastructure with numerous detectors is required to cover a large area and to deal with obstacles.</td>
</tr>
<tr>
<td>Radio frequency</td>
<td>5cm-5m</td>
<td>High</td>
<td>A complex installation is required with more infrastructure components for positioning.</td>
</tr>
<tr>
<td>WLAN</td>
<td>2m-100m</td>
<td>Low-High</td>
<td>Fingerprint location technique is costly because it requires deployment of access points in specific ways.</td>
</tr>
<tr>
<td>Audible sound</td>
<td>1-10m</td>
<td>High</td>
<td>More infrastructure components such as microphones are needed to provide positioning.</td>
</tr>
<tr>
<td>Inertial sensors</td>
<td>1-4m</td>
<td>Low-High</td>
<td>Error-drift increases with walk and additional sensors have to be used to tackle the error.</td>
</tr>
<tr>
<td>Vision</td>
<td>1-5m</td>
<td>Low</td>
<td>Database needs to be updated/maintained over time to avoid errors, which is a time consuming process.</td>
</tr>
</tbody>
</table>
In this work, we aim to provide location information to blind users as a smartphone application in a cost effective way. Machine vision technology seems to be the optimal choice due to following reasons:

- Machine vision based positioning systems can use off-the-shelf camera phones. Most mobile phones have a reasonable camera these days, which eliminates the requirement of any special hardware or infrastructure. On the other hand, WLAN and Inertial systems need extra sensors or infrastructure deployment in specialised ways for large indoor environments.

- The cost of vision based positioning system will be low because a blind person only needs a mobile phone. About 80% population of the world has a mobile phone these days\(^3\) and the prices of mobile phones have reduced significantly in the last few years\(^4\). So we can expect most of the blind users to have a mobile phone.

- Images of more buildings need to be acquired to extend the positioning service from one building to another, which makes machine vision based systems easy to scale at the cost of no extra infrastructure requirement.

The drawback with vision based positioning system is that their positioning accuracy is in meters, which is lower than Infrared, Radio frequency and Ultrasonic. Moreover, such systems also require a large database of tagged images for better performance. However, the associated low cost and good success rate of these systems make them a preferable choice for our work.

We mentioned different studies on visual SLAM which use a camera as a primary sensor to provide localisation and navigation in environments. The main purpose was to identify a suitable localisation technique for our work. But most of these techniques use other data such as laser, odometer readings etc., along with vision for localisation. Finally, we presented several studies on image matching in indoor and outdoor environments (Filliat, 2007; Kang \textit{et al}., 2009; Nister and Stewenius, 2006; Sattler \textit{et al}., 2011; Werner \textit{et al}., 2011; Zhang and Kosecka, 2006). We realised that image geometry, visual BoW or pose estimation are commonly used methodologies for image matching. Few research works have addressed the problem of indoor image matching but experiments are conducted on a small number of indoor places (Filliat, 2007; Kang \textit{et al}., 2009; Kawaji \textit{et al}., 2010; Mulloni \textit{et al}., 2009; Werner \textit{et al}., 2011). Indoor image matching is challenging in self-similar larger environments where locations look alike. We decide to use a 2D image matching approach, particularly visual BoW for our smartphone based indoor localisation system as it has been reported to provide reasonable image matching and retrieval speed in many works.

In the next chapter, we discuss the datasets and performance metrics used for experiments and system evaluation in the thesis.

\(^3\)http://www.go-gulf.com/blog/smartphone/
Chapter 4

Datasets and Performance metrics

This chapter provides an overview of the datasets and performance metrics used in this thesis for experimental evaluations. These datasets will be referred to in the coming chapters.

4.1 Datasets

We have used eleven datasets in this thesis for performance analysis. Out of eleven, five datasets are used for an offline evaluation of components which are later used to build Indoor Positioning System (iPoS). The remaining datasets have been used to analyse the performance of iPoS in real time. All non-indoor dataset images are (subject to) copyright by authors who developed them and are available for use in research work with the requirement of citing the corresponding research papers.

4.1.1 Offline evaluation

For offline evaluation, we used two indoor and three outdoor datasets. One of the indoor datasets was developed by capturing images from our Computer Science building, which is an office building. This collection provides a good platform to test the performance of proposed components in this thesis in an office environment with self similar locations. The offline evaluation datasets are as follows:

- **David Nister (DN)**
  This database was collected by Nister and Stewenius (2006) and includes images of 2550 different objects or scenes. Each object or scene has four images taken from different viewpoints therefore leading to a total of 10200 images. The dataset contains a variety of images of different places as shown in Figure 4.1. The dataset is also referred to as the UKbench dataset and is used as a standard to evaluate the image retrieval performance of systems.

- **Computer Science (CS)**
  I developed this dataset in 2010 and it contains indoor images (Khan et al., 2011a,b). This collection contains images of the Computer Science building at
Otago University, New Zealand. All images were captured with a Sony Cybershot Camera (DSC-W90)\(^1\) with a resolution of 8 MP.

This dataset contains 700 images and covers about 30 indoor locations from three floors of the building such as halls, offices, labs etc. I have used 70 images for testing and 630 images for training. Our building is a standard office building with some classroom size offices and computer laboratories. The developed dataset is challenging as many different locations within the building look similar. Some images from the dataset are shown in Figure 4.2. This dataset has been made available for other works and can be obtained from: http://www.cs.otago.ac.nz/pgdweb/nabeel/download.html.

**Hongwen (HW)**

This dataset was collected by Kang et al. (2009) and contains about 8000 indoor images covering one floor of an office building. The images were collected over a period of time to account for changes in the environment. The collection contains separate images for testing and training purposes. The dataset contains a large number of different corridor images which are quite similar and it provides a good way to test indoor localisation systems. Sample images from this dataset are shown in Figure 4.3.

**Caltech Buildings (CB)**

This dataset was collected by Aly et al. (2009) and contains images of 50 building exteriors around the Caltech campus, California Institute of Technology. Five

\(^1\)http://store.sony.com/p/DSC-W90/en/p/DSCW90
different images were taken for each building from different angles and distances resulting in a total of 250 images. Sample images from this collection are shown in Figure 4.4.

- **Pasadena Buildings (PB)**

  This dataset is reported by Aly et al. (2011). It contains 6 photos of the facades of 103 houses in the Pasadena area and 22 buildings from the Caltech campus taken at different times with varying viewpoints leading to a total of 750 images. Three images were taken in the morning and three were taken during the afternoon with two different cameras. Sample images from this dataset are shown in Figure 4.5.

### 4.1.2 Realistic evaluation

The above datasets are used for the offline evaluation of components. I have also developed four large realistic indoor datasets for testing of iPoS. The training images were captured by the same Sony Cybershot camera (8MP). However, all query images were
acquired with an HTC Wildfire S smartphone (5MP). The query camera is different from the collection camera as this is likely to be the case in any practical navigation system. These datasets have been made available for other research works and can be obtained from: http://www.cs.otago.ac.nz/pgdweb/nabeel/download.html.

These datasets are as follows:

- **Owheo (OW)**
  
  This is the newer version of the CS dataset. I created this dataset in May, 2012 and it contains 1534 indoor images of our Computer Science building for training and 750 images for testing (Khan and McCane, 2012; Khan et al., 2012a,b). The test images were captured with a smartphone while navigating inside the building. The training images represent 25 indoor locations consisting of corridors, labs and central halls as shown in Figure 4.6.

  ![Figure 4.6: Sample images from the Owheo dataset.](image)

  The training images were captured during the afternoon and the day time, while the test images were captured after two weeks during the day, afternoon and night times. Therefore, one can expect some degree of changes in the captured test images compared to the training images.

- **Commerce (CM)**
  
  I collected this dataset in May, 2012. The collection contains images of Commerce building in the University of Otago, New Zealand and covers 14 indoor locations consisting of corridors, halls and main atrium as shown in Figure 4.7. The dataset contains 864 images for training and 234 images for testing.

  ![Figure 4.7: Sample images from the Commerce dataset.](image)

  The training images were captured during the day time while the test images were captured at different times after 2 weeks. The building contains open public
spaces and the test images have some degree of changes compared to the training images such as change in the position of chairs or change in lighting conditions.

- **Otago Museum (OM)**
  I made this dataset in June, 2012 and it contains images of the Otago Museum. The dataset contains 1045 images for training and 135 images for testing. The training images cover a total of 7 indoor locations on the first floor of the museum consisting of rooms, atrium and exhibition halls. The test images were taken one month after capturing the training images. The sample images from this dataset are shown in Figure 4.8.

![Figure 4.8: Sample images from the Museum dataset.](image)

- **Dunedin Stadium (DS)**
  This dataset contains images from the Forsyth Barr Stadium, Dunedin, New Zealand and was collected in July, 2012. It contains 455 images for training and 75 images for testing. The dataset covers 4 places within the building including corridors, room and hall as shown in Figure 4.9. The test images were taken a few days after capturing the training images.

![Figure 4.9: Sample images from the Stadium dataset.](image)

- **Negative samples**
  I have developed two datasets whose images are not in the trained collection. These serve as negative samples to analyse the false acceptance rate of iPoS. Ideally, the system should not match any query image from these datasets leading to a zero false acceptance rate. These datasets are as follows:
– **Indoor Dataset**: This dataset contains the first 50 training images taken from the Hongwen dataset used by Kang *et al.* in his work (Kang *et al.*, 2009). Some of the sample images are shown in Figure 4.10.

![Figure 4.10: Sample images from the Indoor dataset.](image)

– **Benchmark Dataset**: The collection contains 120 images taken from the standard benchmark datasets collectively, such as David Nister (Nister and Stewenius, 2006); first 50 training images, the Holidays (Jegou *et al.*, 2008); first 50 training images, and the Caltech (Aly *et al.*, 2011); first 20 training images. The sample images are shown in Figure 4.11.

![Figure 4.11: Sample images from Benchmark dataset.](image)

The purpose of using the Indoor dataset is to analyse the performance of the system if an indoor image of a building which does not exist in training data is used as a query image. We suspect the system to may give many false positives in such cases because different indoor images may have a high degree of structural similarity among them especially corridor images, which increases the chances of matching two different indoor images. On the other hand, the Benchmark dataset is used to provide an outdoor image as a query image. An outdoor image is generally different from an indoor image and we suspect the system to may give fewer false positives because the chances of matching an outdoor query image with an indoor image are less. Therefore, the reason for using the above two datasets is to analyse the performance of the system for incorrect image matching in two different scenarios.

A table summarising statistics of all datasets such as numbers of the used training and test images in the thesis are shown in Table 4.1.
### Table 4.1: Statistics of offline and realistic datasets

<table>
<thead>
<tr>
<th></th>
<th>Training images</th>
<th>Test images</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Offline datasets</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DN</td>
<td>3000</td>
<td>1000</td>
<td>Non-indoor</td>
<td>Scene or object images</td>
</tr>
<tr>
<td>CB</td>
<td>200</td>
<td>50</td>
<td>Non-indoor</td>
<td>Building exteriors</td>
</tr>
<tr>
<td>PB</td>
<td>625</td>
<td>125</td>
<td>Non-indoor</td>
<td>Houses and buildings exteriors</td>
</tr>
<tr>
<td>CS</td>
<td>630</td>
<td>70</td>
<td>Indoor</td>
<td>Office building covering three floors</td>
</tr>
<tr>
<td>HW</td>
<td>3000</td>
<td>100</td>
<td>Indoor</td>
<td>Office building covering one floor</td>
</tr>
<tr>
<td><strong>Realistic datasets</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OW</td>
<td>1534</td>
<td>750</td>
<td>Indoor</td>
<td>Office building covering three floors</td>
</tr>
<tr>
<td>CM</td>
<td>864</td>
<td>234</td>
<td>Indoor</td>
<td>Office building</td>
</tr>
<tr>
<td>OM</td>
<td>1045</td>
<td>135</td>
<td>Indoor</td>
<td>Non-office building covering one floor</td>
</tr>
<tr>
<td>DS</td>
<td>455</td>
<td>75</td>
<td>Indoor</td>
<td>Office building</td>
</tr>
</tbody>
</table>

### 4.1.3 Challenges with existing indoor datasets

Different indoor datasets have been developed and used in various indoor localisation works. However, the problems with most indoor datasets are:

- not made publically available for research purposes (Kawaji *et al.*, 2010; Ravi *et al.*, 2005).
- cover few indoor locations, such as Arth *et al.* (2009) used images from one long corridor, Hile and Borriello (2007) performed experiments on images of a hallway, or (Menegatti *et al.*, 2004) developed a database covering images from five corridors of the building.
- contain images of non-office buildings (Filliat, 2007; Kawaji *et al.*, 2010)

Image matching becomes challenging in large office buildings due to a high degree of self-similarity among images. I managed to find a challenging indoor dataset (HW) developed by Kang *et al.* (2009), which contains images of an office building. It is a large scale indoor dataset covering 8.8 thousand images, which are repeatedly captured from one floor of an office building over time. But authors have not provided any details regarding the number of used indoor locations, which indicates the minimum number of images required to be captured for every location on average. This prompted us to develop the indoor dataset (CS) for our work. The CS dataset contains images from our computer science building covering 30 different indoor locations, such as corridors, toilet, foyers, or rooms. Moreover, our building has glass windows which makes the CS dataset challenging compared to HW dataset (no glass windows). Indoor image matching becomes particularly hard in office buildings having glass windows because varying intensity of sunlight results in varying light effects in corridors/halls hence posing a challenge for an image matching system.

The CS indoor dataset is used during offline experiments. To test the robustness of our smartphone based localisation system, I have developed four realistic indoor
datasets containing images of four buildings. While developing these datasets, I have taken the following steps in an attempt to make them unbiased and challenging:

- The test and training images are captured from different cameras, such as smartphone camera having a resolution of 5MP is used to capture test images while Sony Cybershot camera with a resolution of 8MP is used to capture training images.

- The captured test images are reduced in size by 50% on the smartphone and are then transferred to the server for matching, which introduces some artifacts hence making query data more challenging.

- The test images are taken few weeks after capturing the training images. This reduces the chances of taking query pictures of an indoor location from the same viewpoint as was used while taking training images. The position of some objects also have changed in query images compared to training images such as location of chairs in coffee room, which further tests the robustness of an algorithm.

- For one dataset (OW), I have taken test images at different times such as morning, noon and night while training data is only captured around noon. Though OW dataset is new version of CS dataset but it is large and contains around 1500 training and 750 test images.

- The collective number of training images in all datasets is around 4 thousand representing 50 different indoor locations. This is a reasonable size to test the robustness of any image matching algorithm.

All developed indoor datasets are available for research purposes and can be sued to test the robustness of any indoor image matching algorithm.

4.2 Performance metrics

We have used different performance metrics to analyse the performance of the components and iPoS itself. The performance of the components is first analysed and is followed by the evaluation of iPoS towards the end of this thesis. All implementations were developed using Visual Studio 2010 on a Windows 7 operating system and all evaluations, are carried out on a single core of 3.6 GHz Intel Core 2 Duo machine.

All performance metrics are listed in this section for reference. Some definitions which define performance metrics, such as similarity threshold, verification method, or voting module are not discussed in this section. These definitions will be discussed in later chapters, where the performance metrics will be used to evaluate the relevant methods.

4.2.1 Definitions

The following definitions are used to define the performance metrics:
\(Q_t\) Total number of query images.

\(Q_c\) Total number of query images corresponding to a particular indoor location such as coffee room, seminar room, graphics corridor etc in the image database.

\(Q_s\) Total number of images for a location.

\(V_t\) Total number of query images passed to the verification method.

\(N_v\) Number of query images correctly matched by the voting module i.e. not passed to the verification method. By definition, \(N_v \leq Q_t - V_t\).

\(N_d\) Number of query images for which no decision is made.

\(M_v\) Number of query images correctly matched by the verification method: \(M_v \leq V_t\).

\(M_t\) Total number of correctly matched query images: \(M_t = N_v + M_v\).

\(M_c\) Number of query images correctly corresponded to a particular indoor location or place during image matching.

\(C_s\) Total number of images of a location grouped on the basis of similarity threshold.

\(W_s\) Total number of a location, which are wrongly grouped.

### 4.2.2 Metrics

The performance metrics used in this thesis based on the above definitions are:

**\(C_a\)** Refers to the correct acceptance rate and indicates the image matching accuracy of a system. Higher is generally better for this metric.

\[
C_a = \frac{M_t}{Q_t},
\]  
(4.1)

This metric is used in Chapters 5, 6, 7, 8 and 10.

**\(W_m\)** Refers to the wrong match rate and indicates the number of wrong image matches made by the system against query images. Lower is generally better for this metric and must be used in conjunction with \(C_a\). A low value of \(W_m\) and high value of \(C_a\) indicates good image matching performance of a system.

\[
W_m = 1 - \left(\frac{N_v + M_v}{Q_t}\right) + \frac{N_d}{Q_t}
\]  
(4.2)

This metric is used in Chapters 7, 9 and 10.
\( W_c \) Refers to the consistency of the weighting scheme used in visual Bag of Words (BoW). Higher is generally better for this metric, which indicates that voting module makes most of the image match decisions without calling the verification method. This metric must be used in conjunction with \( N_c \).

\[
W_c = \frac{Q_t - V_t}{Q_t}.
\]  

(4.3)

This metric is used in Chapter 6.

\( N_c \) Refers to the matching accuracy of voting module. A higher values of this and \( W_c \) metrics indicate good performance of a weighting scheme in visual BoW.

\[
N_c = \frac{N_v}{Q_t - V_t}.
\]  

(4.4)

This metric is used in Chapter 6.

\( R_{nd} \) Refers to the no-decision rate and indicates the number of query images for which visual BoW cannot make an image match. This metric must be used in conjunction with \( W_m \) and \( C_a \). Such value of \( R_{nd} \) is required, which offers a reasonable trade-off between \( C_a \) and \( W_m \).

\[
R_{nd} = \frac{N_d}{Q_t}.
\]  

(4.5)

This metric is used in Chapter 7.

\( S_{cm} \) Refers to the scene confusion matrix and groups the place type as opposed to simply location. The matrix gives the grouping probability of an indoor query location against all trained indoor locations. Higher probability value is better for a particular location and can be obtained as follows:

\[
\text{Probability value} = \frac{M_c}{Q_c}
\]  

(4.6)

A good scene confusion matrix is one which has high probability values for each query location and indicates the success of a localisation system in the context of navigation. This metric is used in Chapter 7.

\( S_c \) Refers to the scene clustering rate. It is the average grouping of images for each location based on overlapping features between images and a similarity threshold \( T \). A high value of this metric shows that more similar images are grouped for each location and the resulting total number of groups will be fewer for each location. Each group of a location may have single or multiple images.

\[
S_c = \left[ \sum_{i=1}^{n} \frac{C_s}{Q_s} \times 100 \right] / n
\]  

(4.7)

where \( n \) refers to the total number of locations. This metric is used in Chapter 8.
**E_s** Refers to the scene clustering error rate. It is the average grouping of non-similar images along with similar images in groups for each indoor location. Lower is better for this metric and is used in conjunction with **S_c**. A low value for **E_s** is required for any value of **S_c** to generate good reduced feature sets via proposed clustering algorithm.

\[
E_s = \left( \sum_{i=1}^{n} \frac{W_s}{C_s} \ast 100 \right) / n
\]  
(4.8)

This metric is used in Chapter 8.

**R_c** Refers to the registration rate and indicates the image matching accuracy against query images. This metric is the same as **C_a** but will be referred to as **R_c** when 3D based approaches are used to match indoor query images.

This metric is used in Chapter 9.

**B_p** is the number of times a pose estimation method produces the best results (i.e. maximum number of inliers) compared to other methods during pose estimation with 3D based image matching approach. Higher value is better for this metric, which indicates the success of a particular pose estimation method.

This metric is used in Chapter 9.

**F_a** Refers to the false acceptance rate and indicates the system performance against negative samples i.e. those query images which are not present in the trained collection. A zero or lower value is better for this metric, which indicates that system matches zero or fewer query images.

\[
F_a = \frac{Q_t - N_d}{Q_t}
\]  
(4.9)

This metric is used in Chapter 10.

### 4.2.3 Selection of metrics

Precision and recall are the two commonly used metrics in many visual BoW or image recognition works. Precision refers to the fraction of number of retrieved positive images out of the total number of retrieved images. While recall refers to the fraction of number of retrieved positive images out of total number of positive images in the collection. These metrics are more useful when the performance of a system is evaluated for retrieval of more than one image against a query image. A system with very good precision and recall rate may not be useful for localisation. Let’s say a system retrieves five images against a query image and produces few wrong matches. However, it often wrongly matches the first retrieved image against a query image. The top matched (first) image is the most important among all retrieved images as it dictates the success or failure in localisation. Therefore, it is not necessary that a system which has high precision and recall rate also suits localisation tasks.
For our work, we want the system to find the top match correctly most of the time while giving few wrong image matches. A wrong location is a disaster for blind people, such as stair is recognised as room. It is better not to find an image match rather than making a wrong match during localisation. Therefore, we have carefully selected the metrics in our thesis, such as correct acceptance, wrong match or no-decision rates as the intended use of our system is to assist blind people indoors with minimum location errors. Moreover, precision rate behaves similarly to correct acceptance rate when only one image is retrieved against a query image.

Indoor localisation research works have used different metrics to evaluate their systems, such as Kang et al. (2009) uses distance information to retrieve top images followed by computation of precision rate, Filliat (2007) uses correct/incorrect/no-decision image match information against query images, Werner et al. (2011) uses distance metric between query and stored images, or Kawaji et al. (2010) uses precision and retrieves only one image against a query image. The unavailability of most indoor datasets and use of different metric makes it hard to compare our system with any of the existing systems. Therefore, we have compared our proposed system with standard visual BoW for a baseline comparison in this thesis.

4.3 Conclusion

We first mentioned all the datasets used for the experimental evaluations followed by the performance metrics used in this thesis for analysis. We used five datasets for offline evaluation of proposed components and the remaining six realistic datasets for realistic evaluation of iPoS.

The camera used to gather query and training data is different for realistic indoor datasets. These datasets contain images of 50 self-similar locations such as corridors, halls etc. This makes these datasets quite challenging and offers a good platform to evaluate indoor localisation performance of any positioning system.

In the next chapter, we briefly discuss and compare several features. We also present proposed shorter version of SIFT features which are used for image matching in iPoS.
Chapter 5

Feature evaluation

Note: Some portions of this chapter are taken from (Khan et al., 2011b).

Indoor Positioning System (iPoS) uses visual Bag of Words (BoW) based on features for the retrieval of relevant images against a query photo. The features are used for further verification and post-verification if required. Therefore, features are the basic building block of visual BoW and an image representation via appropriate features is the key to the success of the proposed system.

In this chapter, we review two popular ways of detecting and describing features, namely SIFT (Scale invariant feature transform) and SURF (Speeded up robust feature) which are widely used in many different works. We also propose variants of SIFT features, which are shorter in size compared to normal 128D SIFT features.

The contribution of this chapter is the comparison of proposed SIFT variants with normal SIFT and SURF features and four other well known feature descriptors. This chapter also thoroughly compares SIFT variants, normal SIFT and SURF features under different image transformations.

5.1 Image descriptions

Vision based scene recognition relies on processing and matching captured images with a database of mapped images to find the required answer. Scene recognition methods typically use features to describe the images and this description is then used for image matching. Images of the same place may be taken from different viewpoints and the images may also undergo different transformations, such as change in scale, blur or translation, which makes it highly likely that images of the same place will tend to be different. Therefore, suitable features are required to describe an image. Good image descriptions can handle image transformations such as illumination changes, addition of noise, and rotation and play an important role in the performance of an image matching system.

There are different methods for selecting a set of image features. Good feature extractor methods aim to find repeatable features i.e. the algorithm should find the same keypoints from similar images taken from varying viewpoints to ensure good
matching performance. The “keypoints” represent unique parts of the image such as edges, blobs or corners, which carry unique information and are used to construct a feature. Each feature is basically a vector of values computed from the keypoint region.

To ensure that features are repeatable, some combination of the following properties are required:

- **Rotation invariance**: The same features should be detected regardless of the orientation of objects in images.
- **Scale invariance**: The same features should be detected regardless of the scale of images.
- **Perspective invariance**: Despite changes in viewpoint the same features should be detected.
- **Affine invariance**: The presence of multiple transformations such as translation or rotation in images should not affect the detection of features.
- **Illumination invariance**: Changes in illumination level in images should not affect the extracted features.
- **Noise invariance**: The presence of motion blur, Gaussian noise etc., in images should not affect the detection of features.

Absolute invariance is seldom possible with any feature. However, we aimed to use those features for iPoS which can achieve many invariance properties in order to obtain robustness in image matching.

### 5.2 Features

A variety of feature descriptors are presented in the literature for reliable image matching such as GIST (Oliva and Torralba, 2001), MSER (Matas et al., 2002), SIFT (Lowe, 2004), PCA-SIFT (Ke and Sukthankar, 2004), HoG (Dalal and Triggs, 2005), SURF (Bay et al., 2008), CENTRIST (Wu and M. Rehg, 2009), FREAK (Alahi et al., 2012).

SIFT (Scale invariant feature transform) is one of the most popular feature extraction algorithms, which was introduced by Lowe (2004). These features provide reliable image matching between different views of an object or image even among clutter and under partial occlusions. The SIFT algorithm applies Gaussian blurring to images at different scales to detect possible keypoints and then develops a distinctive feature descriptor, which is inspired by the human visual system.

SIFT features offer good performance during image matching, but the main problem is that the keypoint detection step is complex and time consuming, which decreases the overall efficiency of the algorithm. Bay et al. (2008) proposed approximate SIFT, which is called SURF (Speeded up robust feature) to address this problem. SURF algorithm uses an efficient way to represent an image and then detect keypoints by using box filter approximations. The authors report SURF to be more efficient in computation and provide better image matching performance than SIFT.
SIFT and SURF features give good performance in image recognition tasks, but both are still computationally expensive and are not suitable for real-time systems such as robot positioning during navigation or image matching on low-power mobile devices. Therefore, efforts have been made to propose feature extractors which can provide more efficiency than SIFT and SURF, but with a comparable performance. CENTRIST (Census transform histogram), is a more efficient visual descriptor than SIFT, which was presented for recognising place categories (Wu and M. Rehg, 2009). It offers a holistic representation and scene categories can be estimated without explicitly detecting and recognising objects in the scene. The authors demonstrated in experiments that CENTRIST is faster to compute than SIFT and also performs better than SIFT for image matching on five datasets including both place and scene category recognition tasks (Oliva and Torralba, 2001).

In one study, Calonder et al. (2010) presented BRIEF (Binary robust independent elementary features), a binary descriptor which uses binary strings for efficient performance. It is shown to perform better than SURF for image matching against different image transformations on benchmark images and is reported to be more efficient than SURF. This feature descriptor can be computed using simple intensity difference tests and offers high image recognition rates. The descriptor similarities are evaluated using Hamming distance. Therefore, BRIEF is very fast both in building and matching.

In a recent work, Alahi et al. (2012) presented FREAK (Fast retina keypoint), a novel keypoint descriptor which is inspired from the retina of the human visual system. The algorithm detects keypoints from an image by an algorithm, such as SURF or BRIEF and then computes the FREAK descriptor for each detected keypoint. The authors report that FREAK descriptor is two orders of magnitude faster in computation than SIFT and SURF descriptors, which improves the overall efficiency during image matching. FREAK is also demonstrated by the authors to outperform SIFT and SURF features during image matching tasks with respect to precision and recall rates.

Up until now, SIFT and SURF are the two most popular invariant descriptors which describe image patches around detected blobs such as corners, edges etc., and use these patches for descriptor computation. Both of these descriptors are highly repeatable and have been used in many applications due to good image matching performance. For this reason, researchers often compare their proposed features against these two benchmark features.

In the following subsections, we briefly discuss different features which are used on indoor and outdoor datasets in this thesis to analyse image matching performance. The purpose is to identify the best feature and use it for image matching in this work.

### 5.2.1 Scale Invariant Feature Transform (SIFT)

SIFT was developed by Lowe (Lowe, 2004) and is a continuation of his previous work on invariant feature detection (Lowe, 2001). It has the ability to robustly identify scenes even among clutter and under partial occlusions because its feature descriptors are invariant to scale and rotation, and partially invariant to illumination changes and viewpoint. A cascade filtering approach is presented to detect the features that transforms image data into scale invariant coordinates relative to local features. There are four major computational stages of SIFT:
1. **Scale Space Extrema detection:** This stage detects possible keypoints from the image and takes most of the computation time. This step ensures invariance to scale.

2. **Keypoint localisation:** This stage involves the filtering of those keypoints which are unstable. Low contrast keypoints and those which are poorly localised along an edge are removed.

3. **Orientation assignment:** The remaining stable keypoints are assigned one or more orientations based on local image gradient directions. This results in rotation invariance because the keypoint descriptor can be represented relative to this orientation.

4. **Keypoint descriptor:** The previous steps ensured invariance to image location, scale and rotation. This stage computes a highly distinctive descriptor vector that is partially invariant to remaining variations such as illumination, viewpoint etc.

First, a 16x16 window around each keypoint is broken into sixteen 4x4 windows followed by computation of gradient magnitudes and orientations for each window. A set of orientation histograms are then created for every window with 8 bins each. The amount of orientation, which is added to a bin not only depends upon its magnitude, but also depends upon the distance from the keypoint. So gradients that are far away from the keypoint add smaller values to bins and have less weights than other gradients. To do this, a Gaussian weighting function is applied to generate a gradient, which is then multiplied with magnitude of orientations to add weighted magnitudes or Gaussian weights to corresponding bins. The descriptor is finally represented by a vector of values from 8 bin histograms from all 4x4 windows. This descriptor vector is then normalised to unit length to achieve invariance to illumination.

For more details about these stages, refer to Appendix A. The SIFT descriptor size is controlled by its width i.e. the array of orientation histograms \((n \times n)\) and number of orientation bins in each histogram \((r)\). The size of resulting SIFT descriptor is \(r n^2\). Lowe experimented with various sizes of SIFT descriptor by varying values for \(r\) and \(n\). It was concluded by Lowe that SIFT is superior in matching precision with \(n = 4\) and \(r = 8\), which gives a 128 dimensional feature vector.

**Implementation**

I re-implemented the SIFT algorithm for my thesis in C++ using Visual Studio. The purpose of implementing my own algorithm was to modify the SIFT algorithm if required to improve the efficiency or accuracy during image matching tasks.

The images are resized to a resolution of 640 x 480 pixels before extracting the SIFT features and the same image resolution is also used for extracting other features. Moreover, I do not double the input image size to avoid a very large number of features from the training data. Other than that, the whole implementation is similar to the original SIFT algorithm.
5.2.2 SIFT variants

Since the SIFT algorithm was formally proposed, researchers have been trying to improve its performance either in terms of image matching accuracy or speed. Some of the popular variants of the SIFT are:

- **PCA-SIFT**: Ke and Sukthankar (2004) proposed the use of Principal Component Analysis (PCA) instead of histogram to normalise the gradient patch in the last stage of SIFT to generate a compact feature description. PCA is an effective data dimensionality reduction technique which performs orthogonal transformation to convert a vector with correlations between components, to a new random vector with no correlations between components. In PCA-SIFT, a vector of 3042 elements is first created by concatenation of the horizontal and vertical gradients for the 41x41 patch centered to the keypoint. This is followed by normalisation and then dimensionality reduction of the vector to 20 elements via PCA. Therefore, the resulting SIFT feature descriptor contains only 20 elements.

- **GSIFT**: SIFT only describes local information and does not make use of global information while generating a descriptor. Mortensen et al. (2005) introduced a SIFT descriptor with global context (GSIFT), which adds a global texture vector to SIFT. For each detected keypoint, GSIFT establishes a vector that consists of two parts. One part is the SIFT descriptor of a local feature (128 elements) and another part is a global texture vector (60 elements) to distinguish similar local features. GSIFT generates a curvature image by computing the maximum curvature of each pixel. The curvature is the largest absolute eigenvalue of the Hessian matrix. For every detected feature, the global shape context accumulates curvature values in a log-polar bin, which is basically a 5 x 12 histogram. Therefore, the resulting descriptor contains 188 elements. Both PCA-SIFT and GSIFT make changes in the last stage of SIFT algorithm.

- **HSV-SIFT**: While detecting keypoints, SIFT only uses grayscale information of an image and discards a lot of color information. Bosch et al. (2008) introduced HSV-SIFT, which calculates the traditional SIFT descriptors on each channel of HSV-color of an image and produces a 128 x 3 dimensional SIFT descriptor for each point. In another work, Abdel-Hakim and Farag (2006) presented CSIFT, which adds color invariance to the basis of SIFT to overcome the shortcomings of SIFT for color images. The authors use color invariance model based on Kubelka-Munk theory of reflectance to generate a color image from the input image, which is invariant to surface direction, light intensity and reflection. The invariant color image is then used to establish a DoG pyramid followed by keypoints detection and generation of feature descriptors in the same way as SIFT.

- **ASIFT**: Affine changes affect the image matching performance of SIFT. To address this problem, Morel and Yu (2009) proposed an affine transformation model which performs rotation and tilt transformations to correct the input image. ASIFT then detect keypoints and generates features from the affine image in the same way as done by SIFT.

58
Among above variants, PCA-SIFT seems to be a good choice due to the reduced size of SIFT descriptors. However, it has been shown in comparison works that SIFT performs better for image matching than PCA-SIFT in most of the image transformation experiments (Juan and Gwon, 2009; Wu et al., 2013). The remaining variants such as CSIFT and GSIFT perform comparable or sometimes better than SIFT but the size of color based SIFT feature descriptors is very large (Wu et al., 2013). On the other hand, ASIFT only performs pre-processing and does not reduce the feature size. This motivated us to investigate some other way to reduce the size of SIFT descriptor (if possible) and still get a performance comparable to the standard 128 dimensional SIFT descriptor.

In traditional 128D SIFT, there is a 4x4 array of orientation histograms with each one having 8 values as shown in Figure 5.1. As discussed before, the gradients that are far away from the keypoint center add smaller values to bins and have less weights than the other gradients in traditional SIFT. To achieve this, a Gaussian weighting function is applied to generate a gradient, which is then multiplied with magnitude of orientations to add weighted magnitudes or Gaussian weights to corresponding bins. This means that regions which are far from the keypoint center have less weighted magnitudes than nearby regions. We decided to use this information to reduce the size of SIFT descriptors by performing a post-processing step, which involves skipping the orientation values from some regions of the 4 x 4 block. To our knowledge, no one has used this way to reduce the size of a SIFT feature.

![Figure 5.1: 4 x 4 orientation histogram each with 8 bins.](image)

We have generated three variants for SIFT as shown in Figure 5.2. The SIFT variants are as follows:

1. **96D SIFT**: We ignored the corner regions. The rejected regions have the least Gaussian weight because the distance of these regions is the highest from the center of keypoint i.e. $\sqrt{2}$.

2. **64D SIFT**: We computed the correlation and normalised Euclidean distance between the vectors (8 dimensional) of neighboring outside regions for 100 images. Correlation is a statistical relationship between two vectors. The value of 1 means perfect correlation and -1 means no correlation. On the other hand, less distance indicates that vectors are similar. The purpose was to identify a similarity between neighboring outside region vectors. We realised from experiments
that on average the correlation was 0.6 and normalized distance was 0.3, which indicates a good similarity between the vectors of neighboring outside regions. Therefore, we decided to further reduce the 96D SIFT feature vector by averaging the neighboring outside regions.

3. **32D SIFT**: We used only the central 2 x 2 block. These regions are nearest to the center of keypoint and have highest Gaussian weight.

![Customised 4 x 4 orientation histogram array configurations used to generate shorter SIFT descriptors.](image)

**Figure 5.2** Customised 4 x 4 orientation histogram array configurations used to generate shorter SIFT descriptors.

### 5.2.3 Speeded up Robust features (SURF)

SURF is another robust visual descriptor for image matching and is partly inspired by SIFT (Bay et al., 2008). SURF replaces complex functions of SIFT with simple box filters during keypoint detection stage, which improves its efficiency. The standard version of SURF is several times faster than SIFT and is claimed by its authors to be more robust than SIFT. There are two main stages in SURF:

1. **Keypoint detection**: This stage detects possible keypoints from the images via efficient use of integral images. An integral image represents the sum of all pixels in the input image or rectangular regions in the input image. The input image is subjected to scale space construction after conversion into an integral image. During scale space construction, the integral images allow the fast computation of approximate Laplacian of Gaussian (LoG) images using an efficient box filter representation. The computational cost of applying the box filter is independent of the filter size because of the integral image representation. SURF therefore uses an approach opposite to SIFT by keeping the image size the same and varying the filter size. The determinant of Hessian matrix is then used to detect the blob-like structures at different scales. This results in location and scale invariance.

2. **Keypoint description**: In this stage, a highly distinctive descriptor is computed. The SURF descriptor is computed by constructing a square window centred around the keypoint and oriented along the orientation obtained before. This window is then divided into 4x4 regular sub-regions and Haar wavelets (Graps, 1995) are calculated within each sub-region. Each sub-region contributes four values thus resulting in 64 dimensional descriptor which is then
normalised to unit length. The resulting SURF descriptor is invariant to rotation, scale, contrast and partially invariant to other transformations.

Shorter SURF descriptors can also be computed however the best results are reported with 64D SURF descriptors (Bay et al., 2008).

**Implementation**

I used OpenSURF implementation (Evans, 2009) to compute the SURF features from the images. The source code is written in C++ and is based on OpenCV (Bradski and Kaehler, 2008).

### 5.2.4 Histogram of Gradients (HoG)

HoG features work by counting the occurrences of gradient orientation in localised portions of an image (Dalal and Triggs, 2005). These features are popular for object detection and are used in different works (Scherer et al., 2010; Velmurugan and Baboo, 2011). The basic idea behind HoG descriptors is that local object appearance in an image can be described by the distribution of edge directions or intensity gradients.

To generate the HoG feature, the input image is first divided into a grid of small rectangular, square or circular regions called cells. The descriptor is formed by combining the histograms of gradient directions or edge orientations for the pixels in all cells. The local histograms are further normalised based on contrast to improve the descriptor invariance to illumination or shadowing. Such normalisation is achieved by calculating a measure of the intensity across a larger region of the image (a block) and then using this value to normalise the cells within that block. The HoG descriptor upholds invariance to geometric and photometric transformations, except for orientation because it operates on localised cells.

The algorithm for HoG is similar to edge orientation histograms (Freeman et al., 1996), SIFT, and shape context (Belongie et al., 2001) but differs in that it is computed on a dense grid of uniformly spaced cells.

**Implementation**

I used HoG code (in MATLAB) from the MASH project (Fleuret, 2011). The code generates a HoG feature from each image having 81 values which results in fast image matching.

### 5.2.5 GIST

Oliva and Torralba (2001) presented GIST descriptors to represent the spatial envelope of a scene for recognition. The spatial envelope is a set of holistic scene properties (naturalness, openness, roughness, expansion, ruggedness), which can be used for inferring the semantic category of a scene without the need for segmentation and processing of objects. These holistic properties represent the dominant spatial structure of a scene and provide a meaningful description for scene categorisation.
To generate the GIST descriptor, the image is first divided into 4x4 sub-images followed by the computation of Gabor responses at different orientations and scale in each of these sub-images. The Gabor filters are basically bandpass filters which are used for edge detection (Risojević et al., 2011). Histograms of Gabor responses are then computed from each sub-image, leading to a GIST feature vector for the image which is further subjected to Principal Component Analysis (PCA) for dimension reduction.

The holistic properties considered in GIST are briefly discussed:

1. **Degree of Naturalness**: The scenes with a distribution of edges commonly found in natural landscapes would have a high degree of naturalness compared to man made scenes with vertical and horizontal orientations.

2. **Degree of Openness**: A scene can have a closed spatial envelope full of visual references e.g. a forest, a mountain etc or it can be vast and open to infinity e.g. a coast, a highway etc. The existence of a horizon line and the lack of visual references lead to a high degree of Openness for a scene.

3. **Degree of Roughness**: It refers to the size of its major components at each spatial scale and their abilities to build complex elements. Basically, it refers to the scene complexity.

4. **Degree of Expansion**: It refers to convergence of parallel lines which give perception of depth gradient of the space. For example, a flat view of a building would have a low degree of expansion. On the other hand, a street with long vanishing lines would have a high degree of expansion.

5. **Degree of Ruggedness**: It refers to the deviation of the ground with respect to the horizon. A rugged environment produces oblique contours in the picture and hides the horizon line. Most man made environments are built on a flat ground. Therefore, natural environments are more rugged compared to others.

**Implementation**

I used the source code provided by the authors to generate GIST descriptors (Oliva, 2006). The code is in MATLAB and each image is represented by a GIST feature vector having 512 values.

**5.2.6 Oriented FAST and Rotated BRIEF (ORB)**

Rublee et al. (2011) developed ORB, a very fast binary descriptor which is rotation invariant and resistant to noise. The descriptor is shown to perform comparable to SIFT/SURF, while being almost two orders of magnitude faster. ORB is based on the FAST keypoint detector (Rosten and Drummond, 2006) and the BRIEF descriptor (Calonder et al., 2010). BRIEF is a recent feature descriptor that offers a bit string description of an image patch constructed from a set of binary intensity tests. The BRIEF descriptor vector is obtained by comparing the intensity of 512 pairs of
pixels after applying a Gaussian smoothing to reduce the noise sensitivity while positions of the pixels are pre-selected randomly according to a Gaussian distribution around each patch centre.

The ORB descriptor starts by detecting FAST keypoints based on Harris corner measure (Harris and Stephens, 1988). To detect features at different scales, an image pyramid is used and features are computed at each level. A 9x9 patch is used with the keypoint as its centre and patch orientation is computed using first order moments. Calonder et al. (2010) computed the BRIEF descriptor for a set of rotations and perspective warps of each patch. However, the ORB authors propose to compute the descriptor based on measured patch orientation, which is more efficient than the method used in BRIEF. A learning method is then used for de-correlating the feature descriptors under rotational invariance leading to ORB features. ORB is claimed by its authors to be an order of magnitude faster than SURF and over two orders faster than SIFT in feature detection and description on images with 640x480 resolution. The authors report ORB features to outperform SIFT/SURF in nearest neighbor matching over large databases of images in their experiments.

Implementation

I used the C++ source code provided by the ORB authors. In ORB, the number of features are fixed per image and I have used 200 features per image, where each image has 640x480 resolution.

5.2.7 Fast Retina Keypoint (FREAK)

Alahi et al. presented a novel key point descriptor inspired by the human visual system and more precisely the retina (Alahi et al., 2012). The descriptor involves a cascade of binary strings, which are computed by efficiently comparing image intensities over a sampling pattern similar to the distribution of retinal cells in the human retina. These features are two orders of magnitude faster to compute than SIFT and SURF descriptors. FREAK features require lower memory than SIFT and SURF and are also claimed to be more robust than SIFT and SURF for image matching with respect to precision and recall (Alahi et al., 2012).

FREAK is a descriptor only and relies on a robust feature detector algorithm for keypoint detection. The sampling pattern around the detected keypoint is circular and has higher density of points near the centre. The sample points are smoothed to reduce the sensitivity to noise by using Gaussian kernels. The authors use different sized kernels for every sample point, which leads to overlapping receptive fields. The binary descriptor is then created by thresholding the difference between pairs of receptive fields with their corresponding Gaussian kernel. However, receptive fields can lead to thousands of pairs, which can result in a large descriptor. A strategy is needed for the selection of suitable pairs to describe an image. Therefore, FREAK descriptor is computed from the selected pairs by running an algorithm similar to ORB. The authors demonstrate in their work that selected pairs yield a highly structured pattern that mimics the sporadic search of human eyes. The orientation of the keypoint is finally
obtained via local gradients over the selected pairs leading to a feature descriptor which offers good image recognition.

**Implementation**

I used the C++ source code provided by the authors to compute the FREAK features. The SURF detector is first used to compute the keypoints followed by the extraction of the FREAK features.

5.3 Evaluation of features

We have measured the performance of the above features using an image matching task on a number of standard datasets using a naive matching algorithm. In naive matching, all feature are extracted from each training image in the dataset and are stored in a kd-tree (Moore, 1991). To perform image classification, all nearest neighbors of features of a query image are found in the image collection based on some distance computation such as Euclidean or Hamming in the space of feature descriptors. The nearest neighbor of a query feature is defined as the training feature from the collection, which has minimum distance against that query feature. If the nearest neighbor of a query feature is within a distance threshold, then a feature correspondence is recorded between a query image and the training image from the collection to whom that nearest neighbor belongs. The training image with most the feature correspondences from the image collection is selected as the best match for a query image.

Therefore, a suitable distance threshold is required which ensures many correct feature matches and rejects most wrong feature matches between images. Based on this criterion, we used different distances and thresholds (determined empirically) with each feature:

- **SIFT**: We used the Euclidean distance for the nearest neighbor search. To determine the suitable Euclidean distance threshold for feature comparison, we performed a test. We picked 1500 training and 100 test images from the David Nister (DN) dataset. We extracted SIFT features from test images and compared them with features of each training image with different Euclidean distance values. The purpose was to analyse the correct image matching performance with test data against different Euclidean distances. We then picked those distance values as thresholds for our SIFT variants, which gave the highest correct image matching performance. We have used 170, 160, 150 and 90 distance thresholds respectively for 128D, 96D, 64D and 32D SIFT descriptors. These are not optimal thresholds as it is quite hard to find one which gives all correct feature correspondences between two images. However, the used thresholds attempt to reject many wrong SIFT feature correspondences while matching two images and have worked well across all datasets in our experiments.

- **SURF**: We used the default threshold provided in the OpenSURF code i.e. \((d_1/d_2 < 0.65)\); where \(d_1\) and \(d_2\) refer to first and second nearest neighbors against a query feature. The idea is to find the two best matched features (first
and second nearest neighbor) against a query feature based on euclidean distance i.e. \( d_1 \) and \( d_2 \). A correspondence between query and best matched features is accepted only if the above ratio test passes.

- **FREAK and ORB**: We used the Hamming distance for the nearest neighbor search for both FREAK and ORB features. A distance threshold of 50 is used for both. A similar way is used to find threshold as used for SIFT variants.

- **HoG and GIST**: We used the Euclidean distance for the nearest neighbor search for both HoG and GIST features. Both algorithms generate one feature vector for the whole image. Therefore, nearest neighbor is simply picked as the best match in this work.

### 5.4 Datasets and performance metrics

We used four datasets to evaluate the performance of features. In each dataset, the training images are different from the test images. For every test image of a scene, there are some training images of the same scene taken from different viewpoints. To perform image matching, naive matching finds an image match for a test image from training images in the same way as discussed in the previous section. The image match is only considered right if the matched training image refers to the same scene or object as of the test image. Otherwise, the image match is considered wrong for a test image.

These datasets are briefly discussed below. The details of these datasets can be found in Chapter 4.

- **David Nister (DN)**
  We used the first 4000 images from this dataset to get a reasonable number of training features. The first image of every scene was used for testing while the remaining three were used for training i.e. 1000 test and 3000 training images.

- **Computer Science (CS)**
  We used all images from this dataset, 630 images for training and 70 images for testing.

- **Hongwen (HW)**
  We used the first 3000 and the first 100 images for training and testing respectively.

- **Caltech Buildings (CB)**
  This dataset contains five images each for a total of 50 building exteriors. We used all images from this dataset in experiments. The first image of every building was used for testing while the remaining four were used for training.

The training images are different from test images. The large number of training images from DN and HW datasets results in a reasonable number of training features for a fair analysis in our experiments. Each feature extractor generates different number
of features from training images as stated in Table 5.1. The table clearly shows that HoG and GIST are extremely compact as they generate few features i.e. one feature vector per image. This leads to very high efficiency but precision needs to be good as well. On the other hand, the ORB features are fewer than SIFT, SURF and FREAK. SIFT features are less in number than SURF and FREAK features because I have not doubled the input image size in the SIFT algorithm while generating features.

Table 5.1: Statistics indicating number of training features for datasets, M means million.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Images</th>
<th>SIFT</th>
<th>SURF</th>
<th>FREAK</th>
<th>ORB</th>
<th>GIST</th>
<th>HoG</th>
</tr>
</thead>
<tbody>
<tr>
<td>DN</td>
<td>1000</td>
<td>3000</td>
<td>0.99 M</td>
<td>2.0 M</td>
<td>1.0 M</td>
<td>0.60 M</td>
<td>3000</td>
</tr>
<tr>
<td>CS</td>
<td>70</td>
<td>630</td>
<td>0.17 M</td>
<td>0.43 M</td>
<td>0.27 M</td>
<td>0.13 M</td>
<td>630</td>
</tr>
<tr>
<td>HW</td>
<td>100</td>
<td>3000</td>
<td>1.61 M</td>
<td>1.90 M</td>
<td>0.67 M</td>
<td>0.60 M</td>
<td>3000</td>
</tr>
<tr>
<td>CB</td>
<td>50</td>
<td>200</td>
<td>0.08 M</td>
<td>0.20 M</td>
<td>0.15 M</td>
<td>0.04 M</td>
<td>200</td>
</tr>
</tbody>
</table>

We have used the following performance metric in this chapter to evaluate and compare the features:

\[ C_a : \text{refers to the correct acceptance rate.} \]

The expected value for this metric ranges from 0-1. A higher value is better for this metric and indicates a high image matching accuracy against test images.

Please refer to Chapter 4 for more details about this metric.

5.5 Results

In this section, we compare different features and report the classification results. This is followed by an evaluation of SIFT and SURF features against different image transformations.

5.5.1 General matching

We evaluated the classification performance of all features on the four datasets and the results are summarised in Figure 5.3. The results show that the HoG features perform the worst and the corresponding correct acceptance rate \( C_a \) remains below 0.6 for all datasets, which is due to a very compact feature representation of each image i.e. a feature vector with 81 values. GIST features are normally preferred for indoor image matching (Quattoni and Torralba, 2008). But our results show that they have not performed well on indoor datasets with \( C_a \) below 0.62. We suspect the reason is still the compact feature representation of an image, such as a vector of 512 elements per image. On outdoor datasets, GIST features only perform well with the CB dataset and give a \( C_a \) of 0.78 because the number of training images is small compared to other datasets i.e. 200 training images. Good algorithms should generate features, which
can maintain distinctiveness to give good image matching performance not only with small but also with large data sets.

ORB features perform well across all datasets except the CB dataset, where the $C_a$ goes below 0.7. The CB dataset contains building exterior images, which are captured at different times of the day. We suspect that non-geometric transforms in the CB dataset, such as blur, exposure, or illumination result in a poor performance of ORB features (Heinly et al., 2012). Non-geometric transforms consist of those that are image-capture dependent and do not rely on the viewpoint. These transforms result in less-distinct image gradients and changes to the relative difference between pixels due to exposure or lighting change hence affecting the performance of ORB.

![Figure 5.3: Features matching performance on Benchmark Datasets.](image)

FREAK is a descriptor only and uses SURF for feature detection. FREAK offers fast image matching and was claimed to perform better than SURF by its authors (Alahi et al., 2012). However, our experiments show that SURF and FREAK features perform similarly on most datasets for correct image matching and outperform each other by a good margin only on one dataset. SURF performs better across the CB dataset with a $C_a$ of 0.98 compared to FREAK that gives a $C_a$ of 0.84 on the same dataset. On the other hand, FREAK gives a $C_a$ of 0.72 across the DN dataset while SURF gives a $C_a$ of 0.62. FREAK is a new descriptor and it’s performance still needs to be more thoroughly evaluated against different datasets or transformations. On the other hand, SURF seems to be a better choice over FREAK because it has been used across different types of datasets and has performed well.
All SIFT features including 128D, 96D and 64D SIFT have performed well across all datasets and their average $C_a$ are 0.84, 0.83 and 0.81 respectively across four datasets as shown in Table 5.2. 64D SIFT has also performed well and we suspect that averaging of corresponding regions minimises the sudden change in two region vectors, which may happen due to noise, blur or occlusion, which has resulted in equally good performance. The 32D SIFT features perform poorly across all datasets because the size of the descriptor is quite small and it fails to capture the distinctive information. As we can see, 32D SIFT has performed poorly on the HW dataset because of very large number of training features i.e. 1.61 M compared to the other datasets.

Table 5.2: Average correct acceptance rate ($C_a$) of features across all datasets.

<table>
<thead>
<tr>
<th>SIFT32</th>
<th>SIFT64</th>
<th>SIFT96</th>
<th>SIFT128</th>
<th>SURF</th>
<th>FREAK</th>
<th>ORB</th>
<th>GIST</th>
<th>HoG</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.61</td>
<td>0.81</td>
<td>0.83</td>
<td>0.84</td>
<td>0.79</td>
<td>0.78</td>
<td>0.78</td>
<td>0.64</td>
<td>0.46</td>
</tr>
</tbody>
</table>

Table 5.2 indicates that 32D SIFT, HoG and GIST do not perform well and are, therefore, not a good choice for image matching. On the other hand, SIFT variants (excluding 32D SIFT) perform better than SURF, ORB and FREAK. Though it is not clear whether this is a significant performance improvement due to a small difference among the $C_a$ of these features. Therefore, we decided to perform a one tailed paired t-test to further evaluate the significance of our results. We have used $C_a$ of features across the four datasets as data for the test. Since the number of datasets are few and there is not a very large difference in performance among some variants, the test is underpowered. The power of a statistical test is the probability that the test will reject the null hypothesis, when the alternative hypothesis is true. The statistical power of a test increases with the number of samples and it becomes easier to detect effects in performances. We have not used 32D SIFT in the test because it has not performed well across all datasets. Our null hypothesis is: means of SIFT variant and the other features are equal. While our alternative hypothesis is: there is a significant difference between the means of SIFT variant and the other features i.e. SIFT does better. We have used the default value of alpha (α) i.e. 0.05 to reject our null hypothesis. The computed results in Table 5.3 show that:

- The $p$-values for SIFT variants against GIST and HoG features are less than $α$, which indicates the rejection of the null hypothesis. Therefore, we can say with confidence that SIFT variants perform better than GIST and HoG.
- The $p$-values are higher than $α$ between SIFT variants and SURF, which indicates that we cannot reject the null hypothesis. Same is the case with ORB and FREAK features.

Despite the fact that the results are not significant, some choice needs to be made and on balance it seems that SIFT variants are a little bit better than SURF, FREAK and ORB. Ideally, more datasets should be used to fully tease apart the performance of all features.
Table 5.3: \( p \)-values for SIFT variants versus other features.

<table>
<thead>
<tr>
<th></th>
<th>SURF</th>
<th>FREAK</th>
<th>ORB</th>
<th>GIST</th>
<th>HoG</th>
</tr>
</thead>
<tbody>
<tr>
<td>SIFT128</td>
<td>0.256</td>
<td>0.093</td>
<td>0.246</td>
<td>0.004</td>
<td>0.007</td>
</tr>
<tr>
<td>SIFT96</td>
<td>0.326</td>
<td>0.156</td>
<td>0.300</td>
<td>0.001</td>
<td>0.020</td>
</tr>
<tr>
<td>SIFT64</td>
<td>0.365</td>
<td>0.156</td>
<td>0.300</td>
<td>0.001</td>
<td>0.020</td>
</tr>
</tbody>
</table>

5.5.2 Image transformations

For our indoor localisation system, the query image is captured from the smartphone carried by the user. There is high probability that the captured query image may be slightly rotated, have blur introduced due to a hand jerk, have different scale, or have Gaussian noise introduced due to lightening/exposure compared to the captured training images. SURF features have performed next best after SIFT variants as discussed in the previous section. So we decided to further evaluate the performance of SIFT variants and SURF features against different image transformations on images of our indoor datasets. The purpose of this evaluation is to figure out which features perform better under different transformations and select suitable features for our work.

We have used 100 test images from HW and CS indoor datasets each and applied different image transformations to generate training images for each dataset. Once we have test and corresponding transformed training images, we computed \( C_a \) across both datasets. Some of the examples of test and corresponding training images are shown in Figure 5.4.

The comparison of SIFT variants and SURF features under different image deformations is discussed as follows:

- **Rotation invariance:** To test rotation invariance, test images were rotated at six different angles in a clockwise direction to generate corresponding training images. The `imrotate` function of MATLAB with ‘bilinear’ interpolation was used to perform rotations. The experiments tested various values for rotation, such as 40, 90, 135, 215, 250 and 300. The rotated images contain a black portion around its borders, which is due to padding of zero values by the MATLAB function. This does not effect SIFT variants and SURF, because no keypoints are detected due to the lack of blobs in those areas. The \( C_a \) of SIFT variants and SURF across CS and HW indoor datasets is shown in Figure 5.6.

  The results show that SURF slightly performs better than SIFT variants in rotation experiments. The \( C_a \) for SURF is 0.987 on average for both datasets, while the best \( C_a \) for SIFT variants is 0.97 as shown in Table 5.4. We decided to perform a one tailed paired t-test with \( \alpha=0.05 \) to check the significance of these results. The \( C_a \) of features from both datasets is used as data for the \( t \)-test. Therefore, the data for each feature contains a total of 12 observations collectively from two datasets. Ideally, observations from multiple datasets should be used to increase the statistical power of the test. Our null hypothesis is: means of SIFT variants and SURF are equal. While our alternative hypothesis is: there is a significant difference between the means i.e. SURF is better than SIFT variants. The results in Table 5.5 show that \( p \)-values are very small, which indicates...
Figure 5.4: The actual and corresponding deformed training images.

<table>
<thead>
<tr>
<th>Actual Images</th>
<th>Transformed Images</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Rotated images</td>
</tr>
<tr>
<td></td>
<td>Angle: 40</td>
</tr>
<tr>
<td></td>
<td>Angle: 250</td>
</tr>
<tr>
<td></td>
<td>Blurred images</td>
</tr>
<tr>
<td></td>
<td>Blur: 5</td>
</tr>
<tr>
<td></td>
<td>Blur: 10</td>
</tr>
<tr>
<td></td>
<td>Illuminated images</td>
</tr>
<tr>
<td></td>
<td>Sigma: 0.5</td>
</tr>
<tr>
<td></td>
<td>Sigma: 1.8</td>
</tr>
<tr>
<td></td>
<td>Noisy images</td>
</tr>
<tr>
<td></td>
<td>Gaussian: 0.01</td>
</tr>
<tr>
<td></td>
<td>Gaussian: 0.1</td>
</tr>
<tr>
<td></td>
<td>Scaled images</td>
</tr>
<tr>
<td></td>
<td>Scale: 0.25</td>
</tr>
<tr>
<td></td>
<td>Scale: 3.0</td>
</tr>
</tbody>
</table>

Figure 5.5: SIFT versus SURF performance on rotated images across the CS dataset.
Figure 5.6: SIFT versus SURF performance on rotated images across the HW dataset.

the rejection of null hypothesis i.e. SURF performs better than SIFT variants in rotation experiments.

Table 5.4: Average correct acceptance rate \( (C_a) \) across rotated images.

<table>
<thead>
<tr>
<th>SIFT64</th>
<th>SIFT96</th>
<th>SIFT128</th>
<th>SURF</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.929</td>
<td>0.958</td>
<td>0.970</td>
<td>0.987</td>
</tr>
</tbody>
</table>

Table 5.5: SURF versus SIFT variants \( p \)-values for rotation experiments.

<table>
<thead>
<tr>
<th>SURF</th>
<th>SIFT64</th>
<th>SIFT96</th>
<th>SIFT128</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.406e-07</td>
<td>1.154e-07</td>
<td>1.527e-07</td>
<td></td>
</tr>
</tbody>
</table>

- **Illumination invariance:** To change illumination of images, we applied Gamma correction on test images to generate training data with different luminance. Gamma correction is a nonlinear operation used to code and decode luminance. The `imadjust` function of MATLAB was used with different Gamma values, such as 0.3, 0.5, 0.8, 1.3 and 1.5, to generate training data with varying luminance. A Gamma value of less than 1 increases the brightness of an image while a large value darkens an image. The performance of features on both datasets are shown in Figures 5.7 and 5.8.

The results show that SURF performs slightly better than SIFT variants. The SURF gives an average \( C_a \) of 0.985 across both datasets while SIFT variants give \( C_a \) up to 0.964 as shown in Table 5.6. Due to a difference in performances, we performed one tailed paired t-test with \( \alpha = 0.05 \). The data for the test uses \( C_a \) of features from both datasets and there is a total of 10 observations for each feature from two datasets. Our null hypothesis is: means of SIFT variants and SURF are equal. While our alternative hypothesis is: there is a significant difference between the means i.e. SURF is better than SIFT variants. The results in Table 5.7 show that \( p \)-values are smaller than \( \alpha \) indicating the rejection of null
hypothesis. Therefore, we can say that SURF performs better than SIFT variants in illumination experiments.

Table 5.6: Average correct acceptance rate ($C_a$) across illuminated images.

<table>
<thead>
<tr>
<th></th>
<th>SIFT64</th>
<th>SIFT96</th>
<th>SIFT128</th>
<th>SURF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.947</td>
<td>0.960</td>
<td>0.964</td>
<td>0.985</td>
</tr>
</tbody>
</table>

Table 5.7: SURF versus SIFT variants $p$-values for illumination experiments.

<table>
<thead>
<tr>
<th></th>
<th>SIFT64</th>
<th>SIFT96</th>
<th>SIFT128</th>
</tr>
</thead>
<tbody>
<tr>
<td>SURF</td>
<td>4.817e-05</td>
<td>1.187e-04</td>
<td>9.708e-05</td>
</tr>
</tbody>
</table>

- **Blurring:** To test the blurring invariance, we first used \textit{fspecial} function of MATLAB to generate a Gaussian filter (20x20). We then used it to filter the
test image by using the *imfilter* function of MATLAB to generate the training data. The experiments tested various values of Gaussian blur i.e. $\sigma = 5$, $\sigma = 10$ and $\sigma = 20$. The value of 20 for $\sigma$ is very large and resulting training image has a huge amount of blur, which poses a challenge for feature matching. The performance of SIFT and SURF variants across both datasets is shown in Figures 5.9 and 5.10.

![Figure 5.9: SIFT versus SURF performance on blurred images across the CS dataset.](image)

![Figure 5.10: SIFT versus SURF performance on blurred images across the HW dataset.](image)

Above results show that SIFT variants and SURF both handle blurring equally well. The results in Table 5.8 also confirm that that is no difference in the average performance of SIFT variants and SURF. Therefore, we can deduce that SIFT variants and SURF are blurring invariant and perform equally well.

- **Noise invariance:** To test the noise invariance, we used the *imnoise* function of MATLAB to apply Gaussian noise on test images to generate training data.

73
Table 5.8: Average correct acceptance rate \( (C_a) \) across blurred images.

<table>
<thead>
<tr>
<th></th>
<th>SIFT64</th>
<th>SIFT96</th>
<th>SIFT128</th>
<th>SURF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.98</td>
<td>0.99</td>
<td>0.99</td>
<td>0.99</td>
</tr>
</tbody>
</table>

We added Gaussian white noise with \( \sigma^2 = 0.01 \), \( \sigma^2 = 0.05 \) and \( \sigma^2 = 0.1 \). For value of \( \sigma^2 = 0.1 \), the degradation in the image is significant due to the addition of a lot of noise. The performance of SIFT variants and SURF on both datasets using noise transformation is shown in Figures 5.11 and 5.12.

Figure 5.11: SIFT versus SURF performance on noisy images across the CS dataset.

Figure 5.12: SIFT versus SURF performance on noisy images across the HW dataset.

The results show that SURF seems to perform slightly better than SIFT variants only across the CS dataset. On HW dataset, there is not much to differentiate between them. SURF gives an average \( C_a \) of 0.97 while 128D SIFT gives an average \( C_a \) of 0.96 for both datasets as shown in Table 5.9. Due to a difference in
performances, we performed one tailed paired t-test with $\alpha=0.05$. The C\textsubscript{a} of features from both datasets is used as data for the t-test. The data for each feature is limited to 6 observations from two datasets hence making the test underpowered. Our null hypothesis is: means of SIFT variants and SURF are equal. While our alternative hypothesis is: there is a significant difference between the means i.e. SURF is better than SIFT variants. The results in Table 5.10 show that $p$-values are greater than $\alpha$ between SURF and all SIFT feature variants, which indicates the rejection of the null hypothesis. Since the test is underpowered and the results are not significant, we can say on balance that SURF seems to perform better than 64D SIFT but comparable to other SIFT feature variants in noise experiments.

Table 5.9: Average correct acceptance rate (C\textsubscript{a}) across noisy images.

<table>
<thead>
<tr>
<th></th>
<th>SIFT64</th>
<th>SIFT96</th>
<th>SIFT128</th>
<th>SURF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.90</td>
<td>0.95</td>
<td>0.96</td>
<td>0.97</td>
</tr>
</tbody>
</table>

Table 5.10: SURF versus SIFT variants $p$-values for noise experiments.

<table>
<thead>
<tr>
<th></th>
<th>SIFT64</th>
<th>SIFT96</th>
<th>SIFT128</th>
</tr>
</thead>
<tbody>
<tr>
<td>SURF</td>
<td>0.06</td>
<td>0.313</td>
<td>0.336</td>
</tr>
</tbody>
</table>

- **Scale**: To test the scaling, we used the \texttt{imresize} function of MATLAB to change size of test images at different scales, which generated training images of different sizes. We have resized the test images at five different scales, such as 0.25, 0.5, 1.5, 2 and 3. The performance of SIFT variants and SURF on both datasets is shown in Figures 5.13 and 5.14.

Figure 5.13: SIFT versus SURF performance on scaled images across the CS dataset.

All features perform equally well and there is nothing much to differentiate between them. The results in Table 5.11 further support this hypothesis, which
shows that all features give a similar average $C_a$ across both datasets. Therefore, we can deduce that SIFT variants and SURF are scale invariant and perform equally well.

Table 5.11: Average correct acceptance rate ($C_a$) across scaled images.

<table>
<thead>
<tr>
<th>Feature</th>
<th>SIFT64</th>
<th>SIFT96</th>
<th>SIFT128</th>
<th>SURF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>0.96</td>
<td>0.97</td>
<td>0.97</td>
<td>0.97</td>
</tr>
</tbody>
</table>

5.6 Conclusion

The contribution in this chapter is the investigation of matching performance of different features across different image datasets. The main purpose is to identify the best among them and use that feature for visual BoW in iPoS. The results indicate that all SIFT features (excluding 32D SIFT) perform best compared to other features and so are used for the remainder of the work. SURF features are found to perform next best in the experiments.

An analysis of SIFT and SURF features under different image transformations shows that SIFT variants (excluding 32D SIFT) and SURF features perform equally well against different image transformations except for rotation and illumination, where SURF performs better. But the statistical power of the tests is not high and it will be worthwhile to use multiple indoor datasets in the future for a better statistical analysis. However, we managed to find some other comparison works, where SIFT and SURF have been compared more rigorously on a large number of images for image transformations (Bauer et al., 2007; Heinly et al., 2012; Juan and Gwon, 2009; Khan et al., 2011b; Wu et al., 2013). These works report that SIFT performs better than SURF for correct image matching. Moreover, most well known indoor image matching works have preferred SIFT features for indoor image matching in their works (Arth et al., 2009; Filliat, 2007; Kang et al., 2009; Ledwich and Williams, 2004; Li, 2006)
Therefore, we decided to use SIFT features for our work.

The proposed 96D SIFT and 64D SIFT features require less memory compared to 128D SIFT. 96D SIFT performs almost similar to 128D SIFT in all experiments compared to 64D SIFT which slightly underperforms. In our work, the correct image matching is the most important to avoid as many localisation errors as possible. Therefore, we have selected 96D SIFT features for visual BoW. However, 64D SIFT could be another good choice for other systems due to its lower memory requirements, reasonable performance and efficient image matching.

In the next chapter, we discuss visual BoW based on 96D SIFT features.
Chapter 6

Visual Bag of Words for localisation

Note: Some portions of this chapter are taken from (Khan et al., 2011a).

This chapter discusses the visual Bag of Words (BoW) scheme, which is the main building block of Indoor Positioning System (iPoS) and makes use of 96D SIFT features presented in the previous chapter. The visual BoW used for iPoS comprises of a voting module and a verification method to perform precise scene localisation from an indoor database of captured images. First, a small subset of images most similar to a query image is retrieved via an inverted index followed by ranking of retrieved images. The verification method is used for image matching only if the voting module fails to make a decision.

The contributions of this chapter are the evaluation of standard visual BoW with different ranking functions (weighting schemes), an analysis of the voting module and a comparison of hard assignment and soft assignment in standard visual BoW. The verification methods will be discussed in the next chapter.

6.1 Related work

Standard visual BoW was introduced for the first time in vision by Sivic and Zisserman (2003) for video retrieval. The system was able to localise all occurrences of a user outlined object in a video and was demonstrated on two full length feature films. Since its introduction, standard visual BoW has became very popular due to its efficiency and has been used in many image retrieval systems.

Standard visual BoW allows images to be compared efficiently by using a concise method to represent an image. Visual BoW applies clustering on features, which are extracted from all training images and this step is referred to as feature quantisation. The set of cluster centres is called a “visual vocabulary” and a cluster centre is called a “visual word”. Visual words are discrete features. These words may represent “trees”, “doors”, “tables” etc., and all descriptors arising from one of these features in the world should be mapped to the corresponding visual word. The idea behind the use of a visual vocabulary is that the same feature viewed in multiple images should map to the same visual word each time and different looking features should map to different visual
words. Each feature is extracted from a training image and is mapped to its nearest visual word, which results in vector representations for each training image, such as frequency or inverse frequency of visual words. These vector representations of training images are used during image matching. Therefore with visual BoW approach, training images are represented by a bag having visual words rather than features and images can be matched rapidly as comparing images consists of simply comparing their BoW representations, which are sparse vectors of each visual word frequency.

To classify a query image, its vector representation is first obtained from the visual vocabulary. The most similar images to a query image are first found by iterating through the BoW database. Then vectors of found training images are compared with a query image, which ranks the found images according to their similarity to the query image, such as the first top ranked image matches best with the query image, the second top ranked image matches next best with the query image and so on. The higher ranked images are most likely to show the same place as the query image and the top ranked image is considered the best match in standard visual BoW. The recognition accuracy of standard visual BoW is high despite geometric information being ignored, and results are robust to small changes in the scene. However, in complex environments the first top ranked image may refer to a wrong place and this often leads to wrong image matches. Therefore, standard visual BoW has been used in different studies with varying configurations to provide reliable image matching, such as use of geometric verification to re-rank the retrieved images (Philbin et al., 2008), use of spatial information to extract visual BoW representations (Lazebnik et al., 2006) or use of additional attributes, such as colour along with features (Botterill et al., 2008; Filliat, 2007).

A large visual vocabulary is required with a large number of features often in millions to get refined clusters, which improves the overall image retrieval performance of a system (Aly et al., 2011; Philbin et al., 2007). However, it takes a long time to perform clustering on very large numbers of features while building a big visual vocabulary and at the same time, the look-up time for visual words also increases while generating visual word distributions for images. Nister and Stewenius (2006) addressed this problem and developed a hierarchical clustering scheme to build an effective tree-based visual vocabulary for fast retrieval of relevant images from large datasets. The scoring function used by the authors leads to a significant improvement in image retrieval with respect to speed and matching accuracy. The authors report that their system matches a query image in about 0.027 seconds on average from the self-developed dataset of 10200 images and retrieves 3.29 correct matches out of 4 on average. The used database contains four different images of 2550 different objects or scenes. In another work, Botterill et al. (2008, 2010) used the same idea of hierarchical clustering for image matching but with composite features. The scheme uses SURF and colour information as a feature, where colour information is extracted from the region around each SURF keypoint. The system targets real-time detection of identical places for robots and is shown to correctly detect scene matches regularly on a video stream from a mobile robot. The localisation in an office indoor environment is not the focus of this work.

Standard visual BoW is an orderless representation of an image and spatial information is ignored while generating representations of images, which results in generation
of visual BoW representations from the whole image. In one study, Lazebnik et al. (2006) introduced the idea of a spatial pyramid in visual BoW to improve the recognition of scene categories. This method works by repeatedly subdividing an image and computing histograms of image features over resulting subregions. The histograms computed at different scales from the image are used for image matching. Therefore, the work uses spatial information by computing a bag of features from sub-regions of an image rather than the whole image. The system is reported to provide an image matching accuracy of 0.65 on the Caltech-101 dataset (Fei-Fei et al., 2004), which exceeds the highest image matching accuracy of 0.54 reported by Zhang et al. (2005) on the same dataset.

Most works have focused on the problem of outdoor image matching. Indoor image matching is challenging because indoor locations are visually similar. Filliat (2007) in his work addressed the problem of indoor image matching and presented a visual BoW system. The system uses additional attributes, such as colour and texture along with features from training images to perform image match. The system extracts SIFT features, colour and texture histograms from a query image and uses them to identify corresponding visual words from the visual vocabulary. These visual words vote for the rooms in which they are perceived at least once followed by computation of vote quality for a possible localisation decision. The system was tested in a non-office building with four different rooms where it gave 83% indoor image matching accuracy. The performance of the system may be affected in office buildings due to similar textures or colour schemes in many places.

In an office building, similar texture, colour or pattern is often followed in almost every part of the building, which makes image matching more challenging. In one study, Kang et al. (2009) presented a novel algorithm comprising two phases to perform reliable image matching in a large office environment with self-repeating structures. In the first phase, training images similar to a query image are retrieved via visual BoW based on the rankings of all images. The retrieved images are then used to generate new rankings in the local context during the second phase. This makes features more discriminating and provides good image matching performance on indoor images. For an input query image, eight of the most similar pre-captured images are retrieved and a potential localisation is suggested by the system if there is a cluster of pre-recorded images captured less than 3m away from each other among the retrieved images. The system was tested on an indoor dataset with two test cases and it improved the image matching accuracy from 0.97 to 0.98 and from 0.70 to 0.85 compared to image matching schemes which use only one stage.

In visual BoW, error is introduced when features from images are mapped to the nearest visual words. The mapping of features to visual words provides a very coarse approximation to the actual distance between the two features that is zero if assigned to the same visual word and infinite otherwise. In practice, such mapping leads to errors known as quantisation errors, which arise due to variability in the feature descriptors. This variability arises from image deformations and causes the same features to be assigned to different visual words, which decreases the image retrieval performance. Different ways have been proposed to reduce such quantisation errors in different studies: a feature is mapped to a visual word only if its distance to that visual word is not greater than a threshold; otherwise this feature is removed (Cai et al., 2012), visual
words are extracted from patches and are stored in a hierarchy based on contextual information (Bai et al., 2012) or features are mapped to multiple visual words while generating visual word distributions for images (Philbin et al., 2008). These systems are tested on urban datasets and are reported to provide good image matching performance compared to visual BoW systems, which suffer from quantisation errors.

Most of the visual BoW work focuses on outdoor image matching. Few works address the problem of indoor image matching and these indoor places were limited to a few indoor locations such as rooms, corridors etc. within a single building (Botterill et al., 2008; Filliat, 2007; Kang et al., 2009). Indoor positioning becomes hard in a large office environment due to the visually similar nature of many indoor locations. In this thesis, we propose a system which uses visual BoW, a voting module and a verification method to perform fast and precise indoor positioning based on image matching in large indoor environments of any type. Figure 6.1 shows an example of our proposed system in which a captured indoor image is compared with the stored database images via visual BoW to identify the best matched image. Each database image is annotated manually with location IDs and the corresponding location information of the best matched image indicates the current indoor place.

Figure 6.1: Scene localisation by the system. Colored circles indicate the identified indoor places against the corresponding input images.

The system presented in this thesis for indoor localisation can be viewed as an extension of that of Robertson and Cipolla (2004). They presented one of the first successful image matching systems for urban environments having building images. In their work, cameras are assumed calibrated and database images are assumed rectified. The features of a query image are identified using Harris Corner detector (Harris and
and a Random Sample Consensus (RANSAC) (Fischler and Bolles, 1982) based algorithm for image registration is applied to match a query image against each database image to obtain a nearby database view. Pose estimation is then performed between the nearby database view and a query image to return the location.

On one hand, we updated the approach of Robertson and Cipolla (2004) using Scale Invariant Feature Transform (SIFT) features and the visual BoW algorithm. We also realised that plane homographies can be used in many environments and not just with building facades as used by Robertson and Cipolla (2004). In this work, only coarse localisation is necessary; therefore we hypothesise that camera calibration is not required for localisation.

6.2 Proposed System

The proposed system is based on visual BoW and differs from the work of Robertson and Cipolla (2004) because it does not require camera calibration, which is ideal for a smartphone based localisation system. Otherwise, camera calibration is required to be performed before using the system on a mobile device. Kang et al. (2009) performed indoor image matching by using the distance information of captured training images and conducted experiments on images of one floor of a building. The authors assumed that database images were annotated with distance information. Our work also differs from the work of Kang et al. (2009) because it does not use distance information between training images and provides good indoor image matching accuracy up to 93% during experiments across four indoor datasets having images of different indoor locations such as halls, offices, coffee room etc. Only training images along with location information, such as IDs are needed to be stored, which reduces the amount of human efforts required to map an indoor environment compared to the work of Kang et al. (2009), where distance information needs to be determined between each pair of a large number of captured images hence making the mapping process difficult.

Standard visual BoW does not take into account the spatial configuration of features or other attributes (such as colour, geometry etc.) and this often leads to spurious matches. This happens frequently when testing and training images are taken at different times such as night, morning etc., and from different cameras. Nevertheless, the correct matching image is often in the top few candidate matches, and incorporating a voting module and verification method should significantly improve the performance.

In visual BoW, features are extracted and an inverted index is developed during the training phase. To classify a query image, visual BoW uses the inverted index to retrieve images most similar to a query photo and ranks them based on similarity. The voting module then plays its role and checks the top three ranked images. If the top three ranked images refer to the same place then the corresponding location is simply returned. Otherwise, verification is performed on the top 50 ranked images one by one to find the best match. If the verification method fails to find a match, the system returns a “no-location” message. In indoor environments, places are quite similar and chances of finding wrong image matches are high. So if the system cannot make a confident decision, then it is better not to make an image match because that image match may be wrong. Instead, the user can use another query photo of the
indoor location for localisation in such cases. The working of the system is stated in Algorithm 1.

**Algorithm 1** Proposed system.

**Input:** Captured image of the current scene (query photo).

**Output:** Location information of best matched image from the database.

1. **Offline**
2. Extract 96D SIFT features from training data (images from the collection).
3. Use approximate k-means clustering to quantise the features.
4. Generate the visual vocabulary.
5. Generate the histograms for all training images via vocabulary
6. Generate inverted index.

7. **Online**
8. For query photo, use the inverted index to retrieve top 200 similar images.
9. Compute the image rankings against the query image via histograms comparison.
10. Take the top 50 ranked images in increasing order.
11. **if** top 3 ranked images vote for the same location **then**
12. return that location *(Best Case)* .
13. **else**
14. Perform verification on top 50 ranked images one by one *(Worst Case).*
15. If any image matches with query; return location.
16. if no match found in the 50 images; this means “no-location” i.e. no match found.
17. **end if**

The voting module results in significant speed up as the correct match for most of the query images is determined without any verification, which is an expensive operation. We have experimented with different number of images for the voting module and experimental results in section 6.4.2 show that the top three images configuration works best because more localisation decisions are made without the use of a verification method while most of these decisions are correct. Therefore, we have used three images configuration for our voting module.

On the other hand, any number of images can be used for verification. We have used 50 top ranked images for verification due to following reasons:

1. We realised from experiments that the verification method always found a match for a query image within the first top 20 ranked images across the CS Indoor dataset. Otherwise, it did not find a match and rejected a query image. But this may differ across another especially large indoor dataset having thousands of images, which prompted us to include more images to search for an image match, such as top 50 ranked images.

2. The chances of getting similar images against a query image get decreased with a decrease in image ranks hence reducing the likelihood of matching a query image. It is better to search among a specific number of top ranked images and then acquire another query image if a match cannot be made. Because
searching among all top ranked images is time consuming as the verification method performs an expensive operation to compare two images. Therefore, the key is to have a reasonable rejection time with a verification method as some time is also required to generate image rankings, such as 2-5 seconds depending upon the vocabulary size. With top 50 images, our homography and fundamental matrix verification methods took about 4 and 8.75 seconds on average to reject a match for a query image (timings presented in the next chapter), which is reasonable for our smartphone based localisation system.

The key phases of the visual BoW are discussed as follows:

### 6.2.1 Extraction of features

Visual BoW is inspired by the bag of words used for document classification in natural language processing and information retrieval (Manning et al., 2008). In bag of words, each document is represented as a vector which is an unordered collection of words disregarding the grammar and even word order. The vector representations of documents are compared during document classification. But there are no text documents in computer vision and there are only images, which carry the visual information of a scene. Sivic and Zisserman (2003) proposed to extract features from images and use these features as visual words for the first time.

Features from training images can be extracted from invariant regions, which are first obtained using region detector algorithms, such as Laplacian based (Linderberg, 1998), Maximal Stable Extreme Region (Matas et al., 2002), Harris affine (Mikolajczyk and Schmid, 2004). Then, an invariant descriptor is built for each region using a feature algorithm, such as SIFT (Lowe, 2004), PCA-SIFT (Ke and Sukthankar, 2004), SURF (Bay et al., 2008) or FREAK (Alahi et al., 2012) and resulting descriptions are used during image matching. Alternatively, features can be directly extracted from the whole training image rather than small image patches and can be used for image matching. The region based feature extraction approach is suitable for very large image databases to speed up the image matching process. Since, standard visual BoW is built on features, the choice of features dictates the retrieval performance for visual BoW. We have used 96D SIFT features (presented in the last chapter) from the whole image here.

### 6.2.2 Vocabulary building

This phase reduces the training feature space via generation of a visual vocabulary consisting of visual words. The visual vocabulary is built by applying clustering on the extracted features (also known as feature quantisation) as illustrated in Figure 6.2. The resulting cluster centres are called visual words and represent the prototype patterns which are made up by clustering local features patterns.

Once the visual vocabulary is developed, the features of training images are mapped to the closest visual words in the visual vocabulary to obtain the visual word distributions for each training image. These distributions are obtained using either hard or soft assignment. In hard assignment, training features are mapped to one closest
visual word while soft assignment maps training features to multiple nearest visual words. Let’s say, we have a set of $N$ features and $V$ visual words. The hard assignment algorithm tries to find an optimal set $O$ having $V$ visual words, which minimises the following objective function:

$$H(O) = \sum_{i=1}^{V} \sum_{j=1}^{N} \| x_j^{(i)} - v_i \|$$  \hspace{2cm} (6.1)$$

where $x_j$ represents $j^{th}$ feature, $v_i$ represents the center of $i^{th}$ visual word. The above equation assigns a feature to the single nearest visual word without considering the other most nearest visual words. For soft assignment, the above equation becomes:

$$S(O) = \sum_{i=1}^{V} \sum_{j=1}^{N} \sum_{k=1}^{P} \| x_j^{(i)} - v_i \|$$  \hspace{2cm} (6.2)$$

where $P$ refers to the number of nearest visual words to which every feature descriptor is assigned. Therefore, the use of soft assignment increases the computational cost compared to hard assignment. In practice hard assignment may lead to errors because of variability in the feature descriptor such as image noise, varying scene illumination etc. This may result in the same surface patch being assigned to different visual words in different images. Soft assignment is often used to avoid this problem (Philbin et al., 2008). In Section 6.4.4, we have compared hard and soft assignment with standard visual BoW on the indoor dataset and the results indicate that soft assignment results in only a slight improvement of about 1% in retrieval of correct images than hard assignment but at the expense of computational cost. Therefore, the proposed visual BoW uses hard assignment.

### Clustering methods

Researchers use different clustering methods in their work during visual vocabulary building, such as k-mediods (Botterill et al., 2008), approximate k-means (AKM) (Philbin et al., 2008; Sivic and Zisserman, 2003) and hierarchical clustering (Filliat, 2007; Kang et al., 2009; Nister and Stewenius, 2006; Philbin et al., 2008). AKM is reported
to be superior to hierarchical clustering in terms of image retrieval but is less efficient (Philbin et al., 2008). We have used approximate k-means clustering (AKM) to build the visual vocabulary in this work due to its good image retrieval performance.

**Number of clusters**

The number of chosen clusters during clustering dictates the size of the visual vocabulary and plays an important role in retrieval performance. A large number of clusters results in a big visual vocabulary, which leads to more time to generate the visual word distributions for training and query images, and also to compare their representations because the size of representations (i.e. histograms) is directly proportional to the vocabulary size.

The size of a visual vocabulary varies from application to application. Normally, a small visual vocabulary does not result in refined groupings of the local features and the resulting visual words are not discriminating, which leads to poor image matching performance. On the other hand, the grouping of features is more refined with a large visual vocabulary which results in better image matching performance (Jiang et al., 2007). Researchers have used different vocabulary sizes but the trade off between discrimination and generalisation motivates the use of an appropriate dictionary size. We used 7 vocabulary sizes ranging from small to large (1K to 50K) in the experiments presented here for a thorough analysis in order to find a suitable one for visual BoW.

### 6.2.3 Inverted index

In information retrieval, an inverted index is a data structure which stores the mapping from visual words to their locations in documents. It offers a quick retrieval of relevant data from large amounts of data and is commonly used in search engines (Baeza-Yates and Ribeiro-Neto, 1999). For example, suppose someone types a keyword in a search engine. The search engine uses an inverted index to quickly identify all those documents on the web, which have that particular search keyword. All results are then ranked on the basis of similarity and are displayed to the user.

Visual BoW differs from other approaches in that it performs image matching via an inverted index. The inverted index in visual BoW stores the mapping of visual words against the corresponding images in which these visual words are found. It offers quick retrieval of relevant images against a query image followed by a best match computation. A sample inverted index is shown in Figure 6.3 which illustrates that each visual word entry in the index can be viewed as a linked list which has nodes equal to the number of images having that visual word. An inverted index is developed and stored for later use during image classification.

### 6.2.4 Query image classification

To classify a query image, the system first obtains its visual word distribution from the visual vocabulary. The inverted index is then used to retrieve 200 training images having the most similar visual words to the query image. To rank the retrieved images, the system represents a query image and all retrieved images via histograms, with
appropriate weighting scheme such as frequency or inverse frequency of visual words. The query image histogram is compared with the histograms of retrieved images to obtain a ranked list of potential candidate images. The first top ranked image is the closest in similarity to a query image, second top ranked image is the second most similar training image to a query image and so on. The top ranked images are passed to the voting module and verification method for a localisation decision. However in standard visual BoW, the first top ranked image is picked as the best match.

The histograms are compared using the $\chi^2$ distance which can be computed between the two histograms, $(H1)$ and $(H2)$, as follows (Filliat, 2007):

$$\chi^2(H1, H2) = \sum \frac{(H_{1,i} - H_{2,i})^2}{H_{1,i} + H_{2,i}}$$  \hspace{1cm} (6.3)

### 6.2.5 Weighting scheme

Reliable image rankings depend on effective histograms generated from the visual word distributions of query and candidate images. A weighting scheme is normally used to generate these histograms from the images as shown in Figure 6.4, where generated histograms carry the frequency count of visual words in an image. The choice of weighting scheme varies from application to application. We have used three weighting schemes:

- **Normalised term frequency (ntf):** In term frequency ($tf$), histogram bins refer to the actual count of visual words in an image. We use a $M$ dimensional bin histogram $T_d = [T_{d1}, T_{d2}, \ldots, T_{dM}]$, where each histogram entry ($T_{di}$) is the frequency count of visual words in an image $d$.

Normalisation is an important factor in achieving better performance as it eliminates the difference between images of different sizes with almost similar visual
word distributions. In normalised term frequency (ntf) scheme, each normalised histogram bin is calculated as follows (Jiang et al., 2007):

$$T_{di} = \frac{n_{di}}{n_d}$$

(6.4)

where $n_d$ is the number of visual words in image $d$. This scheme works well with more clusters i.e. a large visual vocabulary.

- **Normalised term frequency inverse document frequency (tfidf):** In term frequency inverse document frequency (tfidf), We use a $M$ dimensional bin histogram $T_d = [T_{d1}, T_{d2}, \ldots T_{dM}]$, where each histogram entry ($T_{di}$) is computed as follows (Jiang et al., 2007):

$$T_{di} = n_{di} \cdot \log \frac{N}{n_i}$$

(6.5)

where $N$ is the total number of images and $n_i$ is the number of images that have visual word $i$. The idea of this scheme is to down weight the visual words which appear in many images and give more weight to those words which appear in few images. Therefore, unique visual words are given a higher weight in order to increase classification performance. $ntfidf$ is computed as follows:

$$T_{di} = \frac{n_{di}}{n_d} \cdot \log \frac{N}{n_i}$$

(6.6)

This scheme also works well with more clusters. It is the most commonly used scheme in computer vision and is reported to outperform the ntf scheme in different vision works (Filliat, 2007; Kang et al., 2009; Nister and Stewenius, 2006; Sivic and Zisserman, 2003).
BM25: BM25 is the best known probabilistic weighting scheme in information retrieval (Jones et al., 2000) and is commonly used for document retrieval due to excellent results (Manning et al., 2008; Robertson et al., 1994). It ranks a set of documents based on the query terms appearing in each document, regardless of the inter-relationship between the query terms within a document. It is a whole family of scoring functions, with slightly different components and parameters.

Let’s say we are given a query image $Q$, containing visual words $q_i, ..., q_m$. With a $M$ dimensional bin histogram $T_d = [T_{d1}, T_{d2}, ..., T_{dM}]$, each histogram entry ($T_{di}$) is computed between $Q$ and a trained image $D$ as follows:

$$T_{di} = \sum_{i=m}^{n} \frac{tfidf(q_i)}{f(q_i, D) \cdot (k_{1+1})} \frac{f(q_i, D) \cdot (k_{1+1})}{f(q_i, D) + k_{1} \cdot (1 - b + b \cdot \frac{|D|}{avgdl})}$$

where $f(q_i, D)$ is $q_i$’s term frequency in the image $D$, $|D|$ is the length of the image $D$ in visual words and $avgdl$ is the average image length in the image collection from which images are drawn. $k_1$ and $b$ are free parameters usually chosen in the absence of an advanced optimisation as $k_1 = 2$ and $b = 0.75$. $tfidf$ refers to term frequency inverse document frequency.

6.3 Datasets and performance metrics

We used three datasets in the course of the experiments for evaluation. In each dataset, training images are different from test images. For every test image of a scene, there are some training images of the same scene taken from different viewpoints. These datasets are briefly discussed below. For more details about these datasets, please refer to Chapter 4.

1. **David Nister (DN):** We used images of the first 4000 objects i.e. 3000 for training and 1000 for testing. The first image of every object was used for testing and the remaining object images were selected for training.

2. **Hongwen (HW):** We used the first 3000 images for training and the first 100 images for testing respectively.

3. **Computer Science (CS):** We used all images from this dataset, 70 images for testing and 630 images for training.

The following evaluation metrics are used in this chapter:

$C_a$: refers to the correct acceptance rate.

The expected value for this metric ranges from 0-1. A higher value is better for this metric, which indicates a high image matching accuracy against query images. This metric illustrates the success of a visual BoW system based on a weighting scheme for image matching.
\( W_c \): refers to the consistency of weighting scheme.

The expected value for this metric ranges from 0-1. A higher value is better for this metric and illustrates the good performance of a weighting scheme used with visual BoW system, which indicates that the voting module matches many query images without using the verification method. Therefore, a high value of this metric shows that a weighting scheme produces good image rankings and the top three ranked images give the same location most of the time, which results in few calls to the verification method.

\( N_c \): refers to the matching accuracy of the voting module.

The expected value for this metric ranges from 0-1. A higher value is better for this metric, which indicates the accuracy of image match decisions made by the voting module against query images.

This metric is used in conjunction with \( W_c \). Therefore, high values for both \( W_c \) and \( N_c \) metrics indicate a good performance of a weighting scheme used with visual BoW system for retrieving of relevant images against query images.

For more details about these metrics, please refer to Chapter 4.

6.4 Results

In this section, we compare different weighting schemes with standard visual BoW, analyse the performance of the voting module and compare hard with soft assignment and report the results.

6.4.1 Comparison of weighting schemes

We evaluated the matching performance (\( C_a \)) of the visual BoW system without the voting module and verification method across all datasets. The top ranked image is considered the best match for a query image in the experiments and the resulting visual BoW behaves like a standard visual BoW without voting and verification. The results in Figure 6.5 indicate that the \( \text{ntfidf} \) weighting scheme underperforms compared to the others. The \( \text{ntf} \) and the \( \text{BM25} \) weighting schemes both do well and there is not much to differentiate between them. However, the \( \text{ntf} \) scheme is simple to compute compared to the \( \text{BM25} \) scheme, which is complex and requires more computation time. The main purpose of this comparison was to find out the scheme which gives the correct top ranked image most of the time because top ranked image is the most important in localisation. That’s why we have only used \( C_a \) metric to get the required information in this experiment.

6.4.2 Voting module analysis

We have experimented with different number of ranked images for the voting module, such as 1, 2, or 7 and the results are reported across the CS dataset as shown in Figure 6.6. In the experiment, the voting module only makes a match if all used images refer
to the same location during voting. Otherwise, the decision is not made for a query image. The results show that the number of localisation decisions goes down but there are fewer wrong matches. With $n=1$, the voting module produces 20% wrong image matches because no query image is rejected at all. The performance gets better with the use of two images for voting module due to the decrease in the number of localisation decisions. With $n=3$, the voting module makes about 42% localisation decisions with an accuracy of 98%. With $n=5$ or more, the accuracy of the voting module becomes 100% but the number of localisation decisions go below 20%. A reasonable trade-off is required between accuracy and the number of localisation decisions in order to make many correct localisation decisions without invoking the verification method. The results indicate that using three images with the voting module offers the best trade-off and is therefore used in our work.

Figure 6.5: Correct acceptance rates $C_a$ for ntf, ntfidf and BM25 schemes on all datasets using standard visual BoW.

Figure 6.6: Voting module performance analysis with different number of images across the CS dataset.
6.4.3 Weighting scheme analysis

An analysis is then performed to identify the performance of the voting module (with top three images) across the CS indoor dataset with different weighting schemes. A good weighting scheme should generate good image rankings and should result in few calls to the verification method i.e. allowing the voting module to make the most of the localisation decisions. The results in Figure 6.7a show that the ntfidf scheme performs poorly compared to others because it generates poor rankings for query images and results in more calls to verification methods. Each weighting scheme’s performance (Wc) gets better with an increase in the number of clusters because more clusters result in better grouping of local features, which improves the overall image retrieval performance; hence leading to fewer calls to verification methods. Figure 6.7b shows that the voting module matching accuracy (Nc) is equally good for all schemes. The results in both Figures 6.7a and 6.7b show that overall ntf and BM25 schemes perform better. We have used ntf scheme for remaining experiments with the visual BoW because this scheme is more efficient than BM25.

![Weighting scheme consistency and accuracy](image_url)

(a) Weighting scheme consistency. (b) Weighting scheme accuracy.

Figure 6.7: Weighting scheme consistency and accuracy for ntf, ntfidf and BM25 schemes.

The voting module also finds wrong image matches and such wrong matches cannot be avoided. However, the number of wrong matches is very small as shown in Figure 6.7b. Some correct and wrong examples of image matching on the CS dataset by the voting module are shown in Figures 6.8 and 6.9.

6.4.4 Soft versus hard assignment

The CS dataset is a complex one and contains multiple images of every place. The chance of variability in feature descriptors is therefore high. Therefore, we used the CS
dataset to evaluate the performance of soft assignment in standard visual BoW against hard assignment in this subsection.

We performed a soft assignment on the CS dataset by mapping the query or training image features to 2, 3, and 5 nearest visual words respectively resulting in three different soft assignment based configurations i.e. s2, s3 and s5. The correct acceptance rate \( C_a \) was computed for standard visual BoW based on s2, s3 and s5 and the results were compared with standard visual BoW based on hard assignment (h-BoW) as shown in Figure 6.10.

The results in Figure 6.10 indicate that soft assignment does not make a substantial improvement in \( C_a \) with the \( ntf \) scheme. Soft assignment increases the computational cost because features are mapped to more than one visual word. This indicates that soft assignment can be used but on an ad-hoc basis and at the expense of computation. Therefore, we have used hard assignment with visual BoW. However, it is worthwhile to do more experiments in the future, such as varying the number of nearest visual words during mapping of features for more thorough evaluation of soft and hard assignments.
Figure 6.9: Wrong matches by the voting module \((ntf\) scheme) on CS dataset.

Figure 6.10: Soft vs Hard assignment in visual BoW with the \(ntf\) scheme on CS dataset.
6.5 Conclusion

We started with the performance review of standard visual BoW with the three weighting schemes. Surprisingly, we found that the simplest scheme, $ntf$, was as good as the more sophisticated $BM25$ scheme. Both $ntf$ and $BM25$ schemes did well which is interesting. $BM25$ scheme is the standard baseline comparison for document-retrieval and to our knowledge has not been previously used in any vision work. On the other hand $ntfidf$ which is the default weighting scheme for visual BoW underperforms on all datasets compared to the other weighting schemes. It is unclear why $ntf$ performs so well in these experiments, but we suspect it is related to the specific nature of the localisation problem with many images for a relatively small number of locations (numbering in the tens rather than thousands). It will be interesting to repeat these experiments on very large scale databases in future. We hypothesise that on balance $ntf$ is the best scheme due to its efficiency and good image retrieval performance and is therefore used with visual BoW in this thesis.

The performance analysis of the voting module shows that it provides good precision and localisation decision is made by the voting module itself most of the time without the need for an expensive verification operation, which makes the proposed visual BoW system efficient during indoor image matching.

The contribution of this chapter is the presentation of an image matching system based on visual BoW, a voting module and a final verification method. Such a tiered approach is necessary when there are several visually similar locations in the image database and it allows the system to make a precise localisation decision based on accurate indoor image matching with only a single query image without requiring additional information, such as camera calibration, distance information etc. The proposed system offers a reasonable trade-off between image matching accuracy and efficiency.

We discuss the verification methods in the next chapter.
Chapter 7

Verification methods for Visual Bag of Words

Note: Some portions of this chapter are taken from (Khan and McCane, 2013).

We presented the visual Bag of Words (BoW) system in the previous chapter. It comprises a voting module and a verification method to provide fast and accurate image matching to obtain high success rate during indoor localisation. The verification method is used to verify ranked images if the voting module fails to make a localisation decision.

The problem with standard visual BoW is that it sometimes identifies a wrong top ranked image for a query image as shown in Figure 7.1. In such cases, the top three ranked images often refer to different locations and the voting module fails to make a localisation decision. It then calls the verification method to make a decision. These situations often arise when visual BoW generates poor rankings because a query image is dark, blurred or is not very discriminating. A pre-processing involving filtering of images can be performed to minimise image deformations up to some extent but nothing can be done with less discriminating images. Therefore, the main motivation behind the use of a verification method is to find a correct match for a query image when the top ranked images refer to different locations. In a localisation system, wrong location results are not desirable and a verification method capable of reducing wrong image matches is required. Hoover, we discuss verification methods in this chapter.

The main contribution of this chapter is the analysis of various verification methods to identify the best among them and use it with visual BoW.

7.1 Verification methods

Verification can be performed on every candidate ranked image retrieved by visual BoW, but verification can be an expensive operation. Hence, the proposed system only performs verification if the voting module fails to identify a consistent match in the top three ranked images. Verification is performed on the top 50 ranked images one by one. Once the match is found, the verification method stops searching regardless
Figure 7.1: An example showing the top ranked image retrieved against a query image by standard visual BoW.

how many ranked images are left out of 50 and returns the location information. In the case of no match, the verification method reports “no-location”. In such cases, another query photo of the scene can be acquired as this is better than giving a wrong answer. Therefore, the use of a voting module and a verification method offers a reasonable trade-off between efficiency and accuracy.

7.1.1 SIFT distance based (SD)

This verification method is based on SIFT feature correspondences between a query and candidate image. A feature correspondence indicates two matched features between images and is represented by a Euclidean distance value. A small distance indicates a high similarity between features and vice versa. The verification method selects those matched features between a query and a candidate image whose distances values are less than 150, an empirically determined value. The candidate image is picked as the best match if the verification method finds at least three feature correspondences against a query image. Otherwise, it moves to the next candidate image to perform verification. The working of the verification method is stated in Algorithm 2.

Algorithm 2 SIFT distance based verification method.

| Input: The query image and the potential candidate image |
| Output: true (location) or false (image not matched) |
| 1: do |
| 2: Compute SIFT correspondences between the query and candidate image using 150 distance threshold. |
| 3: if correspondences $\geq 3$ then |
| 4: return the location of the matched image; |
| 5: else |
| 6: candidate image not matched, so return false and pick next ranked image; |
| 7: end if |
| 8: Note: 150 distance threshold is kept same in all verification methods to compute SIFT correspondences. |

As discussed in Chapter 4, the distance threshold of 170 worked well during matching of 128D SIFT features and resulted in a good image matching performance. A slightly reduced value, such as 160 worked slightly better with 96D SIFT features. We
have used both feature variants in coming chapters. Therefore, we decided to use one threshold i.e. 170 for both SIFT variants because this threshold did well during feature matching for both of them.

In the verification method, we are concerned that a query image should match with the candidate image (retrieved via visual BoW) of the corresponding indoor location. Let’s say, if query image belongs to kitchen then it should match with one of the candidate images of kitchen. In other words, the focus is scene matching rather than image matching. We performed an experiment to analyse the performance of the distance threshold in such a scenario. In the experiment, we compared five query images with the 50 images retrieved via visual BoW against three different thresholds to determine the average number of correct and wrong features matches. The correct feature matches are legitimate feature matches i.e. between a query and a retrieved image of the same indoor location. While wrong feature matches are between a query and a retrieved image of different indoor location. Ideally, there should be very few wrong and many correct feature matches for a given distance threshold. The experimental results in Table 7.1 show the average correct and wrong features matches with the default threshold of 170 and two other thresholds. The number of wrong feature matches dropped with the 150 threshold compared to 170 but with fewer correct feature matches. The threshold of 130 gave few wrong feature matches but the number of correct feature matches reduced to 5.4. In most cases, the verification method failed to match a query image with retrieved images of the same indoor location because the number of correct feature matches was very low, such as 0-2. This prompted us to use 150 distance threshold with our verification methods. However, our experiments were limited to only three thresholds and it would be worthwhile to experiment with other thresholds, such as 135, 140, 145 or 165 in the future.

Table 7.1: Average correct and wrong feature matches between query and retrieved images.

<table>
<thead>
<tr>
<th></th>
<th>130</th>
<th>150</th>
<th>170</th>
</tr>
</thead>
<tbody>
<tr>
<td>Correct feature matches</td>
<td>5.4</td>
<td>14.5</td>
<td>20.6</td>
</tr>
<tr>
<td>Wrong feature matches</td>
<td>5.0</td>
<td>7.1</td>
<td>9.0</td>
</tr>
</tbody>
</table>

7.1.2 SIFT+ global hue based (SH)

Filliat (2007) and Botterill et al. (2008) both used hue information along with SIFT and SURF features in their visual BoW and claimed a better classification performance with the incorporation of colour information. In an office building, many of the places especially corridors are visually similar. The chances of finding wrong SIFT feature correspondences between the images belonging to two different indoor locations are quite high. Figure 7.2 shows two different corridor images, which are wrongly matched by SD verification method. It can be observed from Figures 7.2b and 7.2d that the colour (hue) information is different for both images. The colour information therefore can be used as an additional attribute to increase the likelihood of rejecting such wrong image matches.
This verification method computes the hue histogram in a similar way to Filliat (2007). The image is first divided into 16x16 windows and hue values from these windows are used to populate the 180 bin hue histogram, which represents the colour information of the whole image. We used the built in function of openCV (Bradski and Kaehler, 2008) to get the hue information from an image, which ranged from 0 to 180. The computed hue histogram of an image is then normalised so that the sum of all bins is equal to 1. The verification method then compares hue histograms of query and candidate image using the $\chi^2$ measure. If two histograms are identical then the distance between them will be 0. Otherwise, it will be 1 for totally different histograms. The candidate image is selected as the best match by the verification method if it has at least three feature correspondences against a query image and the difference between the hue histograms of two images is less than or equal to a histogram threshold.

We performed an experiment to determine a suitable value for the histogram threshold. In the experiment, the proposed visual BoW system with the current verification method was used for image matching on the images of the Computer Science (CS) dataset. The main aim was to analyse the correct, wrong and no-image matches for the current verification method with different values of histogram threshold for query images (not matched by the voting module). A good histogram threshold should result in more correct and few wrong image matches along with few no-image matches. The results in Table 7.2 indicate that a large number of query images were not matched for low values of histogram threshold, such as 0.15 and 0.25. Despite candidate and query
images belonging to the same location most of the time, there exists a large difference between their hue histograms because query and candidate images may have different illuminations, scales, or viewpoints. This results in large differences between the hue histograms hence rejecting many legitimates matches based on SIFT feature correspondences between query and candidate images. On the other hand, a large value i.e. 0.7 significantly reduced the number of no-image matches and slightly increased the number of correct matches than 0.5 value. However, the number of wrong matches also increased because chances of wrong image matches equally increase with more decisions. The value of 0.5 seemed to be a better choice for histogram comparisons than other values due to a reasonable performance across the query images as shown in Table 7.2. Therefore, we have used the value of 0.5 for histogram threshold to compare histograms in the current and all verification methods.

Table 7.2: Performance analysis of SH with different values of histogram threshold across the query images of the CS dataset.

<table>
<thead>
<tr>
<th>Histogram threshold</th>
<th>Correct matches</th>
<th>Wrong matches</th>
<th>No-image matches</th>
</tr>
</thead>
<tbody>
<tr>
<td>0.15</td>
<td>18%</td>
<td>5%</td>
<td>77%</td>
</tr>
<tr>
<td>0.25</td>
<td>37%</td>
<td>24%</td>
<td>39%</td>
</tr>
<tr>
<td>0.50</td>
<td>63%</td>
<td>24%</td>
<td>13%</td>
</tr>
<tr>
<td>0.70</td>
<td>65%</td>
<td>32%</td>
<td>3%</td>
</tr>
</tbody>
</table>

The details of the verification method are given in Algorithm 3.

Algorithm 3 sift + global hue based verification method.

Input: The query image and the potential candidate image
Output: true (location) or false (image not matched)
1: do
2: Compute SIFT correspondences between the query and candidate image.
3: if correspondences >= 3 then
4: Compute hue histograms for query and candidate image.
5: if $\chi^2(\text{Hue}(\text{QUERY}), \text{Hue}(\text{CANDIDATE})) \leq 0.5$ then
6: return location of the matched image;
7: end if
8: else
9: candidate image not matched, so return false and pick next ranked image;
10: end if
11: Note: For an image of size $m$ by $n$, the time to compute the hue histogram will be $O(mn)$ additional to other operations.

7.1.3 SIFT+ local hue based (SSH)

The previous method uses the colour information from the whole image. In an office building, a similar colour scheme is often used in many places e.g. floor tiles have the same colour or pattern, corridor interior walls have the same pattern etc. In such
environments, the colour information from the whole image is not very discriminating and the chances of rejecting wrong image matches are low as shown in Figure 7.3, where $SH$ has wrongly matched a ranked image with a query image.

![Image](image.png)

(a) Query Image.  
(b) Best matched image.

Figure 7.3: Wrong image match found by the verification method $SH$.

This verification method computes the hue histogram from keypoint regions of the image rather than the whole image. This method is inspired by the work of Botterill et al. (2008, 2010). This verification method works in a similar way to Algorithm 3 with the only exception that hue histograms are computed in a different way. The hue information is extracted from the 5x5 regions around every key point for both query and candidate ranked images. On average, an image of 640x480 resolution generates 250 features. The use of 5x5 region around each keypoint gives colour information from 6250 pixels of an image, which is used to populate the 180 bin hue histogram for the whole image. We chose 5x5 region arbitrarily for our verification methods because we hoped that this region would be good enough to extract the desired information. However, an interesting experiment will be to vary the size of regions around keypoints and analyse the performance across different types of indoor datasets to determine an optimal region size in the future.

$SSH$ is more efficient than $SH$. For an image of size $m$ by $n$ (640x480), the time to compute the hue histogram will be $O(pq)$ which is less than $SH$ i.e. $O(mn)$:

- where $p$ is the number of detected features (about 250 on average).
- where $q$ is 25 (i.e. 25 pixels around the key point).
- $pq$ will be much smaller than $mn$.

### 7.1.4 Local binary pattern based (LBP)

Local binary patterns (LBP) offer good texture classification. It has proven to be highly discriminative and its key advantages, namely its invariance to monotonic gray level changes and computational efficiency, make it suitable for demanding image analysis tasks (Ojala et al., 2002).

This verification method computes LBP histograms from the keypoint regions of the query and candidate images in the same way as done by Zhang and Mayo (2010). The
authors proposed three novel techniques to capture more refined spatial information between visual words in visual BoW and one of the techniques used LBP to obtain the shapes of visual words. The authors reported a performance gain with the proposed techniques over the spatial pyramid representation of standard visual BoW (Lazebnik et al., 2006).

To compute LBP histogram, this method selects a 5x5 region around each key point and moves a 3x3 window to capture possible pattern information as shown in Figure 7.4. In each 3x3 window, the neighborhood values are compared with the centre pixel to determine a pattern. The values greater than the centre pixel are given a value of one and the others are given a value of zero. This leads to a 8-bit binary pattern from each 3x3 window, which is then converted into a 8-bit number. Each 5x5 region is covered by nine 3x3 windows which results in nine 8-bit numbers which are used to populate the 256 bin histogram for the whole image which is called the LBP histogram. On average, each training image (640x480 resolution) generates about 250 features, which results in about 2300 8-bit numbers for each image. The method then uses a $\chi^2$ measure to compare LBP histograms for image matching as shown in Algorithm 4. The candidate image is selected as the best match if it has at least three feature correspondences against a query image and the difference between the LBP histograms of both images is less than or equal to 0.5.

**Algorithm 4** local binary pattern based verification method.

**Input:** The query image and the potential candidate image  
**Output:** true (location) or false (image not matched)

1: do  
2: Compute SIFT correspondences between the query and candidate image.  
3: if correspondences $\geq$ 3 then  
4: Compute LBP histograms for query and candidate image.  
5: if $\chi^2(LBP_{QUERY}, LBP_{CANDIDATE}) \leq 0.5$ then  
6: return location of the matched image;  
7: end if  
8: else  
9: candidate image not matched, so return false and pick next ranked image;  
10: end if

7.1.5 Homography based (HM)

Planar homography or simply homography is a relationship between image points on planes and is independent of the scene structure. Image points on a plane in one view are related to corresponding image points in another view by a homography. If $x$ and $x'$ are points in two images, a homography ($H$) relates the pixel co-ordinates in two images by the following equation:

$$x' = Hx$$  

(7.1)

Here $x$ and $x'$ are represented as homogeneous coordinates, that is 2D points in image plane are represented as $(x,y,w)$ and the corresponding Cartesian coordinates
are \((x/w, y/w)\). The advantage of using homogeneous representation is that it allows common operations such as translation, rotation or scaling as matrix operations. For each pairs of corresponding points, equation 7.1 gives two independent linear equations (often called constraints) such as:

\[ x^T H x = 0 \]  

(7.2)

Since \(H\) has 8 degrees of freedom (DOF), four pair of corresponding points in two images are required to determine \(H\). The concept of homography has been used to filter out wrong image matches (Li et al., 2008; Vincent and Laganiere, 2001; Yukhuu and Hwang, 2009). The homography verification method uses homography as an approximation to match two images. The justification for this algorithm is that although a homography is not expected to work for all feature correspondences, it should work for several correspondences especially indoors. The method first identifies the 10 best SIFT feature correspondences between a query and the candidate image. Every correspondence is in fact a distance value between the two features. The use of 10 correspondences with minimum distances results in picking the best matched features between the two images. Moreover, the chances of picking the wrong feature matches also gets decreased with this configuration. Two sample images (i.e. query and candidate) with the 10 best SIFT correspondences are shown in Figure 7.5.

The method applies RANSAC (Fischler and Bolles, 1982) to select four of these ten correspondences randomly for 15 iterations and makes sure that the same combination is not picked twice. A total of 15 iterations is found good enough in the experiments to estimate different possible homographies between two images.

Let \((u_k, v_k)\) and \((x_k, y_k)\) be four corresponding points or features in query and candidate image for \(k=0, 1, 2, 3\) respectively in one iteration, then selected pairs yielding the following 8 x 8 system of linear equations:

Figure 7.4: Generating a 8-bit number from a 3x3 window.
where $H = [h_{11}, h_{21}, h_{31}, h_{12}, h_{22}, h_{32}, h_{13}, h_{23}]^T$ are unknown coefficients and $X = \begin{bmatrix} x_0, x_1, x_2, x_3, y_0, y_1, y_2, y_3 \end{bmatrix}$ are known coordinates in the candidate image. The verification method determines the eight unknown coefficients by solving the linear system and obtains the homography matrix (Wollberg, 1994). With the homography matrix, all candidate features are transformed to new locations as shown in Figure 7.6. If both images are of the same scene, then most transformed features will be approximately at the same location as the corresponding query features. But sometimes feature correspondences are wrongly computed or the resulting homography matrix is not good. So, a number of transformed features may be computed at the same location as the query features even when images are of different places. Therefore, the method first identifies the query and transformed features coming at the same location after applying homography and then performs a feature similarity check (using Euclidean distance threshold) to determine the correct feature correspondences between query and candidate images, which are called perspective correspondences.

In the next RANSAC iteration, the verification method again picks four other points to compute a new homography and calculates perspective correspondences. The candidate image is selected as the best match if a sufficient number of perspective correspondences is found with different homographies. The working of the homography verification method is stated in Algorithm 5.
7.1.6 Fundamental matrix based (FM)

The fundamental matrix is another alternative to estimate the similarity between two images because it encapsulates the epipolar geometry which is the intrinsic projective geometry between images. If \( x \) and \( x' \) are corresponding points or features in two images, the fundamental matrix \( (F) \) relates the points by the following equation:

\[
x'^T F x = 0 \quad (7.4)
\]

\[
F = \begin{bmatrix}
f_{11} & f_{12} & f_{13} \\
f_{21} & f_{22} & f_{23} \\
f_{31} & f_{32} & f_{33}
\end{bmatrix} \quad (7.5)
\]

\( F \) has seven degrees of freedom which is defined only up to scale. Each pair of corresponding points contributes only one constraint, therefore at least seven pairs of corresponding points are required to compute a fundamental matrix. Different approaches have been proposed in the literature to estimate the fundamental matrix, such as 7-point algorithm (Hartley and Zisserman, 2004), 8-point algorithm (Hartley and Zisserman, 2004), iterative estimation or robust estimation (Armangue and Salvi, 2010).
Algorithm 5 homography based verification method.

Input: The query image and the potential candidate image
Output: true (location) or false (image not matched)

1: do
2: Find 10 best SIFT correspondences between query and candidate image.
3: Declare numPerspective = 0.
4: Use RANSAC for random picking of 4 SIFT correspondences 15 times.
5: for every set of four SIFT correspondences do
6: Compute perspective transformation matrix.
7: Transform all candidate features to new locations.
8: Identify the transformed features coming with in (3 x 3) window of corresponding query features.
9: Check all such feature similarities with the query features (using a distance threshold of 150) and keep the “COUNT”.
10: if COUNT >= 3 then
11: numPerspective ++;
12: end if
13: end for
14: if numPerspective >= 3 then
15: return location of the matched image;
16: else
17: candidate image not matched, so return false and pick the next ranked image;
18: end if

2003; Huang et al., 2007). Pairs of corresponding points are first determined between two images followed by the computation of a fundamental matrix and inlier points. Inlier points refer to correctly matched correspondences and outliers refer to mismatched correspondences. Inlier points match according to the fundamental matrix equation:

\[ x_i' T F x_i < \epsilon, i = 0, 1, ..n \]  

(7.6)

where \( x_i' \) and \( x_i \) are points in two images, and \( \epsilon \) is an empirically determined threshold. A larger number of inliers indicates a more reliable image match between two images and this information is used as the basis for decisions regarding an image match. The fundamental matrix has been used by many researchers to refine feature correspondences during image matching (Agarwal et al., 2011; Irschara et al., 2009; Sattler et al., 2012; Snavely et al., 2006).

This verification method uses the fundamental matrix because it uses a full structure match between query and candidate images and is more physically plausible than homography. The method first computes a number of best SIFT correspondences between a query and the candidate image followed by the computation of the fundamental matrix using the 8-point algorithm via RANSAC (Hartley and Zisserman, 2004):

- Repeat for \( N \) samples, where \( N \) is determined by adaptive algorithm defined in (Hartley and Zisserman, 2004):
Select a random sample of 8 correspondences.

Compute the fundamental matrix using 8-point algorithm:

* The input data is normalised by transforming the image coordinates according to \( x_i = T x_i \) and \( x'_i = T' x'_i \), where \( T \) and \( T' \) are normalising transformations consisting of a translation and scale.

* From a set of \( M \) points, equation 7.4 can be written as set of linear equations:

\[
Af = \begin{bmatrix}
x'_0 x'_0 & x'_0 y'_0 & x'_0 & y'_0 x_0 & y'_0 & x_0 & y_0 & 1 \\
\vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \vdots \\
x'_M x'_M & x'_M y'_M & x'_M & y'_M x_M & y'_M & x_M & y_M & 1 \\
\end{bmatrix} \begin{bmatrix} f \end{bmatrix} = 0
\] (7.7)

A solution is obtained from the vector \( f \) corresponding to the smallest singular value of \( A \). Then solution \( f \) is replaced with \( f' \) such that \( \det f' = 0 \) using the singular value decomposition.

* Denormalise \( f' \), so it corresponds to original data.

With the determined \( f' \), the number of inliers consistent with correspondences are computed. A distance threshold of 1.0 is used which refers to the maximum distance from the point to epipolar line in pixels beyond which the point is considered an outlier.

- Choose \( f' \) with the largest number of inliers as fundamental matrix \((F)\).

The working of fundamental matrix verification method is shown in Algorithm 6.

---

**Algorithm 6** fundamental matrix based verification method.

**Input:** The query image and the potential candidate image  
**Output:** true (location) or false (image not matched)

1: do  
2: Find \((Z)\)% of best SIFT correspondences between query and candidate image.  
3: Estimate the fundamental matrix using RANSAC (with 8 point algorithm).  
4: if at least \((Z1)\)% inliers are detected then  
5: return location of the matched image;  
6: else  
7: candidate image not matched, so return false and pick the next ranked image;  
8: end if

We experimented with different values for \( Z \) and \( Z1 \) such as 25%, 50%, 75% etc. \( fm\)-BoW variants tested in this thesis are described below:

**FM\(_{25,25}\)**

1. Find top 25% SIFT correspondences between query and the candidate image.
2. If query and candidate have at least 25% inliers; consider it a best match.

\[
\text{FM}_{25,40}
\]
1. Find top 25% SIFT correspondences between query and the candidate image.
2. If query and candidate have at least 40% inliers; consider it a best match.

\[
\text{FM}_{50,20}
\]
1. Find top 50% SIFT correspondences between query and the candidate image.
2. If query and candidate have at least 20% inliers; consider it a best match.

\[
\text{FM}_{10,20}
\]
1. Find top 10 SIFT correspondences between query and the candidate image.
2. If query and candidate have at least 20% inliers; consider it a best match.

\[
\text{FM}_{10,75}
\]
1. Find top 10 SIFT correspondences between query and the candidate image.
2. If query and candidate have at least 75% inliers; consider it a best match.

\[
\text{FM}_{30,20}
\]
1. Find top 30 SIFT correspondences between query and the candidate image.
2. If query and candidate have at least 20% inliers; consider it a best match.

7.2 Datasets and performance metrics

We used the Computer Science (CS) dataset for the evaluation of visual BoW with all verification methods. We applied RANSAC to pick randomly test and training data 15 times. Therefore, a 15-fold cross-validation with different test and training sets i.e. 70 test and 630 training images, was performed in our experiments to compute the average performance with the CS dataset. For more details about this dataset, please refer to Chapter 4.

In this dataset, training images are different from test images. For every test image of an indoor location, there are different training images of the same indoor location taken from different viewpoints. To match a test image, the visual BoW first retrieves images similar to a query image and ranks them. The top 200 ranked images are then passed to the voting module, which calls the verification method (if required) to find an image match from the training dataset. The image match is considered right if the matched training image refers to the same indoor location as the test image. Otherwise, the image match is considered wrong for a test image.

We used the following performance metrics for analysis:
1. $C_a$ : refers to the correct acceptance rate.
   The expected value for this metric ranges from 0-1. A higher value is better for this metric and indicates a high image matching accuracy against query images, which illustrates the success of proposed visual BoW system (based on voting module and a verification method).

2. $W_m$ : refers to the wrong match rate.
   The expected value for this metric ranges from 0-1. A lower value is better for this metric and indicates the number of incorrect image matches against query images. A low value of $W_m$ and a high value of $C_a$ reflects good image matching performance of the proposed visual BoW system.

3. $R_{nd}$ : refers to the no-decision rate.
   The expected value for this metric ranges from 0-1. A lower value is better for this metric, which indicates that the proposed visual BoW system makes image match decisions against most of the query images. A high value of $R_{nd}$ results in smaller $W_m$ because decisions are not made against most query images. However, the corresponding $C_a$ goes down due to fewer image match decisions.

4. $S_{cm}$ : refers to the scene confusion matrix.
   The expected value for this metric ranges from 0-100% and it indicates the grouping probability of an indoor query location against all training indoor locations, such as how many times query corridors are classified as corridors, rooms or halls. A higher value is better for this metric and indicates that the place type of query indoor locations are recognised correctly most of the time.

For more details about these metrics, please refer to Chapter 4.

7.3 Results

In this section, we analyse variants of fundamental matrix based verification method, compare the performance of all verification methods with the standard visual BoW and test the performance of the system for image matching with multiple query images.

7.3.1 Fundamental matrix based variants analysis

In this section, we compared variants of the fundamental matrix based verification method. The results in Figure 7.7 show that both $FM_{25,40}$ and $FM_{50,20}$ give very low wrong match rates ($W_m$) compared to other variants. The main reason is that these variants have very high no-decision ($R_{nd}$) rate as shown in Figure 7.8. So a localisation decision has not been made on most query images which results in a reduced $W_m$, but the corresponding correct acceptance rate ($C_a$) is also very low, which makes these variants unsuitable for image matching systems, where a high recognition rate is expected.
Figure 7.7: *fm-BoW* variants performance comparison. The curves that are higher and further to the right indicate a better performance.

A similar argument can be made for *FM10,20*, *FM10,75* and *FM30,20*, which give better \( C_a \) than other variants but with high \( W_m \). These variants make decisions on almost all query images, which leads to more wrong image matches. These variants give similar performance, which has merged the corresponding curves in Figure 7.7 and it is hard to distinguish among them. These variants are suitable for image retrieval systems to retrieve relevant images against a query image but cannot be used for indoor positioning where localisation errors must be avoided.

Figure 7.8: *fm-BoW* variants no-decision rates.

Results show that \( C_a \) of *FM25,25* is less than *FM10,20*, *FM10,75* and *FM30,20* variants
but at the expense of lower $W_m$ along with reasonable $R_{nd}$. For our smartphone indoor localisation system, we need a method to match many query images but with few wrong matches as discussed in Chapter 2. The wrong location is a disaster and must be avoided. So it is better to reject a match rather than making a wrong match because another query image can be used in such situations. $FM_{25,25}$ offers a reasonable trade-off between the correct acceptance, wrong match and no-decision rates, which makes it more suitable for our indoor image matching system compared to its variants. Therefore, we have used $FM_{25,25}$ in remaining experiments. We will refer to $FM_{25,25}$ simply as $FM$ in the remainder of this section.

### 7.3.2 Verification methods analysis

The performance of all verification methods with the proposed visual BoW system (presented in the previous chapter) is compared with standard visual BoW as shown in Figure 7.9. It is not surprising to see that the verification methods with the best recognition rate ($C_a$) also have the highest wrong match rate ($W_m$). Verification methods such as $SD$, $SH$, $SSH$ and $LBP$ make a firm recognition on almost all query images and hence have the highest recognition rates. These methods use simple information such as colour, local binary pattern and feature similarities to match two images. The chances of finding wrong image matches are high if a correct matching image is not present among the top few retrieved candidate images. In such cases, these verification methods often match the wrong candidate image with a query image. In the experiments, $SSH$ performs best and the system achieves a recognition rate of 90% with 8% of wrong matches for vocabulary size of 50K clusters. $SD$ and $LBP$ both also give a recognition rate of 90% with 50K clusters but with a wrong match rate of 9%.

Conversely, $FM$ and $HM$ verification methods exploit the image geometry information between images. These methods accept an image match only if a good geometric relationship is established between images, and hence are more reliable. Therefore, their overall recognition rates are lower (including no-decision images), but also their wrong match rates are significantly lower as shown in Figure 7.9. However, when they do make a recognition decision, they make the correct decision most of the time. The overall recognition rate of about 80% is achieved with these verification methods with a $W_m$ of 2%. Both $FM$ and $HM$ produce fewer wrong image matches than simple verification methods with all visual vocabulary sizes because of a reasonable $R_{nd}$, which is suitable for indoor localisation. The $C_a$ of these two verification methods can be increased by reducing $R_{nd}$ via changing parameters of verification methods such as using a lower inliers ratio in $FM$ or using one perspective correspondence to accept an image match in $HM$. However, the chances of getting wrong image matches will also increase.

Figure 7.9 shows that $FM$ gives a slightly better $C_a$ than the $HM$. However, the $W_m$ of $FM$ is higher than $HM$. We decided to check the significance of these differences by performing one-tailed paired t-test with $\alpha = 0.05$ using two configurations:

1. In the first configuration, we used the $C_a$ of both methods across seven vocabulary sizes as data for the test. Our null hypothesis is: there is no difference in the means of two methods. While our alternative hypothesis is: there is a significant
difference i.e. \( FM \) does better than \( HM \). The test gives a \( p \)-value of 0.0006, which indicates that \( FM \) gives better \( C_a \) than \( FM \).

2. In the second configuration, we used the \( W_m \) of both methods across seven vocabulary sizes as data for the test. Our null hypothesis is: there is no difference in the means of two methods. While our alternative hypothesis is: there is a significant difference i.e. \( HM \) does better than \( FM \) by producing fewer wrong matches. A \( p \)-value of 0.02 suggests that \( HM \) performs better than \( FM \).

The above results support our hypothesis that there is nothing much to differentiate between the two methods because both methods can be configured to give almost similar performance, such as \( C_a \) and \( W_m \). Although the \( C_a \) for \( FM \) and \( HM \) is low, the CS dataset is difficult and has many visually similar images, which increases chances of finding wrong image matches, such as hall of the second floor is matched with hall of the third floor or corridor on the left side of the building gets matched with corridor on the right side of building. It is not unexpected that our system would fail on such images. Some examples of wrong image matches are shown in Figure 7.10.

**Stability**

Another important aspect is the stability of verification methods. The standard deviations of the wrong match rates (\( W_m \)) with all verification methods are shown in Table 7.3. The results show that \( W_m \) decreases with an increase in the number of clusters used for visual BoW. The reason is that a large number of clusters produces
refined groupings of training features (as discussed in the previous chapter) in visual BoW, which leads to better image rankings and results in an overall improvement in the image retrieval performance of the system.

We have performed 15-fold cross-validation experiments to compute the $C_a$ and $W_m$ for each method with all vocabulary sizes. The low standard deviation for $W_m$ indicates that the verification method always gives a similar $W_m$ in all experiments with a used vocabulary size. The results in Table 7.3 indicate that $HM$ and $FM$ both have the lowest $W_m$ and corresponding standard deviations compared to the other verification methods for all vocabulary sizes. Hence, we can say that both $HM$ and $FM$ are more stable and good choice for our indoor localisation system.

Table 7.3: Proposed visual BoW wrong match rates ($W_m$) and corresponding standard deviations against different vocabulary sizes. The lowest $W_m$ are highlighted in red for all methods.

<table>
<thead>
<tr>
<th>Clusters</th>
<th>SD</th>
<th>SH</th>
<th>SSH</th>
<th>LBP</th>
<th>FM</th>
<th>HM</th>
</tr>
</thead>
<tbody>
<tr>
<td>1K</td>
<td>0.22±0.03</td>
<td>0.17±0.03</td>
<td>0.19±0.03</td>
<td>0.23±0.04</td>
<td>0.15±0.02</td>
<td>0.05±0.01</td>
</tr>
<tr>
<td>5K</td>
<td>0.14±0.02</td>
<td>0.14±0.03</td>
<td>0.12±0.02</td>
<td>0.14±0.02</td>
<td>0.06±0.01</td>
<td>0.02±0.01</td>
</tr>
<tr>
<td>10K</td>
<td>0.13±0.02</td>
<td>0.14±0.02</td>
<td>0.10±0.02</td>
<td>0.13±0.02</td>
<td>0.05±0.01</td>
<td>0.03±0.01</td>
</tr>
<tr>
<td>25K</td>
<td>0.11±0.02</td>
<td>0.14±0.02</td>
<td>0.09±0.02</td>
<td>0.10±0.03</td>
<td>0.04±0.01</td>
<td>0.02±0.01</td>
</tr>
<tr>
<td>50K</td>
<td>0.09±0.02</td>
<td>0.12±0.02</td>
<td>0.08±0.03</td>
<td>0.09±0.02</td>
<td>0.04±0.01</td>
<td>0.03±0.01</td>
</tr>
</tbody>
</table>
7.3.3 Computational Time

We computed the average time (including retrieval, voting and verification) required by our proposed system with the $HM$ and $FM$ verification methods to match one query image with different vocabulary sizes as shown in Table 7.4.

Table 7.4: Average time to match one query image (in seconds).

<table>
<thead>
<tr>
<th></th>
<th>1K</th>
<th>5K</th>
<th>10K</th>
<th>25K</th>
<th>30K</th>
<th>35K</th>
<th>50K</th>
</tr>
</thead>
<tbody>
<tr>
<td>$HM$</td>
<td>0.04</td>
<td>0.20</td>
<td>0.40</td>
<td>0.91</td>
<td>1.08</td>
<td>1.25</td>
<td>1.84</td>
</tr>
<tr>
<td>$FM$</td>
<td>0.04</td>
<td>0.20</td>
<td>0.40</td>
<td>0.93</td>
<td>1.10</td>
<td>1.30</td>
<td>1.80</td>
</tr>
</tbody>
</table>

Results show that there is nothing much to differentiate between $HM$ and $FM$ in terms of efficiency as both almost give the same image matching speed. The results in Table 7.4 show that efficiency decreases with an increase in the size of visual vocabulary. This is because a large visual vocabulary results in large BoW representations (histograms) for training and query images, which increases the comparison time to find relevant images against a query image. However, the correct acceptance rate ($C_a$) of the visual BoW increases with an increase in number of clusters as shown in the previous subsections. Therefore, a vocabulary size offers a trade-off between accuracy and efficiency of a visual BoW system, the choice of which size to use depends upon the type of application. For our smartphone indoor localisation system, any size more than 10K will be suitable because the performance, such as $C_a$ and $W_m$ gets better with more clusters. The computational time also remains low with more clusters because we aim our indoor localisation system to respond within 30 seconds to the blind user as discussed in Chapter 2.

We further investigated the computational time to match two images for homography and fundamental matrix verification methods alone rather than the whole system. We ran both verification methods several times and recorded the average results. The results in Table 7.5 show that:

1. With RANSAC, the fundamental matrix verification method is not as efficient as the homography verification method. However, it gives a very good image matching performance, which is comparable or sometimes better than the homography verification method.

2. Without a RANSAC option, only 8 correspondences are used by the fundamental matrix verification method. This time the fundamental matrix verification method is quite efficient, but it gives a poor image matching performance compared to homography verification method with RANSAC. The homography verification method is also more efficient without RANSAC, but it performs poorly.

The results indicate that the homography verification is more efficient than the fundamental matrix verification, but Table 7.4 shows that $FM$ performs comparably to $HM$. This is because $HM$ rejects about 6% more query images compared to $FM$. Rejecting a query image is a time-consuming step because a query image is compared with top 50 candidate images and then a rejection decision is made. We suspect that
a relatively high no-decision rate of $HM$ makes its overall image matching time similar to $FM$. In contrast the homography verification method alone is more efficient than fundamental matrix based verification because it requires fewer feature correspondences to make a localisation decision.

### 7.3.4 Image matching with multiple query images

We computed the performance of the system for image matching with multiple query images and compared results with the system when it uses only one query image i.e. $HM$ and $FM$. We designed one test and one training set. The test set included four query images for each indoor location, $q_1$, $q_2$, $q_3$ and $q_4$. All query images differed from each other and did not exist in the training set. We conducted the following experiments:

**Using two query images ($HMc & FMc$)**

With this configuration, the system reports the results on three other query images if it cannot make a localisation decision on $q_1$. A localisation decision is made only if $q_2$, $q_3$ and $q_4$ all refer to the same location.

Results in Table 7.6 show that $HMc$ and $FMc$ achieve a $C_a$ of 94% and 95% respectively because the system finds correct matches against the second query image most of the time. The $W_m$ for $HMc$ and $FMc$ is about 3% and 4% respectively. The average time of the system to match the two query images is reported in Table 7.7, which shows that $HMc$ and $FMc$ both are efficient and perform almost similar to $HM$ and $FM$.

Sometimes the captured query image is not very discriminating, have blur or darkness. This results in poor image rankings and hence a match is not found. The chances of having same problems with the second or another query image are low, which results in an image match against the second query image most of the time. Though our system attempts to make many correct matches with a query image but it is not possible to reject all wrong matches. Therefore, the chances of wrong image matches equally increase with an increase in the corresponding $C_a$. The similar behavior is observed with this experiment, where the use of two query images has increased the $C_a$ of the system but with a slight increase in $W_m$. However, the efficiency remains similar to $HM$ and $FM$.

**Using three query images ($HMc & FMc$)**

The system reports the results on three other query images if it fails to make a decision on $q_1$. A localisation decision is made only if $q_2,q_3$ and $q_4$ all refer to the same location.

Results in Table 7.6 show that a $C_a$ of 94% is obtained with both $HMc$ and $FMc$, with
Table 7.6: Performance with multiple query images. The bold values indicate a large increase in $C_a$ or large decrease in $W_m$ compared to $HM$ and $FM$. Note that $HM$ and $FM$ results are different to Figure 7.9 because of different datasets.

<table>
<thead>
<tr>
<th>Clusters</th>
<th>1K</th>
<th>25K</th>
<th>50K</th>
<th>Three query images</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$HM_e$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$C_a$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$W_m$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$R_{nd}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$FM_e$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$C_a$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$W_m$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$R_{nd}$</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Clusters</th>
<th>1K</th>
<th>25K</th>
<th>50K</th>
<th>Four query images</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$HM_c$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$C_a$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$W_m$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$R_{nd}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$FM_c$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$C_a$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$W_m$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$R_{nd}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$HM_f$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$C_a$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$W_m$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$R_{nd}$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$FM_f$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$C_a$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$W_m$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>$R_{nd}$</td>
</tr>
</tbody>
</table>

Table 7.7 shows that both $HM_c$ and $FM_c$ are slightly less efficient than $HM$ and $FM$.

This experiment also increases the $C_a$ of the system with small or no increase in the corresponding $W_m$ if compared to $HM$ and $FM$. But the average time to match a query image of the resulting system is slightly high. The $C_a$ with this experiment remains lower than the previous experiment because a decision is only made if multiple query images (i.e. 3) match to the same location. This is highly unlikely that all query images will refer to the same wrong location hence resulting in more image rejections. This is also reflected in the results, where no-decision rate ($R_{nd}$) is slightly higher for both $HM_c$ and $FM_c$ than $HM_a$ and $FM_b$. However, this experiment does not affect the $W_m$ of the system, which remains almost similar to the previous experiment.

Using multiple query images for every localisation decision

$HM_d$ & $FM_d$: The system reports the results on two query images. A localisation decision is made if two queries refer to the same location, otherwise a decision is not made.

$HM_e$ & $FM_e$: The system reports the results on three query images. A localisation decision is made only if three query images refer to the same location.

a $W_m$ of 3% and 4% for both $HM_c$ and $FM_c$ respectively.
**HMf & FMf:** The system reports the results on all query images. A localisation decision is made only if $q_1, q_2, q_3$ and $q_4$ all refer to the same location.

The results in Table 7.6 show that use of multiple queries to make every localisation decision gives the best $C_a$ of 83% along with a $W_m$ of 0% for 50K clusters, which is due to high $R_{nd}$ i.e. up to 17%. At the same, the efficiency of the system also decreases with an increase in the number of query images used for localisation decisions. The results in Table 7.7 show that the average time to match a query image increases with an increase in the number of multiple query images, which are used to make every localisation decision.

Table 7.7: Average time to match a query image with one and multiple images (in seconds). *Note that HM and FM results are different to Table 7.4 because of different datasets.*

<table>
<thead>
<tr>
<th></th>
<th>1K</th>
<th>25K</th>
<th>50K</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HM</strong></td>
<td>0.04</td>
<td>0.91</td>
<td>1.83</td>
</tr>
<tr>
<td><strong>FM</strong></td>
<td>0.04</td>
<td>0.93</td>
<td>1.80</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1K</th>
<th>25K</th>
<th>50K</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HMb</strong></td>
<td>0.05</td>
<td>1.03</td>
<td>2.09</td>
</tr>
<tr>
<td><strong>FMb</strong></td>
<td>0.05</td>
<td>1.02</td>
<td>1.95</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1K</th>
<th>25K</th>
<th>50K</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HMc</strong></td>
<td>0.07</td>
<td>1.28</td>
<td>2.61</td>
</tr>
<tr>
<td><strong>FMc</strong></td>
<td>0.06</td>
<td>1.21</td>
<td>2.26</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1K</th>
<th>25K</th>
<th>50K</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HMd</strong></td>
<td>0.08</td>
<td>1.87</td>
<td>3.63</td>
</tr>
<tr>
<td><strong>FMd</strong></td>
<td>0.08</td>
<td>1.83</td>
<td>3.59</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1K</th>
<th>25K</th>
<th>50K</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HMe</strong></td>
<td>0.12</td>
<td>2.91</td>
<td>5.52</td>
</tr>
<tr>
<td><strong>FMe</strong></td>
<td>0.12</td>
<td>2.86</td>
<td>5.40</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th></th>
<th>1K</th>
<th>25K</th>
<th>50K</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>HMf</strong></td>
<td>0.16</td>
<td>3.81</td>
<td>7.30</td>
</tr>
<tr>
<td><strong>FMf</strong></td>
<td>0.17</td>
<td>3.78</td>
<td>7.18</td>
</tr>
</tbody>
</table>

An increase in the use of number of query images to make every localisation decision decreases the $C_a$ because system may match one of the query images wrongly hence rejecting the match. However, the chances of wrong image matches decrease significantly with this configuration because a decision is only made if all query images are matched to the same indoor location, which is highly unlikely. Therefore, we can hypothesise that:

- Use of more query images for every localisation decision decreases the $C_a$ but the corresponding $W_m$ decreases.
- Use of extra query images in case of rejection only increases the $C_a$ but the corresponding $W_m$ slightly increases.

### 7.3.5 Scene Confusion Matrix ($S_{cm}$)

A scene confusion matrix ($S_{cm}$) groups the place type as opposed to simply location. During navigation, it is important to know the type of place in which the person or robot is present even if the specific location information is not available. Therefore, a good scene confusion matrix is desired from a localisation system.

We realised from experiments that using fewer clusters (such as 1K, 5K and 10K) gave poor image rankings with visual BoW, which reduced the overall recognition performance of the system. Use of more clusters (such as from 25K-50K) resulted in higher correct acceptance rate ($C_a$) and lower wrong match rate ($W_m$), because better image rankings were generated. 25K clusters offered a reasonable trade-off between efficiency and image matching performance of the system. In this section, we have
therefore reported the scene confusion matrix for \( HM \) and \( FM \) with 25K clusters as shown in Tables 7.8 and 7.9. The results show that the type of place is recognised well with few wrong image matches. These results are good and are comparable with the results shown in Espinace et al. (2010), which show a good scene confusion matrix for four indoor locations, such as office, hall, conference and bathroom during indoor image matching based on object recognition.

Table 7.8: Scene Confusion Matrix for \( HM \). LEGEND: L, LABS; CR, CONFERENCE ROOM; CoR, COFFEE ROOM; C, CORRIDORS; H, HALLS; W, WASHROOM; O, OFFICES; NL, No-Location; \( W_m \), wrong match rate.

<table>
<thead>
<tr>
<th># query</th>
<th>Places</th>
<th>Measured performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>CR</td>
<td>CoR</td>
</tr>
<tr>
<td>22</td>
<td>L</td>
<td>73% 2% 0% 0% 0% 0% 0% 25% 2%</td>
</tr>
<tr>
<td>2</td>
<td>CR</td>
<td>0% 83% 0% 0% 0% 0% 0% 17% 0%</td>
</tr>
<tr>
<td>2</td>
<td>CoR</td>
<td>0% 0% 87% 0% 0% 0% 0% 13% 0%</td>
</tr>
<tr>
<td>30</td>
<td>C</td>
<td>0% 0% 0% 82% 0% 0% 1% 17% 1%</td>
</tr>
<tr>
<td>6</td>
<td>H</td>
<td>0% 0% 0% 0% 86% 0% 0% 14% 0%</td>
</tr>
<tr>
<td>2</td>
<td>W</td>
<td>0% 0% 0% 0% 75% 0% 25% 0% 0%</td>
</tr>
<tr>
<td>6</td>
<td>O</td>
<td>0% 0% 0% 0% 0% 80% 20% 0% 0%</td>
</tr>
</tbody>
</table>

Table 7.9: Scene Confusion Matrix for \( FM \). LEGEND: L, LABS; CR, CONFERENCE ROOM; CoR, COFFEE ROOM; C, CORRIDORS; H, HALLS; W, WASHROOM; O, OFFICES; NL, No-Location; \( W_m \), wrong match rate.

<table>
<thead>
<tr>
<th># query</th>
<th>Places</th>
<th>Measured performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>L</td>
<td>CR</td>
<td>CoR</td>
</tr>
<tr>
<td>22</td>
<td>L</td>
<td>76% 1% 0% 0% 0% 0% 0% 23% 1%</td>
</tr>
<tr>
<td>2</td>
<td>CR</td>
<td>0% 90% 0% 0% 0% 0% 0% 10% 0%</td>
</tr>
<tr>
<td>2</td>
<td>CoR</td>
<td>0% 0% 87% 6% 0% 0% 0% 7% 6%</td>
</tr>
<tr>
<td>30</td>
<td>C</td>
<td>0% 0% 0% 93% 1% 0% 1% 5% 2%</td>
</tr>
<tr>
<td>6</td>
<td>H</td>
<td>0% 0% 0% 74% 0% 0% 26% 0% 0%</td>
</tr>
<tr>
<td>2</td>
<td>W</td>
<td>0% 0% 0% 0% 100% 0% 0% 0% 0%</td>
</tr>
<tr>
<td>6</td>
<td>O</td>
<td>0% 0% 0% 0% 0% 83% 17% 0% 0%</td>
</tr>
</tbody>
</table>

\( FM \) gives a better overall confusion matrix than \( HM \). The reason is that \( FM \) rejects fewer query images and make more localisation decisions than \( HM \), but this also increases its \( W_m \) compared to \( HM \), which has very low \( W_m \) as shown in Tables 7.8-7.9. Other than that there is not much to differentiate between them.

### 7.4 Conclusion

The current chapter and the previous chapter show that standard visual BoW performs well in terms of recognition accuracy indoors, but suffers somewhat from a high wrong match rate. We experimented with different verification methods for image matching in this chapter. Unsurprisingly, as the verification algorithms reduce the wrong match
rate, the overall recognition rate also goes down. The optimal choice of which scheme to use is dependent on the application.

Other verification methods, such as SSH and LBP provided very good $C_a$ but they gave around 5% more wrong image matches than FM and HM. One way to take benefit from the good $C_a$ of SSH and LBP is to combine one of these methods with either FM or HM. An image match can be found first either using SSH or LBP followed by a validation of that image match either by HM or FM. But this will increase the computational cost for the system. Moreover, it is also not necessary that HM or FM will always validate the correct match due to the lack of image geometry between two images. A better solution can be to use some confidence level with SSH or LBP, such as if a decision is made with good confidence then no validation based on image geometry is needed. Otherwise, a validation is performed by using image geometry. However, this requires extensive experiments across different indoor datasets to identify the parameters for confidence level and should be the focus of the future work.

The focus of our work is indoor localisation with smartphones to guide blind people while they navigate indoors. Wrong positioning information can be a disaster to blind people, such as if stairs are recognised as a corridor or a room is recognised as a lift. Therefore, it is better not to find a match instead of giving the wrong location information because the system can use another picture of the current location to make a decision, if a match is not found. HM and FM are both found to give the best performance among all verification methods and are used with visual BoW in this thesis.

We hypothesised that fundamental matrix verification method would be more accurate but less efficient than homography verification method. We found that it was indeed less efficient, but surprisingly not much more accurate. However, the extra processing cost is minimal compared to the whole retrieval system when used in a tiered fashion. Although the fundamental matrix verification method is more physically plausible, for indoor localisation, there are many planar surfaces, and we strongly suspect that this allows the homography verification method to do as well as the fundamental matrix verification method. We believe that these results would transfer to many built environments, but probably not to less structured ones. The homography verification method offers a good trade-off between accuracy, wrong match rate and computational efficiency.

We further analysed the performance of the system with multiple query images. We realised that use of multiple query images in varying ways offered reasonable trade-offs between the correct acceptance rate, wrong match rate, no-decision rate and efficiency of the system. The use of two query images significantly improved the correct acceptance rate of the system. The use of four query images decreased the wrong match rate of the system to zero in some cases at the expense of a decrease in the efficiency of the system. Indoor localisation with one query image is the main focus of the work in this thesis. However, we expect that it will be interesting to use multiple query images in future to give better recognition performance and to further reduce the number of wrong image matches.

The contribution of this chapter is the analysis of several verification methods which can be used with visual BoW for indoor image matching. The use of multiple query images with the system is also explored, which seems to have a promising effect on the
overall recognition performance of the system.

We discuss a technique to reduce the number of trained SIFT features used with visual BoW in the next chapter.
Chapter 8
Reduced feature sets

Note: Some portions of this chapter are taken from (Khan et al., 2012a).

The previous two chapters presented the proposed visual Bag of Words (BoW) for Indoor Positioning System (iPoS). The size of a visual vocabulary (i.e. number of visual words or clusters) greatly affects the recognition performance and run time of the visual BoW approach. It has been shown that using large visual vocabularies, in the order of hundreds of thousands, improves the recognition performance for large databases with thousands of images (Aly et al., 2011; Nister and Stewenius, 2006; Philbin et al., 2007, 2008). However, a large visual vocabulary increases the time to compute visual words for an image and results in large visual BoW representations (histograms) for each training image. Therefore, more time is required to compare large histograms of training images with a query image, which decreases the overall efficiency of the retrieval system. Such efficiency drops are particularly significant if the whole processing is carried out on a mobile device which has limited memory and processing power compared to a desktop system. Therefore, a compact feature representation of a large environment is desirable for the efficiency of an image retrieval system on both desktop systems and mobile devices.

In a large scale indoor environment, every location has several associated images. In this chapter, we investigate the ways to generate a reduced feature set of training features (SIFT) from large indoor environments, where every location has a number of images. The motivation behind reducing the number of training features is to represent large environments with reduced feature sets to allow our proposed visual BoW system (presented in the previous chapter) to use a small number of clusters during quantisation, which will result in small sized histogram representations for images. This will result in fast generation of image rankings against query images followed by the use of a voting scheme and a verification method to find an image match efficiently for our smartphone based indoor localisation system. Moreover, reduced features require less memory and can therefore improve the efficiency of our system if the proposed visual BoW has to be run entirely on a smartphone.

The main contribution of this chapter is the presentation of reduced feature sets, its analysis and comparison with some well known features.
8.1 Large scale image matching

When a training database contains millions of images, the problem of efficiently searching for a relevant image against a query image becomes challenging. One of the important challenges is to use an approach that scales well with the size of the database and selects relevant images out of a large number of images in an acceptable time. The traditional approach of image matching compares the query image features with the features of every training image to find relevant images. This process may work with small datasets, but it does not scale well with an increase in the number of training images. Imagine a database with thousands of images, such as the “Tiny” dataset with 80 Million images which are collected from the Web and stored as 32 x 32 colour images (Torralba et al., 2008), the “Flickr1” dataset with about 99 thousand high resolution images which are collected from Flickr (Philbin et al., 2008), city datasets with up to 0.2 Million images (Li et al., 2010; Sattler et al., 2011; Snavely et al., 2008) etc., where there are many millions of training features, and traditional image matching approaches may take days to find relevant images from such datasets.

As shown in Chapter 7, the efficiency of the proposed visual BoW decreases with an increase in the size of visual vocabulary. The average retrieval speed for a query image was 1.84 seconds for a visual vocabulary with size of 50K compared to visual vocabulary with size of 1K clusters, which took 0.04 seconds. Ideally, representation of large datasets with a small number of effective training features is desirable so that visual BoW can be used with a visual vocabulary of reasonable size. Therefore, the best approach is to reduce the number of training features and then use them with visual BoW. Li (2006) proposed a scheme to reduce the number of training features for precise image matching in an indoor environment. The scheme first extracts SIFT features from images after partitioning the environment into locations. Then it uses a novel feature selection strategy, such as posterior probability to estimate the ranks of features for each training image and selects the top 10% of the ranked features from each training image in the experiments. During image matching, the features of a query image are compared with the reduced features of locations for an image match. In experiments, the authors reduced the number of features from a database of 296 training images of 18 indoor locations. The system is reported to give a correct image recognition performance of 0.74-0.80 with two test sequences, which is an efficiency drop of 2-9% over unreduced features. Although they report very good results with the reduced number of training features, it is not clear how well the technique will scale to very large databases as features are kept based on their discrimination ability.

Contrary to that, Ledwich and Williams (2004) presented a method to generate fewer SIFT features directly from training images rather than reducing the number of extracted training features. The SIFT algorithm generates multiple keypoints at a single location with different orientations to achieve rotation invariance. The authors assumed that the view point for images would remain relatively stable to the rotation around the view axis and the orientation of keypoint descriptors (located on vertical surfaces) will not rotate. The system therefore ignores the rotational invariance of the SIFT features by removing the orientation assignment and the descriptor alignment steps, which reduces the computation cost along with the overall number of features because multiple keypoints are not generated at a single location. The system was
tested on two test environments with 13 and 10 indoor places respectively. The authors report that their feature reduction approach improves the efficiency of image retrieval and also has a minimal affect on the retrieval rate of images compared to unreduced features.

The above works attempt to reduce the number of training features. On the other hand, some works have proposed schemes to reduce the size of feature vectors (Khan et al., 2012; Stommel, 2010). Such works decrease the overall memory requirements to store features and are useful with very large databases because it is not possible to load millions of features in memory at a time for image matching. Khan et al. (2012) presented a feature compression scheme based on reducing the number of bits for SIFT features for large scale image retrieval systems. The scheme makes use of the significant bits from feature bytes and reduces the size of SIFT descriptor from 128 bytes to 512 bits. During experiments, the authors tested reduced features against different image transformations on standard datasets. The authors report that image matching performance with compressed features is almost similar to the approach reported in (Stommel, 2010), which reduces the size of SIFT feature from 128 bytes to 128 bits. This scheme is not tested on any indoor environment.

Recent advancements in structure-from-motion (sfm) research (Snavely et al., 2008) have made it possible to construct effective 3D models for environments on a large scale thus making it possible to perform 2D to 3D matching for image localisation. Each 3D point has a list of features, which are seen in registered images. During image matching, features of a query feature are compared with all features of 3D points to find 2D to 3D correspondences followed by pose estimation to find an image match. A large 3D model may have millions of 3D points and there may be thousands of features for each 3D point, which greatly increases the time to search for 2D to 3D correspondences. Different approaches have been proposed to reduce the size of 3D models to perform efficient image matching (Irschara et al., 2009; Li et al., 2010; Sattler et al., 2011). In one notable work, Sattler et al. (2011) reduced the size of 3D data by taking the average of all features, which generated a single feature for each 3D point. A visual vocabulary was then developed on compressed data and was used during image localisation. The system is reported to match more query images with the reduced 3D data than Li et al. (2010), who reduced the same 3D data in a different way.

In section 8.2, we have proposed an approach which aims to reduce the number of features in training data based on the concept of tracks. The feature reduction methodology is a 2D track based approach inspired by the work of (Li et al., 2010; Sattler et al., 2011), who reduced the size of 3D points in a point cloud by averaging corresponding features related to 3D points. The proposed approach differs from theirs in the way that we have used 2D images to reduce the number of training features and the approach does not need reconstructed 3D models. The approach differs from the work of (Ledwich and Williams, 2004), because no assumptions are made about the rotation of images and more indoor locations are used during the experiments. It also differs from other works because the focus is kept on reducing the number of training features needed to represent large indoor environments and different datasets (including indoors and outdoors) are used to evaluate this reduction.
8.2 Track based feature reduction

In feature matching schemes (e.g. visual bag-of-words), each image is represented as a set of features. The central idea of the approach is to represent each scene (e.g. the kitchen, the laboratory etc) as a bag of features, where each scene would typically have several images associated with it. The reduction in the number of features for a bag comes from identifying similar features in multiple images of the same scene and storing that feature only once. Tracks represent similar features that are matched across multiple images of the same scene as shown in Figure 8.1. The essential idea is very simple, but there are multiple ways in which the tracks can be generated, which can lead to different types of reduced feature sets per scene. We used sets of 96D SIFT features in this chapter, however the approach can be used with any robust descriptor.

Figure 8.1: An example of a track showing one similar feature traced across three images of an indoor scene.

8.2.1 Track generation

A greedy method is used in this work to identify feature matches for each pair of images in a scene in order to generate tracks. Two features are said to match if the Euclidean distance between them in SIFT feature space is less than 170, an empirically determined distance threshold. This threshold ensures correct feature matches and rejects most wrong feature matches. The initial tracks only contain matched features from two images. However, these tracks are expanded to include matched features from other images in two ways:

1. **Strict**: if a newly found feature matches all existing features of a track, then it is added to the existing track.

2. **Less Strict**: if a newly found feature matches with the original feature, then it is added to the existing track.

Non-matching features of a scene are stored in singleton tracks, where a single feature forms a track.

8.2.2 Reduced feature set

All possible tracks are generated first and each track is then represented as the average of all features contained within it. A scene is therefore represented as a set of tracks.
We evaluated four different scene description variants:

- **ST**: All strict tracks.
- **LT**: All less strict tracks.
- **STF**: Strict tracks excluding singletons.
- **LTF**: Less strict tracks excluding singletons.

The argument for removing singleton tracks is that since they have not matched between training images of the same scene, it is less likely they would match a query image. Eliminating singleton tracks also produces a large reduction in the number of features needed to represent a scene.

The approach used to generate tracks is shown in Algorithm 7. The algorithm requires that each image be labeled with its corresponding location information and this information has been provided manually to the algorithm. This information can be determined automatically by using a place categorization method, such as (Ranganathan, 2012). The problem with such approaches is that correct place categorisation rate is not very high, for example Ranganathan (2012) reported 46% average correct place categorisation for indoor environments in his work. Therefore, a better approach is required for automatic place categorization and is not the focus of the work in this thesis.

While identifying feature correspondences between images, we also record the proportion of features from each image that are matched with the remaining images of the same scene. The process is done for every image of the scene and information is stored in an image correlation matrix of size $n \times n$, where $n$ is the total number of images for a particular scene. This matrix indicates the similarity between scene images on the basis of overlapping features and is used for image grouping as discussed in the next section.

### 8.2.3 Scene representation

Scenes can be represented by one of the four proposed reduced feature sets. Every scene can have a single reduced feature set and this makes sense for image matching in simple scenes, where all images are quite similar. However in complex scenes, some images within a scene can be quite different, such as images of two corners of a room. Therefore, it makes more sense to first group similar images in each scene on the basis of overlapping features and then generate a reduced feature set for each group. The groups of a scene can have either single or multiple images and the number of groups per scene depends upon the similarity threshold ($T$), which is used to identify the similarities between images. Two images are considered similar if their overlapping features are greater than $T$ either ways. Therefore, multiple reduced feature sets per scene may lead to better classification performance in complex environments. However, the number of reduced features per scene remains the same regardless of the number of feature sets.

We used the image correlation matrix to generate multiple feature sets per scene as shown in Algorithm 8. The splitting method starts with the first available scene
Algorithm 7 Algorithm to generate the tracks for a scene.

**Input:** All images along with corresponding scene information and SIFT features.

**Output:** Tracks for each scene of training data.

1: do
2: Find SIFT features for all images of a scene.
3: for all images of a scene do
4: Start with one image at a time referred as $I$.
5: for all features of $I$ do
6: for all remaining images of that scene do
7: Find non-duplicate feature matches across all images using a distance threshold of 170.
8: if (No track “exists” for the matched feature) then
9: Make a new track; add the matched features to the track.
else
11: if (Strict option ==1) then
12: Only the matched feature matching with all existing features a track is added to that track. This produces *Strict* tracks (ST or STF).
else
14: Features are added with the existing features a track without a similarity check. This produces *Less Strict* tracks (LT or LTF).
end if
16: end if
17: end for
18: Update the image correlation matrix by storing feature overlapping values (% of) of image $I$ with all remaining images of a scene.
19: end for
20: for all non-matching features of the $I$ do
21: Generate singleton tracks.
22: end for
23: end for

image and finds corresponding similar images on the basis of $T$. The resulting initial group may be expanded, as some images in the initial group may have other similar images. An incremental approach is used and the group is considered final once no further expansion is possible. This is a simple image clustering method. However, the main problem with this method is that sometimes a non-similar image may occur in a group along with other similar images and that non-similar image may add more non-similar images in that group, which may result in bad reduced feature sets for a scene. We used different similarity thresholds in the experiments to generate different numbers of reduced feature sets per scene for performance analysis.

For complex scenes, it is better to generate multiple feature sets per scene than simple scenes. But the important question is how many feature sets per scene are required across any dataset. In our results, we did not find any specific pattern, such as use of 20% or 40% similarity threshold gave better performance across complex
or simple datasets. Different performances were observed against different datasets with the same thresholds in our experiments. Therefore, we hypothesise that it is not possible to automatically select a similarity threshold value for any dataset. A suitable value for the similarity threshold can only be decided after the performance evaluation on a particular dataset.

**Algorithm 8** Algorithm to generate reduced feature sets for a scene.

**Input:** Tracks and an image correlation matrix (for a particular scene).

**Output:** Reduced feature set (single or multiple files) for a particular scene.

1: do
2: if (Single feature set ==1) then
3: Generate average features per track and return single reduced feature set;
4: else
5: Add all images of the scene in a set referred as $SET_{MAIN}$.
6: while $SET_{MAIN}$ ≠ ∅ do
7: Start with the first available image of scene referred as $I$.
8: Move $I$ to a set referred as $SET_{GROUP}$.
9: repeat
10: Find overlapping features of $I$ against other images of the scene via correlation matrix.
11: for all retrieved overlappings do
12: if (overlapping (%) > $T$) then
13: Select corresponding image as $J$.
14: if ($J$ “exists” in $SET_{MAIN}$) then
15: Move the $J$ from $SET_{MAIN}$ to the $SET_{GROUP}$.
16: end if
17: end if
18: end for
19: Delete $I$ from the $SET_{MAIN}$ and make it unavailable from $SET_{GROUP}$.
20: Select the next available image from $SET_{GROUP}$ i.e $I$.
21: until the $SET_{GROUP}$ has any available elements
22: $SET_{GROUP}$ contains the images needed to be grouped.
23: Generate reduced feature set for $SET_{GROUP}$.
24: end while
25: end if

### 8.3 Datasets and performance metrics

We used four datasets to evaluate the performance of reduced feature sets. In each dataset, the training images are different from the test images. For every test image of a scene, there are some training images of the same scene taken from different viewpoints. To perform image matching, we have used visual BoW and naive matching in the same way as was done in previous chapters:
• **Visual BoW:** The system first retrieves images, which are similar to a test image and then ranks them based on similarity against the query image. The top ranked image is picked as the best match for a test image.

• **Naive matching:** The system identifies feature correspondences between the test image and each training image. The training image giving the maximum number of feature correspondences against a test image is picked as the best match.

Both approaches find an image match for a test image from training images. The image match is only considered right if the matched training image refers to the same scene as the test image. Otherwise, the image match is considered wrong for a test image.

These datasets are briefly discussed below. More details of these datasets can be found in Chapter 4.

1. **David Nister (DN):**
   We used the first 500 objects from the database with 1500 images for training and 500 images for testing. The first image of each object was the test image and other images were used for training.

2. **Pasadena Buildings (PB):**
   We used all images from this dataset, which contains 6 images of 125 houses and buildings collectively. The first image of every building was used for testing while the remaining images were used for training.

3. **Owheo (OW):**
   We used all images from this realistic indoor dataset, with 750 images for testing and 1534 images for training.

4. **Commerce (CM):**
   We used all images from this realistic indoor dataset. 234 images were used for testing and 864 images were used for training.

The following evaluation metrics are used:

- $C_a$ refers to the correct acceptance rate.
  The expected value for this metric ranges from 0-1. A higher value is better for this metric and indicates a high image matching accuracy against query images, which illustrates the success of a visual BoW or naive matching during image matching.

- $S_c$ refers to the scene clustering rate.
  The expected value for this metric ranges from 0-100%. A value of this metric indicates on average how many images in each scene are grouped together based on similarity threshold ($T$).
A 0% value for $S_c$ means that all images of a scene are grouped together and a single reduced feature set is generated per scene. The use of other values results in multiple reduced feature sets per scene and the number of reduced feature sets per scene depend upon the value of $T$. Low $T$ results in more grouping of images per scene and results in fewer reduced feature sets per scene. This metric is used in conjunction with $E_s$.

$E_s$ refers to the scene clustering error rate.

The expected value for this metric ranges from 0-100%. This metric indicates on average how many images are wrongly grouped for each scene. A lower value is better for this metric.

More grouping of images per scene increases the chances of the existence of non-similar images in groups. So a high value of $S_c$ may increase the corresponding $E_s$, which will result in bad reduced feature sets for a scene. Therefore, a smaller value of $E_s$ is better.

For more details about these metrics, please refer to Chapter 4.

8.4 Results

We first report feature reduction statistics to highlight the effectiveness of the algorithm in terms of reducing the overall number of features used to represent the training data. We then evaluate the performance of reduced feature sets against unreduced features.

8.4.1 Feature reduction statistics

The proposed scene variants reduce the actual number of SIFT features used to represent training images. Some algorithms produce fewer features than SIFT and are claimed by their authors to be efficient and reliable for image matching. Therefore, we compared reduced feature sets with three well known features: HoG (Dalal and Triggs, 2005), GIST (Oliva and Torralba, 2001) and ORB (Rublee et al., 2011), which were presented in Chapter 5. The feature reduction statistics of the track based approach and the other features on the datasets are shown in Table 8.1. The table shows that feature reduction is significantly higher for the indoor environment due to a higher similarity between the images. A higher similarity threshold ($T$) leads to fewer groupings of images per scene and results in a large number of reduced feature sets for every scene. On the other hand, zero similarity threshold means maximum grouping between the scene images and therefore results in a single reduced feature set per scene. Table 8.1 also shows that HoG, GIST and ORB generate fewer features compared to normal SIFT features.

As can be seen from Table 8.1, the reduced feature sets which exclude singleton tracks are much more aggressive in reducing the number of features per location. It should also be noted that in image collections with relatively few images per location (DN and PB), there is a very large reduction when singletons are excluded - predominantly because there is very little overlap between the images.
Table 8.1: The feature reduction via track based approach on all datasets. LEGEND: THR, Similarity threshold; FS, Feature sets; M , Million.

<table>
<thead>
<tr>
<th>Datasets</th>
<th>Scenes</th>
<th>THR</th>
<th>FS</th>
<th>Features</th>
<th>Total features</th>
<th>Reduction</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Unreduced</td>
<td>Reduced</td>
</tr>
<tr>
<td>Indoor datasets</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OW</td>
<td>25</td>
<td>0%</td>
<td>25</td>
<td>ST (SIFT)</td>
<td>0.399M</td>
<td>0.335M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0%</td>
<td>25</td>
<td>LT (SIFT)</td>
<td>0.225M</td>
<td>0.198M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0%</td>
<td>25</td>
<td>STF (SIFT)</td>
<td>0.399M</td>
<td>0.088M</td>
</tr>
<tr>
<td>CM</td>
<td>14</td>
<td>0%</td>
<td>864</td>
<td>ST (SIFT)</td>
<td>0.315M</td>
<td>0.261M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0%</td>
<td>864</td>
<td>LT (SIFT)</td>
<td>0.178M</td>
<td>0.164M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0%</td>
<td>864</td>
<td>STF (SIFT)</td>
<td>0.315M</td>
<td>0.084M</td>
</tr>
<tr>
<td>Non-indoor datasets</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>DN</td>
<td>500</td>
<td>0%</td>
<td>500</td>
<td>ST (SIFT)</td>
<td>0.482 M</td>
<td>0.422M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0%</td>
<td>500</td>
<td>LT (SIFT)</td>
<td>0.418M</td>
<td>0.084M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0%</td>
<td>500</td>
<td>STF (SIFT)</td>
<td>0.482 M</td>
<td>0.080M</td>
</tr>
<tr>
<td>PB</td>
<td>500</td>
<td>0%</td>
<td>125</td>
<td>ST (SIFT)</td>
<td>0.248M</td>
<td>0.238 M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0%</td>
<td>125</td>
<td>LT (SIFT)</td>
<td>0.236M</td>
<td>0.017M</td>
</tr>
<tr>
<td></td>
<td></td>
<td>0%</td>
<td>125</td>
<td>STF (SIFT)</td>
<td>0.248M</td>
<td>0.015M</td>
</tr>
</tbody>
</table>

8.4.2 Naive matching

We first evaluated the performance of the variants of reduced feature sets via naive matching in the same way as done in Chapter 5, where the nearest neighbor is based on the distance in the feature space to identify feature correspondences between the query image and each training image. A nearest neighbor of a query feature is defined as the feature from the pool of training features, which has minimum distance from that query feature. The training image which gives most of the feature correspondences for
a query image is selected as the best match. Therefore, a suitable distance threshold is needed which ensures many correct feature correspondences and rejects most wrong correspondences between query and training images. Based on this criterion, we used the following empirically determined distance thresholds:

1. **SIFT**: We used the Euclidean distance for the nearest neighbor search and made use of 170 distance threshold.

2. **ORB**: We used the Hamming distance for the nearest neighbor search and made use of 50 distance threshold.

3. **HoG and GIST**: We used the Euclidean distance for the nearest neighbor search for both of them. Both algorithms generate one feature vector for the whole image. Therefore, We simply picked the nearest neighbor as the best match.

As discussed in Chapter 4, we performed an experiment to figure out the distance thresholds for different features. We analysed the correct image matching performance across the DN dataset with varying thresholds for different features. We picked those thresholds which gave the highest image matching performance. These are not optimal thresholds as it is quite hard to find one which gives all correct feature correspondences between two images. However, the used thresholds attempt to reject many wrong feature correspondences while matching two images and have worked well across different datasets in Chapter 4. Therefore, we have used the same thresholds for all features in the current experiment. An argument can be made that the use of a different threshold may give a better performance for a feature on the same dataset. That is true but we suspect that it is unlikely to get a very big difference in the performance (such as more than 5% or 10%) with a different threshold on the same dataset.

The matched scene is the one with the most feature correspondences from the training collection for a query image. In naive matching, either single or multiple feature reduced sets per scene can be used, as we normally look for a 1-1 feature correspondence between a query image and each scene. We therefore used a single reduced feature set per scene and the correct acceptance rate ($C_a$) for all features on the four datasets via naive matching as shown in Figure 8.2. Normal SIFT features are found to give a stable performance over HoG, GIST and ORB features across all datasets. ORB features have not performed well across indoor and PB datasets. We suspect that the reason is the existence of non-geometric transforms, such as exposure or illumination changes in these datasets because images have been captured at different times of the day/night. The DN dataset does not have these transforms because images are mostly captured with small changes in illumination or blur. Mainly, the captured images of each object have scale, rotation or viewpoint changes. That’s why ORB features have done well across DN dataset and have given comparable performance to SIFT features. On the other hand, HoG and GIST features have not performed well across OW and DN datasets because the number of training images is large i.e. over 1000. We observed similar behavior in Chapter 4, where HoG and GIST performed poorly with large training data. Though, these features have done well across CM and PB datasets due to fewer training images. Good features maintain their distinctiveness regardless of the size or type of images and give good performance across different types of datasets.
The results in Figure 8.2 and Table 8.2 show that the reduced feature sets perform almost comparable or even sometimes better than unreduced SIFT features on indoor datasets and offer up to 78% reduction in the overall number of features. \( ST \) and \( LT \) feature variants perform well across all datasets and in some cases can produce significantly smaller feature sets (a maximum of 44% reduction). The \( STF \) and \( LTF \) feature variants underperform on the DN and the PB datasets due to extreme reduction of features (more than 80%). \( STF \) and \( LTF \) cannot represent the scenes effectively and this leads to a poor matching performance. In the indoor datasets, each location has a large number of images; about 50 on average. This results in a feature reduction of up to 59% for \( STF \) which is sufficient to represent every indoor scene effectively. However \( LTF \) again underperforms due to a higher feature reduction (more than 70%). This highlights that at best, the number of training features can be effectively reduced by about half. This also indicates that our scheme is useful for highly redundant image collections and will suit large scale indoor environments.

Table 8.2: Average correct acceptance rate \( (C_a) \) for normal SIFT and reduced features across indoor and outdoor datasets.

<table>
<thead>
<tr>
<th></th>
<th>SIFT</th>
<th>ST</th>
<th>LT</th>
<th>STF</th>
<th>LTF</th>
</tr>
</thead>
<tbody>
<tr>
<td>Indoor datasets</td>
<td>0.833</td>
<td>0.889</td>
<td>0.864</td>
<td>0.8755</td>
<td>0.828</td>
</tr>
<tr>
<td>Outdoor datasets</td>
<td>0.897</td>
<td>0.916</td>
<td>0.920</td>
<td>0.662</td>
<td>0.653</td>
</tr>
</tbody>
</table>

Figure 8.2: The correct acceptance rate \( (C_a) \) for normal unreduced and compact features on all datasets.
8.4.3 Visual Bag of Words

We analysed the performance of reduced feature sets variants via naive matching first as it has a smaller error rate compared to visual BoW. The reduced feature sets did well with naive matching for image matching, and we expected them to do well with standard visual BoW as well.

The final set of experiments was conducted using standard visual BoW with the ntf weighting scheme in the same way as was done in Chapter 6. We first used an inverted index to retrieve 200 images most similar to a query image. The retrieved images were then ranked by visual BoW and the top ranked image was considered the best match for a query image. The following terms were used in these results:

1. **LT, ST, STF and LTF** refer to the cases when every scene was represented by a single feature set.

2. **LT-8, LT-10, ST-20 etc.** refer to the cases when a specific similarity threshold (8%, 10%, 20%) was used in $C_s$ and the scene was represented by multiple feature sets.

3. **SIFT** refers to the case when normal unreduced features were used.

We used a visual BoW with different reduced feature variants, such as LT-20, LT-40, ST-20, STF-20 or ST-80 and reported the best results in Figures 8.3a to 8.3d. Results show that reduced feature sets do not perform consistently as well as normal SIFT features across all datasets. However, for highly redundant collections such as OW and CM, the number of features can be reduced by 50% or more with the STF feature variant with only a small reduction in performance in the case of OW, and slightly improved performance in the case of CM. For less redundant collections however, only a modest reduction in the number of features appear feasible. The results show that feature reduction can be useful with standard visual BoW in some situations but not all and when there are a large number of clusters, some variants do just as well.

8.4.4 Scene clustering ($S_c$)

As discussed earlier, a higher scene clustering rate ($S_c$) may result in a large scene clustering error rate ($E_s$). The high value of $E_s$ will produce bad reduced feature sets which may lead to a poor performance in any classification algorithm. In this section, we analysed the performance of the proposed clustering algorithm with the earlier version of the CS dataset because it contained more scenes though the number of images per scene was 10-15 on average. The error rate $E_s$ is identified manually by checking the images in each group of a scene against some similarity thresholds ($T$) as shown in Table 8.3. An image in a group is considered to be grouped wrongly if it refers to a different part of a scene than other images of that group, which all refer to another part of that scene. The results show that the clustering algorithm groups images with an error rate ranging from 0-8%, which is not very high. Some examples showing the wrong and correct clustering of images are shown in Figures 8.4-8.5.
Figure 8.3: The correct acceptance rate ($C_a$) for unreduced normal SIFT and reduced features on all datasets. Similarity thresholds (T) of 8%, 20%, 40%, 80% are used for images clustering in our experiments and the best results are reported.

Table 8.3: Scene clustering and error rates for the CS dataset with ST reduced features.

<table>
<thead>
<tr>
<th>Reduce Feature Sets</th>
<th>Scene Grouping rate ($S_c$)</th>
<th>Grouping Error rate ($E_s$)</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST-8 (SIFT)</td>
<td>76%</td>
<td>8%</td>
</tr>
<tr>
<td>ST-20 (SIFT)</td>
<td>42%</td>
<td>4%</td>
</tr>
<tr>
<td>ST-40 (SIFT)</td>
<td>17%</td>
<td>0%</td>
</tr>
<tr>
<td>ST-80 (SIFT)</td>
<td>10%</td>
<td>0%</td>
</tr>
</tbody>
</table>
Figure 8.4: Correct grouping example for some scenes of CS dataset.

Figure 8.5: Wrong grouping example for some scenes of CS dataset.
8.5 Conclusion

We investigated the effectiveness of reducing the number of SIFT features for the proposed system in this chapter. The benefit of reduced features is that visual BoW can use a small number of clusters, which will generate small sized histograms for every image hence resulting in quick retrieval of top ranked images against a query image. As we have already seen in Chapter 7, where the use of small number of clusters resulted in an efficient image matching performance compared to large number of clusters.

We had hoped that a significant reduction in the number of features could be obtained with only a small reduction in the matching performance of images. Unsurprisingly, we found that the size of the reduction depended heavily on the collection used. For image collections with many redundant images, the number of features can be reduced by more than 50% with a small improvement in the performance for naive matching. Curiously, an improvement in naive matching does not necessarily carry over to improvements in visual BoW, indicating that the ranking function used is not optimal for reduced features and alternative ranking functions should be investigated. It is hard to determine a reduced feature variant which gives good accuracy and is suitable for visual BoW. Therefore, we have used the reduced features for post-verification along with the verification method in iPoS. The details of the post-verification method based on reduced features will be discussed in Chapter 10.

We discuss indoor image localisation based on 3D models and compare it with 2D image based localisation in the next chapter.
Chapter 9

3D based scene localisation

Note: Some portions of this chapter are taken from (Khan et al., 2013).

3D based image localisation methods estimate the pose of a query image, using 3D models of locations to match a query image. 3D models carry rich information, which can be utilised during localisation. Recently, many research works have used pose estimation to localise images in different environments with few wrong image matches (Li et al., 2010; Lim et al., 2012; Sattler et al., 2011; Snavely et al., 2010). Most of these localisation works use high quality query images captured by digital cameras and experiments have been conducted in large urban environments.

The previous five chapters discussed the 2D image matching approach for indoor localisation. It is not possible to avoid all wrong image matches in 2D image matching. On the other hand, the use of 3D models for localisation is increasing due to fewer wrong matches than 2D image matching approaches (Irschara et al., 2009; Sattler et al., 2011; Skrypnyk and Lowe, 2004; Snavely et al., 2010). In this chapter, we analyse the performance of 3D based localisation in indoor environments with query images captured from different mobile devices with varying quality cameras. The main motivation was to compare 3D based localisation with 2D based localisation for indoor environments and figure out the better approach. So that the better approach can be preferred for indoor positioning systems in the future works. A better localisation approach must offer good efficiency, few wrong matches and a very good correct acceptance rate.

The main contribution of this chapter is the comparison of 3D indoor localisation with 2D indoor localisation on low resolution images and a proposed hybrid matching method for indoor localisation.

9.1 3D models

A purely image based representation only provides a rough position of the camera. However, using a 3D model to represent the scene offers the additional advantage that the full camera pose can be determined, which can be used to obtain exact location information within a place such as the distance of the user from a wall. Recent advancements in structure from motion (SfM) has made it possible to construct 3D models
of environments of any scale effectively with available tools such as Bundler (Agarwal et al., 2011; Snavely et al., 2006) or Visual SfM (Wu, 2011). These 3D models are then used for image localisation based on pose estimation from 2D to 3D feature correspondences identified between 2D features of a query image and features of 3D models. We will refer to 2D features of images simply as features and features of 3D models simply as points in the remainder of this chapter.

The 3D model of an environment is a dense cloud of points obtained from reconstructed cameras as shown in Figure 9.1. Scene reconstruction accuracy is proportional to the number of registered cameras or the number of images successfully registered with a 3D model. The 3D model contains information about the registered cameras, corresponding camera intrinsics and points. Camera intrinsics are the camera specific parameters (focal length, image format and principal point) that produce a given picture. For each point, there is a list of:

1. Corresponding features from the registered images which are used to triangulate that particular point.

2. Views/images in which the point was detected. This is also referred to as “visibility” and can be used to reduce the number of points from 3D models (Li et al., 2010).

![Figure 9.1: Point cloud reconstruction of Trafalgar Square (London) from several thousand photos. The reconstructed cameras are shown as pyramids and one of the input photos taken from approximately the same viewpoint is shown on the top. (Adopted from Snavely et al. (2010) with permission.)](image)

### 9.1.1 Tools

Some of the popular tools which are used to generate 3D models are:

- **Bundler**

  Bundler is a SfM system for unordered image collections which takes a set of images and produces a 3D reconstruction of camera and scene geometry as output (Snavely et al., 2006). It reconstructs the scene using a modified version
of the Sparse Bundle Adjustment package (Lourakis and Argyros, 2009) as the underlying optimisation engine.

- **Autodesk 123D Catch**

  Autodesk 123D Catch was released as a freeware in 2011 to produce 3D models from input images (Autodesk123D, 2012). The company website reports the system to perform effective scene reconstruction, but the software lacks documentation and also does not produce output files containing the generated scene information, such as features of registered images or visibility information of 3D points, which are needed to perform scene localisation.

- **Visual SfM**

  Visual SfM is a graphical user interface application for 3D reconstruction based on SfM (Wu, 2011). The software allows images to be selected in any order and generates 3D models using dense reconstructions. The output files produced by the software contain customised information regarding scene geometry.

- **Agisoft Photoscan**

  Agisoft Photoscan is a 3D reconstruction software, which automatically builds textured 3D models using digital photos of the scene (Agisoft, 2013). The company website reports that system generates good 3D models. However, it is a commercial software and its license needs to be purchased before using it.

**9.1.2 3D localisation**

For 3D localisation, an image localisation system extracts points of all 3D models and stores them. During localisation, the system extracts features from a 2D query image and compares them with 3D points of every 3D model. This results in 2D to 3D correspondences, which are used for pose estimation. During pose estimation, a transformation is applied to points to generate re-projected features, which are then compared with corresponding features of a query image. The number of re-projected features which spatially overlap with corresponding query features are called inliers. The system selects that 3D model which gives the maximum number of inliers against a query image, and the registered 3D model indicates the corresponding location. There are two matching strategies:

1. **Feature to point matching (F2P):** is known as direct matching where features of query images are matched with points of 3D models to identify correspondences (2D to 3D) followed by pose estimation to register a query image. This approach has been used in many works (Arth et al., 2009; Lim et al., 2012; Skrypnyk and Lowe, 2004).

2. **Point to feature matching (P2F):** is known as indirect matching where points of 3D models are matched with the features of query images. This approach compares a subset of points with query features to determine correspondences (3D to 2D) efficiently and then performs pose estimation to register a query image (Li et al., 2010).
F2P matching seems to be a natural choice for matching as it compares query features with points of 3D models, but its efficiency decreases when 3D models contain thousands of points because of an increase in the search time for 2D to 3D correspondences. In one important work, Irschara et al. (2009) addressed this problem by developing a fast location recognition technique. The scale information of each point is used to generate a minimum set of “synthetic views”, which can effectively represent the whole point cloud. Synthetic views are generated from real views, which contribute to the actual point cloud reconstruction using SfM. The concept of spatial distribution of points is then used to reduce the number of points from synthetic and real views, which results in a compressed representation. During online localisation, the system uses this compressed representation to retrieve 2D to 3D correspondences for a query image to compute the pose and a pose is only accepted if has at least ten inliers. The system is reported to reduce the point cloud by 50% and gives image registration performance (correct image matching) of 0.95 and 0.42 in experiments on two outdoor video sequences compared to a pure image based method (no compression at all), which gives a performance of 0.92 and 0.31.

On the other hand, P2F matching makes better use of information, such as the visibility of points and which points are likely to appear together. This information can be used to select a small subset of points for comparison with features of a query image efficiently. Li et al. (2010) used this idea and developed an efficient localisation method based on P2F matching for urban datasets. The system uses a prioritisation scheme to select a representative set of features from 3D models (which are constructed using SfM) and matches them with query images for localisation. The priorities of all points are initially set to be proportional to their visibility. During localisation, the system starts with a subset of points with high priority that covers all cameras. Every point that can be matched against features of the query image increases the priority of other points that can be seen in the same camera. Therefore, the system matches model points to query features in priority order, always choosing the point with the highest priority as the next candidate for matching. The system stops the matching process when enough correspondences are found. The pose is estimated from the correspondences and is accepted only if it has at least 12 inliers. The system was tested on urban datasets with millions of points and registered a query image within 2 seconds on average. The authors report that their system can register more query images than a F2P, vocabulary tree approach (Nister and Stewenius, 2006) and the approach presented by Irschara et al. (2009). The authors also report their system to be about two times more efficient than F2P on three large urban datasets.

Different direct matching approaches have been proposed to increase efficiency during image localisation (Irschara et al., 2009; Lim et al., 2012; Xiao et al., 2011), but P2F still works faster on large urban datasets. This motivated Sattler et al. to develop a F2P approach, which could perform faster than P2F (Sattler et al., 2011). In this technique, a vocabulary tree is developed and 3D points are assigned to visual words during an offline phase where each visual word maintains a list of points. During online localisation, query image features are used to find corresponding visual words. Possible correspondences for query features are then identified in a linear search based on an efficient priority scheme. The number of points which are stored in a visual word gives a good estimate of the matching cost for a query feature. The system processes
query features in ascending order of their matching costs, starting with features whose activated visual words contain only few points. The system stops the search when 100 correspondences are found and estimates the pose. The pose is accepted only if it has at least 12 inliers. The system was tested on the same datasets used by Li et al. (2010). The authors report that their system can register 4-56 more query images on the datasets and is shown up to 60 times faster than the approach presented by Li et al. (2010).

Pose estimation has been used for image matching in many works (Bielicki and Sitnik, 2013; Garg et al., 2011; Li et al., 2012; Sattler et al., 2012; Xiao et al., 2011). However, most works focus on localisation in outdoor or urban environments. The work in this chapter is closely related to Sattler et al. (2011) who used indirect matching to localise the image in urban environments. This work differs from theirs and others because the focus is kept on indoor localisation and results are reported on query images, which were captured from mobile devices having varying quality cameras.

9.2 Building 3D models

The building of 3D models requires capturing of images followed by extensive processing on those images, such as feature computation, image geometry and pose estimations to generate 3D points, which are used by 3D based localisation approaches. On the other hand, 2D based localisation approaches require a 2D database comprising of feature representations for every captured image. Therefore, more processing is required to generate a database of 3D models than for a 2D database.

We generated the 3D models for four indoor locations of the Computer Science building to analyse the indoor localisation performance with the query images taken from low resolution mobile device cameras. We used Bundler for 3D reconstruction, which used detailed SIFT feature matching to find feature correspondences followed by 3D geometry reconstruction (Agarwal et al., 2011; Snavely et al., 2006). It is often difficult to build a 3D model from office scenes due to the lack of textures. We were able to recover a good 3D model only for one room as shown in Figure 9.2. For other indoor places, we did manage to recover partial 3D models as shown in Figure 9.3. We used MeshLab (MeshLab, 2012) to visualise the 3D models. If a 3D based localisation approach does not work well for few indoor places (such as 4), then it probably won’t work well for many indoor places. However, it will be worthwhile to include more indoor places in experiments in the future for a thorough analysis.

Bundler is well known system to generate 3D models because it is freely available for research purposes. We also tried the trial version of Agisoft Photoscan and got slightly better 3D models on the same indoor images compared to Bundler. But we could not use those 3D models because all information was kept hidden by the trial version of the software. Therefore, we suggest to use such commercial software in the future, which can produce 3D models suitable to be used with a 3D localisation approach. This may help in selecting a better system to generate indoor 3D models in the future works.
Figure 9.2: 3D model of one room showing the view from the top. The green dots indicate the recovered cameras.

(a) North direction; a view from the top.  
(b) South direction; a view from the front.

Figure 9.3: 3D models for one corridor in two directions. The green dots indicate the recovered cameras.

9.2.1 Compact model

Each point in the 3D model has a list of associated SIFT features from the registered images. 3D models of large environments may have million of points and each point may have thousands of features, which results in sparse 3D models. The compressed representations of 3D models offer two advantages: a model with reduced points is more discriminative if the point cloud becomes sparser which leads to better localisation; and a smaller memory footprint also has a positive impact on the overall runtime efficiency (Irschara et al., 2009). The goal of this research is to perform indoor positioning in environments of any scale. Therefore, we decided to reduce points from
3D models and analyse the image matching performance with and without reduction. The points are reduced from 3D models based on the work of Sattler et al. (2011). Each point has a list of features seen in different images or registered cameras. We represent each point by averaging the associated SIFT features. Therefore, each point is represented by a single feature, which leads to a compact 3D model for each indoor location.

9.2.2 Points reduction

We use visibility information and propose three ways to further reduce points from the compact 3D models to represent the indoor locations:

1. All included (A):- All points from the compact 3D models are used i.e. no reduction at all.

2. Less-visible included (L):- Only those points from the compact 3D models, which were seen in exactly two images are used.

3. High-visible included (H):- Only those points from the compact 3D models, which were seen in more than two images are used.

9.3 Scene localisation

For scene localisation, we use the direct matching methodology. We put all points from the compact 3D models into a tree data structure and use a nearest neighbor search based on Euclidean distance to identify 2D to 3D correspondences between a query image and 3D models. We propose three approaches to localise a query image against 3D models from these 2D to 3D correspondences and compare the results with 2D image matching approach. All approaches are discussed below:

9.3.1 2D naive matching (NM-2D)

2D Naive matching (NM-2D) is performed in the same way as done in Chapter 5. For NM-2D, we treat all registered images of 3D models as training data. NM-2D extracts 2D features from each training image and stores them in a kd-tree (Moore, 1991) data structure. To classify a query image, NM-2D finds all nearest neighbors of query features from the kd-tree based on Euclidean distance to determine feature correspondences against every training image. A threshold of 170 is used to accept a feature correspondence. This threshold has also been used before in Chapter 5 during image matching and ensures many correct feature matches.

The number of feature correspondences indicate the number of matching features between a query and a training image. The training image with the most feature correspondences is selected as the best match. The matched training image is considered “wrong” if two training images give the same maximum feature correspondences against a query image.
9.3.2 3D naive matching (NM-3D)

NM-3D puts all points from the compact 3D models into a kd-tree and performs nearest neighbor search based on Euclidean distance to find 2D to 3D correspondences between query image and each 3D model. A 2D to 3D correspondence is accepted only if the two nearest neighbors pass the SIFT ratio test with the threshold set to 0.7 (Sattler et al., 2011). The ratio is determined through experiments and is kept the same in all experiments. The ratio ensures finding many correct 2D to 3D correspondences.

The 3D model whose points give the maximum number of similarity correspondences against the query image features is considered the registered indoor location as shown in Algorithm 9. The registered indoor location is considered “wrong” if a similar maximum number of 2D to 3D correspondences are found for a query image against two different 3D models.

Algorithm 9 Algorithm for the NM-3D.

Input: Query Image.
Output: Registered indoor location of the query image.

1: do
2:    for all 3D models of indoor locations do
3:        Compute the 2D to 3D correspondences.
4:    end for
5: return location of the 3D model giving maximum number of 2D to 3D correspondences against the query image.

9.3.3 Pose matching (PM)

Pose estimation is referred to as Perspective-n-Point (PnP) problem and involves the determination of the position and orientation of the camera. It has been a popular way to register a query image with 3D models (Li et al., 2010; Sattler et al., 2011; Skrypnyk and Lowe, 2004). PM determines 2D to 3D correspondences and then computes the pose between a query image and each 3D model with three pose estimation methods one by one as shown in Algorithm 10. The results of three pose estimation methods are compared, and the best result (i.e. the pose giving maximum inliers) is selected for each 3D model. The indoor 3D model giving the maximum number of inliers against a query image is considered the best match. The registered indoor location is considered “wrong” if two indoor locations give the same number of inliers or if “no-inliers” are detected during pose estimation. All thresholds in PM, such as use of top 50 2D to 3D correspondences and use of value of 5 as re-projection error threshold, are chosen empirically after experiments to get a high image registration rate and low number of wrong matches.

Each pose estimation method has its own advantages and disadvantages. Some work for non-planar scenes (Lu et al., 2000) and some work for planar scenes (Ansar and Daniilidis, 2003; Schweighofer and Pinz, 2006). We used three methods to estimate the pose between a query image and 3D model. The purpose of using three pose estimation methods was to find the best possible pose and to increase the chances of getting good
**Algorithm 10** Algorithm for the PM.

**Input:** Query Image.

**Output:** Registered indoor location of the query image.

1: do
2: for all 3D models of indoor locations do
3: Compute 2D to 3D correspondences.
4: Pick top 50 correspondences based on ratios.
5: Estimate pose via RANSAC using ITR, EPnP and P3P pose estimation methods one by one with $E_t=5$; where $E_t$ is the re-projection error threshold.
6: Store the maximum number of inliers.
7: end for
8: return indoor location giving maximum number of inliers.

localisation decisions for each query image. These pose estimation methods are briefly discussed below:

1. **Iterative (ITR):** It is an iterative method based on Levenberg-Marquardt optimisation and requires at least eight 2D to 3D correspondences (Hartley and Zisserman, 2004). The method finds a pose that minimises the reprojection error which is the sum of squared distances between features and points.

2. **Efficient Perspective-n-Point Camera Pose Estimation (EPnP):** It requires at least four 2D to 3D correspondences (Moreno-Noguer et al., 2007). The basic idea is to express $n$ points as a weighted sum of four virtual control points, and to estimate the coordinates of these control points. This method offers an accurate and fast non-iterative solution to the PnP problem.

3. **3-pt absolute pose problem (P3P):** In this method, algebraic and geometric approaches are used to solve the pose problem (Gao et al., 2003). The analytical solution is obtained first via decomposition of equation system followed by the use of a set of formulas to obtain robust solutions for pose.

**Variant**

The PM approach can be made efficient by using only one pose estimation method. We propose a variant, which is called PMv in which we use only one of the above mentioned pose estimation methods at a time with RANSAC i.e. step 5 of Algorithm 10 is only changed.

**9.3.4 Hybrid matching (HM)**

This approach uses both registered images and 3D models to localise a query image. HM approach first uses 2D based naive matching (NM-2D) to identify the indoor location. The location information is then used to select the corresponding 3D model of the indoor location. HM identifies 2D to 3D correspondences between a query image and 3D model to perform pose estimation via three pose estimation methods. The
location is considered “not-found” if no inliers are detected during pose estimation. Otherwise, HM uses the best pose to re-project all matched points, which are then compared with corresponding query features. HM returns the location only if at least 8 re-projected points have almost the same spatial location as corresponding query features. Otherwise, location is considered “not-found”. The working of the algorithm is given in Algorithm 11, which uses a threshold of 170 to check the 2D to 2D correspondences and a threshold of 0.7 to check the 2D to 3D correspondences. $R_t$ is the distance threshold and $T_p$ refers to the minimum number of re-projected points, which overlap with corresponding query features in the algorithm.

<table>
<thead>
<tr>
<th>Algorithm 11</th>
<th>Algorithm for HM.</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Input:</strong></td>
<td>Query Image.</td>
</tr>
<tr>
<td><strong>Output:</strong></td>
<td>Query image is registered or un-registered.</td>
</tr>
<tr>
<td>1: do</td>
<td></td>
</tr>
<tr>
<td>2: Find the best match for the query image via NM-2D to find the location.</td>
<td></td>
</tr>
<tr>
<td>3: Select the corresponding 3D model of the identified location.</td>
<td></td>
</tr>
<tr>
<td>4: Compute 2D to 3D correspondences.</td>
<td></td>
</tr>
<tr>
<td>5: Pick the top 50 correspondences based on nearest neighbor ratios.</td>
<td></td>
</tr>
<tr>
<td>6: Estimate pose via RANSAC using ITR, EPnP and P3P algorithms one by one with $E_t = 5$.</td>
<td></td>
</tr>
<tr>
<td>7: if No inliers from three algorithms then</td>
<td></td>
</tr>
<tr>
<td>8: Pose Failure i.e. image cannot be registered i.e. $R_n$.</td>
<td></td>
</tr>
<tr>
<td>9: else</td>
<td></td>
</tr>
<tr>
<td>10: Select pose with maximum inliers i.e. the best pose.</td>
<td></td>
</tr>
<tr>
<td>11: Re-project all matched 3D points with the selected pose.</td>
<td></td>
</tr>
<tr>
<td>12: Initialise $counter = 1$</td>
<td></td>
</tr>
<tr>
<td>13: for all re-projected points do</td>
<td></td>
</tr>
<tr>
<td>14: if $(\text{reprojected}<em>\text{point} - \text{image}</em>\text{point}) &lt; R_t$ then</td>
<td></td>
</tr>
<tr>
<td>15: $counter++$;</td>
<td></td>
</tr>
<tr>
<td>16: end if</td>
<td></td>
</tr>
<tr>
<td>17: end for</td>
<td></td>
</tr>
<tr>
<td>18: if $(counter &lt; T_p)$ then</td>
<td></td>
</tr>
<tr>
<td>19: Image is not registered i.e. $R_N$.</td>
<td></td>
</tr>
<tr>
<td>20: else</td>
<td></td>
</tr>
<tr>
<td>21: Image is registered i.e. $R_Y$.</td>
<td></td>
</tr>
<tr>
<td>22: end if</td>
<td></td>
</tr>
<tr>
<td>23: end if</td>
<td></td>
</tr>
</tbody>
</table>

All thresholds in HM, such as use of top 50 2D to 3D correspondences and use of value of 8 for $T_p$, are chosen carefully after an experiment to get a high image registration rate and few wrong image matches. In the experiment, we matched the query images captured by a HTC smartphone using HM with different values for these thresholds. The results in Table 9.1 show that use of different number of 2D to 3D correspondences other than 50 with HM results in lower registration rates due to more no-image matches. However, the wrong match rate remains the same for all 2D to 3D correspondences. Similarly, the results in Table 9.2 show that all values of $T_p$
other than 8 with HM result in lower registration rates, more wrong or more no-image matches. Therefore, on balance top 50 2D to 3D correspondences and $T_p=8$ seems a better choice and has been used for HM. A large difference is not observed with different thresholds in Tables 9.1 and 9.2 because query images are compared with the corresponding 3D models most of the time. More variation can be expected, when query images are compared with 3D models belonging to different locations.

Table 9.1: Image registration, wrong match and no image match rates with different top 2D to 3D correspondences using $T_p=8$ with HM.

<table>
<thead>
<tr>
<th></th>
<th>Image registration</th>
<th>Wrong match</th>
<th>No-image match</th>
</tr>
</thead>
<tbody>
<tr>
<td>Top 30 corres.</td>
<td>72%</td>
<td>0%</td>
<td>28%</td>
</tr>
<tr>
<td>Top 50 corres.</td>
<td>75%</td>
<td>0%</td>
<td>25%</td>
</tr>
<tr>
<td>Top 70 corres.</td>
<td>67%</td>
<td>0%</td>
<td>33%</td>
</tr>
<tr>
<td>Top 100 corres.</td>
<td>67%</td>
<td>0%</td>
<td>33%</td>
</tr>
</tbody>
</table>

Table 9.2: Image registration, wrong match and no image match rates with varying values of $T_p$ using top 50 2D to 3D correspondences with HM.

<table>
<thead>
<tr>
<th>$T_p$</th>
<th>Image registration</th>
<th>Wrong match</th>
<th>No-image match</th>
</tr>
</thead>
<tbody>
<tr>
<td>$T_p=4$</td>
<td>90%</td>
<td>2%</td>
<td>8%</td>
</tr>
<tr>
<td>$T_p=8$</td>
<td>75%</td>
<td>0%</td>
<td>25%</td>
</tr>
<tr>
<td>$T_p=12$</td>
<td>48%</td>
<td>0%</td>
<td>52%</td>
</tr>
</tbody>
</table>

On the other hand, the use of different values of $R_t$, such as 5, 10 or 15 was not found to make any difference in the performance of HM because either very many or very few (such as 0-3) re-projected points were close to the corresponding image points. Therefore, we decided to use value of 5 for $R_t$ with HM.

**Variant**

We propose a variant which is called HMc. The variant uses the camera calibration matrix during pose estimation. Camera calibration is the process of estimating the internal parameters of the camera that produced a given picture and should be used during pose computation to ensure the best results. We used the Camera Calibration Toolbox for Matlab (Bouguet, 2010) to perform camera calibration for each mobile device. We printed out a checkerboard picture and took 15-20 images from each mobile device camera to perform calibration using the toolbox.

**9.4 Datasets and performance metrics**

During image matching, the proposed scene localisation approaches match a query image in the following ways:

- **2D based approach**: This approach, used for NM-2D, matches a query image with registered images. The registered image giving maximum feature correspondences against features of a query image is considered the best match.
• **3D based approach:** This approach, used for NM-3D and PM, matches features of a query image with points of 3D models. SfM generates 3D models from the registered images during an offline processing stage, which are used during localisation. In NM-3D, the 3D model giving the maximum number of 2D to 3D correspondences against a query image can be picked as the best match.

For the PM approach, pose estimation is performed on determined 2D to 3D correspondences between a query image and every 3D model. The 3D model giving the best pose (i.e. maximum inliers) against a query image is considered the best match. 3D based approaches try to register a query image with one of the 3D models and registered 3D model indicates the location for a query image.

On the other hand, HM first uses NM-2D approach to find the best matched training image from registered images. The location information of the matched image is used to pick the 3D model of the corresponding location. HM then uses PM to estimate the pose between the query and 3D model via three pose estimation methods and the best pose is used to re-project the matched points. The 3D model is considered to match only if gives at least 8 re-projected points.

A localisation decision is considered correct if the proposed scene localisation approach determines the location for a query image which is the same as of the actual location of that query image. Otherwise, the localisation decision is considered wrong for a query image.

We used the following datasets in the experiments to test the proposed scene localisation methods:-

• **3D models**

I developed the 3D models of four places in our Computer Science building. The statistics for each 3D model are shown in Table 9.3 which indicate that fewer features were obtained for the corridors due to the lack of textures. The less-visible included points represented about 72% of the overall points while the remaining were high-visible included points. All input images for 3D models were captured from the Nikon D3000 DSLR Camera.

Table 9.3: Statistics for our indoor 3D Models

<table>
<thead>
<tr>
<th>Model</th>
<th>Images</th>
<th>#Registered Cameras</th>
<th># of 3D points</th>
</tr>
</thead>
<tbody>
<tr>
<td>Coffee Room</td>
<td>60</td>
<td>56</td>
<td>58985</td>
</tr>
<tr>
<td>Graphics Corridor</td>
<td>15</td>
<td>14</td>
<td>24948</td>
</tr>
<tr>
<td>Graphics Library</td>
<td>63</td>
<td>56</td>
<td>68541</td>
</tr>
<tr>
<td>Office Corridor</td>
<td>60</td>
<td>60</td>
<td>12825</td>
</tr>
<tr>
<td><strong>Total</strong></td>
<td><strong>198</strong></td>
<td><strong>186</strong></td>
<td><strong>165209</strong></td>
</tr>
</tbody>
</table>

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Less-visible included</td>
<td>118950</td>
<td></td>
</tr>
<tr>
<td>High-visible included</td>
<td>46259</td>
<td></td>
</tr>
</tbody>
</table>
• 2D trained data

Training images were required for the NM-2D approach. Table 9.3 shows that a total of 186 images out of 198 were registered with the four 3D models. We selected all registered images as the training dataset, and a total of 53848 features were extracted from training data during experiments.

• Query images

I captured 60 query images for each location from seven different mobile devices. The query images were captured two months after capturing the images used for the 3D model reconstruction. Therefore, some degree of changes in the query images can be expected compared to registered images as shown in Figure 9.4.

Figure 9.4: Sample registered and query images. Noticeable changes can be observed.

9.4.1 Mobile devices

We used relatively low-cost mobile devices in the experiments rather than expensive ones like the iPhone5, the Samsung Galaxy S3 etc for capturing of the query images. The reason for selecting these mobile devices is that these are common in use. The specifications of the mobile devices used are given in Table 9.4.

Table 9.4: Mobile devices specifications

<table>
<thead>
<tr>
<th></th>
<th>Operating system</th>
<th>Type</th>
<th>Camera resolution</th>
<th>Release date</th>
</tr>
</thead>
<tbody>
<tr>
<td>Nokia N95</td>
<td>Symbian</td>
<td>Phone</td>
<td>5 MP</td>
<td>2008</td>
</tr>
<tr>
<td>iPhone 3Gs</td>
<td>iOS 6.0</td>
<td>Phone</td>
<td>3.0 MP</td>
<td>2009</td>
</tr>
<tr>
<td>Galaxy I7500</td>
<td>Android 1.5</td>
<td>Phone</td>
<td>5 MP</td>
<td>2009</td>
</tr>
<tr>
<td>Nokia C3</td>
<td>Symbian</td>
<td>Phone</td>
<td>2 MP</td>
<td>2010</td>
</tr>
<tr>
<td>IDEOS U8150</td>
<td>Android 2.2</td>
<td>Phone</td>
<td>3.15 MP</td>
<td>2010</td>
</tr>
<tr>
<td>Slim S7</td>
<td>Android 2.2</td>
<td>Tablet</td>
<td>3.15 MP</td>
<td>2011</td>
</tr>
<tr>
<td>HTC Wildfire S</td>
<td>Android 2.3</td>
<td>Phone</td>
<td>5 MP</td>
<td>2011</td>
</tr>
</tbody>
</table>

9.4.2 Performance metrics

The following evaluation metrics are used for analysis in this chapter:
$R_c$ refers to the registration rate.

The expected value for this metric ranges from 0-1. A higher value is better for this metric and indicates a high image matching accuracy against query images, which illustrates the success of a proposed scene localisation approach. This metric is the same as correct acceptance rate, but we have used a different name in this chapter because the image registration term is commonly used with 3D based localisation systems (Li et al., 2010; Sattler et al., 2011).

$W_m$ refers to the wrong match rate.

The expected value for this metric ranges from 0-1. A lower value is better for this metric and indicates the number of incorrect image matches against query images, which are made by a proposed scene localisation approach. A low value of $W_m$ and a high value of $R_c$ reflects good image matching performance of the proposed localisation method.

$B_p$ refers to the number of times a pose estimation method produces the best results compared to other methods during pose estimation.

The expected value of this metric ranges from 0-1. A higher value is better for this metric, which indicates the success of a particular pose estimation method with the proposed scene localisation approach. Therefore, it shows which pose estimation method computes the best pose most of the time while localising query images.

For more details about these metrics, please refer to Chapter 4.

9.5 Results

In this section, we first report indoor localisation results of the proposed approaches with all, less-visible and high-visible included points for a comparison with NM-2D approach. Finally, we evaluate all pose estimation methods to identify the method which gives the best pose most of the times during image registration.

9.5.1 Naive matching (NM-3D) analysis

We computed $R_c$ for NM-3D against all mobile devices and compared the results with the NM-2D as shown in Figure 9.5a. The results indicate that the NM-3D gives a very poor localisation performance compared to NM-2D. There is nothing much to differentiate between the performance of NM-3D with all, high-visible, or less-visible included points, as all performed almost the same.

We initially thought that the 0.7 threshold, which was used to determine 2D to 3D correspondences was too strict and resulted in fewer 2D to 3D correspondences between a query image and 3D models. This could be the reason for the poor image registration performance of NM-3D. Therefore, we relaxed the threshold and repeated the experiments for the NM-3D with a threshold of 0.8. The results in Figure 9.5a show that the NM-3D once again underperforms and gives almost the same performance.
The results suggest that the NM-3D may be not suitable for indoor localisation on low cost devices.

The average time to register one query image for all approaches is reported in Table 9.5 which shows that NM-3D is less efficient than NM-2D with all included (A) and less-visible included (L) points. The reason is that these points are more numerous than the number of training image features. Therefore, 3D localisation approaches require more computational time than 2D localisation approaches. Moreover, this computational time will increase with more indoor locations making 3D localisation approaches unsuitable for any indoor localisation system because of slow response time. However, there are ways to improve the efficiency of 3D localisation approaches, such as the use of visual BoW or priority if required (Sattler et al., 2011).

Table 9.5: Average time to register a query image for all approaches in seconds.

<table>
<thead>
<tr>
<th></th>
<th>All included (A)</th>
<th>Less-visible (L)</th>
<th>High-visible (H)</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NM-3D PM</strong></td>
<td>3.55</td>
<td>3.20</td>
<td>3</td>
</tr>
<tr>
<td><strong>HM</strong></td>
<td>9.57</td>
<td>9.25</td>
<td>8.70</td>
</tr>
<tr>
<td><strong>NM-2D</strong></td>
<td>4.95</td>
<td>4.87</td>
<td>4.77</td>
</tr>
</tbody>
</table>

9.5.2 Pose matching (PM) analysis

$R_c$ for PM with all, less-visible and high-visible included points is shown in Figure 9.6. The results indicate that PM works better than NM-3D with an overall improvement of more than 45% against all mobile devices. However, NM-2D still gives better localisation performance and is more efficient than PM as shown in Table 9.5. Pose estimation
is an expensive operation compared to naive matching. Both NM-2D and PM suffer from wrong image matches, such as 2-7% wrong matches for NM-2D and 2-28% wrong matches for PM. PM mainly finds large numbers of wrong matches for low resolution cameras, such as from 2-3.15 MP.

Figure 9.6: The $R_c$ for the PM versus the NM-2D.

High-visible included points perform poorly with PM because these points are fewer in number, which leads to fewer 2D to 3D correspondences most of the time. This leads to poor pose estimation and hence poor image registration performance compared to all included (A) and less-visible included (L) points. The less-visible included points do slightly better than all included points with PM. This seems interesting as it indicates that the removal of high-visible included points does improve indoor localisation performance. The performance improvement of PM is observed to increase with an increase in camera resolution. We suspect that high resolution cameras produce query images of good quality, which leads to reliable 2D to 3D correspondences and hence an effective pose.

We computed $R_c$ for PMv with all included points and the results are reported in Figure 9.7. The results show that PM performs better than PMv. We decided to check the significance of our results by performing one-tailed paired t-test to compare PM with PMv. The data of our test is the registration rates ($R_c$) with seven mobile device cameras. Our null hypothesis is: means of PM and PMv are equal. While our alternative hypothesis is: there is a significant difference i.e. PM performs better than PMv. We have used the default value of alpha ($\alpha$) i.e. 0.05 to reject our null hypothesis. The results in Table 9.6 indicate the rejection of null hypothesis and support our hypothesis that PM performs better than PMv using all included points. Similar results are also observed with less-visible (L) and high-visible (H) included points for PMv and PM.

Table 9.6: $p$-values for PM versus PMv.

<table>
<thead>
<tr>
<th></th>
<th>PMv (ITR)</th>
<th>PMv (EPnP)</th>
<th>PMv (P3P)</th>
</tr>
</thead>
<tbody>
<tr>
<td>PM</td>
<td>0.0001</td>
<td>0.008</td>
<td>0.02</td>
</tr>
</tbody>
</table>

152
Figure 9.7: The $R_c$ for the PMv for all included (A) points with three pose estimation algorithms versus PM.

Figure 9.7 further shows that PMv performs better with the P3P method compared to the remaining methods. This indicates that ITR and EPnP pose estimation methods compute better pose sometimes, but P3P method gives the best pose most of the time. PMv is about 2-3 seconds more efficient than PM in registering a query with a 3D model.

### 9.5.3 Hybrid matching (HM) analysis

The inefficiency of the PM approach was the main motivation to propose HM. We evaluated the HM and HMc approaches and results are shown in Figure 9.8. The results show that HM and HMc underperform compared to NM-2D. However, their wrong match rates ($W_m$) are almost zero. We suspect that the $W_m$ of NM-2D will increase with an increase in the number of indoor locations because it is hard to avoid wrong feature matches between self-similar indoor images. On the other hand, the use of 3D models produces a pose estimation which can reject wrong matches more reliably.

Performance improvement is again observed with higher resolution cameras which makes sense. HM performs best with less-visible included (L) points in the experiments then all included (A) and high-visible included (H) points. PM has similar results. It is unclear why high-visible included points give low registration performance when used along with less-visible included points. We suspect that high-visible included points occur in almost every registered image because similar texture or pattern is used in the office building. Therefore, high-visible included points are not very discriminating, which reduces the image matching performance when used along with less-visible included points. It will be interesting to analyse this behavior of high-visible included points with a large number of indoor 3D models or 3D models generated from different software in the future.
Figure 9.8: The $R_c$ for the HM Vs. the NM-2D. The curves that are higher and are further to the right are better.

The current results show that less-visible included points can be preferred over all included points for image registration in the future, which will not only reduce the overall number of points but will also decrease the overall processing time. The results also indicate that HMc performs comparably to HM with less-visible (L), high-visible (H) and all (A) included points. Therefore, camera calibration does not make a significant difference during image registration.

**HM versus Thresholded NM-2D**

A training image must have a specific number of feature correspondences against a query image in order to be considered the right match. This decreases the overall number of wrong matches but at the same time some correct image matches may be rejected. That is the case when fewer correct feature correspondences exist between a
query image and a corresponding training image.

For more analysis, we decided to use a threshold of 8 with NM-2D i.e. a training image is considered a match with a query image only if there exists at least 8 feature correspondences. The same threshold was also used in HM i.e. $T_p=8$. The results in Figure 9.9 indicate that HM and HMc with all, less-visible and high-visible included points perform better than NM-2D for image registration with mobile devices having 5MP cameras. There is nothing to differentiate between NM-2D, HM and HMc approaches in terms of wrong match rates. To check the significance of our results, we decided to perform one-tailed paired t-test to compare NM-2D with HM and HMc using all, less-visible and high-visible included points. The data of our test is the registration rates ($R_c$) with seven mobile device cameras. Our null hypothesis is: means of NM-2D, HM and HMc are equal. While our alternative hypothesis is: there is a significant difference i.e. NM-2D provides better registration rate than HM and HMc. We have used the default value of alpha ($\alpha$) i.e. 0.05 to reject our null hypothesis. The results in Table 9.7 indicate that null hypothesis cannot be rejected. The test is underpowered due to few observations. But we can say from Figure 9.9 that on balance HM and HMc with all, less-visible and high-visible included points do well for indoor localisation than NM-2D with smartphones having better cameras, such as 5MP.

<table>
<thead>
<tr>
<th></th>
<th>HM (L)</th>
<th>HMc (L)</th>
<th>HM (A)</th>
<th>HMc (A)</th>
<th>HM (H)</th>
<th>HMc (H)</th>
</tr>
</thead>
<tbody>
<tr>
<td>NM-2D</td>
<td>0.29</td>
<td>0.20</td>
<td>0.24</td>
<td>0.27</td>
<td>0.20</td>
<td>0.26</td>
</tr>
</tbody>
</table>

Table 9.7: $p$-values for NM-2D versus HM and HMc.

Figure 9.9: The $R_c$ for the HM versus thresholded NM-2D.

9.5.4 Best pose ($B_p$)

We computed the $B_p$ for HM with less-visible included points, and results are reported in Figure 9.10. The results show that the P3P method performs best and computes the best pose most of the times compared to ITR and EPNP pose estimation methods. We observed similar behavior with the PMv, where P3P pose estimation algorithm produced the best performance.
Figure 9.10: The $B_p$ for the HM less-visible included (L) points.

## 9.6 Conclusion

The proposed visual BoW presented in Chapter 7 worked well but suffered from some wrong image matches. 3D localisation approaches are expected to perform better than 2D approaches. The purpose of this chapter was to perform a comparison between the two types of approaches and identify the better technique, so that the better approach can be used for indoor image matching in the future work.

We were hoping to get a higher image registration rate and fewer wrong image matches with the HM approach due to the incorporation of 3D data. HM gives almost zero wrong image matches but at the same time the image registration rate is low with mobile devices having cameras of low resolutions such as 1-3MP. We suspect that possible reasons of bad performance are, (1) we failed to get effective indoor 3D models because they are difficult to construct for office buildings due to the lack of texture patterns, (2) query images were of poor quality, which produced poor 2D to 3D correspondences between a query and the 3D model, leading to poor pose estimation, and (3) more than one query image might be required for localisation. HM performs better than NM-2D with all less-visible and high-visible included points with 5MP or better resolution cameras. We suspect that high resolution cameras generate a large and reliable set of query features which leads to a good pose estimation. The P3P method is found to perform best and the Nokia N95 device gives the best localisation results in the experiments.

HM performs best with the Nokia N95 mobile device probably due to the better quality of the camera than the Samsung and HTC smartphones tested. The results illustrate that the HM approach is suitable for indoor localisation with less-visible
included points (L) in the presence of good 3D models and mobile devices having a good camera resolution such as 5MP or above, but the problem with the HM approach is that it gives poor image matching performance with mobile devices having low resolution cameras. Therefore for smartphone based indoor positioning systems, 2D approaches seem to be a better choice due to efficiency and good performance on query images captured from a camera of any resolution. However, we suggest a thorough investigation of the feasibility of 3D localisation approaches by generating 3D models with different systems and using high quality smartphone cameras to capture query images.

We discuss our indoor positioning system (iPoS) in the next chapter based on the proposed visual BoW and reduced feature sets, which were presented in previous chapters.
Chapter 10

Indoor Positioning System (iPoS)

Note: Some portions of this chapter are taken from (Khan and McCane, 2012; Khan et al., 2012b).

We presented visual Bag of Words (BoW) based on a voting module and a verification method for indoor scene localisation in Chapters 6 and 7. The reduced features sets were introduced in Chapter 8 for efficient and effective one to one image matching. Indoor image localisation based on 3D models was discussed in Chapter 9 of this thesis. In this chapter, the best 2D based techniques from previous chapters are integrated into one algorithm, which is called “BoWLocator”. BoWLocator is the key building block of the Indoor Positioning System (iPoS) and is used for indoor localisation.

iPoS is based on a client server model where the client is an Android application running on a smartphone while the server runs the BoWLocator algorithm to localise the query photo sent by the Android application. The Android application generates a voice message indicating the current location to its user once it gets the reply from the server. In this chapter, we briefly discuss the interface of the Android application and the BoWLocator algorithm in detail followed by its evaluation.

The contribution of this chapter is the formulation of the BoWLocator algorithm, its analysis using realistic indoor datasets and its comparison with a 3D image based localisation approach across query images of different cameras.

10.1 Android application

The client is an Android application running on an HTC Wildfire S smartphone (Android 2.3). The application can run on any version of Android greater than or equal to 2.3. The application takes a query photo, sends it to the server and waits for the reply. The server uses the BoWLocator algorithm to match the incoming query photo and returns the location information in the form of a text string. Upon arrival of the result, the Android application generates a voice message on the phone indicating the current location to its user as shown in the demonstration video in Appendix B. The application requires Wi-Fi to be turned on in order to establish a connection with the server.
The interface of the Android application is highly accessible because it does not have any buttons on the screen as shown in Figure 10.1a. The user can tap anywhere on the screen to capture and to send the query photo to the server for localisation. The application generates a sound when the user taps on the screen, which indicates that the query photo is captured and is sent to the server for localisation.

(a) Application Interface.  (b) Sending photo to server.

Figure 10.1: The interface of the Android application.

10.1.1 Development

I developed the Android application in Java using the Eclipse tool along with the Android Software Development Kit (SDK). The Android SDK provides built-in classes to capture images or videos from the camera but such classes come up with an interface where the user has to click buttons on the screen many times in order to capture and to store the image.

Therefore, I created my own Java class to capture the image of a location from the smartphone camera, to generate the sound once the user taps the screen to take a photo and to store the captured photo on the phone. The Java program then establishes a connection with the server based on the hyper text transfer protocol (HTTP) and forwards the stored query image to the server for matching.

10.2 BoWLocator

The server uses the BoWLocator algorithm to handle incoming requests from the Android application. The best techniques for the voting module, verification method and reduced features are selected from previous chapters and are integrated to develop the BoWLocator algorithm. 2D image localisation approach is preferred over 3D image localisation approach for BoWLocator for two reasons: firstly the 3D image matching approach matches fewer query images from cameras having resolutions of 1-3MP (shown in Chapter 9), and secondly it is less efficient than 2D localisation.

BoWLocator first uses visual BoW to retrieve ranked images against a query image followed by the use of the voting module, verification method and post-verification method as shown in Algorithm 12. The voting module returns the location if the top three ranked images refer to the same location. Otherwise, it calls a verification method
to find a match. Post-verification is performed after verification if required to confirm the location output produced by the verification method. The main reason for doing post-verification is to further reduce wrong image matches, which are produced from the verification method. For post-verification, BoWLocator extracts the LTF type reduced feature sets from training images and performs scene based naive matching in the same way as done in Chapter 8, the indoor scene giving the maximum number of similarity correspondences against a query photo is considered the best match. A distance threshold of 170 is used to determine similarity correspondences. This threshold was found to give the most correct similarity correspondences.

Algorithm 12 BoWLocator Algorithm.

**Input:** The query photo or the GPS coordinates from the Android application.
**Output:** the location information or data load acknowledgment

1: do
2: if GPS coordinates received from the application then
3: Load the data of the relevant indoor building if required.
4: return “Data load acknowledgment”.
5: else
6: Store the incoming query photo from the Android application.
7: Use inverted index to retrieve the top 200 images.
8: Rank the retrieved images against the query photo.
9: if top three ranked images refer to the same location then
10: return the “Location information”.
11: else
12: Run the fundamental matrix or homography verification method on top 100 ranked images one by one.
13: if verification method cannot make a decision then
14: return “No-location” found.
15: else
16: if verification method finds a match within the top 10 ranked images then
17: return the “Location information”
18: end if
19: Store the location id returned by the verification method i.e. \( id_{verify} \).
20: Perform post-verification via scene based naive matching.
21: Store the location id returned by the post-verification method i.e. \( id_{postverify} \).
22: if \( id_{verify} == id_{postverify} \) then
23: return the “Location information”.
24: else
25: return “No-location” found.
26: end if
27: end if
28: end if
29: end if
On start up, BoWLocator loads the data of a building based on the provided information and starts running. The data of a building consists of:

1. Features of training images i.e. 96D SIFT features.
2. Location annotations of all training images.
3. Inverted index and visual vocabulary. Our visual vocabulary contains 30K words.
4. LTF type reduced feature sets for indoor locations of the building.
5. Location database containing GPS coordinates of indoor environments and is provided by the database creator. This database is used to perform localisation only in the particular building in which the user is present.

Whenever the user starts the Android application, it sends GPS coordinates to the server. To identify the building and load the relevant data, the BoWLocator matches the incoming GPS coordinates with the stored coordinates in the location database. For efficiency, new data is only loaded once the user changes building and is kept in memory as long as the user remains in that building.

To localise an incoming query photo, BoWLocator first uses an inverted index to retrieve the top 200 images against a query photo quickly in the same way as done in Chapter 6. The retrieved images are then ranked using the $ntf$ weighting scheme, which is discussed in Section 6.2.5 of this thesis. BoWLocator then performs verification using the following methods, which were introduced in Chapter 7.

- **Homography verification method**: In this method, the top 10 SIFT correspondences and RANSAC are used to compute potential homographies between query and ranked images. The ranked image must have at least three perspective correspondences based on computed homographies against a query image in order to be considered the best match.

- **Fundamental matrix verification method**: In this method, the closest 20% SIFT correspondences are used to compute the fundamental matrix between query and top ranked images using RANSAC. The ranked image must have at least 20% inliers to be selected as the best match.

We have used both verification methods with BoWLocator. We realised from experiments that the verification method correctly matches a query image with one of the top 10 ranked images most of the time. However, it often finds wrong matches for a query image if a best matched image is found outside the top 10 ranked images. Therefore, the verification method returns the location if a decision is made within the top 10 ranked images and the decision is considered confident. Otherwise, BoWLocator calls post-verification as an extra step to validate the location information produced by the verification method in order to reduce the number of wrong image matches.

The post-verification method compares LTF features of indoor scenes with the query photo features based on naive matching to find the best matched scene and hence the corresponding location. The location output is considered authentic and is returned to
the Android application if both verification and post-verification methods agree on the same location. Otherwise, the location is considered wrong and “No-location” found message is returned to the application. The verification method is found to produce wrong image matches and post-verification method rejects many such wrong matches, hence decreasing the overall localisation errors for iPoS.

We have used thresholds with BoWLocator algorithm, such as use of top 20% SIFT correspondences to compute a fundamental matrix, use of 20% inliers ratio based on fundamental matrix, use of top 10 SIFT correspondences to estimate the homography and calling the post-verification method if a match is found outside the top 10 ranked images, which gives a high correct acceptance rate and fewer incorrect image matches. However, these thresholds can be changed depending upon the application requirements. For example, the focus of this work is to minimise localisation errors based on single query image matching via the use of strict thresholds. However for image retrieval applications, the focus would be to retrieve a pool of similar images based on less strict thresholds, such as use of a lower inlier ratio in the fundamental matrix verification method.

10.3 Datasets and performance metrics

We used the four realistic indoor datasets to evaluate the performance of BoWLocator algorithm. A HTC smartphone camera was used to take all query photos and a Sony digital camera was used to take training images. The datasets collectively contained about 4000 images representing 50 self-similar indoor locations, including 30 corridors, 6 central halls, 12 rooms and 2 atriums.

In these datasets, training images are different from test images. For every test image of a location, there are a number of training images of the same location taken from different viewpoints. To perform image matching, BoWLocator algorithm finds an image match between a test image and the training images. The image match is only considered right if the matched training image refers to the same indoor location as of the test image. Otherwise, the image match is considered wrong for a test image.

These datasets are briefly discussed below. For more details about these datasets, please refer to Chapter 4.

1. **Owheo (OW):** We used 1534 training and 750 test images.

2. **Commerce (CM):** We used 864 training and 234 test images.

3. **Otago Museum (OM):** We used 1045 training and 135 test images.

4. **Dunedin Stadium (DS):** We used 455 training and 75 test images.

5. **Negative samples:** These are images of locations, which are not in the training collection. We used 50 images from the Indoor dataset and 120 images from the Benchmark dataset. Please refer to Chapter 4 for details about these two datasets.

The following evaluation metrics are used:
$C_a$ refers to the correct acceptance rate.

A higher value is better for this metric and indicates a high image matching accuracy against query images, which illustrates the success of BoWLocator algorithm for indoor localisation.

$W_m$ refers to the wrong match rate.

The expected value for this metric ranges from 0-1. A lower value is better for this metric and indicates the number of incorrect image matches against query images, which are made by the BoWLocator algorithm during localisation. A zero or very low value of $W_m$ along with a high value of $C_a$ is desirable with BoWLocator, indicating good performance.

$F_a$ refers to the false acceptance rate.

The expected value for this metric ranges from 0-1. A lower value is better for this metric, which illustrates that BoWLocator algorithm is robust and rejects most query images of a non-mapped environment during the image localisation.

For more details about these metrics, please refer to Chapter 4.

10.4 Results

We used the following two experimental settings:

1. **Combined dataset:** The training images of all datasets (i.e. 50 locations) were treated as a single training collection. The server generated the features, visual vocabulary and inverted index from this single collection. The data was loaded and then used to localise the incoming query photo. Since the incoming GPS coordinates from the Android application did not play any role, this feature was disabled in the BoWLocator algorithm in this configuration.

2. **Single datasets:** The GPS information from the Android application was used to load the appropriate training dataset and the server performed query image matching on the data of a particular building.

There are two benefits of using the GPS information, (1) iPoS becomes scalable because the server does not need to load the data from all buildings, and (2) the localisation performance also increases because it allows the BoWLocator to localise the incoming query image on images of one building only i.e. on a smaller subset of images rather than the big one.

We experimented with several different variants of the BoWLocator. The purpose was to analyse and to find the best variant for iPoS, which could give a high correct acceptance and lower wrong match rates. These variants are as follows:

1. **BoW:** BoWLocator with no voting module, verification and post-verification methods. This is similar to a standard visual BoW in which the top ranked image is considered the best match.
2. **BoW(H):** BoWLocator with the homography verification method only. The post-verification was not performed.

3. **BoW(H+pv):** BoWLocator with the homography verification and the post-verification methods.

4. **BoW(F):** BoWLocator with the fundamental matrix verification method only. The post-verification was not performed.

5. **BoW(F+pv):** BoWLocator with the fundamental matrix verification and the post-verification methods.

We report iPoS performance using the above configurations and BoWLocator variants in the following subsections.

### 10.4.1 Combined versus Single Collections

iPoS performance with all BoWLocator variants is reported in Figure 10.2. The results for **BoW(F)** are not shown as these results were similar to **BoW(F+pv)**. The results show that the BoWLocator variant with the homography verification method i.e. **BoW(H)** does not perform well compared to the case when fundamental matrix verification and post-verification methods both are used i.e. **BoW(F+pv)**. However, a comparable performance is achieved once the post-verification is used with the homography verification method i.e. **BoW(H+pv)**. **BoW(H+pv)** gives almost the same correct acceptance rate \((C_a)\) as **BoW(H)** does, but its \(C_a\) is slightly lower than **BoW(F+pv)**. However, the wrong match rate \((W_m)\) is smaller for **BoW(H+pv)** than other variants and this can be attributed to more “No-location” outputs compared to both **BoW(F+pv)** and **BoW(H)**. The post-verification method reduces the overall rate of wrong image matches for **BoW(H)** from 0.19 to 0.09 and from 0.15 to 0.07 for combined and single dataset configurations respectively but it also slightly reduces \(C_a\). This happens when the post-verification method wrongly disagrees with the correct output produced by the verification method. The results indicate that all BoWLocator variants do well in image matching with single datasets configuration compared to the combined dataset configuration because the search domain is reduced with single dataset configuration.

All BoWLocator variants give a \(C_a\) similar to standard visual BoW, but with fewer wrong image matches across all datasets. The reason is that standard visual BoW selects the top ranked image as the best match but does not attempt to verify this against the query image. All BoWLocator variants performs poorly with the OW dataset. The reason is that 25% of the query images for OW dataset are taken at night. The Owheo building (OW) has many large windows, which significantly change the scene at night due to lighting changes and glass reflections, leading to many image rejections. Some of the sample images taken at night and corresponding images of the same location taken during day time are shown in Figure 10.3.

BoWLocator with the proposed homography verification performs comparably to the fundamental matrix verification method after the introduction of a post-verification method. The average time required to match one image with the query image by the
Figure 10.2: The results for BoWLocator variants used with iPoS during realistic testing. The curves that are higher and further to the right indicate a better performance.

(a) Image at night. (b) Image during day time. (c) Image at night. (d) Image during day time.

Figure 10.3: Sample query images captured at night and corresponding training images of the same location captured during day time in Owheo building.

The homography verification method is 0.03 seconds while the fundamental matrix verification method takes about 0.11 seconds after optimisation, such as use of efficient ways to hold the data matrices or arrays. On the other hand, the average time required to match one query image is 1.27 and 1.16 seconds for \( BoW(H+pv) \) and \( BoW(F+pv) \).
respectively. BoW(H+pv) takes more computation time because it finds a match quite often outside the top 10 ranked images compared to BoW(F+pv) which results in more calls to post-verification method, hence increasing the overall average image matching time. Therefore, homography verification method alone is more efficient than fundamental matrix verification method but this efficiency gain is lost once it is used in a tiered fashion with visual BoW. Both variants offer a good performance than the standard BoW by producing fewer wrong image matches. Moreover, these variants are also efficient, which makes them suitable to be used for our smartphone indoor localisation system.

10.4.2 Voting module analysis

We analysed the performance of the voting module used in BoWLocator. The voting module is found to handle more than 60% of the query photos sent by the Android application with an accuracy of more than 95% across all datasets. We evaluated the BoWLocator performance without a voting module and the comparison results are reported in Figure 10.4.

![Figure 10.4: BoWLocator analysis using single datasets with and without voting module.](image)

The results indicate that the absence of voting module decreases $C_a$ for both BoW(H+pv) and BoW(F+pv). $W_m$ is found to be slightly smaller for the BoW(F+pv) when no voting module is used. In contrast, the voting module not only reduces $W_m$ but also increases $C_a$ for BoW(H+pv). Therefore, experiments show that the voting module not only improves BoWLocator efficiency but also the overall accuracy. We observed similar results with the proposed visual BoW in Chapter 6, where the voting module played an important role in improving the overall system effectiveness.

10.4.3 False acceptance rate

We analysed the false acceptance rate ($F_a$) of iPoS on the Indoor and the Benchmark datasets with all variants of BoWLocator. The results in Figure 10.5 show that BoW(H) does poorly. However, the use of post-verification reduces $F_a$ for BoW(H):

![Figure 10.5: False acceptance rate for BoWLocator.](image)
1. From 0.83 to 0.53 and from 0.61 to 0.43 for Indoor and Outdoor datasets respectively using combined dataset configuration.

2. From 0.78 to 0.38 and from 0.54 to 0.25 for Indoor and Outdoor datasets respectively using single dataset configuration.

Similarly, the post verification also reduces $F_a$ for $BoW(F)$:

1. From 0.44 to 0.33 and from 0.5 to 0.43 for Indoor and Outdoor datasets respectively using combined dataset configuration.

2. From 0.18 to 0.14 and from 0.27 to 0.25 for Indoor and Outdoor datasets respectively using single dataset configuration.

Figure 10.5: $F_a$ for negative samples using combined and single datasets configuration.

The results indicate that post-verification plays an important role in reducing $F_a$ of BoWLocator variants on negative samples. BoWLocator variants perform poorly with the Indoor dataset compared to the Benchmark dataset because query images in the indoor dataset are more similar to training data. We suspect that the main reason for wrong image matches is the incorrect feature matching between query and training images, which results in wrong homography or fundamental matrix computations. A strict threshold could be used to avoid such wrong matches, which could also lower the correct acceptance rate. However, it is still very hard to reject every wrong image match.

10.4.4 BoWLocator versus Hybrid matching

The proposed 3D image based localisation approach, Hybrid matching ($HM$) did not perform well for image matching on query images of low resolution cameras as shown in Chapter 9. We suspected that the proposed visual BoW should provide better image matching performance on query images captured from smartphone cameras of any resolution. Therefore, we decided to evaluate the indoor image matching performance of $BoW(H+pv)$ on query images of different mobile device cameras to check our claim. This dataset is presented in Chapter 9 and the purpose of this evaluation is to compare
the performance of BoW(H+pv), a 2D image based localisation approach with Hybrid matching (HM), a 3D image based localisation approach.

We computed the correct acceptance rate ($C_a$) for BoW(H+pv) against query images captured from different mobile device cameras and compared it with the results of HM approach, which were presented in Chapter 9 of this thesis. The comparison results in Figure 10.6 show that BoW(H+pv) performs better than HM against query images of most of the mobile devices and matches many query images, which supports the claim made in the previous chapter.

![Figure 10.6: BoW(H+pv) versus HM (with less visible; L, high visible; H and all visible; A points).](image)

To check the significance of our results, we also performed one-tailed paired t-test to compare BoW(H+pv) with HM. The data of our test is the $C_a$ for query images using seven mobile device cameras. Our null hypothesis is: means of BoW(H+pv) and HM are equal. While our alternative hypothesis is: there is a significant difference i.e. BoW(H+pv) performs better than HM. We have used the default value of alpha ($\alpha$) i.e. 0.05 to reject our null hypothesis. The results in Table 10.1 indicate the rejection of null hypothesis and confirms that BoW(H+pv) performs better than HM.

<table>
<thead>
<tr>
<th></th>
<th>HM (A)</th>
<th>HM (L)</th>
<th>HM (H)</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoW(H+pv)</td>
<td>0.003</td>
<td>0.008</td>
<td>0.08</td>
</tr>
</tbody>
</table>

We computed the wrong match rate ($W_m$) for BoW(H+pv). The comparison results in Table 10.2 show that HM does well and produces no wrong image matches. On the other hand, BoW(H+pv) does produce some wrong image matches. This makes sense.
because a 3D image based localisation approach can reject wrong image matches more reliably based on pose estimation.

Table 10.2: Wrong match rates ($W_m$) for $BoW(H+pv)$ versus $HM$.

<table>
<thead>
<tr>
<th></th>
<th>C3</th>
<th>3Gs</th>
<th>IDEOS</th>
<th>S7</th>
<th>N95</th>
<th>Samsung</th>
<th>HTC</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoW(H+pv)</td>
<td>0.08</td>
<td>0.07</td>
<td>0.02</td>
<td>0.02</td>
<td>0.05</td>
<td>0.05</td>
<td>0.02</td>
</tr>
<tr>
<td>HM (L)</td>
<td>0.02</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HM (H)</td>
<td>0.02</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>HM (A)</td>
<td>0.02</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

The comparison results between 2D and 3D localisation approaches are interesting. This highlights the requirement of more investigation in the presence of query images captured from cameras of mobile devices having high resolutions, such as 8MP or 10MP, use of other software to generate 3D models to produce effective models, and use of more indoor locations for more thorough comparison of 2D versus 3D based indoor localisations in the future.

10.4.5 iPoS runtime performance

To compute the run time performance of iPoS, we used a set of 30 query images for matching. iPoS made a decision for half of the query images while decisions were not made for the remaining query images. We realised from our experiments that iPoS took 5 seconds on average when the location was found and 14 seconds on average when the location could not be found for query images. These times include times to:

- capture and store the photo on the smartphone. It takes about 1-2 seconds.
- transmit the photo to the server which takes about 1-2 seconds.
- to find the image match using BoWLocator algorithm and takes about 2-15 seconds. The “No location” found is the worst case scenario and algorithm takes the most time in such cases because the query image is compared with all top 50 ranked images one by one.
- transmit location information back to the Android application and takes about 1-2 seconds.
- generate a voice message on the smartphone and takes less than 1 second.

Therefore, there are two possible outputs in iPoS, either “Location found” or “No Location found”. The time for iPoS to match one query image separately for these two scenarios is also shown in Figure 10.7. The results show that most of the time is consumed by BoWLocator algorithm for “No Location” found scenario because query image is compared with top 50 ranked images one by one to make the conclusion. However, in “Location” found scenario, BoWLocator algorithm is efficient and iPoS also works efficiently.
10.4.6 iPoS robustness

iPoS reports the best image matching performance of 96% and the corresponding wrong match rate of 3% with the single dataset configuration. However, for combined dataset configuration the best image matching performance is found to be 93% and wrong match rate is found to be 3%. The decrease in the image matching performance with the combined dataset configuration can be attributed to “No-location” decisions made for a relatively large number of query images.

A robust indoor positioning system based on vision should give a good localisation performance on datasets under challenging conditions. We tested the robustness of iPoS against the query images taken with the following configurations:

1. **Category A:** I made coloured print outs of some of the query photos. I then reused the Android application to capture query photos from these print outs.

2. **Category B:** I opened some of the query photos on the computer screen and took pictures from the Android application.

3. **Category C:** I used portrait instead of landscape to capture some pictures from the Android application.

4. **Category D:** I took few query pictures of the scenes with one or two people in the image. The people pose as objects in the images and change the appearance of the scene.

iPoS showed good performance on such query photos and recognised the query photos successfully. Some of the sample query images of each category are shown in Figure 10.8.

The experiments show that iPoS cannot make a decision when query images are too dark, or when the query photo is damaged while transmitting to the server as shown in Figure 10.9. It is better in such cases not to find a match and to take another query picture because the probability of localising the indoor place from multiple pictures of the same scene from different viewpoints is higher than using a single image. This had
Figure 10.8: Challenging query photos captured to test the robustness of the iPoS.

Figure 10.9: Sample images where a localisation decision was not made by iPoS.
been already shown in experiments in Chapter 7, where the correct acceptance rate of the system increased with the same wrong match rate when more than one query image was used for indoor image matching. Therefore, in case of a “No-location” decision, the application can ask the user to take another query photo to try and identify the current location.

10.5 Conclusion

Blind people require location information during navigation in unfamiliar indoor environments when they feel they are lost. iPoS uses the proposed BoWLocator algorithm and gives few wrong matches, scales well due to the incorporation of GPS data and works robustly in a variety of buildings. The interface of the Android application is designed to ensure maximum accessibility to blind users. Therefore, iPoS can be used by blind people with ease for guidance while they navigate indoors.

The BoWLocator algorithm with the homography verification method performs comparably to the fundamental matrix verification method. $BoW(H+pv)$ should be used in situations where few wrong matches are required. While $BoW(F+pv)$ should be used when efficiency is more important. $BoW(H+pv)$ performs better than the Hybrid matching ($HM$) approach across query images of different cameras, which seems interesting and needs to be further investigated with high resolution cameras and 3D models for a thorough analysis.

The wrong match rate for iPoS is very low, which increases the chances to localise the current place from multiple query images taken from different viewpoints. This already has been shown in Chapter 7, where the correct acceptance rate of the system increases with the use of more than one query image. In cases of “No-location” found messages, the old photo is discarded and iPoS can request the user to capture another picture to localise the current indoor location. A better option will be to instruct a user to turn around for capturing another query photo because it will reduce the chances of similarity between the new and old captured photos. This will enhance the chances to make a localisation decision with a new query photo. As a whole, iPoS uses a single camera image and offers a cost effective solution for indoor positioning based on computer vision in buildings of any scale.

iPoS produces few wrong image matches and generates an output in less than 30 seconds. Therefore, it meets the specifications of our smartphone based indoor positioning system stated in Chapter 2 of this thesis. The interface of Android application is also highly accessible, which makes its easy to use for blind users. However, there are number of things to be done in the future, such as testing with blind subjects or crowd handling before using the iPoS with blind users. We discuss this in the next chapter.
Chapter 11

Conclusion

We envisioned a system at the start of this research work that could run solely on a smartphone and provide an indoor positioning service to its users. However, this PhD project has progressed to the point of having a module for indoor localisation, which runs in a client-server paradigm. The client is a smartphone application, which captures a query image of the indoor location and forwards it to the server. The server runs the proposed algorithm to localise the incoming query photo from the smartphone application. We have described one specific project that reaches toward this ambitious goal:

**Indoor navigation system as a smartphone application.** Such navigation systems first determine the current position and then provide navigation guidance to its users in the buildings. This thesis focused on indoor positioning and resulted in development of Indoor Positioning System (iPoS). We believe that iPoS can be integrated with any indoor navigation system based on a mobile device and the resulting smartphone application will help the blind person to move from one point to the other.

Computer vision technology offers a cost effective solution for indoor positioning compared to infrared, audible sound or inertial sensors due to the use of cheap cameras. However, vision based indoor positioning is challenging in office buildings, which have self-similar locations. Some vision works have addressed the problem of indoor image matching, but experiments are limited to few indoor places. Therefore, my research further investigates image matching in large indoor environments.

This thesis has described several advancements in computer vision techniques, along with analysis and experiments, which are used to develop an indoor positioning prototype application called iPoS. iPoS has been tested on a realistic training dataset having images of 50 different indoor locations such as 30 corridors, 6 central halls, 12 rooms and 2 atriums. iPoS reports the best image matching accuracy of 93% with a wrong match rate of 3% on this dataset. To enable iPoS to function effectively, an algorithm known as BoWLocator has been developed which is used by a server to localise the incoming query image from the smartphone application. BoWLocator uses:

1. **Voting module:** This checks the top three ranked images retrieved by visual Bag of Words (BoW). It simply returns the location information if all three top
ranked images refer to the same location. Otherwise, it calls the verification method.

2. Verification method: This uses a fundamental matrix or homography verification method to match the top 50 ranked images one by one with a query image. If it cannot find a match in the top 50 ranked images, it returns the “no-location” message and the user can take another query photo.

On the other hand, if the verification method finds an image match in the top ten ranked images, the match for the query image is considered confident and the location information is then returned. Otherwise, the verification method calls the post-verification method to validate its location output.

3. Post-verification method: This matches reduced features with the features of a query image to find the best match and hence the location. It returns the location information only if its location output agrees or matches with the location output of the verification method. Otherwise, it returns “no-location” message and the user can take another query photo.

Such a tiered approach is necessary for iPoS to provide good image matching accuracy, few wrong image matches and fast runtime performance, when there are several visually similar locations in the database. The location information is returned by the server and is finally communicated to the user through a voice message by the smartphone application.

This thesis makes the following contributions to the computer vision area:

- **Shorter 96D or 64D SIFT features** described in Chapter 5, offer comparable performance to the normal 128D SIFT features and uses less memory and is faster. The proposed shorter SIFT features can be used in any vision work.

- **Features evaluation** described in Chapter 5, offers a good comparison between well known features and provides guidance regarding their use in vision applications. For example, to get a high image matching accuracy, 96D SIFT features are better. For a good image matching accuracy and fast runtime performance, the ORB features are better.

- **The layered architecture** based on visual BoW described in Chapter 5, offers a reasonable trade-off between image matching performance such as accuracy and efficiency, which makes it suitable for indoor image matching applications.

- **The homography verification method** described in Chapter 6, provides a reasonable trade-off between accuracy and efficiency. Upon comparison with different verification methods, the proposed homography verification method is found to be efficient and offer the best performance for indoor image matching. Therefore, this verification method along with visual BoW can be used for any indoor localisation application.

- **The feature reduction scheme** described in Chapter 8, reduces the number of indoor training features effectively by almost half. The SIFT based reduced
feature sets give a comparable performance to the unreduced SIFT features for image matching on indoor datasets with naive matching. Therefore, the proposed feature reduction approach can be used in any vision work to reduce the number of training features from large scale indoor environments.

- **The indoor 3D localisation** methods described in Chapter 9, introduces techniques to reduce the size of 3D point data, compares 2D localisation with 3D localisation and proposes a hybrid matching approach for indoor localisation. The hybrid matching scheme suits mobile devices, which have 5MP or better resolution cameras and 2D image matching is found to work well with images from all mobile devices. To the best of my knowledge, no vision work has evaluated the performance of a 3D image based localisation approach across query images from different mobile device cameras, such as mobile phones, smartphones and tablets.

- **The BoWLocator algorithm** described in Chapter 10, combines the best techniques from the thesis using a voting module, verification method and reduced features to localise the query images. The proposed vision based algorithm can be used for any indoor localisation task.

- **The realistic indoor datasets** described in Chapter 4, are challenging and can be used to evaluate any vision based indoor positioning system. The existing indoor datasets have one problem as they cover few indoor locations, such as images of 13 indoor locations (Ledwich and Williams, 2004), images of 18 indoor locations (Li, 2006), images of one floor of a building (Kang et al., 2009) or images of 10 indoor locations (Filliat, 2007). Therefore, I have to develop my own realistic datasets in order to evaluate my indoor matching algorithm across a large number of indoor places.

### 11.1 Future work

In this thesis, we conducted research and proposed advancements in several computer vision techniques after experiments and analysis. The best techniques were then used to develop iPoS. iPoS has been found in experiments to provide good localisation success rate with few wrong matches in different buildings. However, this is just an initial step towards vision based indoor positioning as a smartphone application. Each component of iPoS has been developed to the level necessary to successfully provide location information and each of them can be improved further. The contributions of this thesis raise the following issues for future research:

- Due to time constraints, we could not test iPoS with blind subjects. Testing with blind users is essential to validate the method.

- We only tested iPoS with one or two people in a scene, but the number of people in a scene can be more than that. Therefore, handling of a crowd is very important and should be tackled in iPoS. This may require the use of more than one images in such scenarios.
• Indoor places may change over time and the training collection needs to be updated. An incremental approach would allow iPoS to update the training collection with the query images of the scene captured during localisation.

• Indoor images need to be labeled with location IDs and this was done manually in this work. Some systems have been proposed to perform automatic place recognition (Ranganathan, 2012; Wu et al., 2009). Most such systems assume a finite set of place labels, which are learned offline from supervised training data such as kitchen, rooms or corridors. Such systems use a classifier to categorise input images or video stream during runtime. The main problem with such systems is that the image categorisation performance is not good. For example, Ranganathan (2012) in his work correctly labeled 45% of images and remaining images were not labeled. Therefore, a more suitable technique capable of producing about 100% indoor place categorisation needs to be explored and used for automatic labeling of images.

In one work, Snavely et al. (2008) grouped similar images based on the number of features, which are common among them. The initial set of such images refer to different locations and is called a “skeletal set”. The set is later expanded by adding more images and this scheme is used for efficient scene reconstruction. This technique may also be used first to group indoor images and then label them with locations via a classifier, but this technique has been tested on images of cities taken from the Internet.

• iPoS is only tested in indoor environments. It would be a good experiment to evaluate the performance of iPoS on outdoor datasets for analysis and comparison with other techniques.

• iPoS uses a client-server architecture, where the server needs to run continuously to handle incoming localisation requests. We are planning to make a stand alone version of iPoS in the future so that all processing can be carried out on the smartphone, but the components of BoWLocator algorithm will need to be further optimised because smartphones have limited processing power and storage capacity.

• The developed android application in this work suffers from one main limitation. It needs to be searched among the other smartphone applications in order to be launched which is not ideal for blind users. The possible solution is to use some third party application such as Quickdroid1, QuickLauncher2 etc to launch our application. We are planning to explore this in the future and also looking to incorporate the use of voice command to take and to send the query photo to the server in the application’s future version, which will remove the requirement of screen tapping.

---

Bibliography


188


Appendix A

Scale Invariant Feature Transform (SIFT)

The SIFT algorithm uses four stages to generate features from images (Lowe, 2004):

A.1 Scale space extrema

This stage detects all possible keypoints from an image via a scale space construction. In scale space construction, the input image is incrementally convolved with Gaussian filters at different scales ($\sigma$) to produce Gaussian blurred images. The stack of blurred images make up an ‘octave’ and each Gaussian blurred image has some scale value. For the next octave, the Gaussian blurred image that has twice the initial value of scale is picked and is reduced by half to generate Gaussian blurred images at different scales in the same way and this process is repeated. Therefore, this stage first generates octaves for the input image and the image size for each octave is half the previous one. The stage then generates Difference of Gaussian (DoG) images for each octave by taking the difference of successive Gaussian-blurred images as shown in Figure A.1.

The DoG images are edge images and a DoG image is given by:

$$D(x, y, \sigma) = L(x, y, k\sigma) - L(x, y, \sigma) \quad (A.1)$$

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad (A.2)$$

Where $L(x, y, \sigma)$ is the convolution of original image $I(x, y)$ with Gaussian blur $G(x, y, \sigma)$ at scale $\sigma$ and a DoG image $D(x, y, \sigma)$ is computed from the difference of two nearby scales separated by a constant multiplicative factor $k$. An octave corresponds to doubling the value of $\sigma$ and a value of $k$ results in a fixed number of convolved images per octave. I use four octaves with five blurred images in each of them.

The blobs (i.e. local minima/maxima) are identified from DoG images by comparing each pixel in the DoG images to its 8 neighbors at the same scale and 9 corresponding neighboring pixels in each of the neighboring scales as shown in Figure A.2. Finally, the stage selects a pixel value as a candidate keypoint if its value is an extremum among all compared pixels. This keypoint detection step is a variation of one of the blob
Figure A.1: Gaussian blurred images in octaves are on the left side. Images are downsized by factor of 2 in every octave and adjacent Gaussian images in octaves are subtracted to produce DoG images as shown on right.

(Adopted from Lowe (2004) with permission.)

detection methods developed by Linderberg (1998) by detecting scale-space extrema of the scale normalised Laplacian.

Figure A.2: Blob detection by pixels comparison within same and at different scales.

(Adopted from Lowe (2004) with permission.)

In the real world, objects compose of different structures at different scales. Therefore, the motivation for using scale space extrema in SIFT is to detect and compute features from all possible scales. A sample image with detected keypoints after the first stage as an example is shown in Figure A.3.
A.2 Keypoint localisation

The previous stage detects keypoints, but most of them are not stable. Therefore, the current stage rejects unstable keypoints in the following steps:

- **3D Interpolation:** In this step, the interpolation of nearby data determines the position of every candidate keypoint accurately. The process involves the calculation of interpolated location of the extremum by using the quadratic Taylor expansion of the DoG scale-space function $D(x, y, k\sigma)$ with the candidate keypoint as the origin. The Taylor expansion is given by:

$$D(x) = D + \frac{\partial D^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x$$  \hspace{1cm} (A.3)

where $D$ and its derivatives are evaluated at the candidate keypoint by using differences of neighboring keypoints and $x = (x, y, \sigma)^T$ is the offset from this point. The location of the extremum $\hat{x}$ is determined by taking the derivative of this function with respect to $x$ and setting it to zero giving:

$$\hat{x} = -\frac{\sigma^2 D^{-1}}{\sigma x^2} \frac{\sigma D}{\sigma x}$$  \hspace{1cm} (A.4)

If the offset $\hat{x}$ is larger than 0.5 in any dimension then the sample point is changed and the interpolation is performed instead about that point. The final offset $\hat{x}$ is
added to the location of its sample point to get the interpolated estimate for the location of the extremum. Non-converging keypoints are discarded in this step.

- **Low Contrast Points:** This step uses the function value at the extremum \( D(\hat{x}) \) to reject keypoints, which have low contrast i.e. less than 0.03. Note that all thresholds are the same as used by Lowe (2004).

- **Edge Responses:** For stability, this step filters out edge responses. Such keypoints normally have a large principal curvature across the edge but a small one in the perpendicular direction.

The principal curvatures are computed from a 2x2 Hessian matrix \( (H) \), which is computed at the location and scale of the key point. The determinant \( (Det) \), trace \( (Tr) \) of the Hessian matrix and principal curvatures are computed as follows:

\[
H = \begin{bmatrix} D_{xx} & D_{xy} \\ D_{xy} & D_{yy} \end{bmatrix}
\] (A.5)

\[
Tr(H) = D_{xx} + D_{yy}
\] (A.6)

\[
Det(H) = D_{xx}D_{yy} - (D_{xy})^2
\] (A.7)

\[
PrincipalCurvature = \frac{Tr(H)^2}{Det(H)}
\] (A.8)

The derivatives including \( D_{xx}, D_{yy} \) and \( D_{xy} \) are estimated by taking differences of neighboring sample points in horizontal and vertical directions. The keypoints giving principal curvature less than than a value of 12.1 are considered to be selected nearby an image edge and are rejected.

The second stage filters out most of the unstable keypoints. A sample image after the keypoint localisation stage is shown in Figure A.4.

### A.3 Orientation Assignment

The first two stages helped in achieving scale and location invariance. This stage assigns each keypoint one or more orientations based on local image gradient directions to achieve invariance to rotation.

First, the Gaussian-smoothed image \( L(x, y, \sigma) \) at the keypoints scale \( \sigma \) is used to perform all computations in a scale-invariant manner. For an image \( L(x, y) \) at scale \( \sigma \), the gradient magnitude \( m(x, y) \) and orientation \( \theta(x, y) \) are computed using pixel differences:

\[
m(x, y) = \sqrt{(L(x+1, y) - L(x-1, y))^2 + (L(x, y+1) - L(x, y-1))^2}
\] (A.9)
Figure A.4: Keypoints after localisation phase (298 keypoints are retained i.e more than 50% unstable points are rejected).

\[
\theta(x, y) = \tan^{-1} \frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)}
\] (A.10)

The magnitude and orientation for the gradient are calculated for every pixel in a neighboring region around the key point in the Gaussian-blurred image. An orientation histogram (with 36 bins covering 360 degrees) is then formed from the gradient orientations of sample points within this neighboring region. Each sample is first weighted by a Gaussian-weighted circular window centred on the feature location and then added to bins of a histogram.

The peaks in the orientation histogram correspond to dominant directions of local gradients. The highest peak and any other local peak that is within 80% of the highest peak are used to create keypoints with that orientation. Therefore, for a single location with multiple peaks of similar magnitude, there will be multiple keypoints at the same location and scale but with different orientations. The keypoints with locations, orientations, and scale information are stored for processing in the final stage.

A.4 Keypoint descriptor

This stage computes a highly distinctive descriptor for each keypoint. To account for feature orientation, the coordinates of the descriptor and the gradient orientations in a 16x16 window around each keypoint are first rotated relative to its orientation. Each window is then divided into sixteen 4x4 regions. From every region, this stage computes the gradient information and summarises that into 8 bin orientation histograms. The gradients far away from the keypoint are given less weight compared to near ones.
The magnitudes are Gaussian weighted based on distance before adding to the corresponding orientation bins and this is illustrated with a circular window on the left side of Figure A.5. Finally, this stage generates a 4x4 array of orientation histograms each one having 8 values resulting in a highly distinctive 128 dimensional SIFT descriptor as shown in Figure A.5.

![Figure A.5: Building the SIFT descriptor.](Adopted from Lowe (2004) with permission.)

The resulting SIFT descriptor is normalised to unit length to handle illumination differences. SIFT features can robustly identify objects even among clutter and under partial occlusion, because these features are invariant to uniform scaling, orientation and partially invariant to affine distortion and illumination changes.
Appendix B

Demonstration video

A CD-ROM is included, which is an ISO9660 standard volume and contains the demonstration video of the Indoor Positioning System (iPoS).