APPEARANCE-BASED METHODS FOR
OBJECT RECOGNITION AND VISUAL
LOCALISATION FROM HAND-HELD
AND WEARABLE CAMERAS

JOSÉ RIVERA RUBIO

Department of Bioengineering
Faculty of Engineering
Imperial College London

June 29, 2016

A thesis submitted in fulfilment of the requirements
for the degree of Doctor of Philosophy at Imperial
College London and the Diploma of Imperial
College
José Rivera Rubio: Appearance-based methods for object recognition and visual localisation from hand-held and wearable cameras, © London, June 29, 2016

**Supervisor:**
Dr Anil Anthony Bharath
DECLARATION

I herewith certify that this thesis represents my own original work and that I have used no other sources than those clearly referenced.

London, June 29, 2016

José Rivera Rubio

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

Visual localisation and object recognition are key goals of artificial intelligence research that have been traditionally investigated separately. Appearance-based methods can be used to treat both problems from a common perspective. Therefore, the main purpose of this thesis is to explore appearance-based methods in the specific contexts of object recognition and visual localisation from wearable and hand-held devices. Specifically, the contributions of this thesis are as follows:

The first topic of study was the object recognition of grocery products acquired with hand-held and wearable cameras, a use case of particular relevance for the blind and partially sighted people. The main contributions around this topic are a) the SHORT dataset, comprising 100 categories and more than 135,000 images between its training and query sets; and b) an open-source pipeline and complete evaluation of popular bag-of-visual-words (BoVW) techniques when tested against SHORT. The SHORT dataset is novel as it introduces a clear distinction between high quality training images and query images taken in the wild. This is an anticipated scenario in which retailers would acquire images for their online shopping brochures and users would submit images of unpredictable quality for recognition. The performance results of the methods tested demonstrate the challenging characteristics of SHORT.

The second subject of study was indoor localisation from hand-held and wearable cameras. For this topic, the RSM dataset was constructed, containing more than 90,000 video frames along more than 3 km of indoor journeys. An open-source pipeline and evaluation is also contributed in this area. The methods include a selection of custom-created single-frame and spatio-temporal image description methods. These are tested against baseline appearance-based methods such as SIFT and HOG3D and state-of-the-art SLAM. Results show that appearance-based
methods, even in the absence of tracking, can provide enough information to infer location with errors as small as 1.5 m over a 50 m journey. From the methods studied, results suggest that single-frame approaches perform slightly better than spatio-temporal ones.

In third place, I have developed a novel biologically inspired model of artificial place cells based on kernel distance metrics of appearance-based methods between query and database images. Localisation performance was also tested against the RSM dataset, achieving errors as low as 1.4 m over a 50 m trajectory and comparing favourably with the state of the art SLAM.

Finally, I have prototyped an assistive localisation system using wearable or hand-held visual input and tactile feedback to track the localisation of the user over haptic maps. An evaluation of the quality of the tactile feedback using this approach is also provided.
PUBLICATIONS

Some ideas and figures have appeared previously in the following publications:

JOURNAL ARTICLES


PEER REVIEWED CONFERENCE PROCEEDINGS


Listen to advice and accept instruction, and in the end you will be wise.

— Pr. 19:20

ACKNOWLEDGMENTS

First and foremost I would like to express sincere gratitude to my supervisor Dr Anil Anthony Bharath, for accepting me as a PhD student and for his continuous support at every stage of the PhD. I believe I am privileged as his excellence in PhD supervision went beyond than just expert technical advice. He instilled perseverance in me and showed me with his tireless effort that the hard-work and ambition are the best cure against the many insecurities that many PhD students suffer at some point during our studies.

I would also like to thank the members of the Biologically Inspired Computer Vision Group (BICV). Being surrounded by extraordinarily bright people is the richest source of motivation. In particular, I would like to thank Dr Lucas Hadjilucas for his enthusiasm and patience when passing the knowledge on many aspects of systems engineering and servers infrastructure that have been so valuable for me. Also to Dr Ioannis Alexiou, for providing many scientific contributions that helped lay the foundations for my research and for co-authoring many of our publications so far. Also to Christoforos Charalambus, for always being there in the last three years. Special thanks go to Kai Arulkumaran, with whom I share many views on how new technology and modern approaches to scientific research can have a great impact in a PhD and add incredible value to a research group. His enthusiasm and hard-work have been a source of motivation for the late stage of my PhD. Also thanks to all the undergraduate, MSc and visiting PhD students that at some point were part of the BICV group, a source of new and fresh ideas and always willing to lend a hand with my data collection projects. Special within this group is Saad Idrees, a kind
friend and a true example of professionalism, endurance and impeccable work ethic. I also take this opportunity to sincerely acknowledge Dr Riccardo Secoli for his help and guidance at the early stages of my PhD. To him I owe becoming a Linux power user and his advice helped me define many aspects of my PhD and to find the most important priorities in times of struggle.

On the personal side, I am most grateful to my parents, for teaching me the values to achieve success in life. Their integrity, hard-work and love for education have been important pillars of my PhD, adding the motivation, strength and endurance that is needed to accomplish a PhD.

I would like to thank Myriam for always being at my side. For encouraging me to pursue a PhD and always follow my dreams and ambitions, and who has given me unconditional support during these 4 wonderful years in London in which we have achieved so much. It seemed eternal, but we are almost there. Without her, this work would not have been possible.

Finally, I would like to express my gratitude to all my friends that have been supportive throughout the PhD and the writing of this thesis. Special thanks to those friends who also are (or were) PhD students once. They are the best counselling service that can be offered to a doctoral student. A continuous source of support and well-being recipes: Sergio Conde Martin, Aguila Ruiz Sola, Toni Alomar Sitges and Maxi Gandul Galiano.

My research was funded by an EPSRC DTP scholarship and an EPSRC Translation Award. The hosting of the datasets is supported by the EPSRC V&L Net Pump Priming Grant 2013-1.
CONTENTS

1 INTRODUCTION 1
  1.1 Mobile visual assistive apps 2
  1.2 Appearance-based methods 4
  1.3 Biological evidence of place cell visual localisation 6
  1.4 Objectives and thesis structure 8
  1.5 Technical assumptions and limitations of the modelling 11

2 MOBILE VISUAL ASSISTIVE APPS: A DESCRIPTION OF THE PROBLEM AND MOTIVATION 14
  2.1 Introduction 14
  2.2 Where am I? 15
    2.2.1 Related approaches 16
    2.2.2 Visual paths 16
    2.2.3 Visual path descriptions 18
    2.2.4 Pairwise descriptor comparisons 19
    2.2.5 Query descriptor rejection 19
    2.2.6 The $\gamma$ score 20
  2.3 What am I holding? 21
  2.4 Initial experimental results 22
    2.4.1 Navigation 22
  2.5 Discussion & conclusions 24

3 HAND-HELD OBJECT RECOGNITION: SHORT DATASET, BENCHMARK AND THE PROBLEM OF ASSISTIVE RECOGNITION 26
  3.1 Introduction 26
  3.2 SHORT and related datasets 28
  3.3 SHORT technical details 32
    3.3.1 Overview 32
    3.3.2 Image acquisition protocol 33
  3.4 Benchmarks 37
4.6 Results 80
  4.6.1 Localisation error vs ground-truth route positions 80
  4.6.2 Performance summaries 80
4.7 Comparison with SLAM 85
  4.7.1 Description of the experiments 85
4.8 Conclusion 90

5 Modelling hippocampal place cells for visual localisation 94
  5.1 Introduction 94
    5.1.1 Motivation 95
    5.1.2 Summary of contributions 95
  5.2 Background 97
    5.2.1 Biological place cells (BPC) 98
    5.2.2 Appearance-based methods as models of sensory inputs to place cell models 99
    5.2.3 Biologically inspired visual localisation 101
  5.3 A visual input model based on CNNs 102
    5.3.1 Tensor population model 103
    5.3.2 Simple-cell V1 CNN model 106
    5.3.3 Subsampling 108
    5.3.4 Learning CNN weights 109
  5.4 Artificial place cells (APC) 111
    5.4.1 Modelling a single place cell: the tuning curve encoder 111
    5.4.2 Dictionary encoding 112
    5.4.3 Modelling place-cell behaviour 114
  5.5 Creating place fields 116
    5.5.1 Location from APC activations 119
    5.5.2 A generalised regression neural network (GRNN) for sub-APC localisation 120
  5.6 Experiments and results 121
    5.6.1 APC-level localisation 122
    5.6.2 Sub-APC localisation 123
    5.6.3 Parameter tuning 125
    5.6.4 Computational load metrics 125
# Conclusion

## An Assistive Haptic Interface for Appearance-Based Indoor Navigation

## Introduction

1. The problem of navigation
2. Structure of the chapter

## Background on assistive devices: accessible technology

1. The impact of sight loss in navigation
2. Non vision-based solutions for assistive navigation
3. Tactile interfaces for the blind and partially sighted
4. Computer vision for assistive navigation
5. Getting data into a navigation system: crowdsourcing

## System overview

1. The data sources
2. Algorithm choice
3. Interface device

## Visual processing design

1. Appearance-based methods for “live” location inference
2. Localisation using “kernelised” histogram distances

## A tactile interface for a client-server assistive localisation system

1. The Senseg™ App
2. Client-server integration

## Experiments

1. Dataset
2. Experiments on localisation: Live query scenario
3. Blindfolded users with tactile sensing

## Results

1. Blindfolded tactile experiments
6.8 Conclusion 158

7 CONCLUSION AND FUTURE WORK 163

7.1 Summary of contributions 163

7.2 Concluding remarks 164

7.2.1 Appearance-based methods for wearable and assistive applications 164

7.2.2 Artificial place cell model 165

7.2.3 Prototype of an assistive application 166

7.2.4 Datasets 167

7.3 Future work 169

7.3.1 Datasets 169

7.3.2 Appearance-based methods for visual localisation 172

7.3.3 Biologically inspired localisation methods based on place cell models 173

7.3.4 Assistive localisation Apps with visual input and haptic feedback 174

APPENDIX 176

APPENDICES 177

A.1 Algorithm for generating cumulative error distributions 177

A.2 Tensor convolution 178

A.3 LSD-SLAM parameters 182

A.4 A visualisation of the frame distribution with t-SNE 183

A.5 Tactile feedback experiment protocol 187

A.5.1 Context 187

A.5.2 Experiment protocol 187

A.5.3 Informed consent form 190

BIBLIOGRAPHY 192
Figure 1  Illustration of place cells; this figure is from experiments of S.P. Layton in behaving rats. Small coloured circles depict individual biological place-cells having maximum activation (firing rate) compared to all other place cells throughout the navigation space. For instance, yellow circles indicate that the place cell represented by the colour yellow is firing when the rat is at that specific location within the maze. Licensed for use under the Creative Commons Attribution-ShareAlike 3.0 License.

Figure 2  Crowdsourcing indoor journeys (“visual paths”) from multiple users. Users A and B make the same journey at different points in time, but can associate their journeys through storing their visual paths on a server; other users C and D, make different journeys, but again can associate their experiences with each other. The statistical tests reported in this chapter compare the within-path queries and between-path queries, as well as within-path, between-location scores based on image comparisons.
Figure 3  Tests of visual distinctiveness along paths: Distributions for the $\rho$ metric. Locations within a path, illustrating the distribution trends of the $\rho$-metric, all within a single 80 m path, but at different distances either within or outside 50 cm from known query submissions.

Figure 4  The SHORT database contains thousands of query images that form a representative set of examples of smartphone queries containing everyday household or packaged food products.

Figure 5  Fraction of values of $\rho$ exceeding a threshold $T$ in $k$ consecutive database frames.

Figure 6  All the grocery products that compose the SHORT dataset in its final set of 100 categories.

Figure 7  Sample test images. Top row: still-images; bottom row: video frames. Note: the images were cropped to fit the collage, they actually have different resolutions. This query selection contains samples from all the test datasets (see Table 1.)

Figure 8  SHORT training set acquisition set-up

Figure 9  Collage representing the grocery products in the SHORT-30 dataset. This is a selection of items to include cans (shiny), boxes, uneven surfaces, similar shapes, semi-transparent or deformable packaging. These are popular products that are widely available for easy reproducibility and contain snacks, toiletries, medicines, drinks, canned food, dairy products, etc. Figure 6 shows all the products in the final SHORT-100 release.
Figure 10  Visual representation of the detailed performance evaluation of state-of-the-art encoding algorithms using the SIFT descriptor on the SHORT dataset. LLC – locality-constrained linear coding; FV – Fisher vector. The SIFT descriptors can be computed on dense grids with a spacing of $S_x$ pixels or around the SIFT keypoints ($KP$). The last figure in the method identifier indicates the size of the visual vocabulary (256, 500 or 4000 visual words). The red circle indicates the mean average precision across categories. Best viewed in colour.  

Figure 11  LLC-S8-4000 Test. Representative empirical precision and recall curve for a small sample of product classes. Only four classes, including best and worst results, are represented to help visualisation. The test was run with 59,226 queries against the database of 1,080 models. Performance in all categories is summarised in Figure 12.  

Figure 12  BOVW + SVM test. Precision/recall curves for a representative sample of product classes: top four and worst five. The test was run with 59,226 queries against the database of 1,080 models.
Figure 13  SIFT descriptor matching. Analysis of the effect of using different training sets. Figure 13a: *Training set 1* includes all the training images. *Training set 2* contains 18 images of each product taken every 60° at three different elevations. *Training set 3* contains 12 images of each product at different angles but at the same elevation. A sample of *Training set 3* with *prd014: Matzos* is shown in Figure 13b with the % of positive matches per position. This test was performed with 850 queries from the *still-images* set.

Figure 14  Evaluation of the voting metric for classifying sequential video frames. Four videos of *prd003* are submitted as queries and the bars show the distribution of estimates across the different categories of SHORT dataset. In each diagram we can also see which category obtained the majority vote in each case (in red).

Figure 15  Evaluation of sequential video frames: Fraction of correctly matched queries to incorrect matches for videoframe sequences. Note that there is a distribution of incorrect matches across multiple categories as the number of sequential query frames increases.
A sample path (Corridor 1, C1, from Figure 17) illustrating the multiple passes through the same space. Each of these passes represents a sequence that is either stored in a database, or represents the queries that are submitted against previous journeys. In the assistive context, the user at point A could be a blind or partially sighted user, and he or she would benefit from solutions to the association problem of a query journey relative to previous “journey experiences” along roughly the same path, crowdsourced by N users that may be sighted.

Maps of the recording locations.

The stages in processing image sequences from database and query visual paths are illustrated above. This does not show the process behind the estimation of ground truth for the experiments, which is described separately in Section 4.5. Variants of the gradient and pooling operators, quantisation approaches and distance metrics are described in Section 6.4.2.

A maximum projection intensity rendering of 16 pooling regions over space. The x components of descriptor component time series from regions A, B, C and D are shown in Figure 20.
Figure 20: Four (of 144) representative signals acquired from a visual path; these signals encode changes in the red and green channels as a user moves through space. The collection of signal traces at one point in time can be used to build a simple frame-level space-time descriptor: LW-COLOR. The signal amplitudes are spatially pooled temporal and spatial gradient intensities.

Figure 21: The spatial pooling pattern used for single-frame Gabor filtering is based on the regions shown here. These regions were generated by sampling Eq. (10) to create $11 \times 11$ px pooling masks. The masks can be applied to the Gabor filtered video-frame outputs ($9 \times 9$ px) by spatial convolution, followed by sub-sampling the output every 3 pixels. See text for further details.

Figure 22: A “collage” depicting a thumbnail of each corridor of the RSM dataset. From top left to bottom right C1, C2, ..., C6.

Figure 23: Example of a $\chi^2$ kernel produced by hard assignment and using the SF-GABOR descriptors when querying with pass P1 of corridor C2 against a database comprised of passes P2-10.

Figure 24: Diagram illustrating the nature of visual paths and queries. There are different paths recorded in the databases.

Figure 25: Comparison between the error distributions obtained with the different methods. Note the high reproducibility of the performance results. The origin of the variability within each curve is explained in Section 4.5.1.1.
Figure 26  Comparison between the error distributions obtained with the different methods. The results for a random frame test (RANDOM) were introduced as a “sanity check” 82

Figure 27  Estimated location vs. ground truth. Illustrative examples of good/bad location estimation performance. a) Uses the best descriptor and a single-device dataset, b) uses the best descriptor and a cross-device dataset and c) uses the worst descriptor, and a multiple-device dataset. 83

Figure 28  Localisation performance in LSD-SLAM; this shows that in different corridors, the accuracy of LSD-SLAM can change quite significantly. See text for details of the SLAM parameters and the nature of the dataset, but note that these were obtained at an image resolution of 1024 × 576 pixels. At lower resolutions, loss of tracking dominated the experiments. The difference in the x and y axis labelling is because experiments in a) and b) are obtained from two different corridors with different lengths and different number of frames. 87

Figure 29  CDF of SF-GABOR and LSD-SLAM when sampling 10 million random queries from all the error measurements. The width of the curves represent the variability of the result data. The bounds are the maximum and minimum CDF values obtained from the Monte-Carlo sampling. 88
Figure 30  Box-and-whisker plots depicting the errors obtained in two corridors, using either LSD-SLAM or appearance-based matching using SF-GABOR descriptors. The top row corresponds to the appearance-based result. The bottom row corresponds to LSD-SLAM. On each graph, the horizontal positions correspond to different journeys down the same corridor when the remainder of journeys is used as a database of journeys. Each of these positions represents the statistics of 100 random image queries. These graphs suggest that LSD-SLAM and an appearance-based approach are comparable in terms of reproducibility of localisation within the same corridor. Note, however, that much lower spatial resolution (less than 1/4 of the image size) is used for the appearance-based technique than for LSD-SLAM.

Figure 31  Orientation sensitive simple cells which only respond to a bar of certain size and orientation.
Figure 32  This depiction of simple-cell receptive field (RF) model for V1 responses that captures only spatial – rather than temporal – properties in the plane of a captured image, illustrated here for a small patch of pixels. White areas indicate zones where a bright stimulus induces increased firing rate in a single neuron, dark areas represent inhibition of firing, and grey areas have null effect. Note that the centres of the 8 RFs shown here are actually centred at the origin of local image space (indicated by the red circle) in the first layer of the CNN. In a second layer, information is collected from different regions of the local patch and represented as a population code associated with the centre of the circle.

Figure 33  Illustration of the order 3 pooling tensor, visualised using transparency and colours to depict the spatial arrangement of values in the third mode. The 17 pooling regions corresponding to the $11 \times 11 \times 17$ tensor are divided in a) a central region, b) 8 “petals” at a distance $d_1$ from the center with 8 different orientations and c) another 8 “petals” at a distance $d_2$ on the same orientations. Best viewed in the electronic version.
Figure 34  Illustration of the CNN that simulates simple-cell responses and a population code based on neurons in visual area V1 applied to a video sequence captured from a wearable or hand-held camera. In terms of biological complexity, this is quite a crude model: does not include strong non-linearities such as divisive normalisation of phase quadrature responses [19]. Nevertheless, it captures the behaviour of a significant subset of biological cell responses in the V1 area [30]. Sub-sampling in modes \((i_1, i_2)\) is omitted for clarity.  

Figure 35  Firing rates of five BPC recordings covering different place fields of moving rats. Different colors represent different place cells. Adapted from [49].  

Figure 36  Two image frames should have similar representations along the fibre in the order 2 tensor that contains an encoded image sequence. Here, two similar frames are shown with (diagrammatic) representations taken from along the fibre corresponding to particular frames. The elements of the fibre correspond to dictionary terms, and the occurrence of each term is recorded. I used the \(\kappa_{x_2}\) similarity measure that is explained in Section 5.4.3.  

Figure 37  Each curve represents the response – modelling the firing rate – of an individual place cell to position along a path. Using the maximum response as an indicator of location allows a simple decoding of place cell responses, localising a person as being within the “receptive field” of a place cell (coloured horizontal bars).
| Figure 38  | Illustration of first prototypes of place-cell behaviour extraction from image descriptor similarities. | 115 |
| Figure 39  | Single APC tuning curve (raw measurements in red) and a smoothed version (blue trace). The reference frame is located around frame 690. The APC response must be thresholded before using it for accurate position inference. | 117 |
| Figure 40  | The variability of individual artificial place field (APC) responses, displayed using mean and the variability of the response over ensembles. Each line corresponds to the place cell field shape when testing against one specific pass. | 118 |
| Figure 41  | Sub-threshold and supra-threshold regions can be identified by setting a threshold on the amplitude of the $\kappa_{\chi^2}$ similarity measure; the height of the threshold controls the support region (i.e. the place cell width) of supra-threshold region of an artificial place cell. | 118 |
| Figure 42  | Responses from APCs, each colour representing one APC. The APCs are defined every 4 m within a corridor, using ground truth information described in the experimental section. Once defined, these APCs provide two different methods of localisation. In this example, 10 values for $i$ are used for $r_i$. | 119 |
Figure 43: Overview of the training pipeline. The sequences of visual information are processed to generate frame encoding vectors (FEVs). The activation model based on thresholded $\chi^2$ kernel distances between FEVs is used to generate the APCs. A RBF-based regression network is used to learn sub-APC locations from training sequences. The diagram of the neural network is merely illustrative, it does not represent the real architecture used. The represented FEVs are also diagrammatic.

Figure 44: Sub-APC location estimate comparison. Using broad place-cell tuning curves, very accurate localisation can be achieved within a section of corridor. For this corridor and for this journey, absolute localisation errors range from below one metre to 1.49 m. Ground truth is shown in red. In this case, only single-device (Nexus 4) queries were used. The effect of using queries from both devices is shown in Figure 47 and captured by Table 11.

Figure 45: Using the ground truth information acquired with the surveyor’s wheel for the passes in the database, activations from APCs are overlaid onto the floor plan in which video data was acquired. Different colours refer to individual APCs. For this visualisation experiment, 8 APCs were constructed.

Figure 46: Effect of varying the threshold on place cell width, and therefore overlap and average absolute error in metres.
Figure 47  Multi-device sub-APC localisation test. Green represents localisation results when only passes from the Nexus were used for the dictionary learning. Blue represents the localisation results when passes acquired with both Nexus and Glass devices were used for the dictionary learning. In particular, the light blue shade represents the variability of the results for the cross-device scenario for dictionaries created with different combinations of passes from the two devices. The dark-blue line shows the average performance for the cross-device scenario. Note the substantially worse cross-device performance. 127

Figure 48  “By designing for someone with a permanent disability, someone with a situational disability can also benefit.” From Microsoft’s inclusive design program “Independence Day”, stressing the importance of a universal user-centred design [107]. 130

Figure 49  The solid circles indicate the remit of this chapter. I do not suggest that either visual sensing, tactile feedback or knowing one’s position on a map solves the indoor navigation problem. In this chapter, I have deliberately selected one sensing technique, one mechanism of feedback and one inference technique of the many redundant sensors and systems that one would wish to have in a robust navigation device. Evaluating combinations in this combinatorial manner allows redundant and robust systems to be created systematically, and with component-level performance characterisation. 133
Figure 50  Illustration of the usage scenario. The App installed on the user’s tablet submits queries taken from a coupled wearable camera or the tablet’s camera. A server sends location feedback, conveyed via tactile cues over a floor plan scaled to fit onto the device screen. The user is depicted with earphones to illustrate that they could also receive audio feedback; this was not implemented, but see [32] for one suggestion.  

Figure 51  Matching locations by selecting maximum similarity kernel score between query and database frames. Recalling Chapter 4, the scores may be obtained by comparing a BoVW encoding of a current query frame against all previous frames acquired from different journeys having similar start and end points. Because the frames are relatively small, comparisons and descriptor calculation for all frames can be rapid.  

Figure 52  Senseg™ App screen. The yellow outline represents walls. The grey lines form a grid system for relative localisation. The green box identifies the user’s estimated location. The red box depicts the location of the user’s touch, and was used for debugging purposes. The horizontal scale at the bottom indicates relative position in the journey. The camera image is also displayed for debugging purposes.  

Figure 53
Figure 54  In blue, the histogram of drawing $10^6$ samples from a uniform distribution. Overlaying this, in red, the distribution of the users estimated locations when drawing $10^6$ samples from the experiment in random order.  

Figure 55  A comic strip by Jorge Cham from his series “Piled Higher and Deeper” [31]  

Figure 56  Distribution of the BOVW data of the RSM dataset in a reduced 3D space when visualised with t-SNE. Colours refer to different corridors in the dataset. Note that there is some evidence of the locally connected paths in the visual-words space.  

Figure 57  Distribution of the BOW data of the RSM dataset in a reduced 2D space when visualised with t-SNE. It is easy to identify some sets of points that form locally one-dimensional structures. Though they are not always contiguous for one corridor, the concept of visual paths appears at least partially justified.  

LIST OF TABLES  

Table 1  Summary of SHORT test datasets. Still images (ST) and videoframes (VF) acquired by sighted users (SG) or blindfolded (BF).  

Table 2  Recognition performance across the four different subgroups of SHORT. The reduction in quality of match for the blindfolded subgroup is notable.
Table 3  Classification results. Detailed performance evaluation of state of the art algorithms on SHORT. **HA** – hard assignment; **LLC** – locality-constrained linear coding; **FV** – Fisher vector. The SIFT descriptors can be computed a) on dense grids with a spacing of $Sx$ pixels or b) around the SIFT keypoints (**KP**). The last figure indicates the size of the visual vocabulary (256, 500 or 4000 visual words).

Table 4  Dataset comparison. Classification results of baseline performance algorithms on SHORT and other existing datasets.

Table 5  Classification accuracy of different objects for different number of sequential query frames (fr). Accuracy is defined as the number of video sequences correctly classified divided by the total number of sequences queried. Each of these short video sequences contains only one product. The standard deviation (Std) captures the variability of the classification accuracy across videos of the same object when this is computed as the ratio between the correctly and incorrectly classified frames in each video of the same category.

Table 6  A summary of the dataset with thumbnails

Table 7  A summary of the different encoding methods and their relationships to different descriptors. The number of elements of each descriptor is also reported (**Dim**).
Table 8  Summaries of average absolute positional error and standard deviation of positional errors for different descriptor types. \( \mu_\epsilon \) is the average absolute error, and \( \sigma_\epsilon \) is the standard deviation of the error in metres. Top: single-frame methods. Bottom: spatio-temporal methods. 82

Table 9  Cumulative distribution function values against localisation error in metres (\( \epsilon \)). \( P(|\epsilon| \leq x) \), expressed as a percentage. From this table, the best appearance-based method achieves a probability of 90% of localising with an error below 2 m, whilst LSD-SLAM achieves just above 50% accuracy level for that error boundary, and required images 5 times larger. In addition, the performance of LSD-SLAM was significantly worse (compare the minimum performance columns) on some corridors and journeys, bringing the overall average down across the RSM dataset relative to appearance-based localisation. 89

Table 10  Summary of the main properties of the different descriptors used. ST: Spatio-temporal, SF: Single-Frame. 112
Table 11 Absolute error evaluation when using a larger number (40) of APCs of small spatial support (0.61 m), using arg max() to infer spatial position, in contrast to using fewer (16) but larger APCs with substantial overlap and the regression network (sub-APC). The comparison with a state of the art SLAM method (LSD-SLAM) is also included. LSD-SLAM performance is positively affected by the tracking recovery exception, which reduces the error drift by resetting the odometry calculation and the error of the pose-graph optimisation.

Table 12 Summary of the results of tactile feedback experiment. A precision metric can be calculated as prec = \frac{\text{hits}}{\text{estimates}}.
KEY TERMS

Features  A feature is a property of an image that is used to solve a computational task. An example of a feature can be an interesting point, an edge or a shape present in the image.

Keypoints  Also called interesting points, keypoints are a specific type of features that capture special local properties around a coordinate point within the image.

Descriptors  They are descriptions of a visual feature present in the image such as colour, texture or shape. They are related to keypoints as these can be used as locations for the computation of descriptors that measure local properties of the image.

Visual Path  A collection of image frames that are induced by the relative motion of a person in a scene.

Journey  In our context, a visual path that has a start ‘A’ and end ‘B’ points.

Corridor  In the present thesis, a corridor represents a the recording of a segment of a journey that in great proportion traverses a corridor within a building.

Pass  Each of the recording instances of the same corridor in the RSM dataset.

Biological Place Cells  BPCs are a specific type of neuron found in mammals that exhibit an increased firing rate when the subject navigates a previously visited place.

Artificial Place Cells  The computational models of the BPCs presented in this thesis.
INRODUCTION

In recent years, we have witnessed an unprecedented level of presence of technology in our lives. Mobile phones are smarter every day, with computational power that overtakes desktop computers just a couple of years old. At the same time, these devices gain in ubiquity as they extend their functionality to wearable technology, e.g. wearable cameras, wireless earphones or smart watches. The cameras installed in these devices have seen a similar increase in presence, resolution and quality of the lenses and sensors.

The combination of better cameras with improved processing and network connectivity opens the possibilities for computer vision to contribute to diverse applications, some of them having a special role in the inclusivity of people with disabilities and a positive impact in quality of life.

Data is the cornerstone for parameter optimisation and at the same time for applying the learning methods that are increasingly being made in many approaches that rely on artificial intelligence. Within computer vision, data, and in particular annotated data, has driven research very strongly. The availability of annotated datasets has achieved prominence, with the organisation of systematic and objective challenges around object recognition and detection; examples include PASCAL VOC [54] and ImageNet [44]. Through these data collections and open competitions, benchmarks for performance have been established that are now being extended to other fields within computer vision such as localisation (NAVVIS [72] and SLAMbench [113]).

Big data, or the use of massive amounts of structured and unstructured data is helping learning and prediction algorithms improve their performance in challenging scenarios such as object recognition [90], visual localisation [83, 58] and natural language processing (NLP) [157]. At the same time, however, it is
key to keep developing core algorithms and richer models that help solve artificial intelligence problems by gaining a better understanding of their constraints and representational structure. Often the solution is closer than what we think, and biology can provide us with efficient models that solve the problem in an effective way. In particular, in computer vision, models of the visual system have been shown to be effective in key tasks such as object recognition and visual localisation \([98, 109]\).

### MOBILE VISUAL ASSISTIVE APPS

Low vision brings many challenges to an individual, including reduced independence and social exclusion. The World Health Organization (WHO) estimates (2012) that more than 285 million people worldwide suffer from low vision or blindness. Due to changing demographics and greater incidence of disease – e.g. diabetes – blindness and failing sight are increasing in prevalence. The cost to society includes direct health care expenditure, care-giver time and lost productivity. Enabling people with visual impairment to increase participation will help address social exclusion and improve self-esteem and psychological well-being. There is the potential of near-commodity smartphones, backed by appropriate computer vision algorithms and supporting processes, to address this need.

The growth in availability of these camera-equipped smartphones, networks, methods of social networking and crowdsourcing of data offers new solutions to develop assistive systems that could be scaled in performance and capability \([101, 186]\). The services/capabilities that could be offered include:

**Navigation:** Regardless of whether one is outdoors or indoors, navigation in sighted humans relies heavily on the sense of vision \([80, 166]\). When vision is deteriorated or deprived, a person’s ability to navigate – particularly in unfamiliar settings – is greatly diminished. The importance of navigation for visually impaired people features prominently because of its impact
on a person’s independence. Studies have found that nearly half (45\%) of people with visual impairment go out every day and a fifth do not go out more than once a week [47, 141].

A 2012 survey carried out during an accessibility event organised between the Royal National Institute of Blind people (RNIB) and Android London revealed that the most desired mobile application among members of the blind and partially sighted community would be a navigation application with access to important information such as signage or information panels, found mainly in written formats [142]. A combination of visual cues, translated into speech or tactile information, is therefore desirable.

**Shopping:** Other challenges include shopping and product recognition, both in shops and at home. The technology for visual object recognition from mobile devices has arrived for sighted users; the challenges to deployment for visually-impaired users includes a) the existence of accessible label databases that are free from commercial bias; b) changing retrieval algorithms and systems to place more emphasis on strong match confidence; c) techniques for conveying information readily to blind and partially-sighted users.

**Personal Safety:** As a partially sighted user, one is faced with a number of hurdles when undertaking journeys away from a familiar environment, and lack of confidence about the “unseen” can be a significant contributing factor to reduced mobility. Where does the pavement end? Where is the entrance to the bus, and, are there stairs? Are there obstructions at head-height?

In summary, the overarching need is to increase the possibility for independent living; in a hugely visually-oriented built environment, sighted users rely on visual cues, signage, and recognition of structures such as doorways. Can these cues be reliably translated into semantically appropriate information using computer vision? The focus is therefore on the feasibil-
ity of answering two questions with existing technology and using visual cues: “Where am I?” and “What am I holding?”.

**Appearance-based Methods**

In computer vision, image representation methods can be divided in three approaches. Model-based approaches attempt to represent images as a combination of different geometrical shapes, i.e. boxes and circles [21]. In contour-based approaches image representations are defined by the edges of the structures present in them [28]. The last approach and subject of this thesis aims to represent visual information by its appearance.

The appearance of an object or a scene in an image can be represented by a series of views from different distances and angles, although usually humans only need a few to recognise more than 30,000 different objects [21] solely by their appearance, even when this is partially occluded.

The set of image representations generated by appearance-based methods are commonly referred as image features or descriptors since they describe properties of the image. Features can be sub-divided into two groups depending on whether they focus on a point or a small area within the images (local approaches) or whether they represent each image in its entirety (global approaches), taking into account all the pixels in the image. Good feature traits are robustness against noise, rotation and scale or illumination changes.

In this thesis, the interest is on local region features. The process to compute these region features is comprised of two steps. In the first one, features that represent one point in the image due to some particular properties of those points are calculated. These are called interesting points or keypoints. Although keypoints represent interesting points in the image, they can also convey information about scale and orientation to characterise properties of their surrounding local region such as stability. Some of the most widely used keypoint detectors are the Harris corner detector [66], difference of Gaussian (DoG) detector or
scale-invariant feature transform (SIFT) keypoint detector [98]; and maximally stable extremal regions (MSER) detector [103].

Once the keypoints have been detected for an image, the next step in order to compute the region’s features is to perform some image processing algorithms on a patch surrounding the keypoint. The output of this computation is normally stored in a vector, the feature vector or feature descriptor. Descriptors include a considerable amount of information about the region and in this thesis I will show that the algorithms to compute them are suitable for different tasks.

There is an increasing number of feature description algorithms, with most of the research focusing on the invariability of the descriptors with image transformations (rotation, scale and deformations among others). However, two of them are particularly well known, speeded up robust features (SURF) and especially SIFT, nowadays widely and increasingly extended since its main drawback, the computational load (due to the high dimensionality of its descriptor) is being overcome by the recent advances in multi-core CPUs and particularly graphic processing units (GPU) [187]. SIFT seemed a good choice for being a baseline method and has therefore been used as such for performance evaluation throughout this thesis.

The image representations achieved by appearance-based approaches can be used for a multitude of artificial intelligence tasks: object recognition, visual localisation, scene detection, face recognition, human identification and action recognition to name only a few. As mentioned in the previous section, this thesis will cover the first two applications, object recognition and visual localisation, with emphasis in the challenging scenarios of hand-held image acquisition and assistive applications.

My hypothesis is that appearance-based methods can contribute to the solution to more than one type of assistive application. The first, and obvious problem, is that of object recognition. For this type of application, I created a database and performed experiments to test the performance of hand-held cameras to provide accurate object recognition.
A second problem is that of self-localisation. The role of computer vision is perhaps less well appreciated in this role, because a widely held view [179] is that a combination of radio-signal strength indicators (e.g. WiFi, inertial and depth sensors) and inference techniques (e.g. tracking) are both necessary and sufficient components for solving the self-localisation problem indoors. If vision is to be used, the currently accepted solution is based on structure from motion algorithms [51].

To assess the potential contribution of appearance based solutions to localisation, I tested the feasibility of hand-held and wearable visual localisation using the biologically-inspired concept of place cells. Located in regions associated with memory in animals, I showed that place cell behaviour can be modelled by using appearance-based approaches.

For both of these requirements in assistive technology, the performance of appearance-based techniques was characterised when different variants of low-level descriptors were used. For the case of localisation, performance is compared with existing structure from motion algorithms.

BIOLOGICAL EVIDENCE OF PLACE CELL VISUAL LOCALISATION

Place cells are a specific type of neuron found in mammals that exhibit an increased firing rate when the subject navigates a previously visited place.

Place-cell behaviour is usually found by obtaining electrical recordings from several biological neurons (CA1) in specific regions of the hippocampus; by comparing firing rates between neurons and relative to background firing rates, one can infer which neurons are displaying place-cell behaviour. The concept of a biological place cell (BPC) is illustrated in Figure 1. The coloured circles depict locations within a maze in which individual place cells show elevated firing relative to other place cells. Through observing this electrical behaviour over many trials, firing patterns can be decoded, and used as an indication
of where the rat is along the maze: in the simplest case, the BPC which fires maximally is taken as the rat’s approximate location in space.

There is no suggestion that the inference in place cells is purely a binary form of “you are here” type of response. Instead, it is likely that the firing rates of several place cells are combined to more accurately infer position [65], a process that might involve both phase-locked patterns of firing and rate codes [48]. Place-cell selectivity itself is thought to arise from a combination of inputs and cell responses inside and outside hippocampal areas. Rate coding [170] is just one mechanism by which groups of neurons are thought to encode and represent sensory information. A rate code can be interpreted by estimating average firing rates (over trials) within the same neuron. The visual information captured by the eyes of an animal should be seen as only one of the many sensory and internal cues that lead to the spatially selective nature of biological place-cell responses [68]. Nevertheless, in many animals, and certainly in humans and primates, vision is a particularly strong cue as to one’s position [53]. Therefore, another research hypothesis for this thesis is whether place cells can be modelled
with appearance-based methods and be used for inferring locations by only using visual input.

OBJECTIVES AND THESIS STRUCTURE

The main goal of this work is to investigate, formulate, implement and evaluate appearance-based methods that might provide a better representation of the visual inputs produced in the contexts of wearable and hand-held object recognition and visual localisation. During the course of this research, sub-projects were created to focus on specific research hypotheses that contribute towards the main objective. The following chapters of this thesis compile the work carried out in each of these sub-projects.

- **Chapter 2**: In this chapter I present the preliminary work carried out to evaluate the requirements of computer vision Apps in the challenging contexts of hand-held and wearable object recognition and visual localisation, with emphasis in the assistive use case for the blind and partially sighted.

  I provide pilot studies of appearance-based methods for these applications and analyse the research hypotheses and requirements to fulfil them. These studies motivated the work described in subsequent chapters and represent the first prototypes of the methods proposed in this thesis as well as the large benchmarking datasets that were acquired to provide performance evaluations.

  In particular, this pilot work suggests that it is possible to estimate indoor localisation of a user solely by using the input acquired with a smartphone or wearable camera. To support this hypothesis, I define the concept of “visual path”, or user-crowdsourced videos of indoor journeys and their descriptions of visual appearance. Finally, I describe the first analysis and prototype data acquisition for hand-held object recognition using appearance-based methods.
• **Chapter 3**: The hypothesis for the work described in this chapter originates from visual object recognition as an application of camera-equipped smartphones. The ability to recognise objects through photos taken with wearable and hand-held cameras is already possible through some of the larger internet search providers; yet, there is little rigorous analysis of the quality of search results, particularly where there is great disparity in image quality. Chapter 3 describes the development of the Small Hand-held Object Recognition Test (SHORT). This includes a dataset that is suitable for recognising hand-held objects from either snapshots or videos acquired using hand-held or wearable cameras. SHORT provides a collection of images and ground truth that help evaluate the different factors that affect recognition performance. At its present state, the dataset is comprised of a set of high quality training images and a large set of nearly 135,000 smartphone-captured test images of 100 grocery products. In this chapter I also discuss some open challenges in the visual object recognition of objects that are being held by users. I evaluate the performance of a number of popular object recognition algorithms, with differing levels of complexity, when tested against SHORT.

• **Chapter 4**: In this chapter I address the use-case of wearable or hand-held camera technology related to indoor navigation. The main research question is whether it is possible to crowdsourced navigational data in the form of video sequences captured from hand-held or wearable cameras. This work uses the foundation provided in Chapter 2 to test video data for navigational content, and algorithms for extracting that content without using geometric inference techniques (such as simultaneous localisation and mapping, (SLAM)). Tracking is not included in this evaluation as the purpose is to explore the hypothesis that visual content, on its own, contains cues that can be mined to infer a person’s location. This hypothesis
is tested through estimating the positional error inferred during one journey with respect to other journeys along the same approximate path.

The contributions described in this chapter are threefold. First, I propose alternative methods for video feature extraction that identify candidate matches between query sequences against a database of previously acquired sequences. Secondly, I describe an evaluation methodology that estimates the error distributions in position inference with respect to the ground truth. In the evaluation, standard approaches are compared in the retrieval context, such as SIFT and HOG3D, to establish positional estimates. The final contribution is a publicly available database (the RSM dataset) comprising over 90,000 frames of video sequences with positional ground-truth in the form of position along a path. The data was acquired along more than 3 km worth of indoor journeys with a hand-held device (Nexus 4) and a wearable device (Google Glass).

Finally, I describe experimental results showing that image queries against previously acquired visual paths could contribute to positional estimates used in navigation. The error performance using only these appearance-based methods is favourable when compared with a state-of-the-art SLAM method, LSD-SLAM, even without the use of a motion model. The evaluation also yields that single-frame methods work better than spatio-temporal ones in the context of these tests which do not use explicit tracking or self-motion estimation.

• Chapter 5: In the work described in this chapter I used visual information from wearable and hand-held cameras in order to reproduce the rate-coding effect found in mammalian place cells. These models receive the name of artificial place cells (APCs).

I also evaluated the accuracy of localisation of APCs created using different visual descriptors and different place-
1.5 Technical Assumptions and Limitations of the Modelling

Through the different sub-projects presented in this thesis I have made a number of technical assumptions that provide a context to the research and might pose limitations to the modelling. Future work, as we will see in Chapter 7, often addresses these limitations, but I have included them in the Introduction for clarity. I enumerate these assumptions divided per sub-project next.

Hand-Held Object Recognition  In this project, the main assumption is that providing different acquisition set-ups and image qualities for the training and query sets within the SHORT...
dataset provides a more comprehensive insight in the quality requirements for object recognition datasets in challenging acquisition scenarios. The training set is comprised of high quality models of the 100 grocery products, presenting systematic variability in the views to capture multiple angles and elevations. The test set is comprised of a large quantity of unstructured queries (multiple non-calibrated sensors, camera optics, devices, etc.) and provision for assistive testing with the introduction of queries taken by sighted or blindfolded users.

**Visual Localisation from Hand-held and Wearable Cameras** In this sub-project, a number of assumptions were made to facilitate the evaluation of appearance-based methods in isolation and reduce the number of parameters in the different benchmarks. In first place, the version of the RSM released with the thesis did not include people in its sequences and the only occlusions present were cleaning objects and semi-stationary furniture that were present in some sequences and not in the others.

Another assumption in this project was the use of one-dimensional positional ground truth, instead of the customary 6D position (location and pose) that methods from the robotics (SLAM) community use. This simplification was intentional at this stage of the project, as the dataset comprises narrow spaces and contains restricted views (frontal, with low variability in the angle of acquisition). Moreover, the aim was to investigate a solution which is restricted by the ability to convey simple information to a user: how far are they along a planned route. This has proven to be sufficient for the testing of appearance-based algorithms, especially when a comparison against a state-of-the-art SLAM is also provided. At the same time, SLAM is arguably a poorer fit: it does not easily allow crowdsourcing of previous journeys, and comparisons to those journeys. Self-localisation based on a shared memory acquired from previous journeys (by other people), has already been shown feasible using non-visual data sources. Thus, the appearance based method allows
crowdsourcing of shared journeys, a topic I address more fully in Chapter 4.

Apart from the lack of multiple views of the same sequences, crowded spaces and other artifacts such as motion blur were excluded from the study. Tracking algorithms, fundamental in SLAM methods to supply bad image associations, were not used. However, this was a deliberate design choice, as in an assistive context for the blind and partially sighted, SLAM’s localisation and mapping lack relevance unless the current journey can be related to previous passes. The analysis of the appearance-based methods is therefore performed in isolation, although adding tracking is contemplated in future work as we will see in Chapter 4.

**Localisation from place-cell models** The creation of place cells is based on the assumptions that a) there will be sufficient frame rate in the sequences as to create the distinctive concave shape of the place cells via associations from multiple visual paths; and b) the similarity of contiguous frames in the sequence is sufficient to yield a similar description obtained with the appearance-based methods.

The first assumption does not pose a limitation, as the datasets acquired nowadays can easily have an even higher frame rate [162]. The second assumption, might rely up to some extent on the absence of obstacles, as it is on the other hand customary in SLAM research [113]. The experimental work described in Chapter 5 will discuss the use of dense appearance-based methods as a mitigation strategy.

**A prototype of an assistive haptic app for visual localisation** This prototype works as a client-server application that assumes a robust connection between the client and the server. However, just as maps can be downloaded dynamically by using the principle of geofencing or caching (e.g. Google Maps), so too descriptors of previous journeys could be dynamically downloaded for certain regions of a building prior to just before entering a location.
MOBILE VISUAL ASSISTIVE APPS: A DESCRIPTION OF THE PROBLEM AND MOTIVATION

INTRODUCTION

Although the use of computer vision to analyse images from smartphones and wearable cameras is in its infancy, the opportunity to exploit these devices for various assistive applications is beginning to emerge. In this chapter, two potential applications of computer vision in the assistive context for blind and partially sighted users are considered. These two applications are intended to help provide answers to the questions posed in Chapter 1: “Where am I?” and “What am I holding?”.

Taking into account the context of mobile devices and assistive applications, I present the motivation to study appearance-based methods for indoor localisation and object recognition, and describe two pilot studies that lay the foundations for the work described in subsequent chapters:

• First, it is possible to suggest how to go about providing estimates of the indoor location of a user through queries submitted by a smartphone camera against a database of visual paths – descriptions of the visual appearance of common journeys that might be taken with a hand-held or wearable device. My proposal is that such journeys could be harvested from, for example, sighted volunteers. Initial tests using bootstrap statistics do indeed suggest that there is sufficient information within such visual path data to provide indications of: a) along which of several routes a user might be navigating; b) where along a particular path they might be.
• The second pilot presented in this chapter is a study of the need for a new benchmarking database and test set for answering the second question of “What am I holding?”. The database acquisition and evaluation experiments will be discussed in detail in Chapter 3, however in this chapter I will discuss the requirements that are needed for specific mobile context and assistive applications.

WHERE AM I?

Techniques for WiFi localisation are entering mainstream use through, at one level, estimates obtained from the physical locations of WiFi access points, simple measures of signal strength or approaches such as “Walkie-Markie” [152], which use multiple-sensor signatures to infer location. These technologies hold great potential. However, accurate localisation still relies strongly on reasonable accurate motion models, and the collection of other cues, such as accelerometry or gyroscopes [178].

Indeed, no matter how good other sources of information are, few can replace the contextual information of visual inference. During navigation, using natural vision, sighted individuals are able to from one consistent information source: a) recognise their location relative to previous journeys; b) locate entrances and exits; c) detect obstructions; d) recognise people; e) assess human intent; f) identify objects or activities of personal interest.

Invoking computer vision to simultaneously solve all of these tasks is a current challenge. The purpose of the initial work reported in this chapter is to assess the feasibility and accuracy of existing computer vision techniques to meet some of these needs. The primary question addressed in this section relates to the first topic in the list above: can we use computer vision to recognise locations against previous journeys?
Related approaches

In Chapters 4, 5 and 6 a more detailed collection of related methods for indoor localisation is provided with emphasis in the specific context of each chapter. In all of them there are references to simultaneous localisation and mapping (SLAM [50]) and parallel tracking and mapping (PTAM, [85]) as these and derived approaches are the techniques that are near state-of-the-art for monocular robot navigation, allowing geometry of a space to be mapped out dynamically at the same time that self-localisation is achieved. In the assistive context, Pradeep and colleagues successfully applied this to a demonstration for indoor navigation in an assistive device [132].

Several methods of indoor localisation using smartphone-relevant technology have also been the object of study, including RSSI, dead-reckoning, and combinations of techniques that harvest environmental cues [178, 152].

Visual paths

As we will see later in more detail in Chapter 4, methods such as SLAM and PTAM attempt to simultaneously map world geometry and localise a camera within that geometry. The question here is slightly different: we seek to identify where one might be relative to previous journeys taken along the same route, either by ourselves or other people. Thus, I introduce the idea of the visual path, a stream of descriptions captured from visual information as we traverse from location A to location B, or from location C to D. Such streams could be captured from the cameras of other users moving in the same physical space.

The path localisation problem can be split into two distinct tasks. The first is to determine which of $P$ possible paths one is navigating along, and the second is to determine where along a particular visual path one is located. In the context of computer vision, a key question concerns the distinctiveness of information along paths, either as indicators of a particular journey or
as indicators of location along a known journey. Note that localisation with respect to a map is not explicitly attempted – the suggestion is to localise with respect to a journey. In the context of many users, this would appear to be a sensible way to harvest information about locations that might be frequently reconfigured in a manner that would reduce dependence on explicit mapping processes.

Though SLAM and PTAM are strong candidates for assistive techniques, there is also the need to combine mapping with object detection and other types of semantic information. Putting these systems in the category of mapping and localisation, I explore the possibility that rather than mapping out a space, a user might be more interested in merely following a path that has been traversed by others. It is in this long-term, collaborative context that the visual path concept would sit: we wish to allow users to compare their journeys against those of others through these visual paths (see Figure 2).

Figure 2: Crowdsourcing indoor journeys ("visual paths") from multiple users. Users A and B make the same journey at different points in time, but can associate their journeys through storing their visual paths on a server; other users C and D, make different journeys, but again can associate their experiences with each other. The statistical tests reported in this chapter compare the within-path queries and between-path queries, as well as within-path, between-location scores based on image comparisons.
In tracing along different paths, one might ask how distinctive the visual content is along one path relative to the appearance along another. In this pilot study I used a standard keypoint and descriptor type approaches – the SIFT keypoint and descriptor – to describe visual paths captured by users as they walked along indoor environments. The same type of image description will be subject of a dedicated study in Chapter 4.

I first studied the distribution of a similarity metric, $\gamma$, based on a modification of Lowe’s ratio test for discriminating descriptors [98]. The modification takes the form of an $L_\infty$-type normalisation on the distribution of squared Euclidean distances between distinctive descriptors that are close matches between database images along a set of $P$ possible paths $C_p, p = 1, 2, \ldots, P$.

For the preliminary work described in this chapter, sparse SIFT descriptors are used. For a detailed description of how these descriptors, or feature vectors, are computed, refer to Chapter 4.

**Visual path descriptions**

First, consider a number $M_p^{(i)}$ of descriptor vectors, $v_m^{(i)}, m = 1, 2, \ldots, M^{(i)}$ produced from an image, $I_p^{(i)}$, with each vector being of dimension $L \times 1$. These descriptors are stacked into the rows of an $M^{(i)} \times L$ descriptor matrix, $V_p^{(i)}$ associated with image $I_p^{(i)}$. A set of images, $\{I_p^{(i)}\}_{i=1,2,\ldots,N_p}$ is now collected for path $C_p$, and for each of these, a descriptor matrix is produced. A visual path $C_p$ is then encoded by the set of matrices of descriptors, denoted $M_p = \{V_p^{(i)}\}_{i=1,2,\ldots,N_p}$ generated from the set of images taken along that path.

Query images, $J^{(j)}, j = 1, 2, \ldots, N_q$ are now acquired, separately. A particular query image is also mapped to matrix of descriptors $Q^{(j)}$. We wish to know which of the $P$ paths the query image $J^{(i)}$ has been taken on; this is answered by comparing the query descriptor matrix against the set of path descriptors for all paths, $\{M_p\}_{p=1,2,\ldots,P}$.

---

1 Also known as keypoint-based SIFT, or more frequently, just SIFT.
Pairwise descriptor comparisons

Let us first consider the comparison of individual query descriptors, \( v_n^{(j)}, n = 1, 2, \ldots, N^{(j)} \) arising from a single query image. The Euclidean distance metric in \( L \)-dimensional space is widely used in assessing descriptor distances in computer vision. Let \( D^{(i|n)} \) be the \( M^{(i)} \times L \) matrix defined by

\[
D_p^{(i|n)} = \mathbb{1}_{M^{(i)} \times 1} \otimes v_n^{(j)} - V_p^{(i)},
\]

where \( \mathbb{1} \) is a vector of ones, and \( \otimes \) denotes the Kronecker product. Then, the elements along the diagonal of

\[
D_p^{(i|n)} [D_p^{(i|n)}]^T,
\]

are collated into a vector, \( d_p^{(i|n)} \in [0, \mathbb{R}^+]^{M^{(i)}} \) of squared Euclidean distances between the \( n^{th} \) descriptor from a query image and each of the \( M^{(i)} \) descriptors derived from the \( i^{th} \) image along the path \( C_p \).

Query descriptor rejection

Many descriptors in the query image will not be sufficiently distinct to be useful in matching. The distribution of distances contained in vector \( d_p^{(i|n)} \) is used in a first stage filtering for distinctiveness by order-statistic filtering. A query descriptor \( v_n^{(j)} \) is considered suitable for use in assessing similarity between a pair of images only if

\[
d_1^{(i|n)} < \alpha \cdot d_2^{(i|n)}
\]

where \( d_1^{(i|n)}, d_2^{(i|n)}, \ldots \) denotes the sorted elements of the vector \( d_p^{(i|n)} \) in increasing order (the path subscript \( p \) is temporarily suppressed to include the order-statistic of elements). \( 0 < \alpha < 1 \) is set to around 0.7, and any query descriptors that do not satisfy this condition is discarded. This “uniqueness criterion” was chosen by Lowe as the ratio of closest to second-closest neighbours of each descriptor that provides the best ratio of probabilities for correct versus
incorrect matches [98]. All image query vectors are subjected to the same test. Those that pass the test allow an “average” distance based on best matching descriptors to be used to determine how close a single query image is to a single database image. That is, for a single image query

\[
\mu_p^{(i,j)} = \frac{1}{|D|} \sum_{n \in D} d_{(1)}^{(i|n)},
\]

is calculated where \(D\) is the set of query descriptors that pass the distinctiveness test, as described here. Again, note that path subscript \(p\) has been omitted from the right-hand side of this expression to represent the sorted distances.

**The \(\gamma\) score**

I calculated \(\mu_p^{(i,j)}\) across all query images \(J^{(j)}\), \(j = 1, 2, \ldots, N_q\) and all path images \((I^{(i)}_p)_{i=1,2,\ldots,N_p=p=1,2,\ldots,P}\). A score is then defined to produce a measure of similarity \(\gamma^{(i,j)}\) between image pairs \((i,j)\) relative to path \(p\) such that \(0 < \gamma \leq 1\).

\[
\gamma^{(i,j)} = \frac{||\mu_p^{(i,j)}||_{\infty} - \mu_p^{(i,j)}}{||\mu_p^{(i,j)}||_{\infty}}.
\] (4)

This score is calculated between pairs of query and database images, and one may identify two types of categories that these query comparisons fall into. In the first case, the images come from the same path (although query and visual path database are, of course, distinct). In the other case, queries come from different paths.

A second type of score, \(\rho\), was created with a slightly different normalisation criterion based on observing the maximum within-path distance distributions, i.e. for a given path index, \(p\). The behaviour of this score was studied using query images as taken with ground-truth locations, measured with a surveyor’s wheel in preliminary experiments on what it would later be the
RSM dataset described in Chapter 4. Again, probability density estimates of scores are estimated from hundreds of thousands of descriptor comparisons and are represented as $f_\rho$ in Figure 3.

![Figure 3: Tests of visual distinctiveness along paths: Distributions for the $\rho$ metric. Locations within a path, illustrating the distribution trends of the $\rho$-metric, all within a single 80 m path, but at different distances either within or outside 50 cm from known query submissions.](image)

**WHAT AM I HOLDING?**

An increasing use for smartphones involves using visual search in which a photograph taken with the phone is used as a query into a catalogue of database items. Common items include paperbacks (books), compact-disc sleeves and art. A closely related approach is the use of bar codes on items to look up both prices and more detailed product information.

For the visually impaired, bar codes may be difficult to locate, and one would wish to allow recognition on objects and products from different points of view. The quality of a query image might also be below that of a sighted user. For this reason, it is appropriate to assess the ability of visual search algorithms, designed for large-scale categorisation, to perform when the image queries are of low quality, as might occur in poor or variable lighting conditions.
The SHORT database [136] provides such a dataset; and though it might consist of a small sub-sample of the categories in real-world product databases, it is complementary and compliant with other datasets used in computer vision, such as the Pascal VOC database [54] and ImageNet. SHORT includes a query dataset acquired from 30 different smartphone cameras, with varying degrees of resolution, image quality and under two scenarios: sighted and blind-folded. An example of typical queries is shown in Figure 4 below.

The dataset contains a mixture of stills and video clips, including nearly 135,000 video frames and more than 4,000 still images of 100 popular grocery items. Image sizes range from under 100,000 pixels to over 6 megapixels. In Chapter 3 a detailed description of the dataset and a complete evaluation will be provided.

**Initial Experimental Results**

**Navigation**

I acquired a number of visual paths with a mobile phone (Nexus 4) in what was the pilot study for the RSM dataset (http://rsm.bicv.org) described in later chapters. These visual paths take the form of video acquisitions, captured with the phone pointing in the direction of motion, and recording at 30 fps at 1920 × 1080 resolution. The images were then downsampling to a resolution of 192 × 108 pixels. The number of images captured...
Figure 5: Fraction of values of $\rho$ exceeding a threshold $T$ in $k$ consecutive database frames.

along the paths raises the complexity of the image matching problem task: there are typically 2,000 images per path.

For the analysis of the distribution of the scores $\gamma$ and $\rho$, VLFEAT’s [172] implementation of SIFT [98] descriptors has been used.

The use of statistics randomly sampled with replacement, commonly called bootstrap statistics, was appropriate for this study because, for example, in the navigational context, it allows sampling distributions of distances across the whole image database of around 400,000 possible pairings of visual path images.

In the case of the $\rho$ metric, these bootstrapped measurements have revealed the existence of visual distinctiveness between positions that are “close” or “far” along a path from a given query. In Figure 3 I double-filtered the distribution of the $\rho$ values with a one-point moving average. This clearly shows that values of $\rho$ closer to one are useful for discriminating positions belonging to a specific visual path. These results have
motivated the search for a threshold on the values of $\rho$ and the use of consecutive database frames to maximise discriminability, as illustrated in Figure 5.

**DISCUSSION & CONCLUSIONS**

In this chapter I have presented the preliminary studies that laid the foundations for the work described in the remainder of this dissertation. The motivation was the same for both navigation and object recognition applications: mobile, appearance-based solutions with an emphasis in the assistive case. As we will see, the results presented in the following chapters are also applicable to sighted users. However, the design of the experiments always had the blind and partially sighted user in mind, as I believe that all serious design should be inclusive and therefore usable for both sighted and visually impaired.

There are several conclusions to the pilot work reported here. First, in the navigation context, there is an opportunity to use information from visual paths to provide an indication of which path a user might be on relative to previous journeys. Although this study is limited to early findings with basic, standard techniques; it does indeed indicate that distinctive information can be harvested from visual paths with great ease. For example, the resolutions of the images used in Section 2 contained only 1% of the pixels in the captured images! Yet, decisions on $\rho$ do seem to allow reasonably accurate estimates of where one is likely to be along a path, subject to appropriate verification being performed, perhaps using higher resolution images. With extra processing to perform geometric verification of match locations along the path, the idea of mapping images to a location looks quite feasible.

In the navigational context, the possibility of obscured views has not been considered, either during path collection or query collection. However, the density of the queries is also low relative to the number of queries that would normally be taken. For example, at a normal walking rate, one could easily collect
more than 10 frames within 1 metre. Such an image sampling rate would give more opportunity to capture unobscured visual patches along a path. The caveat is that one would have to include modules for recognising obstructions or moving objects, such as people, within the frame, and remove query descriptors at spatial scales that would include such regions. Since a key reason for incorporating computer vision into navigational aides would be to detect path obstructions and hazards, Chapter 4 introduces a set of benchmarks using a collection of standard and custom “dense” descriptors that seem to be specially well-suited for this task.

In the context of hand-held objects from the SHORT database, a realistic database would be expected to have thousands of products. The category depth of SHORT is more appropriate to home use by a single user. I have presented the challenges of object recognition for the blind a partially sighted users and have introduced the SHORT database as a necessary benchmark for recognition algorithms that tackle hand-held, wearable and/or assistive applications. In Chapter 3 I will evaluate the performance of baseline descriptor (SIFT) and bag-of-visual-words (BOVW) methods and establish retrieval comparisons that justify the need for a more challenging dataset that reflects hand-held and wearable vision.
The ubiquity of smartphones with high quality cameras and fast network connections will spawn many new applications. One of these is visual object recognition, an emerging smartphone feature which could play roles in high-street and online shopping, price and product comparisons and similar uses. There are also potential roles for such technology in assistive applications, such as for people who have visual impairment. In the previous chapter I briefly introduced the Small Hand-held Object Recognition Test (SHORT), a new dataset that aims to benchmark the performance of algorithms for recognising hand-held objects from either snapshots or videos acquired using hand-held or wearable cameras. SHORT provides a set of images and ground truth that help assess the many factors that affect recognition performance. SHORT is designed to be focused on the assistive systems context, though it can provide useful information on more general aspects of recognition performance for hand-held objects. In this chapter, I will describe the present state of the dataset, comprised of a small set of high quality training images and a large set of nearly 135,000 smartphone-captured test images of 100 grocery products. In this version, SHORT addresses another usage context not covered by traditional datasets, in which high quality catalogue images are being compared with variable quality user-captured images; making the matching more challenging in SHORT than other datasets. Images of similar quality are often not present in “database” and “query” datasets, a situation that can be encountered in commercial applications.
There are several motivations for the SHORT dataset; one lies in the emerging application of assistive systems which can help people with visual impairment to obtain information about objects in real-world settings. A common usage scenario might involve holding objects whilst either shopping or using items in the house. The familiar platform of camera-equipped smartphones makes image-based query a natural choice for this context. Connected, wearable cameras are, of course, another option.

Image recognition with hand-held phone and objects presents very particular challenges, as the variability of viewing conditions (lighting, point of view, etc.) is large. Using Internet-trawled images against which to perform the query is one approach, but it can be expensive if one requires a large number of server-side object-camera poses to guarantee a good quality match. Barcodes are not always easily located by users, and may also be vendor-specific.

The purpose of the SHORT dataset is to provide a database to test retrieval and object recognition when querying against curated databases of high-quality images. This is because the quality and provenance of product records is very important for assistive applications. The best source of product records is likely to be the manufacturer of an item. Such records are held in databases that are carefully managed (curated) and where multiple views are captured for web-based catalogues, marketing brochures, and websites, forming a standard part of the product manufacturing processes.

The number of catalogue items – distinct products, or objects – in SHORT is currently 100, but the plan is to expand it over time. To some extent, I compensated for this by having a large number of query images – a total of 134,524 – taken by as many as 30 mobile phones and including acquisitions from both blindfolded and sighted users. The number and nature of the queries allow SHORT to be used to design and test recognition systems that must place a guarantee on being able to return a correct match. For example, a query image might have to be rejected as being of too low quality to provide a definitive
match, either because of visual ambiguity or poor image quality. This is likely to be very important in the assistive device context, where rejection of poor quality or ambiguous images would be preferred over simply finding the closest match, which could be catastrophic.

SHORT also represents an updated and practical dataset for studying object recognition and retrieval in the challenging scenarios of hand-held objects and mobile or wearable cameras. In this chapter, in addition to introducing SHORT, I provide baseline performance measurements on the current dataset using several object recognition algorithms with different degrees of recognition complexity when tested against SHORT and discuss the research challenges arising from the particularities of visual object recognition from objects that are being held by users.

In Section 3.2 I review the computer vision datasets related to object recognition and argument the need for SHORT on the basis of an increasingly mobile and inclusive world. In Section 3.3 I describe the experimental set-up for image acquisition and the particularities of the different datasets that comprise SHORT. Section 3.4 describes the benchmarking of the dataset and its results, illustrating the advantages and disadvantages of the set. These will be discussed in Section 3.7, where I will point out the future work.

SHORT AND RELATED DATASETS

COIL-100 [116] and SOIL-47 [88] laid the foundations for the provision of large-databases of objects. However, their formats, image sizes and depth are now slightly dated.

A related database of house-hold products is the Grozi-120 dataset [106]. It contains 120 categories of groceries and is divided into “model” and “query” sets. For every product, the models (database images) were downloaded from the Internet while the query images consist of cropped video frames from recordings of supermarket shelves. SHORT provides a curated
Figure 6: All the grocery products that compose the SHORT dataset in its final set of 100 categories.

database of models to train object recognition algorithms taking also into account the assistive usage context, where the queries are highly variable. Grozi-120, however, lacks the variability and background clutter that would occur in real world scenarios, and the training images only have a limited number of views per product, usually just the frontal one showing the brand. SHORT expands the number of views to 36, with 12 different levels of rotation and 3 elevations. The multiple views allow investigating the importance of viewpoint in recognition accuracy, and in particular in being able to guarantee a definitive match.

The Caltech-101 and 256 databases [56, 63], together with PASCAL VOC [54], have been widely used to train and establish performance benchmarks for object recognition and detection algorithms. Caltech’s datasets increased the depth of previous datasets, with a minimum of 80 images per category to widen the choice of training and test sets size. However, neither dataset is recommended for localisation tests as the images contain “photographer’s bias” in which the objects are usually placed near the center of the image. Nevertheless, the challenging nature of the PASCAL datasets, and the well-defined evalu-
Figures 7: Sample test images. **Top row:** still-images; **bottom row:** video frames. Note: the images were cropped to fit the collage, they actually have different resolutions. This query selection contains samples from all the test datasets (see Table 1.)

*3.2 SHORT AND RELATED DATASETS*

The dataset protocol established by the PASCAL visual object classes (VOC) challenge, has led to it being a widely-cited benchmark for object recognition algorithms in recent years. However, only 2 out of 20 categories have more than 1,000 images per category, while in SHORT the minimum number of images per category is 3,507, outnumbering the latest PASCAL and both Caltech datasets.

*ImageNet* [44] first version was publicly released in 2010 with the aim of increasing the number of categories to the order of human recognition, which is estimated to be in the range of the tens of thousands [21]. *ImageNet’s* object categorisation “at near human scale” database provides 1.2 million images of a broad range of objects belonging to 1,000 categories [56]. This remarkable dataset depth, however, presents certain disadvantages when the scope of the application is more specific, as it is in the context of day-to-day shopping or assistive systems. The 100 products from SHORT, and the ones that will be obtained for subsequent expansions, are widely available. However, only 6 out of the 100 can be found in ImageNet: Coca-Cola, orange Fanta, semi-skimmed milk, orange marmalade, deodorant and OXO chicken cubes. Of these, some presented ambiguities in recognition. For example, the category “milk” contained 231 images; some represented milk bottles valid for a shopping context, but some of them depicted a glass or jug of milk, milk
crates in a factory, or other packages of milk. ImageNet cannot be used to train a system that guarantees a minimum category depth in a shopping context. Another criticism lies in the fact that some specific items are hard to index. In ImageNet, for instance, the orange Fanta is under drinks → soft drinks → orange synset. This makes it difficult to use ImageNet as a benchmark dataset for such a specific application as shopping.

While some of the datasets mentioned above, like PASCAL and ImageNet, offer a high degree of variability, SHORT has been designed to target the specific need of a dataset to develop object recognition systems for hand-held queries, particularly for use in assistive devices.

Another important limitation of existing datasets is that the same images can be used for training and query, therefore training data may contain unsystematic views of an object. Training a classifier with this data may introduce bias and can lead to “solving” the dataset, i.e. over fitting the categorisation model to the particularities of the training set. SHORT, however, offers a set of images for training which systematically cover variations in an object’s viewing angle. This allows the study of recognition performance when viewpoints of queries differ from views held in the database, and also the effect of viewpoint variation in the query.

In addition, the image queries in SHORT have been captured by multiple users with a variety of the latest smartphone cameras covering a wide range of viewing angles and containing images at current typical resolutions. SHORT introduces a new paradigm, in which high quality catalogue images are being compared with variable quality user-captured images; this makes the matching more challenging in SHORT than other datasets. In real-life situations, images of similar quality are often not present in both “database” and “query” datasets.

As an additional feature SHORT also contains test images acquired by blindfolded users and therefore mimics scenarios involving visually impaired users.
SHORT TECHNICAL DETAILS

Overview

SHORT is comprised of separate datasets for training and testing. Currently, the training dataset consists of high resolution acquisitions of 100 grocery items acquired in a very controlled set-up (see Figure 8), with 36 images of the same object from different angles and views. For testing, it is comprised of a set of query images from a subset of 30 grocery items acquired with 30 different smartphones. Lighting, pose, sensors and camera optics were therefore quite varied. This represents a more realistic view of hand-held object queries from hand-held devices than other datasets at the time. The SHORT dataset contains an average of more than 4,200 queries per product, allowing a realistic study of factors that affect recognition quality. In addition, video sequences of hand-held objects contain blur and different background clutter, as the volunteers moved while capturing sequences, relevant to a use case that might be considered as object recognition within video streams.

In addition to the images, SHORT provides ground truth annotations for all the data in terms of its object class label. Binary masks of the objects from the training dataset, indicating the bounding box around each item, are also provided.

SHORT is openly available and it can be downloaded from http://short.bicv.org. The website includes contact information where database users and SHORT curators can exchange impressions for future releases. The website provides access to the data as a single download or through a web file sharing service based on the open-source Owncloud that allows browsing. This contains access to the two releases of the dataset: SHORT-30 and SHORT-100. The original acquisition took place during the summer of 2013. The expansion of SHORT took place in June 2014 after taking into account feedback from the community on the usability of SHORT. The release of the dataset also includes code and evaluation data.
Image acquisition protocol

Training images

The database of models was acquired with a Nikon D7000 SLR camera using a 18-105 mm lens connected to a laptop and using the Nikon live capture software. The 16.2 megapixel captures in raw format were kept, but a JPEG copy of each image was also generated with a resolution of 4928×3264 pixels. A 986×653 resized copy of the high resolution images is also provided.

A total of 36 views were acquired per category. The views used in the product models contain shots at three elevations (17, 47 and 68 cm, at a distance of 1 m) above the object base. 12 degrees of rotation were used per elevation. A professional “chroma key” set-up was used. Both the background and a turntable containing the object were covered with a uniform chroma key backdrop, a “chroma blue” and “chroma green” cloth depending on the main color of the product being shot. Two halogen 125 W (equivalent to 625 W) 5500 K lamps were used to illuminate the background whilst the object was illuminated with a 40 × 40 cm 5400 K LED panel. The set-up is shown in Figure 8.

As described above, the acquisition of the training images was divided in two phases. In the first, model images of 30 products (the same set as the test set) were acquired. After receiving feedback from the research community, a second acquisition phase was run to expand the number of categories to 100. This expansion was devised to provide a number of categories that captured enough variability between the products so a generalisation study could be performed on the algorithms tested with the dataset and thus assess their performance and robustness across categories. The products of SHORT-30 are shown in the collage in Figure 9.

The process described above produces high-quality database (training) images; however no such precautions were taken with the query (test) images. The difference in capture quality makes SHORT very relevant to the case that in which high-quality
Diagram of the SHORT-100 training dataset acquisition setup

Overall View (seen from the left)

Overall View (seen from the top)

Figure 8: SHORT training set acquisition set-up
Figure 9: Collage representing the grocery products in the SHORT-30 dataset. This is a selection of items to include cans (shiny), boxes, uneven surfaces, similar shapes, semi-transparent or deformable packaging. These are popular products that are widely available for easy reproducibility and contain snacks, toiletries, medicines, drinks, canned food, dairy products, etc. Figure 6 shows all the products in the final SHORT-100 release.
product images are used to provide a controlled database of items. We see this factor as very important in order to guarantee the quality of information.

### Test images

Two experimental sessions were conducted. Around 30 volunteers were asked to take a minimum of five shots of every product and a five second video. No other instructions were given on how to acquire the images. During the second acquisition experiment, the images were taken with blindfolded users, reducing the alignment bias that a sighted user might have; this was used as a proxy for the assistive device context.

A total of 30 different camera-equipped smartphones was used, with resolutions ranging from $320 \times 240$ to $3264 \times 2448$ pixels. The variability of camera characteristics, parameters and capture conditions is enormous and a very distinctive feature of this dataset. The collage shown in Figure 7 contains a small sample of the variability present in the test dataset. As can be appreciated, images contain different views, levels of sharpness, background clutter, occlusion, illumination, and specular reflection. These features of the queries, together with the availability of sighted and blindfolded users, help in identifying certain characteristics required for database quality and coverage in or-

<table>
<thead>
<tr>
<th>DATASET</th>
<th>TOTAL IMAGES</th>
<th>IMAGES PER CATEGORY</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>MIN</td>
<td>MAX</td>
</tr>
<tr>
<td>ST-SG</td>
<td>2,797</td>
<td>75</td>
</tr>
<tr>
<td>VF-SG</td>
<td>91,293</td>
<td>2,115</td>
</tr>
<tr>
<td>ST-BF</td>
<td>1,225</td>
<td>32</td>
</tr>
<tr>
<td>VF-BF</td>
<td>39,209</td>
<td>832</td>
</tr>
<tr>
<td>All</td>
<td>134,524</td>
<td>3,054</td>
</tr>
</tbody>
</table>

Table 1: Summary of SHORT test datasets. Still images (ST) and video-frames (VF) acquired by sighted users (SG) or blindfolded (BF).
der to partially meet the needs of performance in the assistive context.

**BENCHMARKS**

*Classification using SIFT descriptor matching*

Object recognition can use several properties of query and database image appearance, ranging from colour distributions to texture and gradient field fingerprints e.g. histograms of gradients (HoG). However, apart from the use of spatial arrangements of putative matches between datasets and images, the discrimination of visual words, or bags-of-visual-words, is unlikely to be able to exceed the discrimination power of the descriptors or fingerprints used.

For this reason, I include an evaluation of the performance of an object recognition algorithm using a variant on pairwise SIFT [98] descriptor matching between descriptors from query images and those from database images. Each match is ranked according to a distance score; the match with the lowest distance score is assigned the highest ranked similarity. A query is then classified into one of the object classes. I now describe the basic matching technique in more detail.

Let us consider \( N \) training images \( T^{(n)}, n = 1, 2, \ldots, N \). A SIFT\(^1\) database was first constructed by computing \( I^{(n)} \) descriptor vectors \( y^{(n)}_i, i = 1, 2, \ldots, I^{(n)} \) using VLFEAT [172]. Each descriptor is of dimension \( 128 \times 1 \), so is in line with the commonly used configuration, as described by Lowe [98].

Consider, now, a query image, \( Q \) yielding \( J \) SIFT descriptor vectors, \( x_j, j = 1, 2, \ldots, J \). The Euclidean distance [172] in descriptor space was first computed between query descriptors, \( x_j \), and the training set descriptors, \( y^{(n)}_i \), for each training image, \( T^{(n)} \). Distances were then sorted for each compared descriptor pair.

---

\(^1\) Sparse or keypoint SIFT, not to be mistaken with dense-SIFT (DSIFT) introduced later in the thesis.
This was repeated for each image, yielding sets \( U^{(n)} \) consisting of tuples of indices \((p, q)\) such that

\[
U^{(n)} := \{(p, q) : f_U(x_q, y_p^{(n)}) = 1\}, \forall n,
\]

where \( f_U \) is a uniqueness criterion on descriptor distances within each image pair. The implementation of VLFEAT [172] was used, based on the approach suggested by Lowe [98] to define the set of matching descriptor pairs between the query image, \( Q \), and each training image, \( T^{(n)} \). \( f_U \) is defined so that descriptor pairs which meet the criterion evaluate to 1, and otherwise to 0.

For each of the \( K^{(n)} \) descriptor pairs \((p, q)\) that pass the uniqueness test \( f_U \) for each training image \( n \), one can calculate the pairwise distances in descriptor space:

\[
d_k^{(n)} = \|x_q - y_p^{(n)}\|, \quad k = 1, 2, ..., K^{(n)},
\]

where \( K^{(n)} = |U^{(n)}| \) is the number of descriptor matches after filtering for uniqueness. A scaled average of the distances \( d_1^{(n)}, d_2^{(n)}, ..., d_{K^{(n)}}^{(n)} \) is then computed [94] to give a single estimate \( \mu^{(n)} \) of the distance between query image and training image:

\[
\mu^{(n)} = \frac{1}{K^{(n)}} \sum_{k=1}^{K^{(n)}} d_k^{(n)}.
\]

Smaller \( \mu^{(n)} \) for a candidate database image, \( T^{(n)} \), implies a higher similarity and thus a stronger match. A query \( Q \) is then classified as belonging to the class \( C^{(p)} \) corresponding to the training image \( T^{(p)} \) if

\[
p = \arg\min_n \{\mu^{(n)}\},
\]
with \( C^{(p)} \in \{1, 2, ..., N_p \} \) and where \( N_p \) is the number of product classes, i.e. types of objects.

This descriptor-based method is likely to be indicative of the discriminating capacity of individual descriptor types; it is likely that this will have some effect on the ultimate performance of an object recognition technique. However, because it relies on descriptor-by-descriptor comparison, it is not readily scalable to large database sizes. Nevertheless, it is used as a “gold standard” method, despite not encoding geometric relationships between keypoints. I compare this with the performance of more scalable techniques in Section 3.6.

**Benchmarking: More scalable approaches**

The challenges posed by the SHORT dataset were also assessed using a fairly standard recognition “pipeline” to provide category (product ID) ranking. The SIFT descriptor was applied in one of two approaches: dense or sparse. The dense approach employs a grid with either 3 or 8 pixel spacing, and a \( 16 \times 16 \) spatial extent for the single-scale approaches. The sparse approach uses keypoint detection with standard SIFT-based scale-selection. The VLFEAT implementation is used for the descriptor and keypoint extraction. Three different histogram encoding methods were applied: hard assignment [40] (using 500 and 4000 visual words), locality-constrained linear coding (LLC) [181] and Fisher vector (FV) encoding [128]. I used the set-up of Chatfield et al. for the FV approach as it is known to perform best [35]. These methods will be described separately below.

On top of LLC and FV, spatial pyramids pooling was applied, using the approach described in [91] and with 3 pyramidal levels (0,1,2). Kernels (as defined in [173]) were first computed for each pyramidal feature [169]. Kernels were then averaged and fed to a support vector machine (SVM) classifier to determine the classification accuracy and average precision. The LibSVM implementation [33] was used to train an SVM for each cat-
category. The average precision values that are reported are in direct accordance with the VOC evaluation protocol [54].

**Hard Assignment (HA)** In hard assignment, a visual word is mapped to the closest matching descriptor within an image [40]. The visual words are generated using the k-means clustering algorithm to generate codebooks of 500 and 4000 visual words: these sizes allow performance comparisons with techniques reported in the literature [35]. A feature vector was constructed for each image by counting the occurrences of the visual words present in that image. For this case, an SVM with a linear kernel was applied to perform classification.

**Locality-Constrained Linear Coding (LLC)** In some studies, LLC [181] has been found to yield better recognition performance than hard assignment. Each descriptor is then encoded with a weight vector, \( w \), based on the covariance matrix, \( C \), of its distances to the 5 nearest neighbours by solving \((C + \lambda I_5)w = 1_{5 \times 1}\) with \( \lambda = 10^{-6} \) and where \( I_n \) is the \( n \times n \) identity matrix and \( 1_{m \times 1} \) denotes a column vector of \( m \) elements containing 1. This method has been shown to work well with linear classifiers, allowing relatively simple implementation.

**Fisher Vector Encoding** Fisher vectors have been shown to produce state-of-the-art performance in popular classification benchmarking datasets [35]. Recent work [128] has shown that the classification performance can be boosted even further by using a dense multi-scale approach. Thus, the set-up of Chatfield et al. [35] was followed to determine the classification rates by using 4 scales of analysis with two different sampling densities of 3 and 8 pixels. The spatial bin size of the SIFT descriptor was changed to 4, 6, 8 and 10 for each of the 4 scales, respectively. The descriptor dimensionality was reduced to 80 elements by using principal components analysis (PCA) and fitting a Gaussian mixture model (GMM) employing 256 components. The Fisher vector encoding was obtained by finding the
closest descriptor for each Gaussian model and aggregating the first and second order statistics. Finally, a Hellinger kernel [35], known to perform best for this type of encoding, was used to train the SVM classifier.

**ANALYSIS**

**Metrics**

Several metrics exist for evaluating a classifier’s performance. When testing for each category, a query can be a positive or a negative depending whether it belongs to the category being tested or not. In this scenario, a one-against-all binary classifier leads to four possible outcomes: true positive (TP), false positive (FP), true negative (TN) and false negative (FN). With these definitions, one can derive common performance metrics in the classification context given P and N the number of positive and negative queries respectively:

- **Classification accuracy** can be defined as the fraction of queries correctly classified \(\frac{TP + TN}{P + N}\).

- **Precision/Recall** (PR) curves, generated by arranging all the queries in descending order of their classification score [54] (i.e. the ranked output or confidence of how strong the prediction is).

- **Recall** can then be defined as the proportion of all positive queries with a classification score higher than a threshold.

- **Average Precision** (AP): the area under the PR curve. Mean Average Precision (mAP): the mean average precision for a set of queries, and ultimately, the mean AP across classes.

**RESULTS AND DISCUSSION**

In order to facilitate performance comparisons, I organised the queries in the SHORT dataset into the four groups in Table 1:
Still images acquired by sighted users (ST-SG) and blindfolded users (ST-BF); and videoframes acquired by sighted (VF-SG) or blindfolded users (VF-BF). From Table 2 we appreciate that these groups showed differences in average quality of recognition. Still-image queries outperform single-frame queries taken from unfiltered video: video frames typically include images which are blurred due to an object being rotated during capture. However, the pilot work to be described in Section 3.6.1 suggests that multiple frames from video can enhance accuracy. Queries from sighted users led to better performance than those from blindfolded users. The “blindfolded” queries appeared less susceptible to some forms of capture-bias: some images contained inadvertent partial object occlusions, making categorisation more challenging. This makes SHORT arguably a better dataset for designing technology for vision-based assistive systems for hand-held and wearable camera recognition of hand-held objects.

<table>
<thead>
<tr>
<th></th>
<th>mAP</th>
<th>ST-SG</th>
<th>VF-SG</th>
<th>ST-BF</th>
<th>VF-BF</th>
</tr>
</thead>
<tbody>
<tr>
<td>HA-4000</td>
<td>45.86</td>
<td>36.86</td>
<td>27.56</td>
<td>26.03</td>
<td></td>
</tr>
</tbody>
</table>

Table 2: Recognition performance across the four different subgroups of SHORT. The reduction in quality of match for the blindfolded subgroup is notable.

Classification results for the different methods are summarised in Table 3 and Figure 10. The low performance of most state-of-the-art methods demonstrates the challenge presented by SHORT. The precision/recall analysis shown in Figure 11 illustrates a remarkable variability in retrieval performance across categories. This fact reflects the complexity of the recognition problem, and the need for more robust algorithms.

For example, recognition algorithms for hand-held objects should be able to either provide a confidence measure, or possibly to request that further image queries be captured, even suggesting suitable, specific object transformations that could be used as a predictive verification step. For an assistive usage
Figure 10: Visual representation of the detailed performance evaluation of state-of-the-art encoding algorithms using the SIFT descriptor on the SHORT dataset. LLC – locality-constrained linear coding; FV – Fisher vector. The SIFT descriptors can be computed on dense grids with a spacing of $S_x$ pixels or around the SIFT keypoints (KP). The last figure in the method identifier indicates the size of the visual vocabulary (256, 500 or 4000 visual words). The red circle indicates the mean average precision across categories. Best viewed in colour.
3.6 Results and Discussion

<table>
<thead>
<tr>
<th>Method</th>
<th>Still images</th>
<th>Video frames</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>mAP Accuracy</td>
<td>mAP Accuracy</td>
</tr>
<tr>
<td>Descriptor match</td>
<td>25.11 86.65</td>
<td>- 19.70 48.98</td>
</tr>
<tr>
<td>HA-500</td>
<td>6.23 66.54</td>
<td>5.86 62.43</td>
</tr>
<tr>
<td>HA-4000</td>
<td>10.40 91.34</td>
<td>9.12 72.24</td>
</tr>
<tr>
<td>LLC-KP-500</td>
<td>0.90 62.31</td>
<td>0.53 57.55</td>
</tr>
<tr>
<td>LLC-KP-4000</td>
<td>0.89 79.06</td>
<td>0.56 70.47</td>
</tr>
<tr>
<td>LLC-S8-500</td>
<td>0.96 59.85</td>
<td>0.43 38.71</td>
</tr>
<tr>
<td>LLC-S8-4000</td>
<td>1.12 65.38</td>
<td>0.67 47.37</td>
</tr>
<tr>
<td>FV-S3-256</td>
<td>4.55 63.19</td>
<td>4.67 49.03</td>
</tr>
<tr>
<td>FV-S8-256</td>
<td>3.40 61.26</td>
<td>3.60 53.78</td>
</tr>
</tbody>
</table>

Table 3: Classification results. Detailed performance evaluation of state of the art algorithms on SHORT. HA – hard assignment; LLC – locality-constrained linear coding; FV – Fisher vector. The SIFT descriptors can be computed a) on dense grids with a spacing of $S_x$ pixels or b) around the SIFT keypoints (KP). The last figure indicates the size of the visual vocabulary (256, 500 or 4000 visual words).

In case, rather than a potentially incorrect match, it would be more appropriate for a system to reject the query.

Recently, the value of well-defined, robustly labelled datasets in both evaluating and training object recognition systems has become clear. Table 4, provides a comparison of different datasets, and methods of recognition. A few interesting observations may be made. For example, HA-4000 yields high average precision in the SHORT dataset, but appears to yield lower performance in Caltech-101 and PASCAL VOC databases. However, the results are the other way around for HA-500. This suggests that a larger vocabulary is needed for the objects in SHORT. In the case of Caltech-101 where the images are of much lower resolution than SHORT, the details are not easily resolvable and hence a larger vocabulary captures irrelevant variations such as noise. Grozi-120 has a lower mAP than most of the SHORT groups, which is possibly due to its rather low spatial resolution. These observations suggest that we are still some way from having a single dataset that adequately represents all use cases for object recognition.

Furthermore, the categories in the SHORT dataset vary in shape, size, colour, etc. and therefore there is a large variability...
Figure 11: LLC-S-4000 Test. Representative empirical precision and recall curve for a small sample of product classes. Only four classes, including best and worst results, are represented to help visualisation. The test was run with 59,226 queries against the database of 1,080 models. Performance in all categories is summarised in Figure 12.

in classification accuracy across categories, which is reflected in Figures 11 and 12. Products with low retrieval accuracy generally have reflective surfaces; the ones displaying high accuracy have clear surfaces with text labels, and are easy to hold and position.

The range of training images in the database – covering almost all faces of a product with three different elevations – will enable further analysis regarding the training images required to ensure a minimum accuracy for recognition. In this context, Figure 13a shows that increasing the training images do not necessarily improve the classification unless this increase conveys distinct information. The key factor is to include training images covering all faces of the product rather than different elevations. Figure 13b shows that most of the matches are made to the front view of an object since users tend to hold and iden-
<table>
<thead>
<tr>
<th></th>
<th>mAP</th>
<th>VOC-2007</th>
<th>Caltech-101</th>
<th>SHORT-ST</th>
<th>SHORT-VF</th>
</tr>
</thead>
<tbody>
<tr>
<td>HA-4000</td>
<td>37.50</td>
<td>18.91</td>
<td>45.86</td>
<td>36.86</td>
<td></td>
</tr>
<tr>
<td>HA-500</td>
<td>-</td>
<td>61.20</td>
<td>24.54</td>
<td>22.39</td>
<td></td>
</tr>
<tr>
<td>LLC-S8-4000</td>
<td>46.01</td>
<td>66.64</td>
<td>17.89</td>
<td>11.20</td>
<td></td>
</tr>
<tr>
<td>FV-S8-256</td>
<td>59.35</td>
<td>77.78</td>
<td>18.39</td>
<td>-</td>
<td></td>
</tr>
</tbody>
</table>

Table 4: Dataset comparison. Classification results of baseline performance algorithms on SHORT and other existing datasets.

Figure 12: BOVW + SVM test. Precision/recall curves for a representative sample of product classes: top four and worst five. The test was run with 59,226 queries against the database of 1,080 models.

...tify products from this pose. Nevertheless, the scenario with blind and partially sighted users could be different, with images taken from different viewpoints. Therefore, SHORT includes a rich set of viewing angles in our training dataset so the computer vision community could benefit from this and in-
crease the robustness of methods in this particular context of image query.

![Graph showing the effect of different training sets on SIFT descriptor matching.](image)

**Figure 13:** SIFT descriptor matching. Analysis of the effect of using different training sets. Figure 13a: Training set 1 includes all the training images. Training set 2 contains 18 images of each product taken every 60° at three different elevations. Training set 3 contains 12 images of each product at different angles but at the same elevation. A sample of Training set 3 with prd014: Matzos is shown in Figure 13b with the % of positive matches per position. This test was performed with 850 queries from the still-images set.
Evaluation of sequential video frames

In this section I analyse how sequential frames from a video of an object can be used in order to improve classification accuracy. First, multiple sequential frames from a video were queried, and each frame was matched to one of the categories of the SHORT database using the descriptor matching method described in Sections 2.2.4 and 2.2.5. A histogram of the matches was computed for several videos of the same object, as shown in Figure 14. Even though the total number of incorrect matches (the blue bins) increases as we query more video frames, they are distributed across a range of object categories in the database.

As shown in Figure 14, the correct object category (the red bin) often has a higher number of hits than the incorrect ones. Therefore I propose to use “individual voting” as a metric to classify an object based on querying sequential images from a video of a hand-held object. For each video frame, a classification result is provided, and the results for each frame count
as a vote for the final decision about the object category. The category with the majority of the votes is selected. Further analysis was undertaken in order to determine the number of video frames that are required for the total number of correct matches to exceed the number of individual total incorrect matches.

Preliminary results, depicted in Figure 15, show that the number of total incorrect matches to individual objects rises slowly while the number of total matches to the correct object increases rapidly. It seems clear that for some of the videos the number of hits grows faster than the number of incorrect matches. However, other videos show similar increase rates, suggesting possible time limits for the use of these sequential methods. These time limits can be justified as after few seconds the object might not be present in the frames as the person holding the object might have returned it to its original place.

The above analyses were undertaken for several videos and object categories under the SHORT dataset. Classification accuracy using different numbers of query frames and the above voting metric is presented in Section 3.6.2.

![Figure 15: Evaluation of sequential video frames: Fraction of correctly matched queries to incorrect matches for videoframe sequences. Note that there is a distribution of incorrect matches across multiple categories as the number of sequential query frames increases.](image-url)
Classification accuracy based on sequential video frames

Classification accuracy using the individual voting metric was computed for seven categories of SHORT dataset (Table 5). As we saw in Section 3.6.1, a video $Q$, comprising of query frames $Q_i$, $i = 1, 2, ..., N$, is classified as the $k$-th category, $C_k$, to which the majority of query frames, $Q_i$, were matched to. The accuracy was calculated for twelve videos of each object based on different limits for the number of queries allowed, $N$.

Table 5 shows that the classification accuracy increases as we increase the limit of query frames, except for occasional dips which occur due to instability as a person rotates the object in their hand. In comparison to the classification accuracy of individual queries, classification based on sequential video frames using voting gives a much higher accuracy. The standard deviation of the classification accuracy across videos of the same category also decreases as more frames are used to provide an object category estimate.

<table>
<thead>
<tr>
<th>ID</th>
<th>Acc 10 fr (%)</th>
<th>Acc Std 10 fr</th>
<th>Acc 30 fr (%)</th>
<th>Acc Std 30 fr</th>
<th>Acc 50 fr (%)</th>
<th>Acc Std 50 fr</th>
<th>Acc 70 fr (%)</th>
<th>Acc Std 70 fr</th>
<th>Acc 90 fr (%)</th>
<th>Acc Std 90 fr</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>37.9</td>
<td>85.7</td>
<td>35.0</td>
<td>85.7</td>
<td>35.0</td>
<td>85.7</td>
<td>35.0</td>
<td>92.9</td>
<td>25.8</td>
<td>92.9</td>
</tr>
<tr>
<td>2</td>
<td>55.6</td>
<td>75.0</td>
<td>43.3</td>
<td>75.0</td>
<td>43.3</td>
<td>75.0</td>
<td>43.3</td>
<td>91.7</td>
<td>27.6</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>40.9</td>
<td>73.3</td>
<td>44.2</td>
<td>73.3</td>
<td>44.2</td>
<td>80.0</td>
<td>40.0</td>
<td>80.0</td>
<td>40.0</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>43.1</td>
<td>62.5</td>
<td>48.4</td>
<td>62.5</td>
<td>48.4</td>
<td>62.5</td>
<td>48.4</td>
<td>87.5</td>
<td>33.1</td>
<td>87.5</td>
</tr>
<tr>
<td>5</td>
<td>80.6</td>
<td>92.3</td>
<td>26.7</td>
<td>100.0.0.0.0</td>
<td>100.0.0.0.0</td>
<td>100.0.0.0.0</td>
<td>100.0.0.0.0</td>
<td>100.0.0.0.0</td>
<td></td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>46.2</td>
<td>78.6</td>
<td>41.0</td>
<td>78.6</td>
<td>41.0</td>
<td>92.9</td>
<td>25.8</td>
<td>92.9</td>
<td>25.8</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>46.2</td>
<td>78.6</td>
<td>41.0</td>
<td>78.6</td>
<td>41.0</td>
<td>92.9</td>
<td>25.8</td>
<td>92.9</td>
<td>25.8</td>
<td></td>
</tr>
</tbody>
</table>

Table 5: Classification accuracy of different objects for different number of sequential query frames (fr). Accuracy is defined as the number of video sequences correctly classified divided by the total number of sequences queried. Each of these short video sequences contains only one product. The standard deviation (Std) captures the variability of the classification accuracy across videos of the same object when this is computed as the ratio between the correctly and incorrectly classified frames in each video of the same category.
CONCLUSION AND FUTURE WORK

I have presented a new publicly available dataset containing queries consisting of single images and video frames of hand-held objects. These were captured by multiple users using a variety of smartphones. The purpose of this dataset is to have a more realistic view of variability in acquisition across images taken of hand-held objects. Unlike other datasets, SHORT also contains a great disparity in the image resolution and quality of its object and query items, which I feel is more representative of the conditions of use we would expect for high-quality, curated models being used to service queries from wearable or hand-held cameras.

The training set of model images systematically captures variations in objects’ viewing angle allowing us to study the effect of the number of viewing angles present in a database on matching quality in a widely varying query. SHORT can be used to develop object recognition algorithms targeted for both sighted and visually-impaired users: comparisons of such will allow a better understanding of the extra technical requirements placed on computer vision systems by users who may not be aware of the quality of the images they are capturing. Additionally, SHORT paves the way for addressing several open questions; for example:

a) A key issue is that of how to minimise – or better yet, eliminate – the chance of a false positive match: such errors could be dangerous when used in the context of household product identification. One way to tackle this would be to reject a query based partly on its quality (sharpness, resolution, etc.). SHORT, with a range of variability in the test images, can be used to establish image quality metrics for accepting a query in this context. Another possibility would be to reject a query based on the degree of visual ambiguity, when assessed against the database. This brings us to the question of database size.
b) Scalability of image search in large databases brings a key challenge: how to ensure that ambiguity between two similar products might be noted and used to either eliminate a candidate search result as being too uncertain a match, or to prompt a user to submit specific queries based around different hypotheses of what the object might be.

SHORT was developed with the intention of piloting a larger dataset to test such strategies, and indeed our future work will aim to increase the number of classes. Future work might explore curated crowdsourcing techniques to involve a larger community in growth of the dataset.

The shopping and assistive contexts pose a real challenge for the current generation of recognition algorithms, but SHORT represents a key step towards being able to assess performance under realistic query conditions.
APPEARANCE-BASED INDOOR LOCALISATION: A COMPARISON OF DESCRIPTOR PERFORMANCE

INTRODUCTION

Self-localisation within indoor spaces has numerous real-world applications, ranging from navigation inside public spaces and large shopping and social environments to assistive devices for people with visual impairment. Harvesting information from radio-strength signals and radio beacons to perform localisation is an emerging technology [179, 55]. However, few potential solutions are as compelling as those using visual information, captured from wearable or hand-held cameras, and conveyed into knowledge about how to navigate a space.

This work proposes an alternative approach to geometric and SLAM-based localisation. Location is, instead, inferred through visual queries against the journeys of other users, rather than by explicit map-building or geometric inference. I test this idea in a new dataset of visual paths [137], containing more than 3 km of video sequences captured through multiple passes along 10 corridors in a large building with ground truth. I compare custom-designed descriptors with SIFT [98], HOG 3D [84] and a state-of-the-art SLAM method: large-scale direct monocular SLAM (LSD-SLAM). Standard bag-of-visual-words (BoVWs) approaches are used to index and associate views between journeys.

The results suggest that, even without tracking, significant cues can be captured and used to infer location. The application to wearable camera technology – whereby image cues are harvested from volunteered journeys, can then be used to help other users of the same space navigate – is the eventual goal of this work, which is a natural extension to recently re-
ported approaches based on harvesting environmental signals from smartphones [178].

RELATED WORK

Matching between visual paths: appearance-based methods for localisation

As I briefly introduced in Chapter 2, a visual path can be seen as a collection of image frames that are induced by the relative motion of a person in a scene. The work reported in this chapter involves matching the visual paths of a new journey instance to previous, similar instances.

Early work by Matsumoto et al. [104] introduced a similar concept of the “view-sequenced route representation”. In this scheme, a robot could perform simple navigation tasks by correlating current views against those held in a database. Ohno et al. [119] also worked on this idea, using the difference between frames of detected vertical lines to estimate changes in position and orientation. Their results were constrained to controlled robot movement, and therefore arguably of limited applicability to images obtained from human ego-motion. Also employing vertical lines as features, this time from omni-directional images, Tang et al. used estimated position differences between sequences to perform robot navigation [164]. To make the inference more robust, they used recorded odometry at training time. This approach would certainly reduce the error in the localisation task. However, it could lead to solving the training route, without truly analysing the performance of feature matching methods. Furthermore, without ground truth available in a crowdsensing setting, the technique of training with ground truth is of limited usability. On the other hand, with many passes through the same space, the reference for a journey could be the visual paths themselves. In this case, one would use ground truth – if available – only to ascertain the accuracy of proposed matching or localisation methods. This is
the approach taken in the current work. Another drawback of these early publications is that they report results in routes of a few metres in length. The evaluation in this work is in a dataset three orders of magnitude longer.

Filliat [57] presents a bag-of-visual-words approach to provide a “qualitative” recognition of the room their robot is in. In summary, a set of features is extracted, a dictionary is built, and using a voting system as a classifier, a label (class) is assigned to a room. This is arguably not a SLAM method, as there is no simultaneous localisation and mapping that is comparable to the mainstream SLAM methodologies that we will see in the next sections. In fact, there appears to be no mapping beyond a register of rooms visited. Secondly, the method is only partially a topological localisation technique, since there is no modelling of the relationships between rooms. However, one important contribution is the incremental approach to database building. Filliat, rather than relying on a given set of categories (the rooms), creates new categories on the fly based on a decision made from the statistics of the visual words contained within an image.

FAB-MAP [59] and its popular open-source implementation, OpenFABMAP [60], have been considered the state-of-the-art appearance-based method for robot navigation. FAB-MAP relies on a BoVW dictionary constructed from a database of speeded up robust features (SURF) features extracted from location images to provide matching between previously visited places as well as a measure of the probability of being at a new, unseen location. The arrival of SeqSLAM [108] brought additional robustness to the FAB-MAP paradigm. By enforcing sequential constraints to the image matching front-end they were able to improve on OpenFABMAP, especially in challenging situations such as night or rainy sequences. To date, however, such appearance-based methods in a SLAM context are more commonly used at large spatial scales in order to address the loop-closure problem; its application is less common in indoor spaces when operating at smaller spatial scales. In addition, there is lit-
tle evaluation of the effect of matching ambiguity when using appearance-based techniques.

The closest approach to the one described in the present work is perhaps that of Schroth et al. [149]. Schroth and colleagues made use of a purely appearance-based method to provide localisation from image sequences acquired with mobile devices. However, their experiments are closer to those of the object categorisation community, where tests are performed in a batch fashion and performance is reported as precision-recall or receiver operating characteristic (ROC) curves. Our view, however, is that localisation error distributions are also a good illustrative metric of performance. In addition, their use of a 360° camera to produce a database for training somewhat constrains the effectiveness of the system to the quality and richness of this database, almost ensuring poor results when certain conditions are not captured in the database, as demonstrated by Milford and Wyeth [108]. Crowdsourced training instances, discussed in Section 4.2.2, provide information that can largely solve the issue [138].

The performance of previously reported methods that use a retrieval-type approach, albeit from the order of tens of cm (in the case of [104, 119] in routes of few metres in length) to few m [149, 108], cannot be taken as representative for the evaluation of the methods presented in this chapter, where I will propose a new performance metric based on localisation error probabilities.

Finally, the use of a tracking method such as Kalman filtering is deliberately excluded from this work, since it can often hide poor performance of the visual processing.

Crowdsourcing visual paths

Using image sequences represents a particularly data-intensive form of crowdsensing in which the image streams from wearable cameras could be volunteered to others as reference paths
for indoor journeys. An illustration of this concept is presented in Figure 16.

This type of crowdsensing approach is gaining interest, with remarkable work from Google’s indoor localisation systems and crowdsourced sensor information and maps [79]. In terms of a retrieval-based visual localisation system, the NAVVIS team [72] released a dataset for evaluating indoor navigation from a camera-equipped robot. They also advanced earlier work on visual localisation based on matching of SIFT descriptors [124] to one using a bag of features that could be stored in mobile phones for quick retrieval [147, 148]. The dataset I introduce in this work is not constrained to robot navigation, as it includes the ego-motion associated with hand-held and wearable devices.

The proposed approach of Figure 16 aims to match images of the current navigation view from wearable or hand-held cameras against crowdsourced journeys made along the same indoor spaces. This technique has some precedence in the literature, with Liu et al. [96] using a database of keyframes registered with 2D positions and orientations that were later used in an “online” mode for servicing queries that consisted of GIST and SURF descriptors. A state estimator based on a Hidden Markov Model (HMM) was also used for state prediction, and to enforce spatio-temporal consistency. The authors, however, did not appear to test their system “in the wild”: for example, the database images were post-processed to reduce the motion blur. For this, they used an external inertial motion unit (IMU) to capture information about the roll and pitch angles of motion.

Alternative methods: non feature-based and sensor merging

François Chaumette’s team has recently explored navigation solutions that do not require geometrical nor pixel intensities’ features but use mutual information (MI) as a similarity measure. Their system performs a maximisation of the MI that is directly
Figure 16: A sample path (Corridor 1, C₁, from Figure 17) illustrating the multiple passes through the same space. Each of these passes represents a sequence that is either stored in a database, or represents the queries that are submitted against previous journeys. In the assistive context, the user at point A could be a blind or partially sighted user, and he or she would benefit from solutions to the association problem of a query journey relative to previous “journey experiences” along roughly the same path, crowdsourced by N users that may be sighted.

connected with the motion of the robot. These results, and others that rely on tracking visual features [150] cannot serve as a comparison for the current methods either, as they would hinder the fair evaluation of the visual features in isolation.

For outdoor navigation, the Global Positioning System (GPS) has been in widespread use for many years. In an indoor context, localisation technology is still rapidly evolving [152, 178, 133]. Using visual information for localisation holds great potential but lacks reliability; therefore one would certainly seek to support this approach with other forms of sensor such as received signal strength indication (RSSI) data, magnetometers, and tracking algorithms [147, 148, 133]. In the work presented in this thesis, we seek to explore efficient techniques that could be used to index and compare the visual path information gath-
related work

Figure 17: Maps of the recording locations.

ered by multiple user journeys, and to measure the potential of vision on its own as a localisation mechanism.

Structure from motion (SfM)

One outcome of the proliferation of digital cameras is the development of structure from motion (SfM) algorithms that can infer 3D models of cities [2] by means of photographs taken by visitors to popular city landmarks. With such images acquired from the internet [156], bundle adjustment can be used to reconstruct the 3D information about buildings in within well-photographed locations, in addition to the camera pose of every photograph. Similarly to our work, Hile and colleagues also crowdsource location information [69]. However, they provide this information through geotagged images from Flickr, to then use Snavely’s SfM algorithm [156] to perform camera pose estimation. Using models of a scene constructed using bundle adjustment, the position and pose of a camera from a sequence of new photographs taken from a mobile device can be used as a source of “visual” navigation information [174]. However, bundle adjustment is an iterative error minimisation algorithm, and its computational load is still large for real-time use at scale. Furthermore, it is not entirely clear how the geometric information that is acquired from such models could be updated as aspects of a scene change.
Simultaneous localisation and mapping (SLAM)

Another important branch of vision-based navigation research can be found in robotics. Visual SLAM [87, 52, 115] provides a real time reconstruction of the scene by using stereo cameras (stereoSLAM) or a single one (monoSLAM). Though SLAM is often described for its ability to infer the geometric model of a scene, it also estimates the camera trajectory as part of the camera pose inference. The combination of the two is a powerful source of navigation information. In addition, in subsequent journeys along the same route, geometric information can be refined and also used for refinement of camera pose estimates.

Apart from the possible computational load issues that solutions growing in complexity can entail, the main challenge of SLAM, or the “SLAM problem” is to produce an accurate map and therefore a precise localisation. Traditionally, this problem has been studied from two different perspectives: from the image processing and recognition perspective (the latest developments including deep learning approaches [38]); from a tracking point of view, relying on technologies ranging from depth and inertial sensors to wireless networking to minimise the error drift in the location estimations.

Davison et al. [43] developed a seminal real time mono SLAM method that used a lightweight Harris feature detector of interesting points within the point of view of the camera. In the next step the system would locate the same feature in a neighbouring patch. Montiel and colleagues, as we will see in the comparison with EKF SLAM of Section 4.7, took the same monoSLAM approach but aiming for its simplification in degrees of freedom and trying to overcome the problem of the initialisation of features. In [111] they devised an inverse depth parametrisation for point features with 3 degrees of freedom (DOF) instead of the expensive 6.

The complexity of the visual features used for SLAM increased with the introduction of SIFT instead of Harris detectors in [37] and [163] to Montiel and Davison’s approaches. This opened
SLAM to outdoor and more challenging applications as SIFT invariances increased the robustness of the SLAM.

This robustness is key, as some visual SLAM algorithms provide a navigation method suitable for use by autonomous robots [87] but relying on static features from the scene that are subsequently matched before the trajectory is estimated. However, in real life conditions, many of these features are dynamic, since they belong to objects or elements of the scene that are moving (e.g. a crowded scene). Additionally, SLAM algorithms, particularly visual monoSLAM, rely on optic flow induced by egomotion in order to infer a geometry and build up a map. In the presence of significant additional motion within the scene, algorithms can begin to fail. One such failure mode is called scale drift and has been studied and taken into account by [159].

Recent vision-based approaches have been developed by Alcantarilla et al. ([6]; [8]) They incorporate dense optical flow estimation into visual SLAM in order to improve the performance of algorithms in crowded and dynamic environments by detecting the presence of objects that are moving relative to the world coordinate system. Additionally, they have developed a fast vision-based method to speed-up the association between visual features and points in large 3D databases [7]. This approach consists of learning the visibility of the features in order to narrow down the number of matching point correspondence candidates.

One of the most recent SLAM algorithms is LSD-SLAM [51]. This semi-dense tracking and mapping method seems to perform well in an indoor SLAM setting. This approach, instead of keypoints and descriptors, uses semi-dense depth maps for tracking by direct image alignment. This is a remarkable step forward, as the semi-dense maps allow lighter frame to frame comparisons, to the point where odometry can be performed on a modern smartphone [146]. This system, as most SLAM methods, relies greatly on a very accurate camera calibration and initialisation routine, and as we will see in the detailed comparison of Section 4.7, best results are often achieved under
specific conditions, such as monochrome global shutter cameras with fish eye lenses.

Methods

In this section I describe the different image description modalities that were evaluated, starting by the feature extraction process, followed by the descriptor quantisation and distance metric used.

Pipeline

I evaluated the performance of several approaches to matching image queries taken from one visual path against the remainder of the visual paths. In order to index and query the visual path datasets, the sequence of processes that is illustrated in Figure 18 was adopted. The details behind each of the processes (e.g. gradient estimation, spatial pooling) are presented in Section 4.3.3. Both descriptors that operate on single frames (spatial) and descriptors that operate on multiple frames (spatio-temporal) were compared.

Frame-level descriptor: One descriptor per frame

Inspired by the use of optical flow in motion estimation [183] and space-time descriptors in action recognition [180] I estimated in-plane motion vectors using a simple approach. Derivative filters were applied along $(x, y, t)$ dimensions, yielding a 2D+t, i.e. spatio-temporal, gradient field. To capture variations in chromatic content from the visual sequence, spatio-temporal gradients were computed separately for each of the three RGB channels of the pre-processed video sequences. This yielded a
Figure 18: The stages in processing image sequences from database and query visual paths are illustrated above. This does not show the process behind the estimation of ground truth for the experiments, which is described separately in Section 4.5. Variants of the gradient and pooling operators, quantisation approaches and distance metrics are described in Section 6.4.2.
$3 \times 3$ matrix at each point in space, effectively a chromatic Jacobian (Eq. 9),

$$J = \begin{pmatrix}
\frac{\partial I_r}{\partial x} & \frac{\partial I_r}{\partial y} & \frac{\partial I_r}{\partial t} \\
\frac{\partial I_g}{\partial x} & \frac{\partial I_g}{\partial y} & \frac{\partial I_g}{\partial t} \\
\frac{\partial I_b}{\partial x} & \frac{\partial I_b}{\partial y} & \frac{\partial I_b}{\partial t}
\end{pmatrix}$$

Temporal smoothing was applied along the time dimension, with a support of 11 neighbouring frames. Finally, the components of the matrix were each averaged (pooled) over 16 distinct spatial regions (see Figure 19), not very dissimilar to those to be described later in this thesis.

![Figure 19: A maximum projection intensity rendering of 16 pooling regions over space. The x components of descriptor component time series from regions A, B, C and D are shown in Figure 20.](image)

The rationale is that ego-motion is likely to be an important visual cue in human pedestrian navigation, particularly in close or cluttered spaces. Rather than explicit computation of optical flow, the collection of gradient information over different pooling regions of space captures and encodes relevant motion in the form of time series. The 16 proposed pooling regions illustrated in Figure 19 were selected to capture elements of the chromatic space-time Jacobian within two sectors at different radii from the centre of the frame (e.g. A and C, and B and D); to capture information in pairs of regions (such as A and B) that are diametrically opposed from each other horizontally, and also vertically. Diagonal regions were also proposed, yielding
Figure 20: Four (of 144) representative signals acquired from a visual path; these signals encode changes in the red and green channels as a user moves through space. The collection of signal traces at one point in time can be used to build a simple frame-level space-time descriptor: LW-COLOR. The signal amplitudes are spatially pooled temporal and spatial gradient intensities.

fairly uniform angular coverage over image space. Spatial and temporal gradients in the centre of the frame were not encoded because this region contains less visible motion over equivalent time scales to the more peripheral regions.

For each visual path, this yielded a total of $16 \times 9 = 144$ separate time-series, or signals, of length approximately equal to the video sequences. An illustration of the time series for one visual path is shown in Figure 20. Effectively, for each frame, a 144-dimensional descriptor is captured. The elements of this descriptor describe the weighted average of the Jacobian elements within the respective pooling regions that can be used for indexing and matching visual path locations.

At each point in time, the values over the 144 signal channels are also captured into a single space-time descriptor per frame: LW-COLOR. Some observations from the components of this descriptor are that a) relative ego-motion is clearly identifiable in the signals; b) stable patterns of motion may also be identified, though changes in the precise trajectory of a user
could also lead to perturbations in these signals, and hence to changes in the descriptor vectors. Minor changes in trajectory might, therefore, reduce one’s ability to match descriptors between users. These observations, together with the possibility of partial occlusion, motivated the use of patch based descriptors, so that multiple descriptors would be produced for each frame. These are introduced next.

Local descriptors: Multiple descriptors per frame

Keypoint-based SIFT (KP-SIFT)

The original implementation of Lowe’s SIFT descriptor follows the extraction of interesting points in the image that are stable to certain transformations, the “SIFT keypoints” [98]. As we saw in Chapter 2 and 3, this descriptor is widely used across many branches of computer vision, from object recognition to motion detection and SLAM. I used the standard implementation from VLFEAT [172] to compute $\vec{\nabla} f(x, y; \sigma)$ where $f(x, y; \sigma)$ represents the embedding of image $f(x, y)$ within a Gaussian scale-space at scale $\sigma$. The parameter $\text{PeakThresh}$, $t_p$ is used to filter out small local maxima in scale-space that might be originated by noise. Given the small size of the frames in the sequences the minimum threshold $t_p$ was set to 0.

Dense SIFT (DSIFT)

The Dense-SIFT (DSIFT) descriptor [91] is a popular alternative to keypoint- based SIFT. It sacrifices some invariance properties available with keypoint-based SIFT, producing descriptors that are densely, rather than sparsely, distributed across the image. This DSIFT descriptor was calculated by sampling of the smoothed estimate of $\vec{\nabla} f(x, y; \sigma)$. The implementation of the VLFEAT toolbox was chosen, setting $\sigma = 1.2$, with a stride length of 3 pixels. This yielded around 2,000 descriptors per frame, each describing a patch of roughly $10 \times 10$ pixels.
Single-frame Gabor descriptors (SF-GABOR)

An alternative single-frame technique based on a tuned, odd-symmetric Gabor-based descriptor is the SF-GABOR. For this, I used 8-directional spatial Gabor filters previously tuned on PASCAL VOC data [54] in order to provide an implicit encoding of the orientation of local image structures. Each filter gives rise to a filtered image plane, denoted $G_k,\sigma$. For each plane, I compute the discrete spatial convolution, $G_k,\sigma \ast \Phi_{m,n}$, with a series of pooling functions, $\Phi_{m,n}$. The latter are produced by spatial sampling of the function:

$$
\Phi(x,y;m,n) = e^{-\alpha\left[log_e\left(\frac{x^2+y^2}{d_n}\right)\right]^2 - \beta|\theta-\theta_m|},
$$

with $\alpha = 4$ and $\beta = 0.4$. The values of $m$ and $n$ were chosen to produce 8 angular regions ($m = 0,1,\ldots,7$) at each of two distances $d_1, d_2$ away ($n = 1,2$) from the centre of a spatial pooling region. These lobes were similar to those shown in Figure 19, with one additional central lobe, and used a spatial weighting pattern similar to the DAISY descriptor [185]. For the central region, corresponding to $m = 0$, there was no angular variation but instead a log-normal radial decay, with a limiting value at $(x,y) = (0,0)$. This arrangement yielded a total of 17 spatial pooling regions (see “poolers” layer in Figure 21). The resulting $17 \times 8$ fields are sub-sampled to produce dense 136-dimensional descriptors, each representing an approximate $10 \times 10$ region, and yielding around 2,000 descriptors per image frame after spatial sub-sampling.

These poolers were suggested by Alexiou for visual object recognition [9], but were thought to be good a good choice for location estimation because of the nature of the RSM navigation dataset (see Section 4.4).

In Chapter 5 I provide another formulation for the SF-GABOR descriptor that makes use of a tensor notation. This will be needed to understand a convolutional neural network (CNN)
Figure 21: The spatial pooling pattern used for single-frame Gabor filtering is based on the regions shown here. These regions were generated by sampling Eq. (10) to create $11 \times 11$ px pooling masks. The masks can be applied to the Gabor filtered video-frame outputs ($9 \times 9$ px) by spatial convolution, followed by sub-sampling the output every 3 pixels. See text for further details.

interpretation of artificial place cell models based on the SF-GABOR descriptors.

**Space-time descriptors**

Given the potential richness available from space-time information, three distinct approaches were explored to generate space-time patch descriptors. When generating the descriptor associated with each patch, all approaches yield multiple descriptors per frame, and all take into account neighbouring frames in time. In contrast to a sparse-sampling approach of a keypoint-based descriptor, all three densely sample the video sequence. The three methods are i) HOG 3D [84]; ii) a space-time, antisymmetric Gabor filtering process (ST-GABOR); and iii) a spatial derivative, temporal Gaussian (ST-GAUSS) filter.

1. The **HOG 3D** descriptor (HOG3D) [84] was introduced with the aim of extending the very successful two-dimensional histogram of oriented gradients (HOG) technique [41], to space-time fields, in the form of video sequences. HOG 3D seeks computational efficiencies by smoothing
using box filters, rather than Gaussian spatial or space-time filters. This allows three-dimensional gradient estimation across multiple scales using integral video representations, a direct extension of the integral image idea [176]. The gradients from this operation are usually performed across multiple scales. I used the dense HOG 3D option from the implementation of the authors, and the settings yielded approximately 2,000 descriptors per frame of video. Each descriptor contained 192 elements.

2. **Space-time Gabor (ST-GABOR)** functions have been used in activity recognition, structure from motion and other applications [25]. One dimensional convolution was performed between the intensity video-sequence \(I(x, y, t)\) and three one-dimensional Gabor functions along either one spatial dimension i.e. \(x\) or \(y\), or along \(t\) (see Eq. 11). The one-dimensional convolution is crude, but appropriate if the videos have been downsampled. The parameters of \(\sigma_x\) and \(\sigma_y\) were set to be equal, and to provide one complete cycle of oscillation over approximately 5 pixels of spatial span, both for the \(x\) and \(y\) spatial dimensions. The filter for the temporal dimension was set to provide around one oscillation over 9 frames. I also explored symmetric Gabor functions, but found them rather less favourable.

\[
I_1(x, y, t) = I(x, y, t) * g_x(x; \sigma_x), \\
I_2(x, y, t) = I(x, y, t) * g_y(y; \sigma_y), \\
I_3(x, y, t) = I(x, y, t) * g_t(t; \sigma_t). 
\] (11)

After performing the three separate filtering operations, each pixel of each frame is assigned a triplet of values corresponding to the result of each filtering operation. The three values are treated as being components of a 3D vector. Over a spatial extent of around \(16 \times 16\) pixels taken at the central frame of the 9-frame support region, these vectors contribute weighted votes into descriptor bins ac-
cording to their azimuth and elevations, with the weight-
ing being given by the length of the vector. The votes
are also partitioned according to the approximate spatial
lobe pattern illustrated in Figure 21. Each frame had ap-
proximately 2,000 ST-GABOR descriptors, each of 221 ele-
ments.

3. A final variant of space-time patch descriptor was de-
gined. This consisted of spatial derivatives in space, com-
bined with smoothing over time (ST-GAUSS). In contrast
to the strictly one-dimensional filtering operation used
for the ST-GABOR descriptor, I used two $5 \times 5$ gradient
masks for the $x$ and $y$ directions based on derivatives of
Gaussian functions, and an 11-point Gaussian smoothing
filter in the temporal direction, using a standard devia-
tion of 2. 8-directional quantisation was applied to the
angles of the gradient field, and a voting process incor-
porating gradient magnitude was used to distribute votes
across the bins of a 136-dimensional descriptor. Like the
ST-GABOR descriptor, pooling functions, similar to those
shown in Figure 21, were applied. The number of descrip-
tors produced was the same as for the other methods us-
ing patch-level descriptions.

Quantisation and histogram encoding

An initial conjecture was that whole frames from a sequence
could be indexed compactly, using the single-frame descriptor
(LW-COLOR). This was found to lead to disappointing perfor-
ance (see Section 4.6). For the case of many descriptors-per-
frame, i.e. descriptors that are patch-based, there is the added
problem of generating around 2,000 descriptors per frame, if
dense sampling is used. Vector quantisation (VQ) was applied
to the descriptors, then histograms of quantised descriptors
were used to encode each frame as a histogram of visual words
[40]. The dictionary was always built by excluding the entire
journey from which queries were to be taken.
The resulting dictionaries were then used to encode the descriptors of the $M - 1$ training passes and the remaining query pass. This sequence of processes is commonly known as a bag-of-visual-words (BOVW) pipeline. Two different approaches to the encoding of descriptors were taken, one based on standard k-means, using a Euclidean distance measure (hard assignment, “HA”), and one corresponding to the vector of locally aggregated descriptors (VLAD) [11]. These histograms were all $L_2$-normalised.

In the HA case, the dataset was partitioned by selecting $M - 1$ of the $M$ video sequences of passes through each possible path. These $M - 1$ sequences have a total of $N$ frames. A dictionary of visual words was created by running the k-means algorithm on the partitioned set of training descriptors contained in the $N$ frames. The dictionary size was fixed to 4,000 in order to achieve a balance between computational time and atom stability, and allowing comparison with the work of others in related fields [35].

For VLAD, a k-means clustering was first performed using a dictionary size of 256 words. For each descriptor, sums of residual vectors were used to improve the encoding. Further advances to the basic VLAD, which include different normalisations and multiscale approaches, are given by [11].

Localisation using histogram distances

Once histograms had been produced, a distance measurement was used to compare the similarity of histograms in a query frame with the database entries. The query operation was simply performed by using the kernel approaches described in [173]. Concretely, to compare encodings, either $\chi^2$ or Hellinger distance metrics [173] were used to retrieve results for HA and VLAD encoding approaches respectively. Distance comparisons were performed directly between either hard assigned bags-of-visual-words (BoVWs) or VLAD image encodings arising from collections of descriptors for each frame.
For the $M - 1$ videos captured over each path in the database, the queries were constructed from the remaining path. The histogram of each query frame, $H_q$, resulted in $M - 1$ separate comparison vectors containing scores. By using these kernel-based comparisons (which are always positive, and act in the opposite way of a distance metric), one can identify the best matching frame, $\hat{f}$, from pass, $\hat{p}$, across all of the $M - 1$ vectors. This may be expressed as:

$$L(\hat{p}, \hat{f}) = \arg\max_{p,f} \{ K_D(H_q, H_{p,f}) \},$$  \hspace{1cm} (12)$$

where $H_{p,f}$ denotes the series of normalised histogram encodings, indexed by $p$ drawn from the $M - 1$ database passes, and $f$ denotes the frame number within that pass. $K_D$ denotes the so-called kernelised version of distance measure [173]. To measure the localisation error, I used the ground-truth estimates that were acquired at the same time as the videos. The estimated position of a query, $L$, was simply taken to be that of the best match given by Eq. (12). However, in a more robust implementation, checks could be done that would require similar matches in neighbouring frames, both in query and pass.

**THE RSM DATASET**

In order to allow different approaches to be compared, and as a community resource to develop appearance-based methods for visual localisation, the RSM dataset was constructed and is made publicly available at [http://rsm.bicv.org][137]. In this section I will briefly review the existing datasets for visual localisation and explain the motivations behind the creation of the RSM dataset.

*Existing datasets*

Datasets for evaluating visual localisation methods have often been limited to demonstrate the performance of particular met-
rics such as point cloud accuracy [72, 113]. This has led to a number of datasets that were difficult to adapt to new work and different performance metrics, or simply unavailable because they were not released to the community [104, 119, 164].

**Historical Datasets**  
The early work described in Section 4.2.1 used custom-planned datasets for their specific evaluation objectives. This led to datasets [104, 119, 164] containing very short sequences, of few meters of length, that could not be used to assess localisation performance at realistic scale of indoor human navigation.

**SLAM Datasets and the Navvis Dataset**  
SLAM datasets, found in the robotics community, have a variety of scopes and recorded distances: large indoor spaces [162], outdoor itineraries [24], and up to the scale of a few km car ride [154]. They are also heterogeneous in terms of the precision and nature of the ground truth: some use GPS, others the Microsoft Kinect to capture depth [162], while others use the Vicon motion capture system. While the ground truth is often precise (up to the level of GPS, Kinect or Vicon precision), these have usually targeted outdoor comparisons; indoor comparisons focused at geometric reconstruction or pose estimation rather than localisation.

To the best of our knowledge, with the exception of NAVVIS [72], SLAM datasets have had rather restricted distances, not addressing real-world navigation on the scale of large buildings. The NAVVIS project described in Section 4.2.2 first introduced a more generic dataset that could evaluate visual localisation and navigation at human scale for robotic applications. The RSM dataset takes the evaluation and the principle closer to the assistive context than the robot-centric approach of the NAVVIS team: our data and evaluation context introduces the particularities of human motion, both from hand-held and a wearable camera.
The RSM dataset of visual paths

I have previously defined a visual path as the video sequence captured by a moving person in executing a journey along a particular physical path. For the construction of our dataset, the RSM dataset of visual paths, a total of 60 videos were acquired from 6 corridors of a large building, the Royal School of Mines at Imperial College London. A layout of the trajectories is reproduced in Figure 17. In total, 3.05 km of data is contained in this dataset at natural indoor walking speeds. For each corridor, ten passes (i.e. 10 separate visual paths) are obtained. These are acquired with two different devices with 30 videos each. One device was an LG Google Nexus 4 phone running Android 4.4.2. The video data was acquired at approximately 24-30 fps at two different resolutions, 1280 × 720 and 1920 × 1080 pixels. The second device was a Google Glass (Explorer edition) acquiring at a resolution of 1280 × 720, and at a frame rate of 30 fps. Table 6 summarises the acquisition. As can be seen, the length of the sequences varies within some corridors, due to a combination of different walking speeds and/or different frame rates. Lighting also varied, due to a combination of daylight/night-time acquisitions, and occasional prominent windows that represent strong lighting sources in certain parts of some corridors. Changes were also observable in some videos from one pass to another, due to the presence of changes and occasional appearance from people. In total, more than 90,000 frames of video are labelled with positional ground truth in a path relative manner.

Ground truth acquisition

A surveyor’s wheel (Silverline) with a precision of 10 cm and error of ±5% was used to record distance, but was modified by wiring its encoder to a Raspberry Pi model B running a number of measurement processes. The Pi was synchronised to network time using the network time protocol (NTP) enabling synchronisation with timestamps in the video sequence. Because
of the variable-frame rate of acquisition, timestamp data from the video was used to align ground-truth measurements with frames. This data was used to access the accuracy of associating positions along journeys through frame indexing and comparison.

**EXPERIMENTS**

*Performance evaluation*

The methods for a) describing spatial or space-time structure, b) indexing and comparing the data are summarised in Table 7. The choice of parameters was selected to allow a) as consistent a combination of methods as possible, allowing fair comparisons of the effect of one type of encoding or spatio-temporal operator to be isolated from others b) to select parameter choices close to other research in the area, e.g. for image categorisation,
4.5 Experiments

<table>
<thead>
<tr>
<th>Photo</th>
<th>Length (m)</th>
<th>No. of frames</th>
<th>Avg</th>
<th>Min</th>
<th>Max</th>
<th>Avg</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Avg</td>
<td>Min</td>
<td>Max</td>
<td>Avg</td>
<td>Min</td>
<td>Max</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1</td>
<td>57.9</td>
<td>57.7</td>
<td>58.7</td>
<td>2157</td>
<td>1860</td>
<td>2338</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>31.0</td>
<td>30.6</td>
<td>31.5</td>
<td>909</td>
<td>687</td>
<td>1168</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C3</td>
<td>52.7</td>
<td>51.4</td>
<td>53.3</td>
<td>1427</td>
<td>1070</td>
<td>1777</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>49.3</td>
<td>46.4</td>
<td>56.2</td>
<td>1583</td>
<td>1090</td>
<td>2154</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C5</td>
<td>54.3</td>
<td>49.3</td>
<td>58.4</td>
<td>1782</td>
<td>1326</td>
<td>1900</td>
<td></td>
<td></td>
</tr>
<tr>
<td>C6</td>
<td>55.9</td>
<td>55.4</td>
<td>56.4</td>
<td>1471</td>
<td>1180</td>
<td>1817</td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>3.042 km</td>
<td>90,302 frames</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Table 6: A summary of the dataset with thumbnails

the chosen \(^1\) dictionary sizes of \(\approx 256, \approx 400\) and \(\approx 4000\) words are common.

**Error distributions**

Error distributions allow us to quantify the accuracy of being able to estimate locations along physical paths within the RSM dataset described in Section 4.4. To generate the error distributions, the following method was used:

Using the kernels introduced in Section 4.3.6, location inference was attempted. One kernel is shown in Figure 23, where the rows represent each frame from the query pass, and the columns represent each frame from one of the remaining database passes of that corridor. The values of the kernel along a row represent a “score” between a query and different database frames.

---

\(^1\) Large dictionary sizes for an ambiguous dataset such as RSM might not seem adequate. However, internal experiments with smaller dictionary sizes (64 for VLAD and 400 for HA) showed poorer results and were not included in the comparison.
<table>
<thead>
<tr>
<th>Method</th>
<th>ST</th>
<th>Dense</th>
<th>Dim</th>
<th>Encoding</th>
<th>Metric</th>
</tr>
</thead>
<tbody>
<tr>
<td>KP-SIFT</td>
<td>No</td>
<td>No</td>
<td>128</td>
<td>HA-4000</td>
<td>$\chi^2$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VLAD-256</td>
<td>Hellinger</td>
</tr>
<tr>
<td>DSIFT</td>
<td>No</td>
<td>Yes</td>
<td>128</td>
<td>HA-4000</td>
<td>$\chi^2$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VLAD-256</td>
<td>Hellinger</td>
</tr>
<tr>
<td>SF-GABOR</td>
<td>No</td>
<td>Yes</td>
<td>136</td>
<td>HA-4000</td>
<td>$\chi^2$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VLAD-256</td>
<td>Hellinger</td>
</tr>
<tr>
<td>LW-COLOR</td>
<td>Yes</td>
<td>No</td>
<td>144</td>
<td>N/A</td>
<td></td>
</tr>
<tr>
<td>ST-GABOR</td>
<td>Yes</td>
<td>No</td>
<td>221</td>
<td>HA-4000</td>
<td>$\chi^2$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VLAD-256</td>
<td>Hellinger</td>
</tr>
<tr>
<td>ST-GAUSS</td>
<td>Yes</td>
<td>Yes</td>
<td>136</td>
<td>HA-4000</td>
<td>$\chi^2$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VLAD-256</td>
<td>Hellinger</td>
</tr>
<tr>
<td>HOG3D</td>
<td>Yes</td>
<td>Yes</td>
<td>192</td>
<td>HA-4000</td>
<td>$\chi^2$</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td>VLAD-256</td>
<td>Hellinger</td>
</tr>
</tbody>
</table>

Table 7: A summary of the different encoding methods and their relationships to different descriptors. The number of elements of each descriptor is also reported (Dim).

given by applying the kernel mapping functions described in [173] for the $\chi^2$ and Hellinger kernel depending on the case. In this experiment, the position of the best matching database image to the query frame was calculated. The absolute error between this and the ground truth $|\epsilon|$, in m, is then determined. I used the ground-truth information acquired as described in Section 4.4.

In order to characterise the reliability of such scores, bootstrap estimates of error distributions were obtained by using 10 million trials. Specifically, permuting the paths that are held in the database and randomly selecting queries from the remaining path, these error distributions in localisation can be obtained. The bootstrap estimates consisted of 1,000 repeated runs with random selections of 10,000 frames each, revealing the variability in these estimates, including that due to different numbers of paths and passes being within the database. In Appendix A.1 I describe the algorithm to generate the cumulative distribution functions (CDFs) in more detail. The distribution of the errors gives us a probability density estimate, from
which one can get the cumulative distribution function (CDF) \( P(|e| \leq x) \).

The outcome is shown in Figure 26, where the variability in the lines indicate the range of the results obtained during permuting the observations as described above.

![Figure 23: Example of a \( \chi^2 \) kernel produced by hard assignment and using the SF-GABOR descriptors when querying with pass \( P_1 \) of corridor \( C_2 \) against a database comprised of passes \( P_{2-10} \).](image)

Let us consider the idea of crowdsourcing journey information from many pedestrian journeys through the same corridors. This approach to evaluating the error thus makes sense: all previous journeys could be indexed and held in the database; new journey footage would be submitted as a series of query frames (see Figure 24).

All the results were generated with a downsampled version of the videos at \( 208 \times 117 \) pixels.
Figure 24: Diagram illustrating the nature of visual paths and queries. There are different paths recorded in the databases.
RESULTS

I calculated the average absolute positional error (in metres) and the standard deviation of the absolute positional errors across the provided dataset, and these are shown in Table 8. For these errors, all queries, by a leave-one-out strategy, have been used, but there is otherwise no random sampling of the queries. Standard deviations of the absolute errors are also provided. Table 8 also provides the Area-Under-Curve (AUC) values obtained from the CDFs of Figure 25.

Localisation error vs ground-truth route positions

As described in the previous section, by permuting the database paths and selecting, randomly, queries from the remaining path that was left out in the dictionary creation, one can assess the errors in localisation along each corridor for each pass, and calculate, also, the average error in localisation on a per-corridor basis, or per-path basis. For these, the ground-truth information acquired as described in Section 4.4 was used. Figure 27 provides some examples of the nature of the errors, showing evidence of those locations that are often confused with each other. As can be seen, for the better method (top trace of Figure 27) whilst average errors might be small, there are, occasionally, large errors due to poor matching (middle trace). Errors are significantly worse for queries between different devices (see Figure 27(c)).

Performance summaries

I calculated the average of the absolute positional error (in m) and the standard deviation of the absolute positional error in a subset of the complete RSM dataset (Table 8). I used a leave-one-journey-out approach (all the frames from an entire journey are excluded from the database). Also using bootstrap sampling, I also estimated the cumulative density functions of the
Figure 25: Comparison between the error distributions obtained with the different methods. Note the high reproducibility of the performance results. The origin of the variability within each curve is explained in Section 4.5.1.1.

error distributions in position, which are plotted separately in Figure 25 and in the same graph in Figure 26. The variability in these curves is shown, although it is hard to appreciate. It has been therefore summarised in the last two columns of Table 8 through the area-under-curve (AUC) values. In the best case (SF-GABOR), AUCs of the order of 96% would mean er-
errors generally below 2 m; in the worst (HOG3D), AUCs \( \approx 90\% \) would mean errors of around 5 m. These mean absolute error estimates are obtained as the queries, the dictionary and the paths in the database are permuted.

<table>
<thead>
<tr>
<th>Method</th>
<th>Error summary (m)</th>
<th>AUC (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>( \mu_\varepsilon )</td>
<td>( \sigma_\varepsilon )</td>
</tr>
<tr>
<td>SF-GABOR</td>
<td>1.59</td>
<td>0.11</td>
</tr>
<tr>
<td>DSIFT</td>
<td>1.62</td>
<td>0.11</td>
</tr>
<tr>
<td>KP-SIFT</td>
<td>2.14</td>
<td>0.17</td>
</tr>
<tr>
<td>LW-COLOR</td>
<td>3.64</td>
<td>1.13</td>
</tr>
<tr>
<td>ST-GAUSS</td>
<td>2.11</td>
<td>0.24</td>
</tr>
<tr>
<td>ST-GABOR</td>
<td>2.54</td>
<td>0.19</td>
</tr>
<tr>
<td>HOG3D</td>
<td>4.20</td>
<td>1.33</td>
</tr>
</tbody>
</table>

Table 8: Summaries of average absolute positional error and standard deviation of positional errors for different descriptor types. \( \mu_\varepsilon \) is the average absolute error, and \( \sigma_\varepsilon \) is the standard deviation of the error in metres. Top: single-frame methods. Bottom: spatio-temporal methods.
Figure 27: Estimated location vs. ground truth. Illustrative examples of good/bad location estimation performance. a) Uses the best descriptor and a single-device dataset, b) uses the best descriptor and a cross-device dataset and c) uses the worst descriptor, and a multiple-device dataset.
Note that tracking algorithms were not used, and so there is no motion model or estimate of current location given the previous one. As it will be described in Section 4.7, incorporating a particle filter or Kalman filter should reduce the errors, particularly where there are large jumps within small intervals of time. This deliberate choice allows us to evaluate the performance of different descriptor and metric choices independently.

The results show that localisation is achieved with good accuracy in terms of CDF and AUC without a large difference between the applied methods, despite the big diversity in their complexity. Absolute errors show significant differences between methods, with average absolute errors in the range of 1.5 m to 4.20 m. Single-frame methods (SF-GABOR, KP-SIFT and DSIFT) perform slightly better than spatio-temporal approaches (LW-COLOR, ST-GABOR, ST-GAUSS and HOG3D). This is not surprising, as the spatio-temporal methods might be strongly affected by the self motion over fine temporal scales.

Dense methods are known to provide a better coverage of the scene and in principle should show better performance [167]. However, the performance of KP-SIFT is in the range of its dense counterpart and above other dense methods such as ST-GAUSS, ST-GABOR and HOG3D, thus casting doubt on the need for such a dense sampling (≈ 2,000 descriptors per image).

In spite of using image retrieval methods in isolation, the attainable accuracy appears to be in line with those of other methods reviewed in Section 4.2. However, previously reported methods include tracking, the use of other sensors, or estimates of motion. In this work, I emphasise that no form of tracking was used in estimating position: this was deliberate, in order to assess performance in inferring location from the visual data fairly. Introducing tracking would probably improve localisation performance, and could reduce query complexity. Yet, tracking often relies on some form of motion model, and for pedestrians carrying or wearing cameras, motion can sometimes be relatively unpredictable. In the next section I will describe the comparison of the appearance-based approaches with
SLAM methods, particularly through comparing performance with large-scale direct monocular SLAM (LSD-SLAM).

**Comparison with SLAM**

*Description of the experiments*

The first attempt to put the appearance-based approaches into perspective consisted of applying one implementation of the simultaneous localisation and mapping technique (SLAM) to the RSM dataset, at the same frame resolution as for the appearance-based localisation discussed in this chapter. I chose the “EKF Mono SLAM” [39], which uses an extended Kalman filter (EKF) with 1-point RANSAC. This implementation was selected for three reasons: a) it is a monocular SLAM technique, so comparison with the single-camera approach is fairer; b) the authors of this package report error estimates – in the form of error distributions; and c) the errors from video with similar resolutions (240 × 320) to the small images (208 × 117) used for the appearance-based matching were reported as being below 2 m for some sequences [39] in their dataset.

The results of the comparison were surprising, and somewhat unsatisfactory. The challenging ambiguity of the sequences in the RSM dataset, and possibly the low resolution of the queries, might explain the results. The feature detector, a FAST corner detector [144], produced a small number of features in its original configuration. I lowered the feature detection threshold until the system worked on a small number of frames from each sequence. Even with more permissive thresholds, the average number of FAST features averaged only 20 across the experiments of the RSM dataset. This small number of features led to inaccuracy in the position estimates, causing many of the experimental runs to stop when no features could be matched. The small number of features per frame is also not comparable with the feature density of the methods described in this work, where an average of 2,000 features per frame was obtained for
the “dense” approaches. Dense SLAM algorithms might fare better, and that was the reason why LSD-SLAM was chosen, as I will describe in the next section.

Comparison with state-of-the-art SLAM: LSD-SLAM

I compared the best performing appearance-based method, SF-GABOR, against the current state-of-the-art in SLAM for indoor sequences, LSD-SLAM. I first ran the LSD-SLAM code over the 60 video sequences of the RSM dataset and retrieved the results from the visual odometry engine in order to obtain position estimates for each processed frame. This provided an estimate of the distance travelled by a user, allowing comparison to the results for the appearance-based method.

The cameras in the devices used to acquire the RSM data are quite different for those suggested for use with LSD-SLAM (recommended: monochrome global-shutter camera with fish eye lens); therefore, the standard LSD-SLAM software was modified to recover from lost tracking when this happened. The semi-dense SLAM parameters were adjusted to adapt best to our conditions and data, specifically, to minimise instances of lost tracking in medium resolution versions (1024 × 576) of the images, i.e. at a considerably higher resolution than the image required for the appearance-based approach. In A.3 I provide the parameter values that were used for the experiments.

Performance of LSD-SLAM on RSM dataset sequences

Figure 28a and 28b illustrate the localisation performance of LSD-SLAM (blue) with respect to the ground truth (red). The two cases that are shown were selected to illustrate that accuracy of localisation can vary, depending on the specific corridor within the RSM dataset (see Section 4.4). As can be seen from the figures, in the better case, the absolute error is below 2 m. However, in some cases, errors rise as high as 5 m. Perhaps equally importantly, these errors were found when operating at image sizes that allowed LSD-SLAM to function without losing tracking.
Figure 28: Localisation performance in LSD-SLAM; this shows that in different corridors, the accuracy of LSD-SLAM can change quite significantly. See text for details of the SLAM parameters and the nature of the dataset, but note that these were obtained at an image resolution of 1024 × 576 pixels. At lower resolutions, loss of tracking dominated the experiments. The difference in the x and y axis labelling is because experiments in a) and b) are obtained from two different corridors with different lengths and different number of frames.
CDF comparisons

CDFs, as introduced in Section 4.5.1.1 illustrate very well the differences in the performance of methods. As we can see from Figure 29 the appearance-based method using Gabor-based descriptors performs better than LSD-SLAM in the RSM dataset.

Another perspective to these results can be seen from Table 9, where we can see a head to head comparison for the probabilities of achieving a localisation error smaller than x in metres. We can see how SF-GABOR achieves smaller localisation errors than LSD-SLAM with a larger probability. For the sample of CDF values obtained, we can see that SF-GABOR can localise with an error below 2.5 m in the 95.95% of the times, whilst LSD-SLAM can only guarantee errors below 2.5 m 59.88% of the times on average.

This does not imply that the appearance-based technique used should replace SLAM or its equivalents – it is merely a different context of usage. A user who is interested in getting from A to B may be less interested in mapping the geometry along that route than whether their trajectory matches that of other users who have previously made the same journey. In the
Table 9: Cumulative distribution function values against localisation error in metres (\(\epsilon\)). \(P(\mid \epsilon \mid \leq x)\), expressed as a percentage. From this table, the best appearance-based method achieves a probability of 90% of localising with an error below 2 m, whilst LSD-SLAM achieves just above 50% accuracy level for that error boundary, and required images 5 times larger. In addition, the performance of LSD-SLAM was significantly worse (compare the minimum performance columns) on some corridors and journeys, bringing the overall average down across the RSM dataset relative to appearance-based localisation.

scenario described, and in which performance is being compared, the journey of a user is assessed against those made by other people who have made the same journey – location becomes journey-relative, not map-relative. The usage scenario also means that loop closure may not be possible. However, localisation systems may aim for a combined solution, as LSD-SLAM seems to perform better given an appearance-based loop closure method [51] such as FabMap, which could be replaced by any of the solutions proposed [139] or a combination of both.

Reproducibility errors

Another comparison is illustrated by Figure 30. These box-and-whisker plots evaluate the comparative reproducibility in localisation between the best appearance-based method, SF-GABOR,
and LSD-SLAM within RSM corridors 1 and 3 (C1 and C3) for multiple “leave-one-out” passes. The plots suggest that whilst LSD-SLAM yielded worse results in terms of error, it has a consistency in performance that is comparable to that of SF-GABOR. Also, the errors for LSD-SLAM rarely go beyond 5 m, with an average of \( \mu_e = 2.48 \pm 2.37 \) m. Conversely, the appearance-based method contains some outliers; even so, for some sequences the error is of the same order of magnitude or lower (\( \mu_e = 1.59 \pm 0.11 \) m) than the best reported for SLAM, supporting the idea of an appearance-based localisation approach for indoor navigation.

**Area-Under-Curve comparisons**

In Section 4.6.2 I calculated the average absolute positional error (in m) and the standard deviation of the absolute positional errors for a variety of methods. Here, I reproduce the AUC results for the method SF-GABOR, which ranged from 96.11% to 96.39%. For the case of LSD-SLAM, the AUC ranged from 89.71% to 90.61% All queries were again performed by adopting the leave-one-out strategy, but because of the high repeatability of results, random frame-level sampling was not applied.

**Conclusion**

I have presented three main contributions to the topic of indoor localisation using visual path matching from wearable and hand-held cameras. In first place, a complete evaluation of six local descriptor methods is provided: three custom designed and three standard image (KP-SIFT and DSIFT) and video (HOG3D) matching methods as baseline. These local descriptions follow a standard bag-of-words and kernel encoding pipeline. The code for both the local descriptors and for the evaluation pipeline is available online [140]. In second place, the RSM dataset is also made available, a large dataset of more than 90,000 video frames with positional ground truth of indoor journeys to complete the evaluation framework. The dataset
Figure 30: Box-and-whisker plots depicting the errors obtained in two corridors, using either LSD-SLAM or appearance-based matching using SF-GABOR descriptors. The top row corresponds to the appearance-based result. The bottom row corresponds to LSD-SLAM. On each graph, the horizontal positions correspond to different journeys down the same corridor when the remainder of journeys is used as a database of journeys. Each of these positions represents the statistics of 100 random image queries. These graphs suggest that LSD-SLAM and an appearance-based approach are comparable in terms of reproducibility of localisation within the same corridor. Note, however, that much lower spatial resolution (less than 1/4 of the image size) is used for the appearance-based technique than for LSD-SLAM.
is freely available at http://rsm.bicv.org. In third place, I introduce the CDFs of the error as the probability of the error being less than a certain threshold. This way of reporting localisation errors is a different performance metric than the usual average error with respect to the ground truth (although this is also provided) found in SLAM and other localisation approaches.

The results show that there is significant localisation information in the visual data, and that errors as small as 1.5 m over a 50 m distance can be achieved, even without tracking. As mentioned above, the results have been reported in two ways: a) average absolute positional errors, and b) error distributions, both of which allow image descriptions to be assessed for their localisation capability. The latter could also be used to build a measurement model for inclusion in a Kalman or particle filter aimed at supporting human ambulatory navigation.

As a baseline for these results, I provide a comparison with the state of the art method in SLAM, LSD-SLAM, with the best appearance-based method, SF-GABOR. It was found, rather surprisingly, that appearance-based localisation appeared to be at least as accurate as that of LSD-SLAM without loop closure over a distance of around 50 m. In particular, SF-GABOR presents a lower average localisation error ($\mu_{e, \text{SF-GABOR}} = 1.59$ m $< \mu_{e, \text{LSD-SLAM}} = 2.48$ m). It was also found that appearance-based localisation was achievable with low resolution images, of around 30,000 pixels per image – a 5% of the number of pixels in the frames used for the LSD-SLAM experiment. This lowers the computational burden, a potentially important factor in an assistive context where device power autonomy can hinder the use of power-hungry computer vision algorithms. In addition, the small file sizes required for the appearance-based localisation approach reduces the bandwidth and storage required for crowdsourcing data. Results show that 1,500-frame sequences, sufficient for a 50 m corridor at normal walking speeds, consumed no more than 2 MB once compressed, meaning that the journey segments required can feasibly be crowdsourced from several users within a building.
We plan to introduce tracking in future work. There are, of course, numerous other enhancements that one could make for a system that uses visual data; integration of data from other sensors springs to mind, such as inertial sensing, magnetometers and RSSI. Although fusing independent and informative data sources would theoretically lead to improvements in performance, we would argue that the methods applied to infer location from each information source should be rigorously tested, both in isolation and as part of an integrated system. This would help ensure that real-world systems would be somewhat robust to sensor failure. We anticipate that using vision to associate locations in the journeys of several users through their visual paths could play an important role in navigation. In fact, in the next chapter I delve deeper into the importance of appearance-based retrieval techniques in localisation and explore the similarities of these techniques with those present in biology. I develop a model of biological mimicry to configure a localisation system inspired in biological place cells found in mammals.
INTRODUCTION

Animals use a variety of environmental cues in order to recognise their location. One of the key behaviours found in a certain type of biological neuron — known as place cells — is a rate-coding effect: a neuron’s rate of firing decreases with distance from some landmark location. In this chapter, I used visual information from wearable and hand-held cameras in order to reproduce this rate-coding effect in artificial place cells (APCs).

In order to achieve place-cell modelling, it was necessary to develop an efficient algorithm for searching tensor representations of the population model for the cortical area V1 in joint space-time-frequency encodings of video sequences. This algorithm is based around the use of a dictionary technique, which is also used for the place-cell modelling. Specifically, the location inference system based on APCs uses dimensionality reduction of the tensor-valued representations of population codes through vector quantisation. Kernel similarity measures are then applied to the descriptions of population codes in order to mimic biological place-cell behaviour.

This computational framework effectively replaces the first elements of the BOVW pipeline described Chapter 4. Additionally, it enhances the BOVW approach — in which localisation was based on the location of the best match — with a population coding approach: a population of artificial place cells.

The accuracy of localisation using these APCs was evaluated using the different visual descriptors presented in Chapter 4 and different place cell widths. Simple localisation using APCs was feasible by noting the identity of the APC yielding the maximum response. I also propose using joint coding within a num-
ber of automatically defined APCs as a population code for self-localisation. Using both approaches it was possible to demonstrate good self-localisation from very small images taken in indoor settings (the RSM dataset). The error performance using APCs is favourable when compared with ground-truth and LSD-SLAM, even without the use of a motion model.

Motivation

This work explores the potential of location-sensitive computational units to achieve self-localisation by using the images from wearable or hand-held cameras. The approach in this chapter is motivated by studies of biological vision which suggest that, in many species, multiple strategies are employed to help animals self-localise. Examples include the use of the visual horizon and path integration in ants [114] and the use of optic flow in insects and birds [89, 18]. Indeed, multiple computational approaches appear to be simultaneously at work even within a single species and for one sensing modality, such as vision. Such approaches can be successfully transferred from biology into computer vision; see for example, the combined use of optic flow and image descriptors that were suggested to mimic the visual homing system of insects [171].

Biological place cells display location-dependent firing. I describe how to mimic place-cell behaviour from appearance information in order to provide localisation within an indoor environment. In terms of computer vision, the research question might be articulated as:

*Given a video sequence taken by a person as they walk, can we generate artificial place cell responses that are able to localise that person with respect to previous journeys?*

Summary of contributions

The approach to indoor localisation described in this chapter is novel for several reasons.
1. First, I describe a convolutional neural network – formalised through using *tensor convolution* – to provide a description of spatial processing in the form of a population code [17]; this code mimics the dependency of average firing rate of simple cells in visual cortical area V1, an important and extensive visual region in primates and man, under different patterns of visual stimuli. This CNN effectively implies a new formulation for SF-GABOR that shows the links between the biology of place cells and convolutional neural networks (CNN). To accomplish this, I describe SF-GABOR in terms of a tensor operation between input images and a series of spatial filters. The work is relevant to recent developments in neural network architectures, and in particular to convolutional neural networks [90, 92]. Specifically, Kendall et al. [83] have applied deep convolutional neural networks to camera pose regression, therefore learning location and orientation simultaneously. Also using monocular commodity cameras, Giusti et al. [58] train a deep neural network as a supervised image classifier to compute the main direction of travel within a trail compared to the viewing direction. These are, to our knowledge, the first applications of convolutional networks for self-localisation from monocular visual data. These methods achieve extraordinary localisation results in the order of 2 m. However, they often lack a theoretical formulation of the operators used in their neural networks. The tensor operations described in this chapter represent an effort towards an unambiguous theoretical tool that can help formalise the operators used in state-of-the-art experimental methods.

2. Second, I demonstrate the use of the V1 population code model and simple learning techniques to model place-cell behaviour; this allows us to model biological place cells (BPCs) with artificial place cells (APCs), producing a gentle decay in the strength of a response (firing rate) with the spatial distance of an observer to particular landmarks.
These landmarks are implicit and consist of orientation patterns within scenes. In a more general view, the computational units of this behaviour, the APCs, are able to use distinctive information from image queries based on the recall of previously visited places.

3. Third, I describe a complete pipeline of visual localisation that incorporates a decoder for the APC activity; this decoder takes the form of a Generalised Regression Neural Network (GRNN).

4. Finally, I demonstrate self-localisation and provide an evaluation of the proposed localisation method using the RSM dataset described in Section 4.4.

BACKGROUND

Navigation is one of the more complex tasks performed by animals; it involves integration of multiple sensory inputs, combining past information (memory) and also the execution of physical actions to perform navigation movements.

John O’Keefe, together with Edvard and May-Britt Moser, received the Nobel Prize in Physiology or Medicine (2014) for their discovery and subsequent elucidation of place and, subsequently, grid cells in the brain, respectively. Their early findings suggested that navigation in the moving rat is based on a cognitive map of the environment; in this map, a set of landmarks is created, and the spatial relationship between these is used to navigate [81]. The implications are interesting, partly for the potential importance in understanding some forms of dementia, but also because the cells encoding an organism’s own spatial position are found in one of the less understood areas of mammalian brains: the hippocampus, with a key role in memory, and its adjacent brain areas.

Humans, like other animals, need a sense of position to perform basic interaction with the environment. This interaction is often finding our way from one place to another and place
awareness is integrated with distance and direction information to navigate.

**Biological place cells (BPC)**

Place cells are special types of neurons located in the hippocampus which attain higher-than-average firing when an animal “recognises” a particular place in its environment. Grid cells, located next to the hippocampus in the entorhinal cortex, provide the brain with a reference system for navigation, a “grid” that is speculated to be used as a form of coordinate system for the creation of spatial maps. Although the existence of place and grid cells is without question, the computational description of what the cells do is debatable. However, the combination of place and grid cells within neural circuitry appears crucial for the execution of navigation tasks.

The physical region within a spatial environment over which a given place cell shows elevated firing is sometimes referred to as its *place field*, though the term may also refer to the mapping between an animal’s location and a cell’s firing rate. The combination of multiple place fields yields a spatial map, and multiple spatial maps formed by combinations of place cell activity patterns are thought to be stored in the hippocampus. It is the unique combination of place cell firing patterns in a specific order during movement that gives rise to a unique spatial representation of a journey.

It was when studying the entorhinal cortex looking for similar place coding cells when the Mosers discovered the grid cell type, with unprecedented properties [65]. Specially, they present multiple-locations firing pattern with hexagonal shape, thought to be part of a navigation or path integration system with distance measuring and coordinate system properties. Apart from grid cells, there are other cells in the entorhinal cortex that have a spatial function. These are head direction cells, which act similarly to a compass; border cells, active in reference to boundaries; and combined cells. The Mosers demonstrated that
all these cells project to the hippocampus, concretely to the CA1 area where place cells are located [190]. This corroborates the studies that show that entorhinal cortex cells that encode spatial information, specially border cells, play a role in the firing activity of place cells [27].

The evidence is that many cues, and perhaps even self-motion itself, may be involved in forming the observed location-selective response of biological place cells. The visual information captured by the eyes should be seen as only one of the many sensory and internal cues that lead to the spatially selective nature of biological place cell responses [68]. Nevertheless, in many animals, and certainly in humans and primates, vision is a particularly strong environmental cue to an organism’s awareness of its location [53].

\textit{Appearance-based methods as models of sensory inputs to place cell models}

\textit{Gradient operators as V1 receptive field models}

In [27], Bush et al. establish the idea that grid cells and place cells are not successive links of a chain when performing localisation and navigation tasks, but complementary and interconnected processing units to encode spatial maps. Importantly, they claim that place cell spatial firing is determined by sensory inputs, among which vision plays a major role.

This motivated the creation of models of place cell encoding patterns from elements of the computer vision research that modelled visual cortex representation. An exceptional case is the one of gradient operations, present in SIFT and SIFT-like descriptors, as models of the pyramidal neurons in the primary visual cortex (V1), which exhibit strong direction selectivity, spatial phase invariance and response inhibition [71, 45, 29].

The seminal work of Hubel and Wiesel [71] proposed that the response of V1 neurons is produced by stimuli of higher complexity than ganglion cells in the retina and the lateral geniculate nucleus (LGN). In particular, it is known that simple and
complex cells in the V1 display orientation selectivity produced by bars or edge stimuli [126].

Orientation selective simple cells can be modelled as the output of a 2D Gabor function [42], and through tuning their parameters, different combinations of orientation and phase can be achieved. In particular, the steerability of the Gabor filter can represent the orientation selectivity of these neurons, whilst the phase parameter of the filter can be viewed as their shape selectivity.

I have therefore studied the use of the 2D Gabor-like filters, which have been used for a great variety of applications in computer vision, from action recognition [153] to face recognition [95] and appearance-based indoor localisation (see Chapter 4 and [139]), and often as part of a biologically inspired system [153]. Mathematically, the over-completeness of Gabor outputs, i.e. multiple Gabor output values to a single pixel, provides advantageous invariance properties for descriptors computed on local image patches. In particular, the main benefit of using Gabor filters resides in the combination of symmetric and antisymmetric responses that yield a description of local regions in an image that are either phase selective or phase invariant.

The real part of these filters presents the simple-cell receptive fields behaviour described above. In Figure 31 I illustrate the orientation selectivity of V1 simple-cell receptive fields, and an example 2D Gabor fit spatial and 2D views.

Figure 31: Orientation sensitive simple cells which only respond to a bar of certain size and orientation.

Also relevant to this study, the difference of Gaussians (DoG) space representation found in the scale invariant feature transform (SIFT) keypoint detection [98] may be seen as an approx-
imation of the spatial receptive field of a retinal ganglion cell. Lowe also suggested that the process behind the computation of the orientation of the frequencies is similar to the behaviour of complex cells in V1. In particular, each bin of the histogram of oriented gradients (HOG) can be seen as a single orientation-selective neuron.

*Biologically inspired visual localisation*

Biologically-inspired methods of localisation from image data are emerging; for example, a few computational models for place-cell behaviour already exist, though they are often rooted in dynamical systems [22], [16]. Some models of place cells use attractor properties of recurrent networks [160], [112]. Whilst interesting and valuable, the role of sensory input is marginalised in these models, a key differentiator of the approach proposed in this chapter.

The boundary vector cell (BVC) model [14] is a popular computational model that describes place-cell response, allowing predictions to be made and experimentally tested [26]. Whilst of substantial interest in computational neuroscience, one criticism of the BVC model is – like the dynamical systems models – a lack of detail in explaining how sensory processing feeds into the computation. The closest work to the approach described in this thesis is Strosslin’s [161], which uses a low-cost and computationally efficient model to navigate a robot. However, place-cell behaviour – in terms of firing rates that vary with the position to some reference location – is not demonstrated except through the policy of a robot and its navigation attempts. The visual processing is also rather limited, using techniques that are also employed in SLAM algorithms [6] to track specific scene locations.

Milford and colleagues have also applied SLAM techniques to biologically inspired localisation systems, setting a seminal precedent with RatSLAM [109], a persistent navigation and mapping model based on modelling the hippocampus of rodents.
RatSLAM continuously performs SLAM while simultaneously interacting with other navigation systems, such as odometry and landmark detection. More recent work [108] has been focused on taking RatSLAM to larger scales and incorporating more complex visual landmark detection models.

A VISUAL INPUT MODEL BASED ON CNNS

Several authors have pointed to the similarities between spatial filter banks employing oriented band-pass filters [177, 19] – many of which are implemented using spatial convolution – and the receptive field patterns found in the primary visual cortex (area V1). These receptive fields are sensitivity patterns to visual stimuli that can be found within individual neurons [122, 135].

The recent interest in convolutional neural networks [90, 77, 82, 83, 58] builds on over two decades of work into architectures and methods for training [92]. Recent results in CNN training tend to converge to yield first-layer weights (filters) that are remarkably similar oriented spatial band-pass filters, and therefore also bear striking resemblance to the receptive fields found in biological visual systems at the level of V1 or equivalent.

I used a two-layer feed-forward convolutional neural network – adapted from a model used in computational neuroscience – to simulate the behaviour of a sheet of tissue in V1 containing just under 200,000 neurons. The first layer of this network represents orientation selective, approximately linear simple cells, and the second layer models a population code for joint position/orientation encoding based on the retinotopic arrangement of oriented cells over a region of cortical tissue. However, rather than learning the weights from random initialisation, neurons were constrained to be orientation selective and with spatially antisymmetric weights about the axis of orientation selectivity (see Figure 32). It is useful now to introduce a tensor description of the data structures and the processing involved in this CNN.
Tensor population model

Tensors have been suggested as a multi-dimensional representation for convolutional neural networks, specifically for representing the stack of spatial filters that is used to perform spatial convolution. Representing a filter set in this way allows various tensor decompositions to be applied [86] to reduce the representation into smaller convolution operators. However, a definition for tensor convolution itself appears lacking from the literature. Two operators that – taken together – can be very useful concepts for reproducibly describing the structure of convolutional networks are tensor convolution and permuted tensor convolution. These are defined next and expanded in Appendix A.2.

Adopting the notation and nomenclature of Kolda [86], tensors are treated as multi-way arrays or multidimensional arrays. The order of a tensor is the number of dimensional indices required to address it; for example, an order 5 tensor $A$ may have addressable elements $a_{i_1,i_2,i_3,i_4,i_5}$, with each index varying from 1 to $I_n$, $n = 1, 2, 3, 4, 5$ in integer steps; note that in contrast with Kolda’s notation, indices are comma-delimited. Since each element of the tensor can be restricted to be real-valued, $A$ may be considered as lying in $I_1 \times I_2 \times I_3 \times I_4 \times I_5$-dimensional real space. The mode of a tensor refers to the tensor elements simultaneously addressed by one of the indices, and is applied to refer to operations that involve, possibly non-exclusively, a particular one of the indices.

Definition 1. Tensor Convolution The tensor convolution operator in modes $\mathcal{M}$ is defined by the following:

$$\mathcal{M}_{\star} : (A, B) \mapsto C,$$ (13)

where $\mathcal{M}$ is a set of $|\mathcal{M}|$ tuples representing paired indices of $A$ and $B$ over which the convolution is performed. These indices associate the modes of the tensors being convolved together, but $A$ and $B$ should be of equal order. Moreover, the dimen-
sions of the modes that are not participating in the convolution should be equal.

The tensor convolution operator maps equal-order tensors, $A$ and $B$, to a tensor $C$ by the following:

$$A^{[\mathcal{M}]}B = \sum_{i_1^M} \cdots \sum_{i_M} a_{i_1, i_2, \ldots, i_M} \times$$

$$b_{i_1, i_2, \ldots, i_M} = i_1^M, \ldots, i_M, \ldots, i_N$$

where $\mathcal{M}$, takes the form of a set of tuples that associate indices in $A$ with those in $B$ for the convolution:

$$\{(m_1, n_1), (m_2, n_2), \ldots, (m_M, n_M)\}.$$  

(15)

The order of the result, $N_C$, will be equal to the orders of $A$ and $B$.

**Comment** Tensor convolution is applicable to mapping a single frame or image through the V1 and population code model. It is therefore applicable to describing taking a single frame and estimating a place-cell response. For much of the work reported in this chapter, I evaluate entire video sequences. For this, permuted tensor convolution is a more appropriate operator for the first level of the model.

**Definition 2. Permutted Tensor Convolution** The permuted tensor convolution operator in modes $\mathcal{M}$ permuted over modes $\mathcal{P}$ can be defined as a mapping taking the form:

$$\mathcal{M}^{[\mathcal{P}]}: (A, B) \mapsto C,$$

(16)

where $\mathcal{M}$ is a set of $|\mathcal{M}|$ tuples representing paired indices of $A$ and $B$ over which the convolution is performed and $\mathcal{P}$ represents the modes of $A$ and $B$ which are permuted, expanding the order of $C$ relative to that of tensor convolution.
The permuted tensor convolution operator maps tensor, \( A \), to the higher-order tensor \( C \) by the following:

\[
A^\mathcal{M} \ast_{\mathcal{P}} B = \sum_{i_{m1}'} \ldots \sum_{i_{M}} a_{i_{1},i_{2},...,i_{m1}',...,i_{M},...,i_{p1},...,i_{pP},...,i_{N_A}} \times
b_{i_{1},...,i_{n1}-i_{m1}',...,i_{nM}-i_{mM}',...,i_{\pi(q1)p1},...,i_{\pi(qP)pP},...,i_{N_B}}(17)
\]

where \( \mathcal{M} \), consists of the tuples:

\[
\{(m_1, n_1), (m_2, n_2), ..., (m_M, n_M)\};
\]

and \( \mathcal{P} \) by the tuples:

\[
\{(p_1, q_1), (p_2, q_2), ..., (p_P, q_P)\}.
\]

The permutation operator \( \pi(i|j) \) denotes that the fibre number of the tensor in a particular mode are permuted. The order of the result, \( N_C \), will depend on the orders of the tensors \( A \) and \( B \), and the modes participating in the operator \( \mathcal{M} \) of \( \ast_{\mathcal{P}} \), according to:

\[
N_C = N_A + N_B - |\mathcal{M}|.
\]

Modes that do not participate in the convolution or the permutation must have equal dimension. Finally, where a third mode is implicit in one of the arguments of the permuted indices (such as for an order 2 tensor that is being permuted with an order 3 tensor), this is indicated by a “dot” symbol (e.g. \( \mathcal{P} = \{(-, 3)\} \)).

**Remark**  The modes of permutation may be thought of as describing the scaffolding of a convolutional network, and is related to the topology of a multi-layered network [34].
Simple-cell V1 CNN model

Having defined these two operations, they can be used to formulate the modelling of primary visual cortex and a simple population code using tensor convolution and permuted convolution.

Recognising that SIFT operates with vector fields of the form:

\[
\vec{\nabla} f(x, y; \sigma) = \frac{\partial f(x, y; \sigma)}{\partial x} \vec{x} + \frac{\partial f(x, y; \sigma)}{\partial y} \vec{y}
\]

\[
= \frac{\partial f(i_1, i_2; \sigma)}{\partial i_1} i_1 + \frac{\partial f(i_1, i_2; \sigma)}{\partial i_2} i_2 \tag{21}
\]

\[
= \bigcup_{k=1}^{2} D_k[f(i_1, i_2; \sigma)]i_k, \tag{22}
\]

where spatial dimensions \((x, y)\) are now represented by modes \(i_1, i_2\) in the tensor notation of Kolda \([86]\), and Eq. 22 follows from Eq. 21 because of the orthogonality of unit vectors \(\vec{x}\) and \(\vec{y}\). \(D_k\) is a derivative operator along dimension (mode) \(i_k\).

More generally, when the directional operators are not necessarily partial derivatives, the discrete spatial orientation tensor at scale \(\sigma\) may be introduced as:

\[
G_\sigma = \bigcup_{k=1}^{K} O_k[f(i_1, i_2; \sigma)]i_k. \tag{23}
\]

The operator \(O_k\) is some form of discrete, directional spatial operator. Eq. 23 generalises a two-dimensional gradient field at scale \(\sigma\); it permits more than 2 directions of peak angular sensitivity, and unlike the operator \(D_k\), there is no requirement that \(O_k\) be linear.

**Level 1: V1-like units**

Using grey-scale image sequences represented as an order 3 tensor, \(F\), an order 4 tensor \(G\) is constructed using the first-level of a convolutional network by:

\[
G_{\sigma,\lambda} \triangleq R_+ \left( F \left[ \frac{i_{1},i_{2}}{\{i,3\}} K_G \right] \right), \tag{24}
\]
where \([\ast]\) represents tensor convolution in the modes \(i_1\) and \(i_2\) (see Definition 4), and \(\lambda\) is a tunable spatial wavelength parameter. The image sequence, \(F\), is of dimensions \(I_1 \times I_2 \times N_F\), where \(N_F\) represents the number of image frames in the sequence. The tensor \(K_G\) holds convolution weights that model simple-cell behaviour. These weights are generally orientation selective in the image plane, one direction per slice of the third mode \((i_3)\) of \(K_G\); directions span the 2D plane. For the experiments reported in this thesis, I took \(K_G\) to be of dimension \(7 \times 7 \times 1 \times 8\); the small spatial scale of the tensor is in line with the filters used in recent convolutional networks, but is also quite appropriate for small images.

The function \(R_+(\cdot)\) is the one-sided ramp function applied element-wise to its tensor-valued argument. For an order 0 tensor, \(x\), (scalar) the ramp function is defined by

\[
R_+(x) = \frac{x + |x|}{2},
\]

i.e. it is a non-linear activation function; for a tensor with elements \(a_{i_1,i_2,...,i_N}\) it creates a tensor of the same order and size with the elements \(|a_{i_1,i_2,...,i_N}|\).

**Level 2: Spatial Poolers**

The second level of the convolutional network groups the orientation-selective outputs of the first level of convolution over different regions of a local image patch. It is obtained by applying a permuted tensor convolution between \(G_{\sigma,\lambda}\) and a pooling tensor \(P\):

\[
D \equiv G_{\sigma,\lambda}^{\{i_1,i_2\}}[\ast]_{\{i_3\}} P.
\]

The pooling tensor is also an order 3 tensor that defines 17 pooling regions with respect to each location in image space, distributed in a radial and angular fashion across a patch. The values in this tensor are visualised over a normalised neighbourhood of unit width and height in Figure 33. The resulting
5.3 A VISUAL INPUT MODEL BASED ON CNNS

Figure 32: This depiction of simple-cell receptive field (RF) model for V1 responses that captures only spatial – rather than temporal – properties in the plane of a captured image, illustrated here for a small patch of pixels. White areas indicate zones where a bright stimulus induces increased firing rate in a single neuron, dark areas represent inhibition of firing, and grey areas have null effect. Note that the centres of the 8 RFs shown here are actually centred at the origin of local image space (indicated by the red circle) in the first layer of the CNN. In a second layer, information is collected from different regions of the local patch and represented as a population code associated with the centre of the circle.

order 5 tensor, D, is of dimension $I_1 \times I_2 \times 8 \times 17 \times N_F$. Because the slices of this tensor create different pooling regions over a patch, it is distinctly different to the pooling operations of standard CNNs, which typically use a single fixed spatial smoothing operator for each pooling layer, and not multiple poolers for which weights are learnt.

Subsampling

Since the poolers are designed to be spatially smoothing, the subsampling operation can be applied directly to dimensions $i_1$ and $i_2$ of D. This maintains D as an order 5 tensor, but reduces the dimensions over the two modes representing in-plane spatial location.

In the notation used until Section 5.3, tensor D was used to generate the SF-GABOR descriptor, by sampling D every 3 pixels along modes $i_1$ and $i_2$, generating around 2,000 descriptor vectors per frame, each of 136 elements.
5.3 A VISUAL INPUT MODEL BASED ON CNNS

Figure 33: Illustration of the order 3 pooling tensor, visualised using transparency and colours to depict the spatial arrangement of values in the third mode. The 17 pooling regions corresponding to the $11 \times 11 \times 17$ tensor are divided in a) a central region, b) 8 “petals” at a distance $d_1$ from the center with 8 different orientations and c) another 8 “petals” at a distance $d_2$ on the same orientations. Best viewed in the electronic version.

Learning CNN weights

Even with the use of weight-sharing [92], deep convolutional networks require learning anywhere of the order of millions to tens of millions of parameters. In contrast, the two-layer network described above, which uses functional spatial forms for the spatial weights layers required learning a much smaller number of parameters. Weight learning was performed by optimising over the Gabor parameters of wavelength and spatial scale, and the pooling parameters used in the second layer of permuted convolution.

Both the pooling patterns and the Gabor parameters $\sigma$ and $\lambda$ were optimised on the PASCAL VOC 2007 database [54] in order to achieve good discrimination between classes of this database. The optimisation used the criterion of area-under-curve metric in an image retrieval task within the labelled im-
Figure 34: Illustration of the CNN that simulates simple-cell responses and a population code based on neurons in visual area V1 applied to a video sequence captured from a wearable or hand-held camera. In terms of biological complexity, this is quite a crude model: does not include strong non-linearities such as divisive normalisation of phase quadrature responses [19]. Nevertheless, it captures the behaviour of a significant subset of biological cell responses in the V1 area [30]. Sub-sampling in modes \((i_1, i_2)\) is omitted for clarity.

age database, i.e. given a query image from a set of data, find the nearest match in a database. Both training and testing images were taken from a partitioning of the PASCAL VOC database, and performance was measured in separate test sets. Stochastic gradient descent was used. In addition, the weight set was constrained so as to sum to zero at each point in space across all convolution filters, and were constrained to be spatially antisymmetric about one axis. No further optimisations were applied to the two early layers of the CNN once this optimisation had been done, and the optimisation did not use any of the images described in experiments in Section 6.6.

The architecture of the CNN and the order of tensors used to represent the data flowing through it is shown in Figure 34.
ARTIFICIAL PLACE CELLS (APC)

Modelling a single place cell: the tuning curve encoder

Given a series of video frames extracted from footage recorded during indoor navigation, I made use of the visual path concept [138, 105, 120], to perform matching, or association, between locations of a physical environment being traversed and a database of previously captured journeys.

The proposal – elaborated in Chapter 4 and an evolution of that suggested for sensor and WiFi-based localisation [179] – is that an appearance-based method using visual features could be easily mined to create a form of virtual landmarks. Such landmarks could be used to retrieve similar image locations around the locale of a landmark. The similarity scores obtained from appearance-based comparison methods, applied between sequences of frames of a journey and these virtual landmarks, should exhibit a behaviour that is similar to those recorded in mammalian place cells (Figure 35). In other words, one should obtain high scores when locations are visually similar or spatially close, and low ones when they are dissimilar, with a concave behaviour of scores with distance to the landmark. Requiring such behaviour of location sensing computational units would make them close to the behaviour of biological place cells.

In order to model the place-cell behaviour based on the visual input provided by cameras, one first needs a way of describing the collection of patches in an image near to a virtual landmark, and a means of comparing two images through their individual patch similarities.

I used a combination of the CNN that simulates simple-cell responses and population coding described in Section 5.3 – single-frame (SF-GABOR) –; and a subset of the visual features described in detail in Chapter 4, namely: keypoint-SIFT (SIFT) [98], dense-SIFT (DSIFT) [172] and spatio-temporal (ST-GABOR) Gabor based descriptors, and spatio-temporal Gaussian based descriptors [138]. Table 10 provides a summary of
The techniques and shows the number of elements of each descriptor. Similar to the comparison established in the previous chapter, a state-of-the-art SLAM method, Engel’s LSD-SLAM [51] was chosen.

In order to model place cells based on the visual input provided by a camera, the collection of patches has to be performed and two images should be compared through their individual patch similarities. When there are several frames involved, the encoding can be computationally expensive, and clumsy to search and compare one frame vs many. This is the topic now addressed.

**Dictionary encoding**

The convolutional network to produce a joint encoding of orientation and spatial position within a patch yields an order 5 tensor once applied to an intensity-only video sequence. Consider one order 4 and one order 5 tensors, $A$ and $B$. One of

<table>
<thead>
<tr>
<th>Method</th>
<th>SIFT</th>
<th>DSIFT</th>
<th>SF-GABOR</th>
<th>ST-GABOR</th>
<th>ST-GAUSS</th>
</tr>
</thead>
<tbody>
<tr>
<td>ST</td>
<td>No</td>
<td>No</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Dense</td>
<td>No</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
<td>Yes</td>
</tr>
<tr>
<td>Dim.</td>
<td>128</td>
<td>128</td>
<td>136</td>
<td>221</td>
<td>136</td>
</tr>
</tbody>
</table>

Table 10: Summary of the main properties of the different descriptors used. **ST**: Spatio-temporal, **SF**: Single-Frame.
these, A, will be treated as a description of a V1-type of population code in a frame captured during a journey. B is treated as a visual memory of a previous journey. The first two modes of A and B correspond to location in the image plane; the next two dimensions provide the population code over a patch of pixel data, and the fifth mode of B corresponds to time. We want the place-cell response to emerge from comparisons between tensors A and B.

However, searching this order 5-space for particular patterns is difficult: for a wide-angle input video of size 208 × 117 pixels, for example, an order 4 tensor (corresponding to one time-point) contains around 230,000 elements. A solution to the search problem is to apply vector quantisation to some of the dimensions, as seen in previous chapters. For example, by treating three of the five dimensions \((i_1, i_2, i_5)\) as observations of variables across the remaining two dimensions \((i_3, i_4)\), a dictionary, \(\mathcal{V}\) can be constructed. The encoding of a single frame using this dictionary can then be achieved using an order 1 tensor, and the sequence held in B by an order 2 tensor. This reduces the frame comparison problem to one that is performed by comparing dictionary-encoded descriptions of the population code. The dictionary encoding acts as a simple technique for dimensionality reduction, and it allows a more efficient comparison between a visual memory and a new visual stimulus within the modelled population code. As we saw in Chapter 4, we used the k-means algorithm for vector quantisation because of its simplicity and reasonable scalability when dictionary sizes range from the order of hundreds of terms to the order of thousands. We chose \(k = 400\) as the dictionary size and allowed a maximum of 20 iterations for computing the dictionary.

Once encoded using the dictionary \(\mathcal{V}\), which contains terms \(t_1, t_2, \ldots, t_{|\mathcal{V}|}\), the modes corresponding to the joint orientation and spatial encoding over patches are collapsed to a single term for each patch, and the set of numbers over an entire frame (modes \(i_1, i_2\)) are converted into a term-frequency representation [188] (see Fig. 36). These frame-encoded vectors take the form of a single vector (or order 1 tensor) for a given frame,
Figure 36: Two image frames should have similar representations along the fibre in the order 2 tensor that contains an encoded image sequence. Here, two similar frames are shown with (diagrammatic) representations taken from along the fibre corresponding to particular frames. The elements of the fibre correspond to dictionary terms, and the occurrence of each term is recorded. I used the $\chi^2$ similarity measure that is explained in Section 5.4.3.

and a matrix (or order 2 tensor) with dimensions $N_F \times |\mathcal{V}|$ for a prior journey sequence held in memory.

Modelling place-cell behaviour

In order to model place-cell behaviour it is necessary to map pairs of frames onto a scalar value that is analogous – perhaps after a non-linearity and affine scaling – to a firing rate. Consider two image frames that are captured at positions $P$ and $Q$, and are spaced a distance, $\ell_{PQ}$, apart (Figure 38). As the distance $\ell_{PQ}$ varies, the mapping should yield a smoothly varying result. In addition, biological place cell (BPC) behaviour often displays a concave relationship between firing rate and distance from the location of peak response (often, the location of some landmark) as illustrated in Figure 37 and Figure 38.

One way to mimic the behaviour of BPCs within APCs becomes patent when revisiting the kernel function introduced in the previous chapter. This kernel mapped a pair of BOVW-encoded image descriptors onto a positive scalar value. These functions can be equally applied to two BOVW vectors (in the
Figure 37: Each curve represents the response – modelling the firing rate – of an individual place cell to position along a path. Using the maximum response as an indicator of location allows a simple decoding of place cell responses, localising a person as being within the “receptive field” of a place cell (coloured horizontal bars).

Figure 38: Illustration of first prototypes of place-cell behaviour extraction from image descriptor similarities.
case of the non-biologically inspired descriptors) and two single-order tensors (when referring to the V1-inspired descriptors or SF-GABOR). As both representations are mathematically equivalent, I unified the terms into frame encoding vectors (FEVs) to denote both BOVW vectors and V1 single-order tensors.

I used the following mapping between two FEVs \( v_q \) and \( v_{db} \) (following the notation of previous sections):

\[
\kappa_{\chi^2}(v_q, v_{db}) = \sum_{j=1}^{400} \frac{v_q(j) \cdot v_{db}(j)}{v_q(j) + v_{db}(j)}.
\] (27)

This maps the two FEVs onto a scalar value that takes a maximum when the two vectors are identical. The behaviour of this kernel mapping between a fixed frame and a series of frames from a sequence is shown in Figure 39. As one moves along the horizontal axis from left to right, vectors from each frame in a section of a video sequence are compared against those of a reference frame (virtual visual landmark) from another sequence. The location of the reference frame corresponds to the location in the new sequence at which one observes the peak in the curve (around frame 690) of Figure 39; this location corresponds to the effective position of the virtual landmark, which is the field of view in front of the camera at that (frame) location. The variability of responses is high, but it is not unlike the types of variation found in biological place-cell behaviour.

Creating Place Fields

The similarity function \( \kappa_{\chi^2} \) is an important component in producing APC behaviour from frame encoding vectors. However, regions of the APC response curve which are almost flat contain very little information from the point of inferring the position of a stimulus that elicits some response. In other words, regions of the curve where the gradient with respect to distance to the peak of the curve, \( \ell \), is small, convey relatively little information about the camera’s location. Other regions of the curve show rapid change with distance from the peak. In order to synthe-
Figure 39: Single APC tuning curve (raw measurements in red) and a smoothed version (blue trace). The reference frame is located around frame 690. The APC response must be thresholded before using it for accurate position inference.

To create useful “tuning curves” for APCs, sub- and supra-threshold regions need to be defined for each APC (Figure 41).

The place cell is modelled by extracting the frame encoding vector, $\mathbf{v}_{r_i}$, when the camera is at position $r_i$ from one or more reference journeys and calculating the supra-threshold response to some frame $\mathbf{v}_\ell$ acquired at location $\ell$. As $\ell$ is varied, $\kappa_{\chi^2}(\mathbf{v}_\ell, \mathbf{v}_{r_i})$ changes accordingly. The set of supra-thresholded response curves, $r_i(\ell)$, is generated using:

$$r_i(\ell) = U(\kappa_{\chi^2}(\mathbf{v}_\ell, \mathbf{v}_{r_i}) - T_i) \cdot \kappa_{\chi^2}(\mathbf{v}_\ell, \mathbf{v}_{r_i}).$$  \hspace{1cm} (28)$$

where $U(\cdot)$ represents the unit step function and $T_i$ represents a suitable threshold. Curves acquired by averaging responses from several journeys with respect to the same APC location may be referred to as an APC tuning curve.

By defining APCs at regular intervals along a corridor, a simple population code for location can be devised. In Figure 42, the average response from each of several such APC cell responses is plotted along the length of one corridor. These
Figure 40: The variability of individual artificial place field (APC) responses, displayed using mean and the variability of the response over ensembles. Each line corresponds to the place cell field shape when testing against one specific pass.

Figure 41: Sub-threshold and supra-threshold regions can be identified by setting a threshold on the amplitude of the $\kappa_{\chi^2}$ similarity measure; the height of the threshold controls the support region (i.e. the place cell width) of supra-threshold region of an artificial place cell.
curves are produced by setting APCs to be spaced every 4 m within the corridor, and constructing the average APC responses. In Figure 42, ground truth was used to register the curves for illustrative purposes.

The responses of the collection of APCs provide important information as a population code [130]. By changing the threshold level, the support region of the APCs can be altered, leading to greater or lesser degrees of overlap, and (see Section 6.6), different performance.

*Location from APC activations*

Conceptually, given a series of APC responses to visual cues of a person’s location along some journey – illustrated in Figure 37 – there are two obvious ways of estimating location, \( \ell \). The first is simply to use the APC which displays maximum activation (firing rate) as a rough indicator of where the person is. That is, given a set of activations \( p(r_i|\ell) \), the location of the
camera that captured a particular image frame is provided by the index, \( i \), associated with maximizing \( p(r_i | \ell) \). This provides a precision that is limited by the width the APCs, but requires little more than ensuring that place cell responses are reasonably well-separated. This is similar to the “best-match” estimate approach of Chapter 4.

The second technique to infer location achieves more accurate localisation of a camera from its captured visual data by using the joint distribution \( p(r | \ell) \), of APC response, \( r \) to infer location \( \ell \) relative to some designated ground truth. I use a single index, \( i \), to refer to the response of a unique place cell, \( r_i \).

First, a rough location may be identified by using \( \text{argmax}(r_i) \) over the index, \( i \); then, the responses of neighbouring APCs can be used to obtain sub-cell localisation. In order to apply this principle, one needs sufficiently accurate estimates of \( p(r | \ell) = p(r_1, r_2, r_3, ..., r_C | \ell) \), where \( r_C \) is the total number of place cells in some region of a path. Given several active cells that are a subset of all place cells in a location, sub-APC localisation is possible using APC responses from previous journeys using empirical Bayes’ techniques. For example, if three cells are active, the chain rule can be used to obtain successively refined estimates of \( \ell \):

\[
p(\ell | r) \propto p(r_3, r_4, r_5 | \ell)p(\ell) \\
  \propto p(r_3 | r_4, r_5, \ell) \times p(r_4 | r_5, \ell) \times p(r_5 | \ell) \times p(\ell) \tag{29}
\]

so that the responses of spatially close APCs can be used to infer sub-APC position. If the width of an APC is set to around 2 m, localisation of the order of tens of centimetres is plausible.

A generalised regression neural network (GRNN) for sub-APC localisation

In the experimental work to be discussed in the next section, a generalised regression neural network (GRNN) was used to provide sub-APC position estimates, obviating the need to con-
struct ad-hoc empirical estimators. This network takes APC responses as inputs, providing a position as a regression estimate.

This GRNN network consists of two layers: the first layer has radial basis functions (RBF) neurons with a spread of 0.1. The second is a linear output layer that calculates real-valued position estimates from the RBF outputs. This supervised phase of training used a subset of videos to infer spatial location from the ground truth camera position of this subset of the dataset. The trained network was then applied to infer the spatial position in videos from the remainder of the dataset. In the experiments, the responses from $C = 16$ place cells were input to the network, and ground truth of location within a section of corridor – up to 4 m long – used to train the network. In all experiments, dictionary generation was performed independently of the APC responses used in training the regression network.

An overview of the pipeline for processing the frames to generate APCs is shown in Figure 43. The performance of different methods will be discussed in Section 6.6.

![Figure 43](image)

Figure 43: Overview of the training pipeline. The sequences of visual information are processed to generate frame encoding vectors (FEVs). The activation model based on thresholded $\chi^2$ kernel distances between FEVs is used to generate the APCs. A RBF-based regression network is used to learn sub-APC locations from training sequences. The diagram of the neural network is merely illustrative, it does not represent the real architecture used. The represented FEVs are also diagrammatic.

**EXPERIMENTS AND RESULTS**

In order to test the APC concept, I performed a series of experiments using the RSM dataset [138]. Recall from Chapter 4 that
this dataset contains visual data of more than 3 km of indoor journeys acquired with two devices: a hand-held Nexus 4 and a wearable Google Glass. Corridors varied in length between 32 m and 60 m and the sequences comprise more than 120,000 frames with ground truth acquired with surveying equipment. The original resolutions of the dataset are $1920 \times 180$ and $1280 \times 720$ pixels per frame which I also downsampled to $208 \times 117$ for the experiments of this chapter.

**APC-level localisation**

The first method used to infer location – based on identifying the APC with maximum activity – was tested in several corridors of the experimental data. First, a visualisation showing the locations of APCs with maximum activation is shown in Figure 45; these are indicated on the floor plan of one of the building sections that was used to conduct the experiment. Locations are staggered across the width of the corridor in order to visualise the individual activations of the 8 APCs defined within this corridor. The second technique to estimate location relies on the overlap of the responses from APCs and the use of a GRNN. Table 11 compares the two localisation techniques for different methods of descriptor generation. A dictionary based on a single device type was used, but all the combinations of remaining passes were submitted as queries. The neural network regressor shows better results, achieving errors as low as 2.49 m for the V1-inspired SF-GABOR, even with the majority of the queries coming from a different device that was not used to learn the dictionary. Very low errors were observed using a single device, as may be seen in Figure 44. The performance of LSD-SLAM, when tested on the same sequences, is also reported, but tracking was lost in roughly 40% of the sequences. This is probably because LSD-SLAM performs best on sequences acquired with global-shutter, fish-eye lenses. The RSM dataset does not use such cameras. The tracking recovery
Figure 44: Sub-APC location estimate comparison. Using broad place-cell tuning curves, very accurate localisation can be achieved within a section of corridor. For this corridor and for this journey, absolute localisation errors range from below one metre to 1.49 m. Ground truth is shown in red. In this case, only single-device (Nexus 4) queries were used. The effect of using queries from both devices is shown in Figure 47 and captured by Table 11.

exception employed for the comparison in Chapter 4 was also used here to keep LSD-SLAM running.

Sub-APC localisation

By arranging for APC responses to be up to 4 m in width, the responses from several cells can be used to perform accurate inference of spatial position using a single section of corridor. The success of this technique is illustrated in Figure 44. Note that average absolute errors are very small compared to distances traversed.
Figure 45: Using the ground truth information acquired with the surveyor’s wheel for the passes in the database, activations from APCs are overlaid onto the floor plan in which video data was acquired. Different colours refer to individual APCs. For this visualisation experiment, 8 APCs were constructed.
Parameter tuning

The GRNN relied on the overlap of the responses of the place cells to perform location inference. By varying the threshold beyond which place cells are considered active, it is possible to have very long-tailed APC responses, spanning several metres. This yields APC behaviour that is similar to that of rate-coding in place cells observed in biology. The blue trace on Figure 46 shows an example of the average cell width in metres as the threshold on the $\chi^2$ comparison metric is varied. The red trace illustrates the average error, also in metres, that is obtained from the GRNN regressor. Having low APC response overlap deprives the network of sufficient non-zero inputs, leading to poor accuracy in position inference.

One of the key problems with using different cameras – even if they are calibrated – is the difference in field of view of one imaging device with respect to another. A Nexus 4 smartphone and a wearable Google Glass were used to conduct experiments into the effect of the dictionary on the localisation using the sub-APC (GRNN) approach.

A set of 10 journeys through one of the corridors of the RSM building with different devices was taken for this experiment. Some of the journeys were included in the database, others were used to conduct queries. The partitioning of the data was permuted, varying the number of journeys in the database and the number of query journeys kept out of the database. Journeys which were used for queries did not contribute to the learning of the dictionary, which was repeated for each permutation. Figure 47 shows the difference in absolute error of Nexus and Glass queries when only passes from one device (in this case, the Nexus) were used for the dictionary learning.

Computational load metrics

The main computational load for the place-cell computation is due to the dictionary learning, which enables fast lookup of
individual frames – encoded as a single vector of terms from the vocabulary. The dictionary is built by randomly selecting 200 frames from each sequence, and randomly selecting 800 examples of the joint spatial-temporal encoding, which consists of D-dimensional frame encoding vectors (see Table 10). Dictionary learning takes around 1 hour for approximately 50 video sequences of the whole corridor dataset on a 64 GB machine using 400 words. Once learnt, encoding a new frame and looking it up takes less than 200 ms using unoptimised MATLAB code.

**CONCLUSION**

Place cell physiology and function is of immense interest for a number of reasons. To my knowledge, there is no computational architecture that reproduces place-cell responses in the form of tuning curves artificially from video sequences, and there is no biologically plausible model for place cell encoding that can be applied to the recorded video sequences I have used in this work. Furthermore, competing camera-based localisation techniques such as SLAM rely on relative object camera
Figure 47: Multi-device sub-APC localisation test. Green represents localisation results when only passes from the Nexus were used for the dictionary learning. Blue represents the localisation results when passes acquired with both Nexus and Glass devices were used for the dictionary learning. In particular, the light blue shade represents the variability of the results for the cross-device scenario for dictionaries created with different combinations of passes from the two devices. The dark-blue line shows the average performance for the cross-device scenario. Note the substantially worse cross-device performance.
Table 11: Absolute error evaluation when using a larger number (40) of APCs of small spatial support (0.61 m), using \( \text{argmax} \) to infer spatial position, in contrast to using fewer (16) but larger APCs with substantial overlap and the regression network (sub-APC). The comparison with a state of the art SLAM method (LSD-SLAM) is also included.\(^{(*)}\) LSD-SLAM performance is positively affected by the tracking recovery exception, which reduces the error drift by resetting the odometry calculation and the error of the pose-graph optimisation.

motion to infer structure and to then either perform odometry or to estimate location: motion is a requirement. The place cell model proposed and evaluated in this chapter does not require motion at location estimation time: a single frame gives a possible response.

A number of questions observed from this study motivates further work in the search of a plausible biological answer. First, it is the nature of the place cells what defines their width after thresholding, and thus their sensitivity to locations. In biology, neurons are known to reach a certain threshold potential before they fire an action potential or impulse that can propagate along the axon and eventually trigger similar responses on other cells. We are planning to study this effect and extend the model of the place cell by also incorporating a model of its firing that takes into account the shape of the firing envelope and the causes of the activity.

Perhaps a greater problem, from a biological perspective, is that it is unclear how grid cells might fit into the model proposed in this chapter. One intriguing prediction is that grid cells are implicit in the existence of place cells; in other words, place cells are not entirely formed from grid cells, but working place cells create a grid-like pattern of activation if one grid cell is wired to several spaced place cells. This observation may
be related to observed loop-like connectivity observed between hippocampus and entorhinal cortex.

In third place, I have found that the best performing methods are SF-GABOR and SIFT (sparse and dense). This poses another question worth investigating and is that of the relationship of better performance with the closeness of the local patch description method with models of early vision. As described in Section 5.3, the 2D Gabor filters in SF/ST-GABOR descriptors present the orientation-selective simple-cell receptive fields behaviour of the primary visual cortex (V1). Similarly, the difference of Gaussians (DoG) space representation found in Lowe’s SIFT keypoint detection may be seen as an approximation of the spatial receptive field of a retinal ganglion cell. Lowe also suggested that the process behind the computation of the orientation of the frequencies is similar to the behaviour of complex cells in V1.

In conclusion, the work described in this chapter demonstrates that computational models of place cells can provide effective estimates of camera location without relying on tracking or construction of a geometric model of the local environment.
AN ASSISTIVE HAPTIC INTERFACE FOR APPEARANCE-BASED INDOOR NAVIGATION

Figure 48: “By designing for someone with a permanent disability, someone with a situational disability can also benefit.” From Microsoft’s inclusive design program “Independence Day”, stressing the importance of a universal user-centred design [107].

INTRODUCTION

I have covered the topic of navigation extensively in Chapters 4 and 5, but have not considered the user’s perspective in detail. In this chapter I take a user-centric approach and focus on the
design of an application that serves those for which navigation present additional difficulties: the blind and partially sighted.

The problem of navigation

Navigation might seem a very natural task: usually, it involves travelling along a path that we have previously visited and learnt. At other times, navigation might require us to follow a new and unseen path, triggering the need for planning and evaluating possible directions of movement. Traditionally, finding our way in unfamiliar environments required certain skills such as map reading, or using a compass. External cues can also help us find our way to a destination: signs, landmarks, directions from other people, etc. Recently, with the emergence of smartphones and other wearables, we now have devices that gather data, interpret it and provide tools through which assessing one’s position and planning a route is almost immediate. These tools are increasingly available on a single device: the problem of navigation is reduced, in outdoor contexts, to the simple act of following the indications of a navigation App on a mobile device, or a “SatNav” [158].

Indoors, the problem can be more complex, and the technologies that make vehicular navigation an easy task do not work inside buildings for ambulatory journeys. Navigation can be more ambiguous, and despite the fact that, in global terms, we are restricted to moving over a relatively small region of the surface of the earth, buildings can have vast internal dimensions. The nature of the navigation problem in some indoor environments – universities, museums, government buildings, shopping centres, airports to name a few – might seem simpler than outdoor navigation. Yet, the frustration of getting lost in indoor environments is more emphasised, perhaps because the immediacy and efficacy that is achieved outdoors with automotive and naval navigation systems cannot be easily matched.

As we saw in Chapters 1 and 2, the blind and partially sighted community has to face additional challenges. Despite demon-
striations of promising technology on small scales of usage [110, 100, 96], the white cane remains the most widely used navigational aid. Helping the blind and partially sighted to navigate in unfamiliar environments is particularly challenging. There are several reasons for this, including immaturity of localisation technology in indoor settings, the cost of installing customised localisation technology, and the challenges associated with keeping mapping information up to date.

As for the case of vehicular navigation, it is likely that the solution to indoor navigation lies within not one, but rather a collection of approaches that work together. For reasons of both precision (a statistical argument, based on acquiring independent measurements) and redundancy (an engineering principle), several possible sources of localisation data, methods of user interaction and algorithms should be developed and evaluated separately. In this chapter, my intention is to take one combination of sensor, one type of inference method and one type of user interface option to provide navigation information (see Figure 49).

Structure of the chapter

In Section 6.2 I will review the state-of-the-art on assistive technologies for navigation. An overview of the proposed system for assistive navigation will be described in Section 6.3. In Sections 6.4 and 6.5 respectively, I describe the elements of visual processing and tactile feedback involved in this work; with Section 6.5.2 summarising the structure of the client-server application that was developed to allow the experiments of Section 6.6. Finally, in Section 6.7 I analyse the results of the experiments to later conclude with final remarks in Section 6.8.
Figure 49: The solid circles indicate the remit of this chapter. I do not suggest that either visual sensing, tactile feedback or knowing one’s position on a map solves the indoor navigation problem. In this chapter, I have deliberately selected one sensing technique, one mechanism of feedback and one inference technique of the many redundant sensors and systems that one would wish to have in a robust navigation device. Evaluating combinations in this combinatorial manner allows redundant and robust systems to be created systematically, and with component-level performance characterisation.
6.2 Background on Assistive Devices: Accessible Technology

Background on Assistive Devices: Accessible Technology

The impact of sight loss in navigation

Clearly, the ultimate goal would be to prevent people losing their sight or to heal sight loss. In the absence of these achievements, both governmental agencies and charities have identified that support for independent living for people with visual impairment is a priority. In the UK, as we briefly pointed out in Chapters 1 and 2, the leading sight loss charity, the Royal National Institute of Blind people (RNIB), has identified two key aims [141] (slightly paraphrased for clarity):

- more people should be able to make journeys safely and independently;
- more people should achieve independence through the use of information technology and mobile technologies.

An engineering solution that supports navigational autonomy of the user is needed. Being able to also access information sources might be considered something of a holy grail for navigation for many people with visual impairment. In the next sections, we will describe some studies that have approached the navigation problem from different perspectives.

Non vision-based solutions for assistive navigation

Classical aids

The two principal navigation aids for visually impaired people remain the guide dog and the white cane. In addition to being good navigational aids outdoors and in complex environments, guide dogs have been recognised as being a source of companionship for some people who feel isolated. However, the cost of training dogs can be high and the potential to get lost remains, particularly when the dog is unfamiliar with a route.
Also, obstacles that are above the height of the dog (such as low-hanging branches) can present a hazard [102].

The other highly successful piece of navigation technology for visually impaired people is the white cane [143]. Whilst it is known to allow more independence, it does not provide navigational information on a spatial scale much greater than a stride length [100]. There are some navigation scenarios, such as those involving environments that are unfamiliar or too complex, which are avoided by some white cane users. Examples of these include walking a route for the first time, using public transport, approaching a building entrance or public transport door, and other such actions requiring relatively small-scale accuracy in body positioning. In particular, public transport usage remains extremely low among visually-impaired people, with just 11% of blind or partially sighted travellers boarding a train or a bus regularly [129].

Radio frequency systems

The latest advances to provide accurate navigation outdoors have been relying on the information provided by satellite-based navigation systems, most commonly Global Positioning System (GPS). Systems based on GPS have changed the current concept of outdoor navigation. Some significant attempts have been developed to target the needs of visually impaired people. For example, the Sendero Group’s Mobile Geo [151] system uses GPS to provide position and navigation directions through an accessible keyboard and a speech synthesis interface. BlindSquare for Apple mobile and tablet devices takes GPS a step further by using crowdsourced data for points of interest (via integration with Foursquare services and data) and OpenStreetMap for fine-grained street information [110]. However, even such customised systems lack the information sources or signal availability indoors to be used by people with visual impairment. For example, the signal strength that reaches devices indoors is relatively weak – of the order of a tenth of a femtowatt [182] – and often unstable. Because of this, a system that is reliant
on GPS indoors would compromise navigation, provide a poor user experience, and potentially compromise safety.

Several other projects have attempted to provide navigational information indoors based on radio frequency technologies. According to the RNIB [186], RFID, Wi-Fi and Bluetooth radio technologies can provide both accuracy and coverage indoors. Additionally, body sensors employing ZigBee signal strength indicators [46] have demonstrated the feasibility of wearable sensor networks to provide navigation information. Such networks usually require some form of infrastructure to be deployed throughout buildings. Signal transmitter locations then need to be tested, associated with indoor mapping information, and subsequently maintained; a process that can be costly.

Finally, Drishti, an integrated indoor/outdoor navigation system, has been proposed [134]; this uses differential GPS (DGPS) for outdoor positioning and an ultrasound positioning device for indoor location measurements. Although reporting sub-metre localisation errors, the indoor subsystem requires the deployment of ultrasound transmitter “pilots”, and the user has to carry ultrasound beacons and specialised hardware, making Drishti a technically feasible but costly and currently impractical prototype.

_Tactile interfaces for the blind and partially sighted_

Today, there are two common sensory channels we use to understand our surroundings: vision and, crucially for visually impaired users, hearing. Much load is already placed on these channels, and they may reach a point of saturation in busy environments. Therefore, it is worth using another sensory channel to convey information. One channel that has been investigated since the 1800s is touch.

In 1897, the Elektroftalm was created using a single block of selenium [36]. Its photoconductivity was used to convey a sensory stimulus to the foreheads of blind people, allowing them to distinguish between dark and light.
Bliss’ technology [23] was a prime example of early assistive user interfaces: they used a combination of a tactile stimulator and an optical sensor to allow the blind to understand their surroundings. The image found by the optical sensor fell onto a $12 \times 12$ phototransistor array and used one-to-one mapping onto tactile stimulators. Each illumination of the phototransistor led to a vibration on the corresponding tactile stimulator; these were spaced 1.25 inches apart. The aim of his experiment was to determine the minimum size of object that could be recognised on a tactile display. Only crude images were produced and it was found that not many visual objects were recognised reliably – even those as large as $2/3$ of the screen. However, Bliss identified that the results of this experiment may have been subject to defects in the intensity of responses in the piezoelectric bimorphs. The experiment did find, unsurprisingly, that larger objects were more reliably detected.

In more recent times, there have been many advances in tactile technologies. Users can experience tactile feedback in displays through several cues including piezoelectric sensors [125], shape memory alloys [175], micromachined devices [93] and air jets [12]. In addition, there are promising directions around the use of electrorheological fluids [125], those that respond to electric fields by changing their viscosity. A common class of technique under exploration is vibrotactile displays. These use a combination of microlinear electromagnetic actuators and piezoelectric ceramics. However, they do not seem to convey the frictional forces which visually impaired users are attuned towards when exploring objects with their fingers [64]. More recently, Hartcher-O’Brien et al. [67] explored a piece of technology designed to enable visually-impaired users to find the distance to an object, such as a wall, by distance sensors worn on the head. In this arrangement, tactile cues were provided through vibrating patterns in a hand-held device [67].

Tactile technologies are also being considered for several applications for a wider range of users (including those with and without visual impairment), ranging from providing cues for pilots in flight [158] to computer mice [4]. In addition, some
companies have considered implementing tactile-feedback technologies on their phones. For example, Motorola found that out of 42 subjects, 35 preferred having a combination of vibrotactile feedback and visual cues [32], an outcome supported by previous research [131]. Poupyrev and colleagues claimed that tactile interfaces were a “peripheral awareness interface”: they provide sensory stimulation on a subconscious level, thereby taking cognitive load off the user. Further uses of tactile technologies include training for surgery, in which it is sometimes necessary for a surgeon to be able to function under circumstances in which there is limited visibility [70].

For the purposes of this work, the Senseg™ tablet is an apt modern example of a tactile display that allows a user to feel something akin to frictional forces. The Senseg™ device passes a low current to an isolated electrode (“Senseg tixel”) that creates a small attractive force to the skin of the finger. By modulating this force, a device can convey the sensation of different textures. This is a rather significant advancement on the mechanical piezo solutions used by Bliss et al. [23].

For the experiment described in this work, a Senseg™ tablet will be used to test the information delivered as the result of an appearance-based location query (see Section 6.6.3).

Computer vision for assistive navigation

Geometry-inferring methods for assistive navigation

The extended use, minimal cost and increasing quality of modern cameras have brought the use of visual information for assistive devices a step closer to reality.

Recently, RGB-D devices, producing both colour images and depth information, have shown promise for robotics. For example, Aladren et al. [5] make use of an inexpensive RGB-D sensor to detect obstacle-free paths. Possible obstacles or architectural elements are detected using the depth-range; both depth and colour information are used to infer the presence or absence of obstacles. Potentially, this is a vital feature for visually impaired
people, as it extends the range of obstacle detection provided by the traditional white cane.

Another stream of computer vision with huge potential for assistive design is SLAM. We saw in Chapter 4 that visual SLAM is a key branch of vision-based navigation and very popular in robotics. We also saw that its main strength is the ability to infer a geometric model (or map) of the environment and the camera trajectory at the same time.

Some SLAM methods have been developed aiming blind and partially sighted users. In particular [6] and [8], incorporate dense optical flow estimation into visual SLAM in order to enhance the performance of the algorithms with obstacle detection and improved performance in crowded environments.

A SLAM-based solution that is closer to our approach was suggested by Ali and Nordin [10], who used a standard EKF-SLAM approach to track SIFT features. The SIFT features are simultaneously used to provide semantic information (i.e. object recognition for obstacle avoidance and path recognition) about the environment. Apart from the fact that the authors do not test vision in isolation (that is, they enforce tracking), their method’s caveat is that instead of building a visual path with crowdsourced image descriptions or sensor signatures, they impose a known constraint on the “true pathway” for the location of the detected features, and also as a prior for the tracking. This constraint limits the scalability of the method; and perhaps more importantly, it does not explore the accuracy of a location estimation system based on crowdsourced visual information.

The novel SLAM algorithm subject to comparison in Chapter 4, LSD-SLAM [51], seems to perform well in an indoor SLAM setting. As we saw, this approach, instead of keypoints and descriptors, uses semi-dense depth maps for tracking by direct image alignment. Although it has not been tested in an assistive context, this is a remarkable step forward for mobile inclusive applications, as the semi-dense maps allow lighter frame to frame comparisons, to the point where odometry can be performed on a modern smartphone [146]. The shortcomings, however, as most SLAM methods, originate on the great depen-
dence on a very accurate camera calibration and initialisation routine, and best results are often achieved under specific conditions, such as monochrome global shutter cameras with fish eye lenses [51].

Appearance-based methods for inclusive visual navigation

As we saw in Chapter 4, appearance-based methods attempt to provide localisation without keeping track of the coordinates of the robot/user or landmarks in metric space.

Schroth et al. [149], as seen in previous chapters, also suggested and developed a prototype client-server application based on appearance-based methods, although using a different approach to the one explored in this chapter: instead of performing all computation on the server, they pre-select some relevant visual words that are sent to the client for matching. Whilst the work of Schroth et al. is relevant to that reported here, the lack of an assistive context makes the challenges different.

However, there are some examples of assistive applications of appearance-based methods. Ali and Nordin [10] developed an appearance-based method that uses SIFT features in order to construct a weighted topological map of the environment stored in a modified electronic white cane. During query, the cane submits SIFT features that are matched with the ones in the database. The degree of similarity or “weight” between the matched images allows direction instructions to be conveyed based on previous knowledge about the environment. Nguyen et al. [117] combined SLAM with FAB-MAP to develop a mapping and visual localisation system based on constructing a route map that contains a database of images together with an odometry model. Their appearance-based method is limited to what FAB-MAP offers (SURF features), but to improve indoor reliability, they introduced markers that are easier to distinguish along the route. Nguyen et al. [118] introduced a standard Kalman filter SLAM approach to track the detected features in order to improve the robustness of location estimation for a robotic aid for visually impaired users.
In the previous work [138] described in Chapter 4, we compared the performance of different appearance-based techniques for indoor localisation, extending the use of these methods beyond loop closure, allowing for positioning on its own. Average absolute position errors of as low as 1.59 m were reported using an approach based on matching images against crowdsourced journeys made along indoor corridors.

In the next sections, we describe how we applied the BoVW approach designed in Chapter 4 to estimate a user’s position during indoor navigation by using images acquired from either hand-held or wearable cameras. Position is estimated with respect to the distance travelled along one-dimensional paths consisting of ambiguous corridors, a difficult use case for techniques such as SLAM, as shown in Section 4.7.

*Getting data into a navigation system: crowdsourcing*

As we saw in Section 6.2, crowdsourced data is already enriching location information through social networking and personalised place recommendations (e.g. Foursquare); and through collaborative maps (e.g. OpenStreetMap). Crowdsourcing sensor data from mobile phones is providing a myriad of applications, from detecting traffic congestions [15] to mapping the real network coverage based on thousands of individual signal strength readings [123]. More relevant to the present work is the work by Wang et al. [179] where indoor localisation is provided by matching inertial and magnetometer readings to sensor signatures stored in a database of crowdsourced data from previous users traversing the same space. We are adding vision to this hypothesis, and we propose two different scenarios for crowdsourcing of visual data:

a) visual data, together with ground truth positioning, is incorporated into mapping information as part of an accessibility measure. There are tools available for this process that standardise and accelerate the acquisition of visual data (see, for example, [73]).
b) individual users contribute recordings of their indoor journeys from wearable cameras and provide some contextual information and ground truth via a Web or mobile application.

These two scenarios are compatible in the sense that users should be able to enrich public indoor maps through crowdsourcing tools and benefit from the availability of this data through accessible Apps installed on their mobile/wearable devices.

Therefore, we consider first the role of an appearance-based technique for using low-resolution images from a hand-held or wearable camera as both a source of query information and a source of database (mapping, localisation) information. Images are compared in order to establish position, and this can be seen as a means of externalising visual memory [168], concept that we introduced in Chapter 5.

**System Overview**

A key contribution of this work is to explore the feasibility and usefulness of an App that provides a haptic interface for appearance-based indoor localisation. Figure 50, illustrates the concept: a blind or partially sighted user wants to travel from a point A to a point B in a building. They launch an App which starts collecting images from the camera of the Senseg™ tablet or from a wearable camera paired with the tablet. These images are sent to the server, which estimates the location of the user based on an appearance-based visual localisation algorithm. The estimated location is sent back to the user’s device where it is interpreted and conveyed in the form of a haptic cue over a pre-loaded floor plan of that part of the building. The device also shows visual feedback for sighted users, as illustrated in Figure 52.
The data sources

As suggested in Section 6.1, both the floor plan and a database that contains previously acquired views (see Figure 2) must be available. The former should be (politically) easily to acquire, particularly if it is considered as a means toward supporting accessibility. One option for the latter is described in Section 4.4, but note that, as shown in Figure 49, the necessary data can be acquired or inferred through a variety of techniques.

A key concern for such databases, and particularly those that would seek to acquire image or video information upon which to base navigation services, is the sheer quantity of information that would need to be stored, acquired and transmitted. This is where the choice of processing technique can make a significant difference. Ideally, we should aim to capture, store and process visual information using as small an image size as is practically feasible to permit location recognition. Basing these calculations on uncompressed video is useful, because most algorithms that generate visual descriptors for object recognition currently operate outside of the compressed domain. Furthermore, the range of descriptor implementations that we are able
to choose from is dramatically increased by working in the image domain. A 15 s clip of video acquired at normal walking speed equates to just over 20 m of distance in real space. UMTS 3G mobile can run at up to 48 kB/s, enough to perform uncompressed image transfer for a 208 × 117 pixels greyscale image – a location query – within 1 s. However, 1 MB/s – the speed of EV-DO [20] – or 9.4 MB/s – the uplink speed of LTE 4G – is more than sufficient for acquiring crowdsourced low-resolution video data through streaming. The storage costs of these videos should also be considered. We found that 80 s of low-resolution video footage at 25 fps occupied between 2 and 3 MB when compressed, with 20 such journeys, each of 80 s duration, coming in at under 50 MB. These low storage costs are achievable if the spatial resolution of image data is sufficiently low, significantly lower than used by current computer vision techniques for localisation. We observed that from small videos, we were able to determine by eye where in a building a person with a wearable camera was walking. In Chapters 4 and 5 the results of the appearance-based methods and artificial place cells models proved localisation to be feasible at such low resolutions. It is now time to assemble the pieces and provide a complete solution to localisation: from the visual sources to tactile feedback.

Algorithm choice

In the previous work described in Chapter 4, image patch descriptors were evaluated for their ability to discriminate location [139]; in this chapter, I use the widely available dense-SIFT (DSIFT) described in Section 4.3.3. The reason behind this choice was the availability of an optimised C code implementation of DSIFT that is more suitable for a client-server prototype [172] in terms of speed. From the point of view of localisation performance, DSIFT ranked second (out of 7) after the single-frame GABOR method in the results reported in Sec-
The detailed performance figures will be included in Section 6.7.

We describe the algorithm choice for visual processing in some detail in Section 6.4. For the RSM dataset, the inference of geometry and camera odometry can operate in real-time on a mobile device. However, running LSD-SLAM at the resolution required for tracking to succeed can quickly deplete a device’s battery [51]. Furthermore, the use of several uncalibrated devices, a likely scenario when trying to crowdsource information, poses a challenge to LSD-SLAM as we saw in Section 4.7. Perhaps more importantly, utilising several sources of navigation information adds robustness, and allows routes to be updated at low cost. This requires a repository of journey sequences; the repository is therefore a key part of the architecture – and of the algorithm – used in creating the prototype App.

An indexing process considers all frames from multiple journeys, using this to build a custom dictionary that can be used to quickly search for matching frames, and which can apply distance measures between candidate BoVW models. Metrics can be used to pass information back to a server about relative distance based on previous journeys along the same route.

The data repository sits on an Ubuntu server and consists of the frames at original and compressed resolutions and binary files for processed data (descriptors, dictionaries and encoded visual words) served with the open source distributed file system GlusterFS. I expand on this in Section 6.4.

**Interface device**

We chose to use the tactile interface of a prototype version of an electrostatic device, the Senseg™ tablet. This tablet is a customised version of a Google Nexus 7 Android tablet, and can provide a fairly rich tactile experience. In order to provide a scalable and real-time localisation service to a person, we utilised a standard client-server model, with a customised App on the Senseg™ tablet as the client. The server was imple-
mented as a Node.js HTTP server, which acts as a proxy for calling the localisation code. This generic, modular design allows us to both extend the HTTP server’s functionality, for example to include capturing of data for the dataset via another phone App, or to change the implementation of the HTTP server or localisation code independently.

Note that not only does the HTTP-based approach allow for relatively fast communication over a building’s Wi-Fi network, but other network communication protocols are often blocked in institutional networks. The server can also, if needed, be extended to use HTTPS, and to operate on standard ports (80/8080 for HTTP and 443 for HTTPS).

**VISUAL PROCESSING DESIGN**

The appearance-based pipeline from Chapter 4, shown in Figure 18 will be used in these experiments, but adapting it to a live query scenario. As we saw, it is similar to standard BoVW pipelines in the sense that is composed by a feature extraction process followed by dictionary creation and encoding techniques. However, it presents important particularities: in the work described in Chapter 4 we followed the same pipeline within a benchmark package or evaluation suite to evaluate several feature extraction techniques [139] in the context of visual localisation. They comprise a mix of single-frame and spatio-temporal descriptors, with an emphasis on dense methods, as we believe these adapt better to the particulars of indoor navigation. Additionally, the distance metrics used here differ from the standard classification metrics present in most BoVW schemes as our intention is not to classify one frame versus the rest of them, but to assess the similarity in the dictionary space of frames that are close to each other in the physical space. In the following sections we will briefly recap the different elements of the pipeline for the case of the appearance-based method used for the application prototype.
6.4 VISUAL PROCESSING DESIGN

Appearance-based methods for “live” location inference

The incoming frames for both the database creation and query branch are first converted to greyscale. The images are then downsampled to size 208 × 117 pixels, which in Chapters 4 and 5 was found sufficient to generate reasonable localisation. Prior to the feature extraction stage, the images were smoothed at a scale of $\sigma = 1.2$, which corresponds to a keypoint scale $s = 2$. This avoids computing a Gaussian scale space: the single scale of descriptor calculation on a dense grid in which a single value of $\sigma$ appeared well-suited to the goal of working with relatively small images. More information on these design choices, and a comparison of sparse versus dense SIFT, and other descriptors, is discussed in Section 4.3.3.

For the query frames submitted to the server, I computed the DSIFT descriptor [97, 91], selected for its wide availability within many (operating system and software) environments. I used the same implementation as in Chapter 4 with a stride length of 3 pixels.

For the vector quantisation corresponding to the bag-of-visual-words (BoVW) model of this experiment, hard assignment (HA) was used to encode each descriptor vector by assignment to a dictionary entry. The dataset was partitioned by selecting $N_v - 1$ of the $N_v$ video sequences of passes through each possible path. This ensured that queries were never used to build the vocabulary used for testing the localisation accuracy. The dictionary was created by applying the k-means algorithm on samples from the video database and loaded on the server ready to accept queries. This time, we also fixed the dictionary size to 4,000 (clusters, words); this allows comparison with the work of others in related fields [35].

The resulting dictionaries were then used to encode the descriptors, both those in the database stored in the server and those from queries originating in the Senseg™ tablet. The frequency of occurrence of atoms was used to create a histogram of visual words “centred” around each frame of the video sequence (visual path) in a database, and the same process was
used to encode each possible query frame from the remaining path. Histograms were all $L_2$-normalised.

*Localisation using “kernelised” histogram distances*

In a similar fashion as the methods described in Chapters 4 and 5, I used “kernelised” histogram distances to provide localisation as illustrated in Figure 51. This technique allows the storing of “pre-trained” kernels in a database computed using leave-one-out validation with all the training data available in the RSM dataset [137].

As a reminder of Chapter 4 we briefly recap that if using $n$ to denote the frame number, and $p$ a particular journey down a corridor, the $\chi^2$ (Eq. 30) kernel

$$K_{\chi^2}(H_q, H_{p,n}) = 2\frac{(H_q \cdot H_{p,n})}{H_q + H_{p,n}},$$

and the Hellinger kernel (Eq. 31)

$$K_H(H_q, H_{p,n}) = \sqrt{H_q \cdot H_{p,n}},$$

are common choices to compare query frames encoded by a BoVW encoded frame with a database containing several frames (here, consisting of different journeys, $p$ and frames, $n$). In earlier work described in Chapter 4 the $\chi^2$-kernel seemed best for the path localisation problem. For a random subset of the $N_v - 1$ videos captured over each path in the dictionary, the query is selected from amongst the frames of the remaining journey. Each histogram, $H_q$, representing a query frame results in $N_v - 1$ separate comparison matrices (Fig. 51), each containing the distances of each database frame histogram to the query in the form of matrix columns.
Recall that the best matching frame, $\hat{n}$ from pass $\hat{p}$ across all of the $N_v - 1$ vectors is retrieved using:

$$L(\hat{p}, \hat{n}) = \arg \max_{p, n} \{ K_{\chi^2} (H_q, H_{p,n}) \}.$$  \hfill (32)

$H_{p,n}$ denotes the series of normalised histogram encodings, indexed by $p$ drawn from the $N_v - 1$ database passes, and $n$ denotes the frame number within that pass. The estimated “position”, $L$, of a query was that corresponding to the best match given by Eq. 32; this position is always relative to that of another journey along approximately the same route; the accuracy and repeatability of this in associating locations between passes was evaluated using distributions of location error and Area-Under-Curve (AUC) criteria derived from these distributions as seen in Section 4.6.2. The method itself is depicted in Figure 51 for clarity.

**A TACTILE INTERFACE FOR A CLIENT-SERVER ASSISTIVE LOCALISATION SYSTEM**

I have described localisation systems that use visual input to provide location information by matching queries against a database of previously acquired images of the environment. I now describe how this information can be conveyed to blind and partially sighted users by means of a haptic interface. In Section 6.6.3 experiments to gauge the quality of the haptic feedback for localisation are presented.

*The Senseg™ App*

The goal of the Senseg™ App was to convey localisation information to visually impaired users. The Senseg™ device allows different textures to be felt at different locations and at a varying range of intensities, as specified by the programmer. It provides enough variation in textures to create discretely identifi-
6.5 A TACTILE INTERFACE FOR A CLIENT-SERVER ASSISTIVE LOCALISATION SYSTEM

Figure 51: Matching locations by selecting maximum similarity kernel score between query and database frames. Recalling Chapter 4, the scores may be obtained by comparing a BoVW encoding of a current query frame against all previous frames acquired from different journeys having similar start and end points. Because the frames are relatively small, comparisons and descriptor calculation for all frames can be rapid.
able objects, and hence impart localisation information through haptic feedback.

**Overview of the App**

Two important criteria for the App are used:

1. Manual intervention from the user should be minimised,
2. the space available for feedback should be maximised.

To address the first criterion, the App was programmed to take photos automatically at fixed intervals. This removed the need for any buttons, allowing the map to be scaled to fit the 7 inch screen of the Senseg™ device. Figure 52 shows a screenshot from the App, with colour-coded information to provide additional visual feedback:

1. The yellow outline represents walls – the limits of the map – and imparted the greatest intensity feedback.
2. The grey lines form a grid system. A grid system was used for two main reasons:
   a) There needs to exist distinct boundaries between haptic feedback positions to allow the user to differentiate between them.
   b) To allow the user to quantify how far they are from reference points. For example, here the map consists of 10 boxes between the entrance and exit. Each box therefore represents 10% of distance between the start and the end of the corridor. This allows the user to estimate their current location by using relative distance between the start and end points.

   The perimeter of the boxes have the same “Edge Tick” haptic feedback assigned to them.
3. The green box represents the user’s estimated position at any given time. The whole area of the box has a “Grainy” texture assigned to it. This allows users to identify their location along their journey.
Figure 52: Senseg™ App screen. The yellow outline represents walls. The grey lines form a grid system for relative localisation. The green box identifies the user’s estimated location. The red box depicts the location of the user’s touch, and was used for debugging purposes. The horizontal scale at the bottom indicates relative position in the journey. The camera image is also displayed for debugging purposes.

4. The red box represents the location of the users’ touch on the screen over any of the boxes in the grid. This was used to ensure that the App was registering touches correctly when experimental data was taken.

The percentage bar at the bottom allows sighted users to obtain feedback of how far along a specified journey they are (in our case between the beginning and the end of the corridor). Distances are measured in a normalised scale from 0 to 1. This normalised scale allows ready adaptation to different tactile screen geometries and methods of user-interaction. As a measure of distance travelled along a desired path, it also eas-
ily conveys a sense of how fast one is making progress along a planned route.

Task flow

The task flow of the user is to obtain location information as they progress along their journey. For this, the App needs to be integrated with the server that takes images as input and outputs location information. It would be inconvenient for visually impaired users to manually request localisation information, so a picture of the user’s surroundings is taken automatically at fixed time intervals. This is then uploaded to the server which in response returns a number between 0 and 1; 1 represents the completion of the journey (to the end of the corridor in this example), and 0 represents the start (of the corridor). The App maps this information onto the grid system, the green box and the percentage bar at the bottom. The user can then identify their whereabouts in relation to the walls and other objects on the grid.

Client-server integration

When requesting localisation information, the user could carry a wearable camera, such as a Google Glass, for visual input. This would be paired with the App on the haptic device. As the user navigates the environment, the wearable camera takes low resolution pictures at regular intervals. The intervals can be chosen to minimise processing/battery usage, whilst still providing responses that are usable in real-time. Each picture is sent to the Node.js HTTP server via a POST request to a specific URL endpoint. The HTTP server asynchronously saves the image and calls the appearance-based matching code. This code returns the estimated location to the Senseg™ tablet via the HTTP response. Under the assumption that the indoors area has a Wi-Fi network, we have chosen to offload the computation to a server at the cost of bandwidth [155]. This arrangement is supported by the bandwidth requirements of the appearance-
based approach that we settled on. This is because, unlike the SLAM techniques, the appearance-based method appeared to work with quite small images, requiring no more than $\approx 40\text{kB}$ per greyscale image, and no more than $120\text{kB}$ per colour image.

**Experiments**

*Dataset*

For these experiments I used the sequences from the RSM dataset which acquisition and details were described in Section 4.4. The dataset is publicly available for download at [http://rsm.bicv.org](http://rsm.bicv.org).

*Experiments on localisation: Live query scenario*

I benchmarked the performance of the appearance-based methods in Chapter 4, and carried out a comparison between SF-GABOR and SLAM in Section 4.7.1. These benchmarks, however, were part of an evaluation pipeline, open-sourced and publicly available [140].

In this case, a “live” scenario is recreated: as briefly introduced in Section 6.2 and depicted in Figure 2, three different users used the client Android app to record sequences of one of the corridors in the RSM dataset and the frames were submitted to the matching server. As described in Section 6.4 this server computes a “streaming” version of the appearance-based BOVW pipeline for the query image and compares it using a kernel distance to a database of pre-computed training kernels. As we will see in the next section, the location estimates given by the highest score (or lowest distance between encoded visual words) are presented in the form of haptic feedback for the user subjects to provide an estimate of where they are along the path.
Blindfolded users with tactile sensing

Aim

The aim of this experiment was to evaluate the quality of the tactile feedback when used with blindfolded users who were attempting to estimate their locations. Blindfolded users received a tactile cue on the tablet that encoded an estimate of their position along a specific journey relative to the start and end points. Given several location estimates conveyed through the Senseg™ tactile interface, the experiment assesses the accuracy of tactile feedback for localisation through the user’s perception of their position.

Experimental protocol

Eighteen volunteers were asked to conduct the following steps:

1. Firstly, the volunteer was asked to ground his/her hands.

2. The volunteer was then given some familiarisation tasks with a Senseg™ demo that shipped with the tablet in the form of an Android App (“HapticGuidelines”). This App allows a user to gain familiarity with the feel of the different textures that they would encounter within our custom localisation App. Volunteers were asked to determine which of their fingers appeared the most sensitive to the haptic effects. They were also asked to find the correct finger movement speed to obtain the most feedback.

3. After the nature of the experiment was announced to each volunteer, they were again given the haptic tablet with the localisation App already launched. Two red rectangles denoted the start and end points and had specific – relatively intense – textures. Each of these boxes had a screen size of $150 \times 75$ “tixels”, equivalent to the same number of pixels in the touchscreen display. According to the Nexus 7

---

1 In reality, the App design uses the Android “Supporting Multiple Screens” API from the software development kit [61], which establishes a density-
(2012) specifications, the screen resolution is 216 ppi (85 ppcm), and so the actual height of each box corresponds to around 0.35 in (0.88 cm).

The volunteer was then asked to search for four landmarks:

a) the beginning of the path,

b) an area with no haptic feedback, this was the area that would represent the path that users had already traversed,

c) an area with haptic feedback, that represents the remaining segment of the path. This feedback would be the same “Edge Tick” texture described in Section 6.5.

d) the end of the path, with highlighted haptic texture.

4. The volunteer was then carefully blindfolded with a clean tissue being placed between the blindfold and their eyes. The experiment then began.

5. Participants were given 20 tactile cues, each spaced 15 seconds apart. In the time between the cues, they were asked to estimate and announce their location estimate to the closest 10%; 0% was the starting point of the journey and 100% was the end point of the journey.

6. After 100 trials (5 users), it was found that participants were finding it hard to distinguish their whereabouts. This was found to be due to a build-up of static charge on the surface of the screen. From then on, (for the next 13 users), the screen was discharged after every two tactile cues.

In the following section, the results of the experiments – described above – are presented and discussed. Since one aim of this work has been to compare potential sources of error, we report the experimental outcomes of the visual position inference independent pixel (dp) for its responsive and multi resolution design. For the case of the Nexus 7 (2012) tablet, with a resolution of 216 ppi (85 ppcm), I used $100 \times 50$ dp, which converts to $150 \times 75$ pixels.
and of the ability to convey the inferred position separately, synthesising the implications towards the end of the section.

**Results**

*Blindfolded tactile experiments*

Two remarks are in order regarding the haptic tablet device. First, the accuracy of the location feedback improved after discharging the tablet’s screen with an electrostatic cloth at regular intervals. Figures 53a and 53b show the results for individual subjects; one can see a notable improvement in the users’ performance when the device was discharged between trials (summarised in Table 12). Secondly, noticeable haptic feedback could only be identified when the tablet was plugged into a USB charging port, i.e. when it was grounded. This currently limits its use as a portable haptic device.

Allowing for these limitations, we can see from Table 12 that there is a 58.18% hit rate and an average error of roughly 4 m. For comparison, the grid size used for a representative test corridor of 30.62 m long with discretised locations 10% apart is \( \approx 3 \) m. If we recall Table 6 from Chapter 4, we can see that within this size of tactile boxes the best appearance-based method (SF-GABOR) would give a correct estimate with a mean probability of 98.25%, whilst SLAM would only achieve 69.59%.

During the experiment the user placed their finger at either the end or the beginning blocks, which were a different texture (“Grainy”) to the map position (“Bumpy”). The position was updated on the map every 15 seconds which provided ample time for confident estimates. However, most users found it helpful to count blocks (edge tactile feedback) from the current position to the end block through the tactile feedback between these points.

Another observation from the experiment is the average time to estimate the location. The users took an average of 11.28 s (\( \sigma = 5.58 \) s) of the 15 s that were given to complete the task. This
### Table 12: Summary of the results of tactile feedback experiment. A precision metric can be calculated as $\text{prec} = \frac{\text{hits}}{\text{estimates}}$.

<table>
<thead>
<tr>
<th>Discharge</th>
<th>meanErr (m)</th>
<th>stdErr (m)</th>
<th>#trials</th>
<th>#estimates</th>
<th>#hits</th>
<th>precision (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>2.63</td>
<td>4.04</td>
<td>260</td>
<td>231</td>
<td>136</td>
<td>58.87</td>
</tr>
<tr>
<td>No</td>
<td>9.28</td>
<td>5.34</td>
<td>100</td>
<td>44</td>
<td>24</td>
<td>54.54</td>
</tr>
<tr>
<td>Overall</td>
<td>4.11</td>
<td>4.33</td>
<td>360</td>
<td>275</td>
<td>160</td>
<td>58.18</td>
</tr>
</tbody>
</table>

might seem a long time in a real navigation scenario. However, this experiment required the user to provide estimations on randomly selected locations, without any correlation between the tactile cues. During a typical real journey which has been planned using the map, once the journey has started, there would be a predictable progression to the order of the cues: a user has to concentrate more – and therefore spend more time estimating position – when large changes of trajectory happen.

A final comment is concerned with the large difference between the errors from the vision system and the errors in the user interface. The vision subsystem used in this “live” scenario had an average absolute error of 2.43% in the sequences of the tactile experiment. This represents sub-metre accuracy of 0.75 m for a journey of 30 m in length. The errors reported in earlier work presented in Chapter 4, 1.59 m, are still nearly 4 times less than errors from estimates of position based only on a user’s tactile perception of distance travelled (4.11 m). The error of the haptic feedback is therefore significantly larger than the error of the live system’s algorithm.

**Conclusion**

In the present chapter I have described a prototype indoor visual localisation system for the blind and partially sighted. This system provides visual localisation using an appearance-based engine for matching views taken from a wearable or hand-held device with an existing dataset contributed by previous users who have traversed the same routes. The feedback to users is
(a) Errors in localisation using tactile feedback via the Senseg™ tablet. Results from individual subjects spaced along the horizontal axis.

(b) Proportion of hits (correct estimates of the portion of the journey completed) together with the number of estimates provided by each subject.

Figure 53

provided through haptic cues via a Senseg™ tablet (a Google Nexus 7 device modified to allow extra haptic feedback).

In previous chapters, I described the use of location appearance to deduce one’s position from low resolution images taken from a wearable or hand-held camera. The main outcome of the work described in this chapter is the mapping of that position through a tactile device to a user. Additionally, I have found that the architecture of such a system can be remarkably simple. Thus I described the components of the prototype: the algorithms behind the computer vision component, the architectural aspects of the App, and the client-server interaction (the
Figure 54: In blue, the histogram of drawing $10^6$ samples from a uniform distribution. Overlaying this, in red, the distribution of the users estimated locations when drawing $10^6$ samples from the experiment in random order.

client being an Android App and the server being a Node.js HTTP server).

Additionally, I evaluated the accuracy of the subjects’ perceived locations supplied via a particular haptic cue located on a map laid out on a tablet’s tactile screen, with additional haptic cues mapping boundaries and start and end points. Blindfolded users were able to perceive their location on a map with a minimal amount of practice, and that the resulting perceived location improved the accuracy of a user’s perception of their own position relative to start and end points of a pre-mapped journey. This supports the hypothesis that the ability of sighted users to see where they are on a map can be analogously provided to visually-impaired users through technology that can present a spatial layout and user position information via haptic feedback.

Taking both the precision of the haptic device and the accuracy of visual localisation into account, I suggested a technique to infer the error in localisation that reflects the limits of the haptic device and the position sensing technology. In effect, this allows us to determine the error in localisation that a given grid size on the tactile map may have for a specific journey.
In future work, two aspects of the work reported in this thesis might be extended: improving the visual processing and further exploring the capabilities of mapping visual information onto haptic devices. Generally speaking, the use of images captured with wearable cameras by a navigating person remains only superficially explored in the literature. Though power consumption and accuracy of detection remain key barriers to wide scale deployment, these barriers will be lessened over time. In addition to location estimation, the possibility of detecting obstructions, people and any deviations in environment from previous journeys holds great promise. In future work, it should be possible to integrate ground-plane detection on a wearable camera in order to detect irregularities in walking surface, or obstructions out to around a 5 m distance. Furthermore, other sources of data, for example Wi-Fi signal strength, can play a key role in both improving the reliability of position estimation, and robustness in the case of indoor lighting failure.

On the haptic side, there is the opportunity to refine the mapping from floor plans to tactile feedback. Returning to the Senseg™ platform as an example, a variety of textures could be conveyed to a user by varying the amplitude and temporal pattern of voltage pulses sent to the haptic interface. By combining the flexibility of this device with prior work on mapping textures to haptic feedback, it should be possible to improve the information conveyed to a visually impaired user by automatically harvesting visual information from an appropriately prepared map. For example, a standard map format that contains hatches or textures to illustrate locations of steps, or different types of rooms could be mapped to different tactile sensations on the device. Combining this information with lists of possible journeys that might be taken allows journey planning to be performed, and more readily opens up exploration of indoor locations.

Finally, a future direction that combines both the visual and haptic elements might motivate the design of an algorithm that extends gradient-based indexing techniques described in pre-
vious chapters. This could permit both scalable location indexing and the representation of textures via a haptic device. In particular, multi-directional spatial Gabor filters are attractive for visual analysis because, amongst their many uses in computer vision, they can be used to characterise image textures [75, 184, 1] and perform face recognition [189]. Both of these are applications of computer vision that hold potential for visually impaired users. Moreover, the same convolution operation can be re-used for both mapping of image texture to tactile textures and for location recognition. Although we did not use direct mapping of texture to tactile feedback in this study, it is a feature that we plan to investigate in the future (see, for example, the texture mapping work of Adi and Sulaiman [1]).
CONCLUSION AND FUTURE WORK

The main goal of this thesis was to explore appearance-based methods in the novel contexts of wearable and hand-held object recognition and visual localisation. In Chapter 5 I also explored a biologically-inspired algorithm for localisation based on visual input.

As means to achieve this objective I collected two large datasets; provided a thorough evaluation of baseline and custom-created image description methods; developed a biologically inspired model of place cells for visual localisation and produced a prototype system for assistive localisation using wearable and/or hand-held visual input and tactile feedback.

In this final section I will extend this summary for each of these contributions and suggest some future directions.

SUMMARY OF CONTRIBUTIONS

1. Appearance-based methods for wearable and assistive applications First, I have analysed the impact of computer vision in mobile and wearable technologies in an assistive context, providing studies of appearance-based meth-
ods for two applications, hand-held object recognition of household products and indoor navigation.

2. **Artificial place cell model** Second, I have provided a novel artificial place cell (APC) model that appear to replicate rate-coding behaviour of their biological counterparts found in the hippocampus. These models were tested under challenging conditions of indoor navigation by using a generalised regression neural network as a training mechanism for learning a positional ground truth from a database.

3. **Prototype of an assistive application** I took these previous findings to the next step and developed a prototype client-server Android application for assistive localisation from wearable and hand-held devices using their visual input and a haptic feedback tablet (the Senseg™) to provide tactile cues to the location estimates. With this work I laid out the foundations for a crowdsourcing approach that extends the idea of using sensor data from wearable devices to localise a person.

4. **Two novel datasets** These contributions are accompanied by two datasets, namely the SHORT dataset for hand-held object recognition and the RSM dataset of visual paths.

**Concluding remarks**

*Appearance-based methods for wearable and assistive applications*

The research question for Chapters 2, 3 and 4 was whether the appearance-based methods extensively used in the object recognition field could be applied to the scenarios of wearable and assistive applications, with emphasis on two key applications: object recognition and visual indoor localisation. Therefore in Chapter 2 I studied the particularities of this use case, and defined simple image matching methods and metrics testing them with pilot data and paving the way for the more thorough evaluation to be carried out in the following chapters.
The work described in Chapter 3 focuses on hand-held object recognition. I described the acquisition of the SHORT dataset and the evaluation of popular appearance-based methods against this dataset. The dataset proved to be extremely challenging even for algorithms that had practically solved other datasets. In the last edition (2014) of the ImageNet Large Scale Visual Recognition Challenge (ILSVRC), almost all groups proposed some form of convolutional neural network achieving an average of 94.6% classification accuracy [145]. This shows that deep learning approaches are beginning to “solve” the ImageNet dataset, highlighting the importance of offering new challenges such as the object recognition from video that SHORT proposes.

In Chapter 4 I studied another important application, indoor visual localisation from hand-held and wearable cameras. We acquired another large dataset, the RSM dataset, built a benchmark and developed a series of appearance-based methods and metrics I believe are more descriptive than the current state of the art. In the benchmark, I showed how the proposed custom methods compared to standard methods such as dense SIFT or HOG3D, but also the state of the art SLAM method for indoor scenarios, LSD-SLAM. Concretely, the top performing one, SF-GABOR, exhibited a mean localisation error $\mu_{c, SF-GABOR} = 1.59$ m whilst LSD-SLAM $\mu_{c, LSD-SLAM} = 2.48$ m. I also argued that the power of SLAM resides in the robustness of its tracking mechanisms. The appearance-based methods, on the other hand have shown good results when tested in isolation, where no tracking was applied; therefore we believe they can work alongside SLAM to reduce estimation errors during SLAM’s optimisation stages, the same way OpenFABMAP is used in the own LSD-SLAM.

Artificial place cell model

In Chapter 5 I enunciated a different research hypothesis, which is in reality twofold. In first place, it was questioned whether it
was possible to find a model of biological place cells using the appearance-based methods described in Chapter 4. In particular, I was interested in analysing the behaviour of the kernel similarity metric when frame encoding vectors, or histograms of visual words, were used as inputs. The second research question was whether this artificial place cell models could provide self-localisation by mimicking the behaviour found in biology.

As shown, the answer to the two questions was positive. It was shown to be possible to construct artificial place cells (APCs), computational models of their biological counterparts, the biological place cells (BPCs). APC localisation results were tested with the RSM dataset. In particular, the models mimic the BPCs firing rate, showing a tuning curve-type function that peaks on locations that correspond to that of the query. Once the APCs were characterised, I presented two localisation models. One is based on the maximum response of the APC tuning curve, and the other is based on a generalistic regression neural network (GRNN) that has the advantage of providing sub-localisation – i.e. training the neural network regressor with a discrete number of APCs it is possible to provide continuous localisation for a given query. I also provide a complete evaluation using the descriptors described in Chapter 4 and establish a comparison with LSD-SLAM.

I believe the findings described in Chapter 5 represent the first model of APCs that relies purely on vision to provide localisation estimates. However, many questions remain unanswered, opening many lines for future work that will be described in Section 7.3.

Prototype of an assistive application

In Chapter 6 I described how, taking the localisation pipeline presented in Chapter 4, an assistive localisation prototype that used vision as input could provide haptic feedback to give user location estimates. With a client-server architecture, the visual input can be provided from a wearable camera or mobile or
tablet camera, processed in the server which sends the location estimation to the Senseg™, a tactile feedback Android tablet that is able to provide a range of different textures that the user can learn to interpret depending on the application.

I provided a description of the prototype and their results in a “live” scenario, and also an experiment on tactile feedback perception. The purpose of this experiment was to assess whether it is possible to use a device such as the Senseg™ to provide location estimations along one dimension, i.e. how far along the path the subject is.

The results showed that up to a certain extent (a precision of roughly 3 m) the users were able to distinguish neighbouring positions on the tactile tablet. However, we believe this only represents a pilot study and many questions should be addressed before this is tested with blind and partially sighted users. We will describe future work in the next section.

Datasets

SHORT dataset

In the case of SHORT, we took a slightly different approach and diverged from the trend in object recognition dataset research. Instead of going towards the domain and dataset depth of big data (although our datasets are not small, containing hundreds of thousands of examples), we emphasise the need of understanding better the constraints of the problem at hand (wearable or hand-held object recognition) and also comprehend why even state-of-the art deep learning approaches that purely learn from examples find it hard to generalise outside the dataset they’re being trained –and tested- on. Torralba and Efros already studied the perils of dataset bias and poor cross-dataset generalisation [165] and these were one of the main motivations to build the SHORT dataset as there were no available dataset that could capture the challenges of wearable or hand-held recognition of groceries.
With the expansion of SHORT from 30 to 100 categories in June 2014 the category depth problem was solved, allowing for enough generalisation challenge from within the dataset. The test set, with more than 130,000 images constitutes an extremely challenging dataset, with some preliminary work being carried out in our group demonstrating the poor performance of deep learning frameworks such as Berkeley’s Caffe [78] when tested against SHORT.

**RSM dataset**

The research hypothesis behind the RSM dataset was to collect a vision-based navigation dataset of indoor spaces that incorporated the particularities of human navigation and specifically, wearable or hand-held camera inputs. It is an innovative dataset for the localisation community, as existing datasets, albeit rather complete such as the NAVVIS [72], do not capture the challenges of realistic human motion and wearable devices or smartphone acquisition. The RSM dataset is also novel in the sense that provides a scenario of crowdsourced visual paths, whereby different users submit their collected sequences while they complete their journeys.

The RSM dataset makes available a large range of video-frame resolutions, from the native 1920×1080 to the lightest one we have tried: 208×117, which have proved its suitability for crowdsourcing and Internet-based applications.

We also provide one-dimensional ground truth along the journey, sufficient to test the accuracy of appearance-based methods in constrained indoor journeys that often traverse narrow corridors. However, it might be too constrained for different SLAM or SfM purposes that rather look to create a map or reconstruct the environment. In order to make the RSM dataset also attractive for these communities we would need to expand the ground truth to have three dimensions. This, and other future directions will be discussed in the next section.
FUTURE WORK

The topics covered in this dissertation remain challenging and active research areas. In this section I will summarise the future directions for each line of work previously described that appear to be particularly promising.

Datasets

One might think that a straightforward future work for a dataset is just to carry out an expansion. I believe that this is one of the key aspects for the growth and dissemination of the data. However, we need to take into account the current challenges and opportunities of crowdsourcing techniques and big data scenarios.

SHORT dataset

A large proportion of the time dedicated to the acquisition of SHORT was invested in prototypes of the acquisition set-up and trials for its testing. The intention was to have a flexible but at the same time reproducible set-up, as it is shown in Figure 8. As the number of categories in the last version of SHORT, SHORT-100 was deemed appropriate in terms of generalisation under our testing conditions, we decided to stop its development there. However, the more categories we have in the training dataset the better for this type of benchmarks (controlled training set vs. natural or wild test set) to be adopted. Therefore it is important to contact large suppliers of product images, supermarket and retail chains and propose collaboration plans beneficial to both the research community (these suppliers are capable of taking our approach to scale) and for the image suppliers, as they can have an important role in acquiring the models for future improved training algorithms.

Regarding the test set, a natural expansion would consist of the set-up of a web repository and the development of a re-
trieval mobile App so images of new grocery products could be contributed to the platform.

In this respect, there are open-source alternatives such as [74] and [62] that would facilitate this task as instead of developing a dedicated App, the SHORT test set acquisition can be a project within these initiatives and attract altruist contributors that might have an interest on this sort of projects.

Alternatively, modern mobile App and web technologies allow for easy deployment of Apps based on client-server communication using a RESTful architecture. By creating an API, posting images would be trivial, and the development of the client would be the final user’s choice.

**RSM dataset**

For the RSM dataset, we envisage that a larger number of corridors could attract more users. The expansion of the RSM dataset should ideally contain more variety of places, lighting conditions and overall, devices. In fact, this is an opportunity for crowdsourcing too, as a collection App or API as described in the previous section could on its own encourage the contribution of many different devices in the process.

We have recently developed synthetic data of similar-looking corridors using the Unity game engine to assess the performance of appearance-based localisation, with and without employing artificial place cell models. The passes of this synthetic journey will be incorporated into the publicly available RSM dataset repository for the community to evaluate their algorithms in both real and synthetic datasets. These passes contain expanded ground truth information – 3D camera pose and camera pitch and yaw – and we are including different camera movements in some of the passes to mimic human head movements that have an effect on wearable camera footage.

The current state of the real sequences of the RSM dataset, despite including different illumination conditions, no changes in the viewpoint were included. This can be a limitation in establishing where the algorithms tested in Chapter 4 might fail.
Nevertheless, descriptor density and a high frame rate ($\geq 24$ fps) were used as a mitigation strategy for the presence of obstacles.

Another limitation of the current version of the RSM dataset is the absence of obstacles and obstructions such as the presence of people in the sequences. This can have an impact in the performance of the algorithms that should be assessed. We are planning to incorporate “natural” sequences to the next version of the dataset, and there is ongoing work within the group to produce automatically blurred faces of people present in the sequences to preserve anonymity and maintain the open status of the dataset.

Finally, as mentioned earlier, one of the biggest limitations of the dataset is the absence of 3D ground truth. Therefore we have the opportunity to expand the dataset with 2D 3D ground truth but maintaining the natural particularities of human motion. Rich 3D ground truth have traditionally been provided by robots, depriving these datasets from real human motion traits. We are currently exploring depth sensor and multi-camera indoor datasets in our research group for the purposes of indoor tracking of human subjects, and we believe this information could be incorporated to our dataset to provide a richer ground truth and motivate additional projects such as the creation of multi-modal signatures between external human tracking, visual input and sensor readings.

The acquisition of this sensor data could add huge value to the dataset. It is easy to collect sensor readings: APIs are mature in mobile phone development kits, and at the same time would attract research from the sensor, the “Internet of Things” (IoT) and big data communities. One of our future ideas to develop within our research group, is the modelling of similar tuning curves as the ones produced by the place cells, but with inertial sensor data. The relationship between both sources of curves could have a huge impact in learning locations without the need of a map or even a database not obtained through crowdsourcing.
Appearance-based methods for visual localisation

As we are open sourcing the evaluation pipeline [140], we welcome different research groups to contribute code to expand the amount of appearance-based methods for visual localisation.

A second future line of work will be to test our best performing method in conjunction with SLAM, the same way Open FAB-MAP is used with LSD-SLAM. We believe the SLAM community should keep relying on appearance-based methods not only for loop closure tasks but to provide their localisation and mapping with some semantic information about the environment.

A third project that would naturally emanate from this work would be a hierarchical appearance-based localisation that would implement the latest advances in scene retrieval and multi-sensor positioning. The same way we are providing a localisation within one journey from a database of similar journeys, we could also provide localisation within a building, within a city and so on. Broader estimates can come from GNSS signals, Wi-Fi and other radio location-based services [179], whilst fine grained positioning can be provided by the appearance-based methods and a combination of these with SLAM when a database of previous journeys is not present. In Appendix A.4 I include a visualisation of the high-dimensional visual word space representing the RSM dataset in 2D and 3D using t-distributed stochastic neighbour embedding (t-SNE), a powerful dimensionality reduction algorithm for visualisations that can help lay the foundations for future studies on retrieval at the building or large scale level, rather than at the journey level as in this thesis.

Finally, a limitation of the evaluation presented in Chapter 5 was the lack of a comparison with “natural” sequences with obstacles and the presence of people in the scenes. This evaluation is forecast to be included in future work, as despite the high frame-rate and descriptor density, the use of sequences with the presence of human obstacles and other obstructions might impact the performance of the methods described including approaches based on structure from motion.
Biologically inspired localisation methods based on place cell models

We are currently exploring different normalisation and matrix equilibration techniques by which the localisation using APCs can be optimised. In particular, divisive normalisation has been proposed as a model to describe non-linear population coding effects observed within biological sensory neurons. Different forms of normalisation have also been applied in machine learning, such as $L_2$ normalisation across features for each sample.

In this current project, we are exploring the relationship between divisive normalisation and matrix equilibration, a technique that scales matrix rows and columns. Using mixed matrix norms, and introducing partial mixed matrix norms, we interpret divisive normalisation in terms of matrix equilibration. There are plans to evaluate the effect of divisive normalisation in our artificial place cells models. We hypothesise that the use of partial mixed norms provide methods of scaling ensembles of neurons and their responses over experiments in a variety of ways, some of which may improve the performance of artificial networks. We believe that the freedom to select different combinations of norms provides the potential to improve performance, but improvements are both highly context dependent, and network dependent. This is the reason for testing these techniques on our recently acquired synthetic corridor to be incorporated into the RSM dataset described in previous Section 7.3.1.2.

Another strand of work that aims to develop the biological motivation behind our model focuses on how different clustering methods can exhibit a closer relationship to biological models of population coding observed in sensory neurons. Specifically, clustering techniques such as Gaussian mixture models (GMMs) are better suited for accommodating clusters with heterogeneous sizes and inner relationships within them than k-means. The work by Perronnin et al. [127, 128, 76] demonstrated in the object recognition field (see Chapter 3) the suitability of a bag-of-words model based on the Fisher kernel, which is computed using the GMM parameters learnt from the data. In fu-
In future work, it is intended to use the Fisher kernel as the encoding technique for the video sequences and evaluate its performance when artificial place cells are constructed using this model.

**Assistive localisation Apps with visual input and haptic feedback**

In future work, there are plans to extend two aspects of the work reported in Chapter 6: improving the visual processing and further exploring the capabilities of mapping visual information onto haptic devices. Generally speaking, the use of images captured with wearable cameras by a navigating person remains only superficially explored in the literature. Though power consumption and accuracy of detection remain key barriers to wide scale deployment, these barriers will be lessened over time. In addition to location estimation, the possibility of detecting obstructions, people and any deviations in environment from previous journeys holds great promise. In future work, the group plans to integrate ground-plane detection on a wearable camera in order to detect irregularities in walking surface, or obstructions out to around a 5 m distance. Furthermore, as discussed earlier, other sources of data like Wi-Fi signal strength, can play a key role in both improving the reliability of position estimation, and robustness in the case of indoor lighting failure.

On the haptic side, one could refine the mapping from floor plans to tactile feedback. Returning to the Senseg™ platform as an example, a variety of textures could be conveyed to a user by varying the amplitude and temporal pattern of voltage pulses sent to the haptic interface. By combining the flexibility of this device with prior work on mapping textures to haptic feedback, it should be possible to improve the information conveyed to a visually impaired user by automatically harvesting visual information from an appropriately prepared map. For example, a standard map format that contains hatches or textures to illustrate locations of steps, or different types of rooms could be mapped to different tactile sensations on the device.
Combining this information with lists of possible journeys that might be taken allows journey planning to be performed, and more readily opens up exploration of indoor locations.
Part I

APPENDIX
APPENDICES

ALGORITHM FOR GENERATING CUMULATIVE ERROR DISTRIBUTIONS

Algorithmus 1: Calculation of the error distribution

Inputs:
- Database of kernels, $\mathcal{K}_{c,p,1}$
- $c = 1, 2, ..., N_c$, // corridor index
- $p = 1, 2, ..., N_p$, // pass index
- Number of permutations, $P$
- Number of random queries, $Q$

Outputs:
- Error Distribution, $X$

// Compute localisation error for all possible queries
for $c \leftarrow 1$ to $N_c$ do
  for $p \leftarrow 1$ to $N_p$ do
    // For each query frame in a pass ...
    foreach $q : q \in P_p$ do
      // Take the corresponding kernel computed by
      // leave-one-out strategy and get closest neighbour
      $\rho \leftarrow \text{getClosestNeighbor}(K)$
      // Given the ground truth for that query, compute the error
      $E_{c,p,q} \leftarrow \text{computeError}(\rho)$
    end
  end
k $\leftarrow$ 1
for $i \leftarrow 1$ to $P$ do
  for $j : j \leftarrow 1$ to $Q$ do
    $e_k \leftarrow \text{randomSampling}(E)$
    $k \leftarrow k + 1$
  end
end
// Compute Cumulative Distribution Functions
$X \leftarrow \text{computeCDF}(e_k)$
TENSOR CONVOLUTION

We have found it useful to adopt the definitions of [86], in which the tensors are interpreted as multidimensional (multi-way) arrays. The authors also introduce or formalise operations upon and between tensors. In Kolda and Bader’s notation and nomenclature, the meaning of a tensor is different to that of classical physics and stress-analysis, in which tensors are mathematical entities that obey strict transformation laws.

In Kolder and Bader’s (K&B’s) terminology, the order of the tensor is the number of dimensional indices required to address it; for example, an order 5 tensor \( A \) may have addressable elements \( a_{i_1,i_2,i_3,i_4,i_5} \), with each index varying from 1 to \( I_n \), \( n = 1, 2, 3, 4, 5 \) in integer steps; note that in contrast with the K&B notation, indices are comma-delimited. Since each element of the tensor can be restricted to be real-valued, we may consider \( A \) as lying in \( I_1 \times I_2 \times I_3 \times I_4 \times I_5 \)-dimensional real space. The mode of a tensor refers to the tensor elements simultaneously addressed by one of the indices, and is applied to refer to operations that involve, possibly non-exclusively, a particular one of the indices. Definitions of tensor-vector and tensor-matrix products follow [86], with tensor contraction as described in [13] and also [3].

In the following definitions, we will refer to the tensors \( A, B \) and \( C \), where \( A \in \mathbb{R}^{\prod_{n=1}^{N_A} I_n} \) is of order \( N_A \), containing elements \( a_{i_1,i_2,...,i_{N_A}} \), and \( B \in \mathbb{R}^{\prod_{n=1}^{N_B} I_n} \) is a tensor of order \( N_B \) with elements \( b_{j_1,j_2,...,j_{N_B}} \), and \( C \) is of order \( N_C \).

**Definition 1: Tensor Convolution** We denote the tensor convolution operator in modes \( M \) by the following:

\[
A^M \ast B : (A, B) \mapsto C,
\]

(33)
where $\mathcal{M}$ is a set of $|\mathcal{M}|$ tuples representing paired indices of $A$ and $B$ over which the convolution is performed. These indices associate the modes of the tensors being convolved together; if single mode indices, rather than tuples $(\cdot, \cdot)$ are provided, then it is understood that the modes are repeated for the second element of a tuple.

The tensor convolution operator maps equal-order tensors, $A$ and $B$ to a tensor $C$ by the following:

$$A \star^\mathcal{M} B = \sum_{i_{m1}'} \ldots \sum_{i_{M}'} \sum_{i_{1}} a_{i_{1},i_{2},\ldots,i_{m1}^{'},\ldots,i_{M}^{'},\ldots,i_{N_{A}}} \times b_{i_{1},i_{2},\ldots,i_{n1}-i_{1}^{'},\ldots,i_{nM}-i_{M}^{'},\ldots,i_{N_{B}}}$$

where $\mathcal{M}$ takes the form of a set of tuples that associate indices in $A$ with those in $B$ for the convolution:

$${\{(m_1,n_1), (m_2,n_2), \ldots, (m_M,n_M)\}}.$$
**Definition 2: Permuted Tensor Convolution** We define the permuted tensor convolution operator in modes $\mathcal{M}$ permuted over the modes $\mathcal{P}$ as a mapping taking the form:

$$\mathcal{M} \big[ * \big] \mathcal{P} : (A, B) \mapsto C,$$

where $\mathcal{M}$ is a set of $|\mathcal{M}|$ tuples representing paired indices of $A$ and $B$ over which the convolution is performed and $\mathcal{P}$ represents the modes of $A$ and $B$ for which permutation is performed, expanding the order of $C$ relative to that of tensor convolution.

The permuted tensor convolution operator maps tensor, $A$, to the higher-order tensor $C$ by the following:

$$A^{\mathcal{M} \big[ * \big] \mathcal{P}} B = \sum_{i'_{m_1}} \cdots \sum_{i'_{M}} a_{i_1,i_2,...,i_{m_1}',...i_{M}',i_{P_1},i_{P_2},...i_{P_P},...i_{N_A}} \times$$

$$b_{i_1,i_2,...,i_{n_1}',...i_{n_M}',i_{\pi(q_1)p_1},...i_{\pi(q_P)p_P},...i_{N_B}}$$

where $\mathcal{M}$ consists of the tuples:

$$\{(m_1,n_1), (m_2,n_2), ..., (m_M,n_M)\},$$

and $\mathcal{P}$ by the tuples:

$$\{(p_1,q_1), (p_2,q_2), ..., (p_P,q_P)\}.$$
Generally, $\mathcal{M} \cap \mathcal{P} = \emptyset$. As for the case of $\mathcal{M}$, if single elements are given for $\mathcal{P}$, it is understood that the second member of the tuple is the same; where no corresponding dimension exists in one argument, the $\sim$ denotes a null mode in the tuple.

This permutation is not across modes, but within the possible values that one mode can take. By way of example, given a definition of an order 2 tensor $A$ of order $2 \times 3$

$$A = \begin{bmatrix} 1 & 2 & 3 \\ 4 & 5 & 6 \end{bmatrix}, \quad (41)$$

and an order 3 tensor, $B$, of size $2 \times 2 \times 2$ where

$$B = \begin{bmatrix} 1 & 1 \\ & 2 & 2 \\ 1 & 1 \\ & 2 & 2 \end{bmatrix}, \quad (42)$$

then the tensor $C$ defined by

$$C = A \bullet_{\{i_1,i_2\}}^{\{[\sim,i_3]\}} B, \quad (43)$$

is of order $N_C = 2 + 1 = 3$, and will be of size $3 \times 4 \times 2$:

$$C = \begin{bmatrix} 1 & 3 & 5 & 3 \\ 5 & 12 & 16 & 9 \\ 4 & 9 & 11 & 6 \end{bmatrix} \cdot \begin{bmatrix} 2 & 6 & 10 & 6 \\ 10 & 24 & 32 & 18 \\ 8 & 18 & 22 & 12 \end{bmatrix}. \quad (44)$$
### LSD-SLAM Parameters

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Definition</th>
<th>Default</th>
<th>Set to</th>
</tr>
</thead>
<tbody>
<tr>
<td>minUser-Grad</td>
<td>Minimal absolute image gradient for a pixel to be used at all. Increase if your camera has large image noise, decrease if you have low image-noise and want to also exploit small gradients.</td>
<td>1.96</td>
<td>5</td>
</tr>
<tr>
<td>camer-aPixel-Noise</td>
<td>Image intensity noise used for e.g. tracking weight calculation. Should be set larger than the actual sensor-noise, to also account for noise originating from discretisation / linear interpolation.</td>
<td>16</td>
<td>2.4</td>
</tr>
<tr>
<td>KFUsage-Weight</td>
<td>Determines how often keyframes are taken, depending on the overlap to the current keyframe. Larger: more keyframes.</td>
<td>4</td>
<td>10</td>
</tr>
<tr>
<td>KFDist-Weight</td>
<td>Determines how often keyframes are taken, depending on the distance to the current keyframe. Larger: more keyframes.</td>
<td>3</td>
<td>10</td>
</tr>
</tbody>
</table>

(*) The values for the keyframe weights, KFUsage-Weight and KFDist-weight were increased following the suggestions of LSD-SLAM authors. By increasing these weights the thresholds to take keyframes are lowered, therefore more keyframes are taken, gaining robustness against tracking at the expense of a larger map, more loop closures and slower processing. Although both affect the amount of keyframes that are selected, KFDistWeight is an indirect weight applied to the distance between frames that has an influence in the keyframe selection threshold. KFUsageWeight on the other hand, directly modifies the keyframe selection threshold.
In the previous sections I have extensively studied the case of the localisation within a journey, answering the question “where am I along the path?” that was introduced in Chapter 2. In a visual path retrieval system divided in different journeys inside a building, to be able to answer the question “in which path am I on?” with precision would give this system the necessary prior information to provide a better location and also suggest path planning, which would be specially relevant in an assistive context as we will see in Chapter 6. Although the journey selection was beyond the scope of this thesis, it was informative to study the behaviour of a state-of-the-art dimensionality reduction technique in a rather challenging scenario of having such highly dimensional data (the BOVW-encoded vectors have 4,000 elements). Therefore I chose t-SNE as a technique for visualising in two or three dimensions the high dimensional descriptor space of the RSM dataset.

_t-distributed stochastic neighbour embedding (t-SNE)_ is a machine learning algorithm for dimensionality reduction developed by Laurens van der Maaten and Geoffrey Hinton [99]. It is a non-linear dimensionality reduction technique that is particularly well suited for embedding high-dimensional data into a space of two or three dimensions, which can then be visualised in a scatter plot. Specifically, it models each high-dimensional object by a two- or three-dimensional point in such a way that similar objects are modelled by nearby points and dissimilar objects are modelled by distant points.

In other words, t-SNE is a dimensionality reduction technique that aims to preserve the local structure of the data. For this reason I wanted to compute the visual path descriptions of the RSM dataset and build the foundations for future work on journey/corridor selection.

In Figure 56 we can see the 4,000-dimension visual word reduced to three dimensions and in Figure 57 we can see the two-dimensional embedding. The embeddings were generated
Figure 56: Distribution of the BOVW data of the RSM dataset in a reduced 3D space when visualised with t-SNE. Colours refer to different corridors in the dataset. Note that there is some evidence of the locally connected paths in the visual-words space.
using more than 50,000 randomly selected examples from all the corridors. Following the method described in Section 4.3.5, I selected for this particular example dense-SIFT descriptors encoded with hard assignment (HA), using k-means to create the visual word examples.

As we can see from the images, the difficulty of generating two or three-dimensional embeddings of such a high dimensional and complex dataset is notable. However, there are patterns showing how examples from the same corridors can display a sequential relationship within the embeddings.

The present thesis gives an emphasis on understanding visual path data from a journey perspective, from a crowdsourced collection of journeys in particular. However, although of limited practical use within journey localisation, this visualisation is the first step in understanding the structure of the data from a “building” perspective.
It is therefore subject of future work the use of these visualisations to understand the important features of visual paths datasets such as the presence of global clusters that reveal remarkable distinctiveness between journeys or give insight on how to optimise the retrieval based on between-journey differences.
TACTILE FEEDBACK EXPERIMENT PROTOCOL

Context

The “Visual localization with tactile feedback” project aims to evaluate the quality of the tactile feedback given by the Senseg tablet in an indoor localization for the visually impaired context. When navigating a physical path, the user receives a tactile cue that encodes an estimate of their position along that specific path, relative to start and end point. Given several location estimate feedback cues through the Senseg tactile interface, the goal of this experiment is to evaluate how accurate this tactile feedback is based on the user perception of the position they are.

Experiment protocol

1. The user will be given some familiarization tasks with the Senseg demo that shipped with the tablet as Android applications. These will be:

   • Familiarize with the different textures with the app “Haptic Guidelines”.

2. The user will be given the following instructions:

   You have agreed to take part in the “Visual localization with tactile feedback” project experiment on tactile feedback quality. The experiment consists of the following tasks:

   a) You will be given the Senseg tablet you used previously to get familiar with its tactile interface.

   b) If you visually inspect the path, you will notice 2 red rectangles that denote starting and end point. These are texture highlighted. As you feel the screen with your finger and move it over the
A 5 tactile feedback experiment protocol 188

path you will notice four haptic “landmarks” or “events” that can be differentiated:

i. the beginning of the path,

ii. an area with no haptic feedback, this is the area that would represent the area that users have already traversed,

iii. an area with haptic feedback, that represents the remaining segment of the path,

iv. the end of the path, with highlighted haptic texture as event (i).

c) You will receive one tactile cue every 15 seconds, making up to 20 cues.

d) Upon the reception of the cue, it will be your task to announce an estimate of your location as a percentage of the total distance. You will only provide estimates that are 10% apart:

• 0%: starting point of the journey,
• 10%
• 20%,
• ...
• 80%,
• 90%
• 100%: end point of the journey.

e) As agreed, you will blindfold yourself for this experiment. Please, proceed to wear the blindfold now, the experiment will start shortly.

3. The experiment will start:

a) The user will receive 20 tactile cues corresponding to 20 randomized location estimates provided by the localization server.
b) The users’ announced estimates will be annotated next to their corresponding index in the following table\(^1\)

<table>
<thead>
<tr>
<th>Trial index</th>
<th>True location</th>
<th>Estimated location</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0.81</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>0.91</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>0.13</td>
<td></td>
</tr>
<tr>
<td>4</td>
<td>0.91</td>
<td></td>
</tr>
<tr>
<td>5</td>
<td>0.63</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>0.10</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>0.28</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>0.55</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>0.96</td>
<td></td>
</tr>
<tr>
<td>10</td>
<td>0.16</td>
<td></td>
</tr>
<tr>
<td>11</td>
<td>0.49</td>
<td></td>
</tr>
<tr>
<td>12</td>
<td>0.80</td>
<td></td>
</tr>
<tr>
<td>13</td>
<td>0.14</td>
<td></td>
</tr>
<tr>
<td>14</td>
<td>0.42</td>
<td></td>
</tr>
<tr>
<td>15</td>
<td>0.79</td>
<td></td>
</tr>
<tr>
<td>16</td>
<td>0.66</td>
<td></td>
</tr>
<tr>
<td>17</td>
<td>0.04</td>
<td></td>
</tr>
<tr>
<td>18</td>
<td>0.93</td>
<td></td>
</tr>
<tr>
<td>19</td>
<td>0.40</td>
<td></td>
</tr>
<tr>
<td>20</td>
<td>0.27</td>
<td></td>
</tr>
</tbody>
</table>

\(^1\) Not present in the volunteer copy.
Informed consent form

Experiment purpose & procedure

The purpose of this experiment is to evaluate the quality of the tactile feedback given by the Senseg tablet in an indoor localization for the visually impaired context.

The experiment consists of 2 parts as detailed in the previous section.

After the experiment, you will be asked to complete a feedback form.

Please note that none of the tasks is a test of your personal intelligence or ability. The objective is to test the usability of our research systems.

Confidentiality

The following data will be recorded: Estimates of the tactile-encoded position along a path based on Senseg haptic feedback.

All data will be coded so that your anonymity will be protected in any research papers and presentations that result from this work.

Finding out about result

If interested, you can find out the result of the study by contacting the researcher Jose Rivera-Rubio, after 1 April 2015.

His email address is jose.rivera@imperial.ac.uk.

Record of consent

Your signature below indicates that you have understood the information about the “Tactile feedback with Senseg” experiment and consent to your participation. The participation is voluntary and you may refuse to answer certain questions on the questionnaire and withdraw from the study at any time with no penalty. This does not waive your legal rights. You should have received a copy of the consent form for your own record.
If you have further questions related to this research, please contact the researcher.

Your Name ______________
Your Signature ____________

Researcher: Jose Rivera-Rubio
Date: 10/03/2015
BIBLIOGRAPHY


[8] Pablo Fernández Alcantarilla, José J Yebes, Javier Almazán, and Luis Miguel Bergasa. On combining visual SLAM and dense scene flow to increase the robustness of localization.


[74] Apple Inc. The researchkit framework an open source software framework that makes it easy to create apps for medical research or for other research projects., 2013-2015. URL http://researchkit.org/. [Accessed May 1st, 2016].


[76] Hervé Jégou, Florent Perronnin, Matthijs Douze, Jorge Sánchez, Patrick Pérez, and Cordelia Schmid. Aggregating


COLOPHON

This LaTeX document was typeset using the typographical look-and-feel classicthesis developed by André Miede.

London – Milton Keynes

Final Version as of June 29, 2016
(classicthesis vo.8).