Author:
Panphattarasap, Pilailuck

Title:
Urban Patterns

Urban Patterns: Using Spatial Arrangement for Vision-Based Place Recognition and Localisation

Pilailuck Panphattarasap

University of Bristol

A dissertation submitted to the University of Bristol in accordance with the requirements for the degree of Doctor of Philosophy in the Faculty of Engineering

March 2019

Abstract

In this thesis, we investigate the impact of using visual landmarks with spatial knowledge to improve the performance of vision-based localisation. We separate our work into two parts. First, we introduce a new place recognition method based on a novel representation called *landmark distribution descriptor* (LDD) which combines landmark identification based on CNN features with their spatial distribution across a view. We use the representation to do matching within an image-to-image place recognition framework, which is to compare test images with single images taken at distinct locations in urban environments. Results on large datasets from 10 different cities obtained from Google StreetView and Bing Streetside demonstrate an average precision of around 70% (at 100% recall), compared with 58% obtained using whole image CNN features and 50% for a comparable landmark method without spatial information. Second, we investigate the problem of localisation in urban environments using only image data and a 2-D map of the area. We employ *binary semantic descriptors (BSD)*: 4-bit binary descriptors indicating the presence or otherwise of salient landmarks at a given location which are indicated on the 2-D map. On their own, these descriptors are not sufficiently distinctive to allow localisation. However, when combined sequentially over routes, the resulting concatenated descriptors prove to be highly discriminative, enabling robust localisation corresponds to the map. Performance can be further improved by incorporating the turn information along with a route. Landmark presence in 360-degree images taken at a given location is detected using a CNN binary classifier, trained using Google StreetView and OpenStreetMap data. Experiments with over 6,000 locations in an urban area show that the approach can give 95% of accuracy using an average route length of 200 metres. Thus, in both works, we demonstrate that landmarks combined with spatial knowledge provide an effective means of improving vision-based localisation.

Declaration

I declare that the work in this dissertation was carried out in accordance with the requirements of the University's Regulations and Code of Practice for Research Degree Programmes and that it has not been submitted for any other academic award. Except where indicated by specific reference in the text, the work is the candidate's own work. Work done in collaboration with, or with the assistance of, others, is indicated as such. Any views expressed in the dissertation are those of the author.

SIGNED: .... DATE:..........................

Acknowledgements

First of all, I would like to give my appreciation to my supervisor Dr Andrew Calway, without his help, I may not be at this point I am now.

Second is my family and friends, especially my parents who always be there when I need even they do not quietly understand what I am doing. I am also glad that you guys are okay with me dedicating this thesis to food and books.

There are many people that I may not put in here as I am afraid that I might miss some names, but I always remember everyone's support.

When I think back to the starting point, I am a little bit fascinating to how much I have changed. This is a really long and tough journey for me and it is time to move forward. I might not suddenly change the world, but I, at least, give some contributions to the field. Thus, I would like to say thank you to myself for never giving up until the end.

This thesis is dedicated to
Salmon, novel books, and the fictional characters
for cheering me up all these years

Contents

List of Figures

Chapter 1

Introduction

Modern technologies have made significant progress and are merging with daily life in all areas, from basic operations, such as household activities, leisure, and entertainment, to the infrastructure level, such as transportation, security, and exploration. Technology development is moving towards the concept of autonomy, which refers to the ability of systems to execute their tasks with limited or no inputs from humans. Examples include mobile robots, self-driving cars, drones, and some wearable applications, as illustrated in Figure 1.1. This thesis focuses on autonomous navigation.

1.1 Autonomous localisation and navigation

To move from one place to another is the common concept of the navigation. To automate the task, which is the large-scale operation, systems should be able to identify where they are in the world and to navigate themselves to new locations. The general processes for autonomous navigation are (i) self-localisation, (ii) path planning, and (iii) navigation. Consider the scenario of an autonomous robot being placed in a city and commanded to go from its current location to a given destination; to accomplish that, the robot needs to know where it is in the world by comparing stored knowledge with data obtained from sensors. Once localised, the robot makes a decision based on that location to navigate itself to the destination. The given example can be integrated into numerous real-world applications, such as the systems that assist human during shopping [32, 33] and the automated system that helps to navigate impaired people [34, 35], as well as into industrial applications, such as an unmanned aerial vehicle for military operations [36].

Therefore, 'Where am I?' is the first question for autonomous navigation. The task of identifying the current location is known as localisation. To do that, the systems rely on their equipped sensors to read their surroundings. For large-scale operations, the sensors

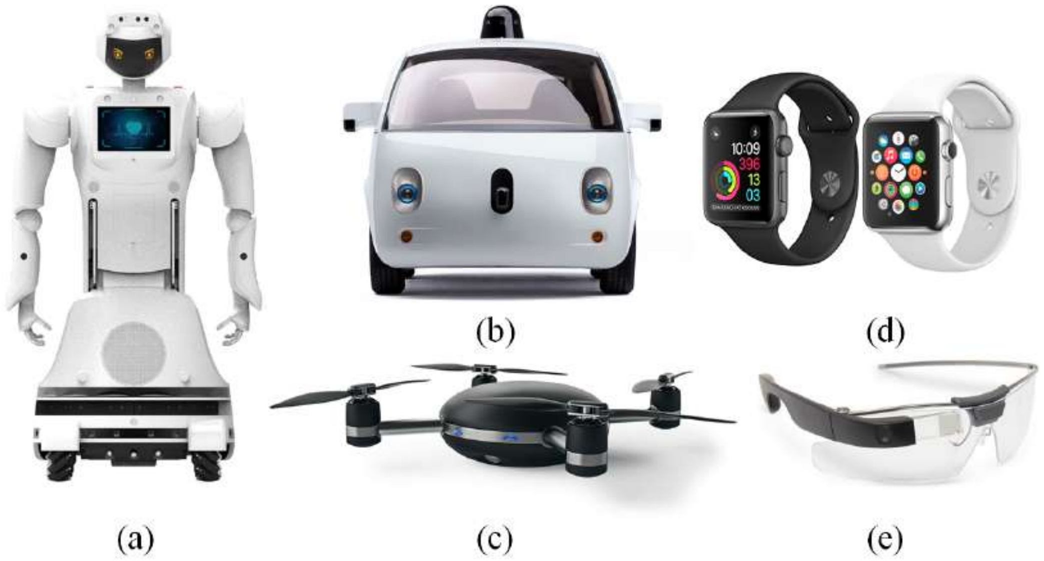
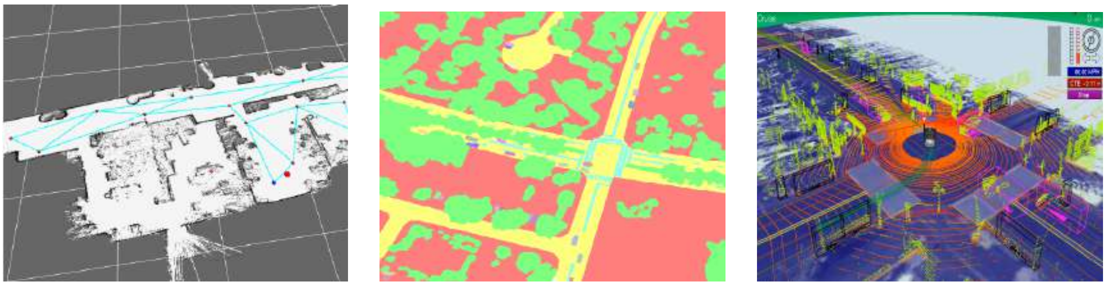


Figure 1.1: Examples of autonomous systems: (a) a mobile robot in a humanoid form, (b) self-driving car, (c) drone, (d) smartwatches and (e) smart glasses. The examples in (a), (b), and (c) are systems with high autonomy. They can operate without human guidance. The examples in (d) and (e) are wearable devices that have autonomous applications, such as health monitoring and exercise planning. Note that these images are taken from [1, 2, 3, 4, 5, 6].

autonomous systems usually rely on are in some form of odometry, such as wheel movement, and a range of sensors, such as sonar, laser, structured light, and global positioning system (GPS). The raw sensor inputs are interpreted via algorithms and compared with records of the surrounding environment, and the systems orient themselves in relation to the records. In general, the records of the environment are known as maps. They can be pre-defined or simultaneously built by the system during the localisation process (often known as simultaneous localisation and mapping [SLAM] in robotics [37, 38]). As illustrated in Figure 1.2, several types of maps contain different data. Choosing the type of map usually depends on the purposes of the systems and their equipment.

For example, self-driving cars (Figure 1.1b) typically make use of LIDAR (pulsed laser) depth sensors, so the map they produce is in a 3-D form (Figure 1.2c). The mobile robot (Figure 1.1a) usually projects the map in topological or metric forms (Figure 1.2a). As for aerial vehicles, like drones (Figure 1.1c), to localise the ground-level area, the most suitable map is in a 2-D form (Figure 1.2b). Using the map for autonomous navigation is similar to how humans orient themselves in an unknown environment. Humans retrieve surrounding information via their sensory organs (eyes, nose, or ears) and combine the information with their mental or physical maps. The techniques of map-making, or cartography, have been around since ancient times [39] and developed in parallel with technology [40]. More details are provided in Chapter 4.



(a) (b) (c)

Figure 1.2: Different types of maps using in autonomous navigation: (a) topological and metric maps from [7], (b) a segmented 2-D map retrieved from drones and segmented using commercial-level API for autonomous vehicles [8], and (c) 3-D map generated by the pulsed laser of the Google self-driving car from [9]. Each map displays different degrees of information stored.

1.2 Vision-based localisation

The well-known technique for localisation is to use positioning systems, such as GPS. However, in a cluttered environment, like urban areas, positioning-based systems are not fully functioning due to the limitations of the signals. As the positioning-based system requires information about the external infrastructure, it creates a sense of infrastructure dependence, meaning the autonomous system is reliant on its availability at all times, which lessens degrees of autonomy. Therefore, sensors with more degrees of independence from infrastructure are preferable. Some applications use LIDAR depth sensors to generate the 3-D representation of the surroundings. The depth sensor's benefits include precision and speed. However, it only provides geometric information, which has limitations for recognising places and objects. In addition, by making use of point cloud data, this technique contains high operating costs that directly affect scalability for large-scale applications.

This thesis focuses on using a vision sensor for localisation, as this approach provides cheaper costs for installation, is more lightweight, is readily available without needing much knowledge, and, once combined with techniques from computer vision, the visual information retrieved from the sensor also provides multiple forms of information, such as colour, motion, and the presence of pedestrians and objects. Tasks of vision-based localisation consist of (i) recognising a place (place recognition) and (ii) identifying where it is located on the map (localisation) using only visual information. In this respect, vision-based localisation can be grouped as a part of content-based image retrieval (CBIR) (reviewed in [41]), with specific tasks for retrieving locations. Apart from image retrieval, the systems are integrated with a variety of computer vision techniques, such as image segmentation [42, 43], object detection [16, 44, 45], and pattern recognition [46, 47]. This is discussed in greater detail in Chapter 2.

The major techniques for vision-based localisation are pose estimation and image matching. The former directly obtains the 6 degrees of freedom (6-DoF) pose estimation of the query (including the location and orientation). The techniques usually combine with 3-D point cloud maps, as in [48, 49]. The latter uses image-to-image matching techniques to find the most similarity between the query and references in the geo-tagged database. There are two types of data manipulation: comparing the query with static databases, as in [50, 51, 52], and sequentially tracking the current location, as in vSLAM [53] and the probabilistic FAB-MAP [54, 55].

These stated methods introduced the ability to identify the current location of a system using only visual information. However, issues of concern are invariance and scalability. For invariance, without positioning infrastructure, automated localisation based purely on visual information highly depends on the location characteristics. Figure 1.3 illustrates the environmental changes, including temporal changes (season, time of day, and lighting direction) and spatial changes (viewpoints and occlusions). These changes can significantly affect the performance of place recognition systems; as reported in several studies [56, 57, 58, 59, 60, 61], the traditional techniques failed to cope with the issue. Figure 1.4 illustrates a simple environmental change: the time of day and the interest features (or salient features) detected in each image are different.

For scalability, the visual information requires massive amounts of storage, which, in practice, are limited by physical resources. Aside from maintaining accuracy, to support long-term operations, the main focus for scalability is speed. One factor that directly impacts the query time is the size of the database. For example, in [54, 55], even when using image features, which is smaller than using whole images, the storage needed for 1000 km routes is 177 GB. For the speed, the stated works compressed data using the vocabulary tree, but the technique is still insufficient for long-term operations. In other words, with an increase in explored areas, the traditional data structures and searching techniques do not entirely cover the problem, especially in terms of space and processing time.

In this respect, autonomous systems require the ability to handle environmental changes in visual information and to maintain performance even in the larger areas. Further details about the problems of invariance and scalability are provided in Chapter 2. To overcome these limitations, the studies of vision-based localisation shift to semantic reasoning approaches, which are more similar to human perception in location-based activities.

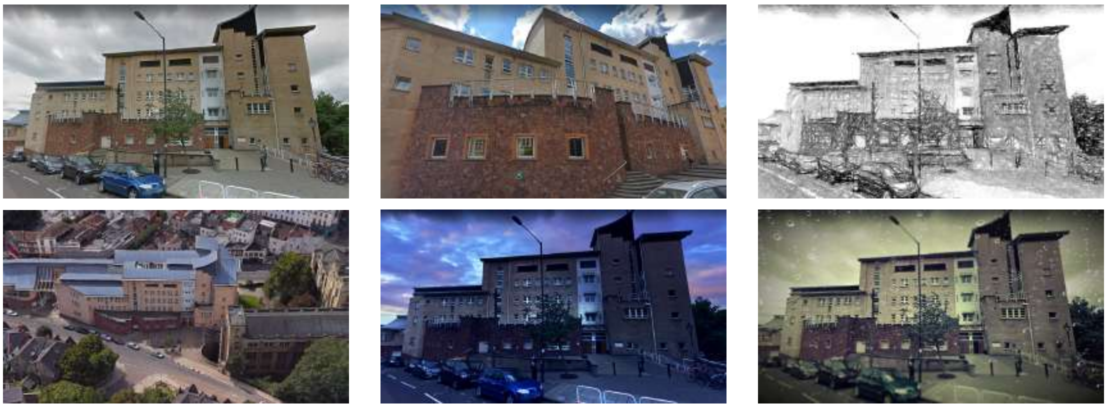
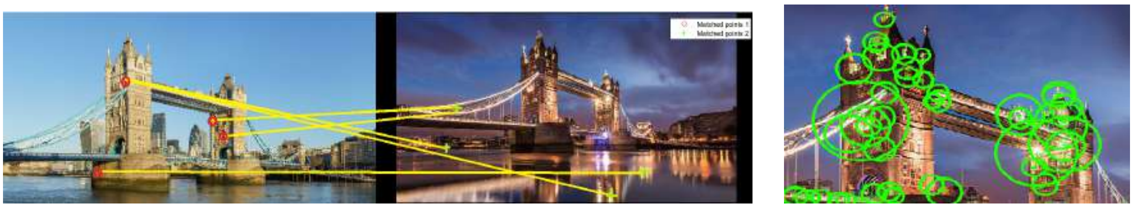


Figure 1.3: Challenges in place recognition using visual information. All images are taken at the same place, with variation, including temporal changes (season, time of day, lighting direction, etc.), spatial changes (viewpoints, occlusions, etc.), and domain changes (from digital to sketch).

1.3 Vision-based localisation using semantic features

Semantic approaches in the location-related study refer to the use of higher-order concepts of linguistic description. For example, a scene can be described in natural language, such as ‘a building A is on the left-hand side of a building B’, ‘there is a junction in front of the building A’, or ‘there is a gap between those buildings’. These semantic features are similar to the way humans describe their surroundings. For example, if a person takes an image of London’s Tower Bridge and presents it to other people, the image can be described by the name of the location (London Tower Bridge), the usage (a bridge to cross the river), and the geographical location in which the image is taken (London, the United Kingdom). Humans can perceive these semantic descriptions by glancing at the image. Additionally, when compared to another image taken at the same location, even with environmental changes, humans can still recognise the similarity. This contrasts with the way feature-based techniques work. As depicted in Figure 1.4b, the feature representation of the image is in a set of points or regions of interest. It does not include any meaningful information about the image.

Therefore, by integrating semantic information, the systems are likely to overcome both invariance and scalability. Regarding invariance, as shown in Figure 1.3, even with the environmental changes, viewpoint changes, and domain changes, the meaning of the objects in all examples (a building) is still the same. Figure 1.4 demonstrates a similar concept; even with changes, semantic information within the pair of images remains unchanged. In this respect, we believe that the use of semantic features is likely to cope better with invariance. Regarding scalability, by using the natural language description, the size of descriptors is



(a) (b)

Figure 1.4: (a) features matching between a pair of images taken at the same location (London's Tower Bridge), but at a different time of day and with a slight change in viewing position, while (b) highlights some of the strongest 150 extracted features. Note that the image has been cropped for visualisation. The yellow lines represent the correspondence between detected points of interest. This demonstrates the difficulty of using feature-based methods for recognising a place in terms of invariance. Both images used in this example are retrieved from [10].

likely to be more compact. For example, as illustrated in Figure 1.4b, describing an image requires a number of features, yet the whole scene can be semantically described using one or two linguistic terms. Therefore, as the size of the database directly affects its querying time, the use of semantic features also improves the sense of scalability. Motivated by this, a number of vision-based localisation systems have imitated the movement of humans using the higher concept of semantic features, such as landmarks [62], road lanes [63, 64], and texts of streets and shopfronts [65, 66], and by segmenting the semantic information from the scene, as in [65, 66]. These systems have demonstrated the robustness of localisation compared to feature-based methods, especially in terms of invariance to environmental changes.

As localisation using semantic approaches is inspired by humans, who describe places in terms of semantic information, it aligns better with human wayfinding, which is the human act of taking visible cues from their surroundings to orient themselves over their mental or physical maps [67, 68]. The visual cues include landmarks (either natural or manmade), texts, and symbols. Many psychology studies have also supported the use of semantic information in human wayfinding, especially spatial knowledge [69, 70, 71]. There are records of integrating spatial distribution in autonomous navigation [72]; however, the study still provides no solid foundation, which leads to the present research focus described in the next section.

1.4 Contributions

The main focus of this thesis is on investigating the use of semantic features in vision-based localisation. We explore this in two areas: image-to-image matching and image-to-2-D-map matching. The main contributions of this research are as follows:

- We investigate the impact of using spatial distribution for place recognition. We propose the use of image descriptors that encode the spatial distribution of objects in the scene. By applying semantic approaches, we aim to handle the invariance (temporal changes, spatial changes) and scalability of the system.
- We investigate the possibility of localisation on a 2-D map using visual information. We introduce the novel technique of using semantic representations. Therefore, we aim for a representation that is robust to invariance (temporal changes, spatial changes, and domain changes), compact for scalability, and closer to how humans perceive this problem.

This thesis includes seven chapters in total. The first contribution is covered in Chapters 2 and 3, while the second contribution is covered in Chapters 4–6. The outline is as follows:

- Chapter 2 provides a review of image-to-image place recognition techniques from the early to the recent stage and discusses the related works.
- Chapter 3 presents a novel approach to vision-based place recognition that enforces the spatial distribution of landmarks, called the landmark distribution descriptor (LDD), and demonstrates the use of LDDs.
- Chapter 4 provides a review of image-based localisation over cross-view references, especially 2-D maps, and discusses the related works.
- Chapter 5 presents a novel approach to using 2-D maps for localisation by applying a binary semantic descriptor (BSD) constructed by making use of the presence of semantic features, such as junctions and gaps between buildings.
- Chapter 6 explains the process of converting four directional images to a BSD by training classifiers to detect the presence of the selected semantic features and demonstrates the use of BSDs for localisation.
- Chapter 7 discusses final conclusions, findings, limitations of the systems, and future research directions.

The aforementioned works have been included in the following publications:

- Panphattarasap, P and Calway, A, 2017, Vision-based Place Recognition Using Landmark Distribution Descriptors. in: Computer Vision - ACCV 2016: 13th Asian Conference on Computer Vision, ACCV 2016, Revised Selected Papers. Springer-Verlag Berlin, pp. 487-502

- Panphattarasap, P and Calway, A, 2018, Automated Map Reading: Image Based Localisation in 2-D Maps Using Binary Semantic Descriptors. in: Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems. Institute of Electrical and Electronics Engineers (IEEE)

Chapter 2

Vision-based place recognition

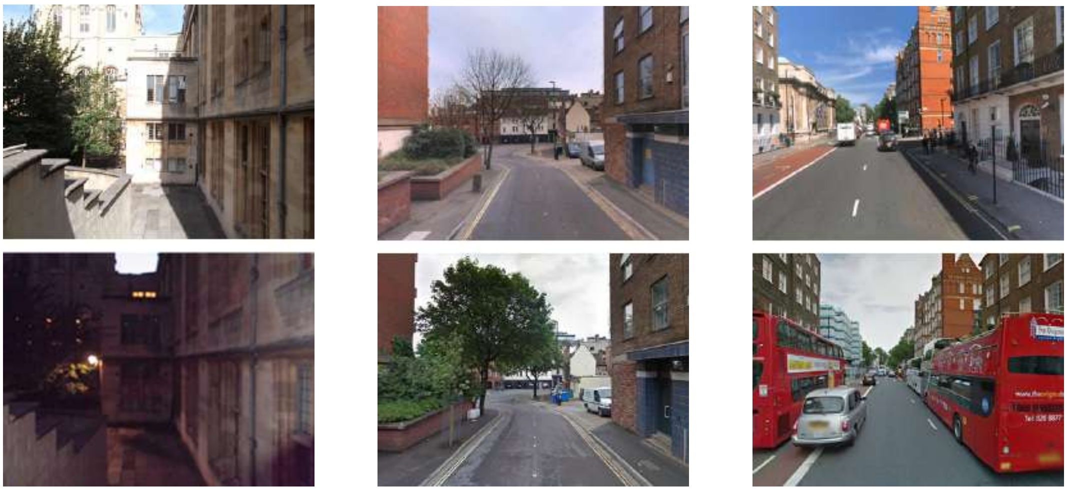
This chapter provides a review of techniques related to place recognition using image-to-image database matching. In Section 2.1, we explain the overall ideas of vision-based place recognition. In Section 2.2, we consider traditional feature-based and learning-based approaches, while in Section 2.3, we discuss the current state and limitations of vision-based place recognition applications. In Section 2.4, we present the integration of semantic information into place recognition systems. In Section 2.5, we examine the use of spatial knowledge in human place recognition. Finally, in Section 2.6, we summarise the reviewed techniques and outline problems we intend to solve.

2.1 Vision-based place recognition using image-to-image matching

The term 'place' generally refers to the knowledge of a location [73]. Recognising a place is an act of identifying a place one has previously visited. Therefore, vision-based place recognition refers to the technique of using received visual information to recognise the revisited location. However, to automatically recognise places purely relying on visual information, rather than using a positioning infrastructure such as GPS, requires more sophisticated methods. The task highly depends on the characteristics of places, the viewing positions and directions, and the environmental conditions, such as light and visibility, as illustrated in Figure 2.1. This further complicates the problem.

2.1.1 Visual information

There are several forms in which the reference images can be stored, such as a single image per view, multiple images per view, or a sequence of images (or video frames) along the finite route. Having more data per location means more chances to obtain the correct



(a) (b) (c)

Figure 2.1: Examples of challenges in recognising places: (a) day/night, (b) changes in vegetation, and (c) the presence of transient objects. We can see that the changes are highly affected by the appearance of images, but we can still see the maintenance of static information using semantic reasoning.

results. Chance, in this case, can be viewed in two aspects. First, at a given location, the database records a set of static images taken at the location, with multiple changes. Based on this, when they perform matching, the system has more chances to retrieve the correct result because there are more reference views, which mean more correct choices. Second, chance can be viewed in the context of recording a location in the form of video sequences. The higher chance results from the higher level of techniques applied. For example, in FAB-MAP [54, 55] , the system applied the probabilistic methods to localise sequentially by tracking the detected visual vocabulary. Therefore, even when the system predicts the wrong location at the earlier state, with more data gathered in the long run, they have more chances to use prior knowledge to recognise the current location. This is also supported by [74], which demonstrated the comparison of image matching using various resolutions of visual data. The results indicate that using longer image sequences provides better performance than using single images or short image sequences. However, using only images does not provide depth information. Another alternative is to integrate 3-D information from other sensors, such as lasers [75] or RGB-D [76, 77], or to use geometric techniques such as stereo vision [78, 79]. Although such techniques might improve the quality of the description, extra information still requires extra space, and this affects the scalability of the system.

In this thesis, we consider using static databases of a single-image database (i.e. one test and one reference). This could reduce the chance of obtaining the correct result; however, it also increases the degree of challenge. In addition, if we can make simpler and harder problems work, there is a chance for better performance using higher-level techniques.

2.1.2 Image-to-image matching processes

Given two images, image features are extracted to construct image representations and compared to indicate the similarity between those features. Figure 2.2 summarises the processes of image-to-image matching. First, each image is represented using interesting points or regions, known as image features. These detected features are then used to construct feature vectors, or image descriptors. In other words, both images are transformed into a set of visual feature information or salient points. Finally, the two image descriptors are compared, most often using the technique of L2 norm or Euclidean distance. The result is a similar score, which indicates the similarity between the two images.

For place recognition applications, as illustrated in Figure 2.3, the given query image is compared with all geo-tagged images stored in the database, and it is common to use the search algorithm to speed up the matching process. The nearest neighbour technique, which is used to compare the whole database one by one, usually applies when the number of references to compare is acceptable. However, in larger-scale applications, more sophisticated methods for data pre-processing and searching are needed. More details regarding the size of the database are discussed in Section 2.3. The geo-tagged image retrieved from the search process is treated as the best candidate, and its geolocation is likely to be treated as the geolocation of the query. This information is further used in other location-based operations, such as path planning and navigation. Note that, in this respect, place recognition can be viewed as a content-based image retrieval (CBIR) problem [41] in the sense that a database is searched to find the most similar images to the query.

2.2 Place representations

Among the processes represented in Figure 2.2, key in vision-based place recognition is finding suitable location-representing features that tolerate environmental changes, as demonstrated in Figure 2.1. In general, the feature extraction approaches fall into two broad categories: traditional approaches and learning-based approaches.

2.2.1 Traditional approaches

For traditional approaches, the central process is to select the most salient features that could differentiate the image from others. In other words, the selected features are used to

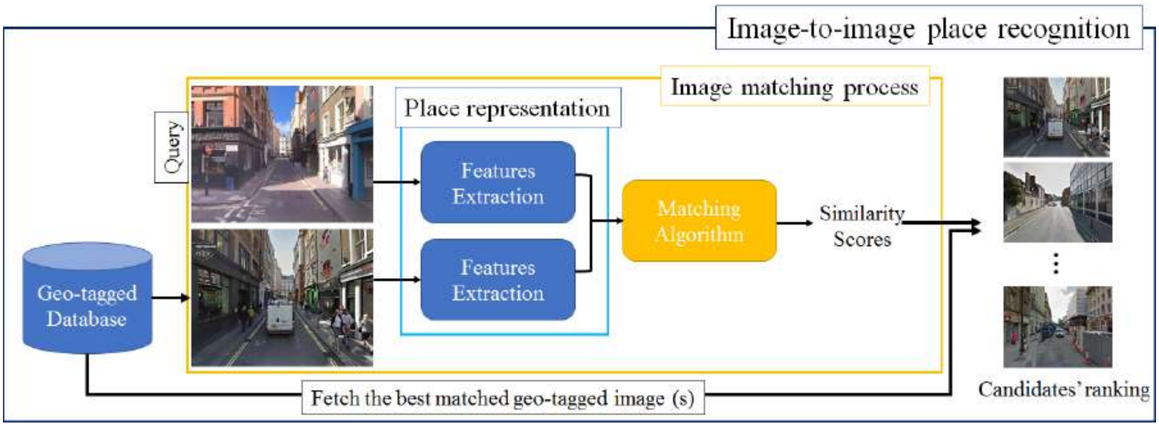


Figure 2.2: The fundamental processes of image-to-image place recognition. The main element is finding the similarity between two images: one from the query and another fetched from the geo-tagged database. To match between two images, each of them is applied the feature extracting methods to create a feature descriptor. The feature descriptors of the two are compared using the matching algorithm. The final product is the similarity score, which represents the similarity between the two.

describe the information that is relevant to the image. Due to the characteristics of feature selection, the traditional approaches are also known as hand-crafted features. Commonly selected features include geometric information, such as corners, edges, blobs, and ridges, and low-level visual information, such as colours, pixel intensity, textures, and contours. The techniques can be divided into two broad categories: local-based and global-based. Figure 2.4 illustrates the concept of each type of approach.

Local-based approaches represent an image by making use of salient points or regions of interest. The techniques dominating the early state are the scale-invariant feature transform (SIFT) [80] and speeded up robust features (SURF) [81]. These techniques are commonly incorporated with a visual dictionary, such as bag of words (BoW) or bag of features (BoF) [82]. The primary process involves treating the repeated patterns of local features and quantising them in the form of visual vocabulary or a visual codebook (dictionary). A common technique for quantisation is to cluster the visual words using an algorithm, such as k-means clustering. Among the systems using local-based features, the probabilistic FABMAP method [54, 55] is the most well-known in vision-based place recognition; the system recorded the scenes using over 100,000 visual words and used them for sequentially tracking the current location. The place recognition process was used for loop closure operation; in other words, the system recognised the revisited places by considering the similarity between the set of stored visual words. Each contained rarity indexing, which boosted the distinctiveness of the scene.

For the global-based approaches, instead of focusing on points of interest, the tech-

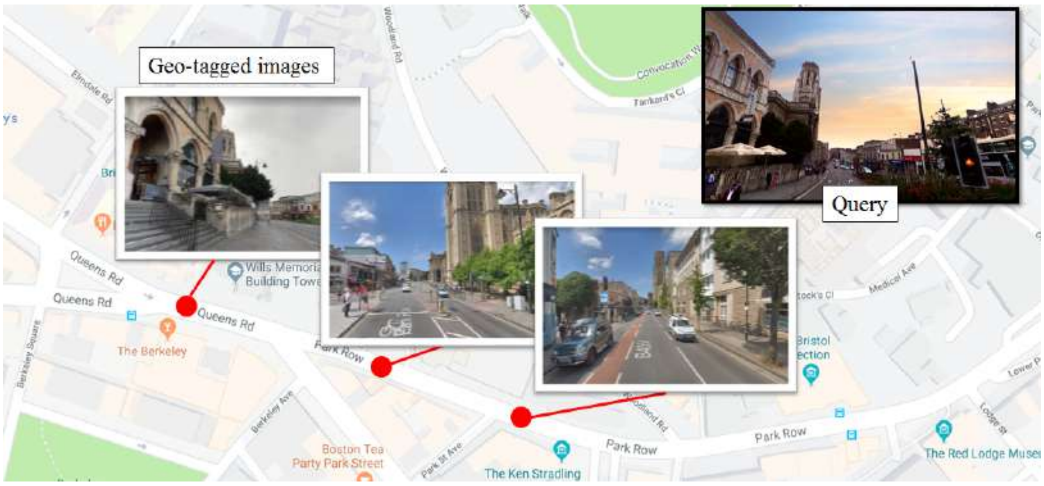


Figure 2.3: The general overview of a place recognition system using the static image-to-image database. Given a query image, the system searches for the best match from the geo-tagged images stored in the database. Once the result is returned, the query image is assumed to be taken at the same geo-location. This information can further be used for localisation.

niques are applied to the whole image. As illustrated in Figure 2.4b, the colour histogram extracted from the whole image is a simple example of a global feature. One of the most well-known global-based techniques is the GIST descriptor [83], which is the low-dimensional holistic scene descriptor using spectral representation spatial envelope features. Compared to local-based techniques, the global-based techniques consume less computational time and space. Several works have applied these types of features for location-based applications, such as a large-scale place matching [84, 85, 86] and localisation and navigation [87, 88, 89, 90].

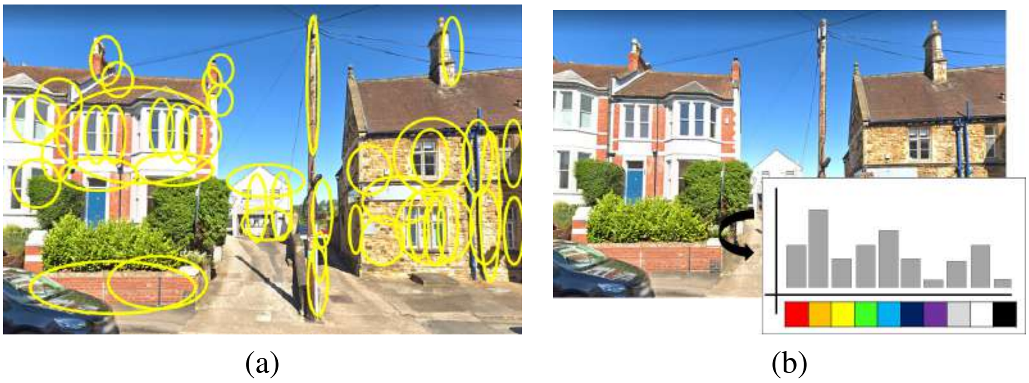


Figure 2.4: Traditional approaches consist of (a) local feature-based descriptors and (b) global feature-based descriptors. The main difference is how each technique is applied to the image. The local-based techniques are only interested in some parts, while the global-based techniques apply equally to the whole image.

It was noted in [91] that both approaches have complementary attributes: local features provide invariance to viewing positions and directions, while global descriptors provide better invariance to changes in viewing conditions; however, neither one can perform both. In response to this, hybrid approaches have been introduced to overcome invariance in both viewing directions and condition problems by making use of both local and global features. The first aspect of hybrid approaches is to use patch-based features combined with global-based techniques [50, 92]. The second aspect is to combine the usage of two or more types of global and local features. For example, a representation is created by combining both local and global features [93], or, in the matching process, GIST is applied first to narrow down the candidates, and SIFT is then applied to find the best match [94]. However, in recent years, these hand-crafted descriptors have been outperformed by the learning-based approaches in both performance and invariance. This is discussed in further detail in the next section.

2.2.2 Learning-based approaches

The key difference between the traditional approaches and the learning-based approaches is that, for the latter, the system makes use of the learned mathematical models for automatically predicting or making a decision without human involvement. The core process of learning is to observe the patterns in data and automatically learn from them. The learning-based approaches generally fall into two categories: unsupervised learning (clustering) and supervised learning (classification). The former does not require any prior knowledge or labelled data; examples include k-mean clustering and the Gaussian mixture model (GMM). Its purpose is for grouping similar data together by using the function to infer the hidden patterns of unlabelled data. The latter works based on the pre-defined image classes and the support vector machines (SVM) is widely used. A group of works [95, 96, 97] have applied SVM to solve the problem of place recognition. These stated techniques were considered state of the art until learning-based approaches using deep learning made substantial progress, especially for image-related problems including recognising places.

Among the deep-learning approaches, the convolutional neural network (CNN) is the most well-known model, especially for image-based problems. This model was introduced in the 1990s; however, its use did not become widespread until the introduction of the *AlexNet* model, which set a new state of the art for object classification over millions of data. This leads to other CNN models [98, 99, 100] which have later been extensively used in many fields, including robotics and computer vision.

To apply CNN to image matching tasks, two common techniques are used: (i) generating similarity functions and (ii) extracting features. The former applies the Siamese model [101] to learn the similarity between fed image pairs, as in [102, 103, 104]. For

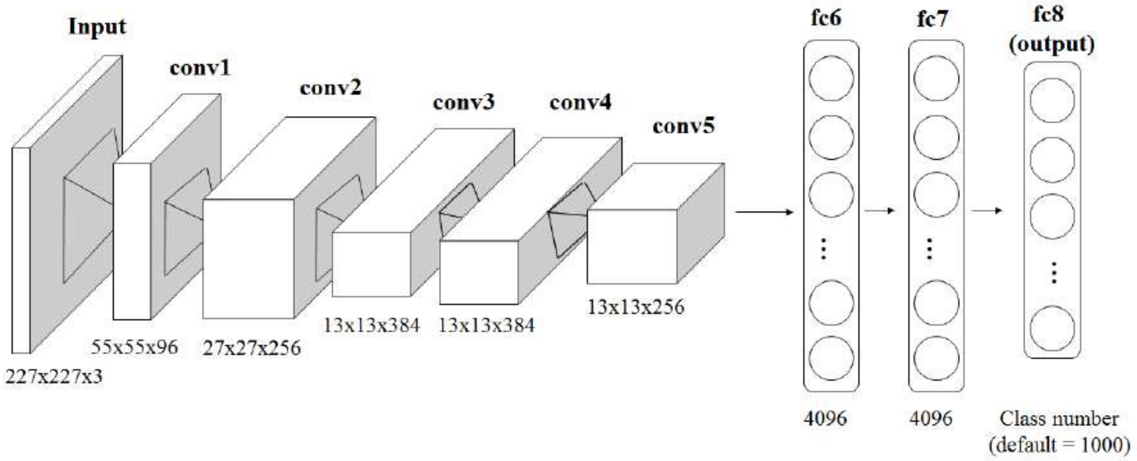


Figure 2.5: Architecture of AlexNet adopted from [11]. The model consists of five convolutional layers and three fully connected layers. Several place recognition applications [12, 13, 14] use the intermediate results retrieved from mid-layers (such as conv3, conv5, and fc6) as the image descriptors.

the latter, the results from CNN layers are used as image descriptors. To provide a better understanding, Figure 2.5 illustrates the structure of the AlexNet model, which consists of five convolutional layers (conv1, conv2, conv3, conv4, and conv5) and three fully connected layers (fc6, fc7, and fc8). Numerous works [12, 13, 14] have demonstrated the use of products from the mid-layers (such as conv3, conv5, and fc6) as the image descriptors. The results indicate that using mid-level features provides benefits for performance. In addition, the approach of CNN as a feature descriptor can be sub-categorised into whole image descriptors [51, 52, 105] and patch-based descriptors [106], depending on how they are applied.

Although these stated techniques demonstrate improvement for image matching, CNN still contains major drawbacks, including the high computational cost and massive amounts of labelled data the network requires for training. Additionally, the structures of networks are still unclear. In particular, the state-of-the-art AlexNet model [11] (the architecture illustrated in Figure 2.5), consists of 600 million parameters and 650,000 neurons. Hence, it is difficult to interpret their learning due to this amount of data. This problem may require further investigation. However, CNN models are still widely used in all areas, including vision-based place recognition.

2.3 Challenges in vision-based place recognition

The previous section reviewed the general techniques of image-based feature extraction for representing a place. However, as in [56, 57], the tendency of vision-based applications has shifted towards the 'lifelong' approach. The key characteristics of lifelong systems include their ability to be used any time, at any location, and in any conditions. In other words, for the long run, the system should be able to operate at any time of day, ideally 24/7. Based on the current state of research, under the fixed-scale area, repetitive visiting, and static environment, it is possible for feature-based systems to operate endlessly. However, the stated situation is not practical in the real world, which is full of changes, as illustrated in Figure 2.1. As reported in [56, 57, 58, 61], the feature-based approaches do not cope with this issue. For instance, in [58], they demonstrated failure in using the FAB-MAP system at night (after 7 pm). Based on this, the challenges in vision-based place recognition can be characterised as invariance and scalability. More details on the impacts of these issues are discussed in the next sections.

2.3.1 Invariance

Invariance refers to the property of remaining unaffected by any variation. For vision-based place recognition, the variation comes from the use of visual information as the main source. Figure 2.1 illustrates examples of data variation, including temporal changes such as seasons, time of day, and lighting directions, and spatial changes such as viewpoints and occlusions. To deal with changes, some researches proposed improvements to the state-of-the-art feature extraction techniques [81, 107, 108] or made use of custom feature-based techniques, such as in [56, 58, 59, 109, 110]. These techniques attempted to present robust place representations that tolerate environmental changes. The key is to identify the static or semi-static points of interest, as those are hardly affected by the spatial and temporal conditions. For example, in [59], given an image taken at the same location but in the different seasons, researchers represented the scene using the static features within that scene. This technique is more similar to how humans recognise a scene. Humans do not rely on non-static information such as trees, because their appearance could be affected by seasonal changes. Instead, they recognise more static elements such as buildings, signs, or posts.

The notable systems challenging for invariance is seqSLAM [111], which is a vision-based place recognition for a day and night dataset using patch-normalised sequences of images. The seqSLAM system was evaluated over the Nordland dataset [112], which included sets of routes over 750 km recorded across four seasons. The findings demonstrate the performance of seqSLAM, yet they also show that the system is poorly conditioned if viewpoint changes are too drastic. This problem was later extended to studies by [113, 114, 115], which showed the improvement of seqSLAM. However, these techniques have been outperformed by the introduction of deep-learning models. A group of works [13, 62, 115, 116] faced the data variation by using CNN as a place representation. One of the notable systems [52] introduced an architecture called NetVLAD. This network is specifically trained for place recognition using images obtained from Google Street View Time Machine, which represent images taking at the same place over time. With this, and subsequent works in [117, 118, 119], the place recognition using NetVLAD shows great potential for vision-based place recognition, especially in terms of dealing with changes in time of day and viewpoints. Although these stated methods have demonstrated signs of progress in terms of invariance, they still relied on the lower-level concept of semantic features, which are still not robust enough to deal with the problem.

2.3.2 Scalability

Scalability refers to the capability of the system to handle expansion. In this case, it implies that when the systems explore larger areas and the image database increases in size, they should be able to maintain their performance, especially in terms of accuracy and speed. The recent state of autonomous navigation shows high potential over a large-scale building. However, when the exploration expands to the city or country level, the autonomous systems are limited by their internal resources. For vision-based applications, which require significant storage, the scalability of the system is of great concern. For example, as reported in [55], to store a single 1000 km route, they required 177 GB of storage. To put this into perspective, in 2016, the explorable roads in England recorded in [120] were 188500 km in length. If the goal were to have a system that covers all roads in England, storage of at least 33 TB space would be needed.

Using the tree structure to reduce the searching time, the average processing time reported in the same paper was around 480 ms to update their 100,000 visual words over the 1000 km routes. Applying this to the England case, the processing time in the average case would be around 90 seconds for the update. However, these assumptions are based on the settings stated in the paper. In practice, having the expansion of the database, the time to search might not grow linearly. Moreover, the 100,000 visual words might not be sufficient to describe the whole England scenes. As a result, increasing amount of vocabulary affects both storage size and processing time. These examples reflect the importance of scalability for image-to-image database matching.

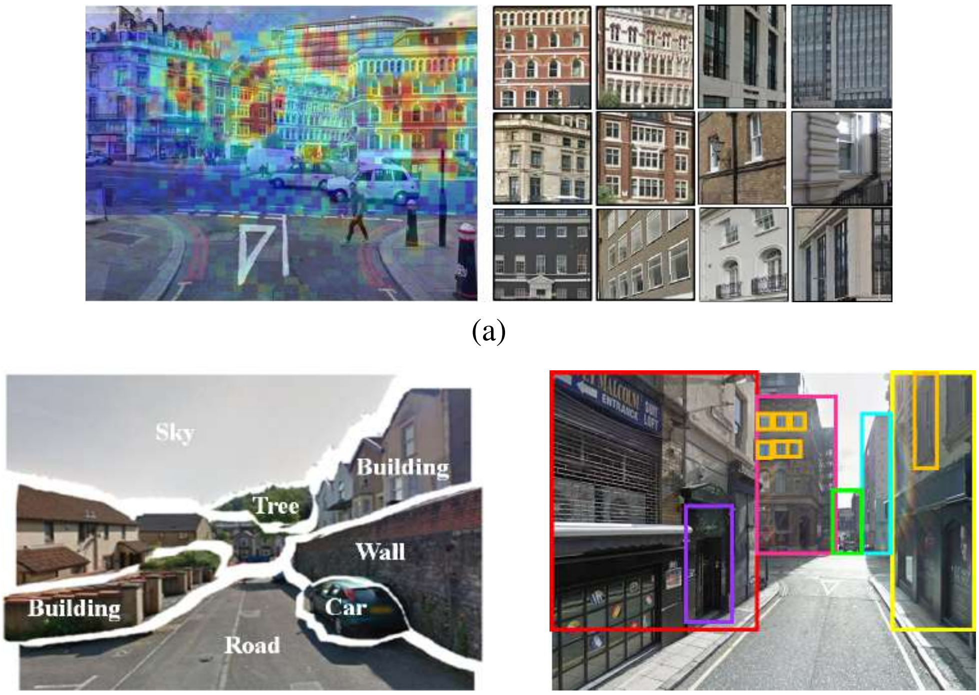
To overcome the data size issue at the hardware level, the techniques for complexity reduction include sparsification, parallel computation, and work distribution [121]. At the algorithm level, the preprocessing algorithm is the dimensional reduction [122, 123], the visual dictionary [82, 124], and indexing techniques [125]. The aim of using these preprocessing techniques is to compress the data so it would consume less storage space. However, even with these techniques, the growth of data is still the main problem for image-based applications.

2.4 Vision-based place recognition using semantic information

In response to the problems of invariance and scalability, recent studies of vision-based applications have moved toward semantic reasoning approaches. The term 'semantic' in computer vision can be interpreted in different degrees, from a low level, such as texture and colour, to a higher level, such as a physical address, types of buildings, and historical eras. Compared to traditional features, semantic information has the potential to cope better with changes in the environment and is likely to be more impact. For instance, Figure 2.1 presents examples of environmental changes in a scene, including (i) day and night, (ii) seasons, and (iii) transient objects. Even with these changes, the semantic information of the scene is the same; given images of a building taken at daytime and nighttime, they still depict the same building. Based on this property, we can handle the invariance problem. For scalability, we rely on the compact nature of the semantic descriptor.

Another example is displayed in Figure 2.1a. One might describe each scene as 'a building', 'a tree', 'at noon' (top row), or 'at night' (bottom row), and most humans would understand these descriptions. This contrasts with the way feature-based descriptors represent a scene as a set of salient points. Given a situation in which the scene is recorded during daytime and nighttime, the detected image features of that scene might change to the degree that the system needs to record a new set of visual words. However, in the same situation, humans can still describe the scene using the same term – 'a building' – with no additional description needed. We believe that this property could support the scalability issue.

For autonomous localisation, a number of techniques have increased the semantic senses in feature-based methods by integrating the visual dictionary or the statistical models, as in [82, 124, 126]. However, these visual words in the human perception do not contain any linguistic information. To recognise a place, humans rely on the visual cues retrieved from their surroundings. The term 'landmark' refers to the visual components of a scene that influence localisation and navigation [127]. Based on this, several autonomous navigation systems have moved towards the higher concept of semantic information. Figure 2.6 illustrates the common semantic-related techniques used for location-based applications. In computer vision, semantic approaches are generally characterised as segmentation or classification.



(a)

(b)

(c)

Figure 2.6: Examples of methods to extract semantic information from a scene. The top row presents (a) a visual pattern using mid-level visual representation. The bottom row shows (b) the concept of scene segmentation, in which components in the scene are segmented and labelled, and (c) the concept of object/landmark detection within a scene; all boxes represent the detected landmark regions.

For segmentation, the common process of retrieving semantic information is to segment images and label each segmented group using semantic labels such as buildings, trees, and cars (Figure 2.6b). The state-of-the-art techniques for segmentation include Markov Random Field (MRF) [128] and superpixels [65, 66], which are used to detect a group of connected pixels with similar intensity (either colour or grey level). Each of them is further processed by feeding to the learned model to obtain a semantic label. Recently, both segmentation and semantic labelling processes have been replaced by CNN models, as in [129, 130, 131].

An alternative to segmentation is to extract the architecture or the style of buildings, which contains a higher level of semantic information (Figure 2.6a), such as building façade segmentation, which is used to detect the surface of each building [46, 132, 133]. A number of studies [134, 135] have applied a method of mid-level visual representation trained using SVM, which contained a higher sense of semantic reasoning than visual words, but lower than actual semantic objects. The others [136, 137] proposed methods using the learning-based techniques to train the systems to distinguish the characteristics of patches taken inside and outside the cities. The final products of these techniques are typically applied to geo-localisation problems, such as city classification [137, 138, 139], scene understanding [140], and cross-domain/cross-view matching [92, 141].

The task of classification is to detect linguistic objects or salient landmarks appearing in the scene, such as cars, trees, signposts, and buildings (Figure 2.6c). Common techniques include salient region detection [142, 143], geometric detection [16, 45, 144, 145], and text detection [146, 147]. Similar to the segmentation approach, the classification technique can also be replaced by CNN's region detection models [148, 149]. In practice, classification approaches are usually combined with segmentation to classify segmented pixels.

Figure 2.7 summarises the process of semantic extraction from visual information for navigational purposes, as in [15]. The systems make use of the 'semantic map', which is the coarse semantic segmentation of the scene using scene segmentation combined with the previously stated classification techniques. In the earlier state, the system only performed feature extraction and quantised them into a set of visual words. For example, in [43], they segmented the image query and quantised them using visual words. These words are matched with a list of visual words stored in the system as a semantic map. However, even these techniques have introduced the use of the higher level of semantic features; in terms of the systems, they only rely on the coloured representation (groups of labelled pixels), rather than the real linguistic semantic context.

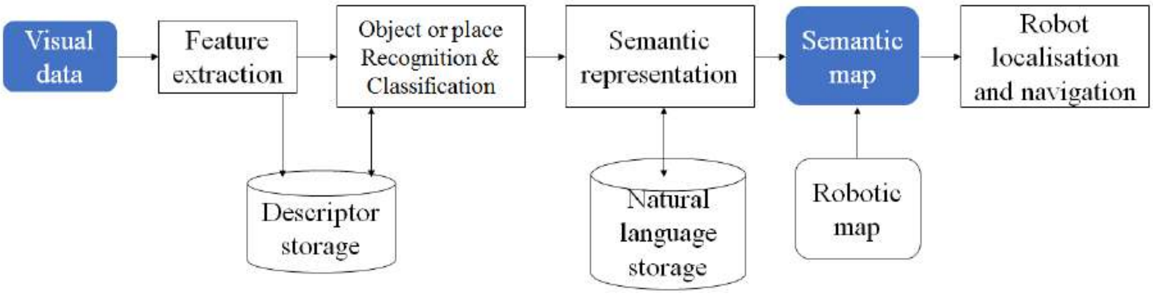


Figure 2.7: The concept of semantic extraction from visual information for localisation and navigation systems derived from [15]. Note that the robotic map depicted in this figure refers to metric and topological maps, which are the general maps used in robotics. More details about these maps are presented in Chapter 4.

Therefore, to integrate semantic information into the system, the extracted features are labelled using natural language descriptors, such as buildings and road segments. The standard method is to assign salient objects to represent the place, such as regions of interest [62], road lanes [63, 64], traffic lights [44], and text [150, 151]. These representations are further combined with the robotic map to create the semantic map. However, for humans, knowing only salient objects may not be enough to recognise the place. The locality of objects is also important. Thus, we address the integration of relative positions between the objects in the scene. To provide a better understanding, Figure 2.8 illustrates the importance of spatial knowledge. Figure 2.8a contains only objects. In Figures 2.8b, 2.8c, and 2.8d, road segments are added to shape the spatial relation between objects. Therefore, if we compare Figures 2.8b, 2.8c, and 2.8d, which contain the same objects, with the spatial relationship provided, we differentiate Figure 2.8c from Figure 2.8b and Figure 2.8d. Given this information, we believe that the spatial arrangement of landmarks is vital to semantic-based place recognition. This is further discussed in the next section.

2.5 Spatial awareness in the human place recognition

For humans to recognise a place, they rely on their mental representations for spatial knowledge, known as the cognitive 'map' [67, 68, 152], which relates to the field of automated reasoning. Spatial knowledge refers to the acquisition of the orientation in a large-scale space. A review in cognitive studies [69] concluded that humans recognise a place by relying on their spatial knowledge, which consists of (i) local landmarks, (ii) spatial relationships among the local landmarks, and (iii) how local landmarks are oriented in relation the observer. This suggests the integration of semantic information in human place recognition.

To gain a better understanding of human abilities, experiments were conducted in which participants were prompted to draw a directional map that led to a certain location [153]. The continuity of the studies indicates that the varying representations in the drawings did not affect the map interpretation ability of the participants. For example, one person might project the building as a square, while another one draws it in a circular shape. Based on its locality (where it is placed) in the scene, both the square and the circle would be successfully interpreted as a building. Another study based on the human sketch map was detailed in [72]. They asked the participants, who regularly visited a place, to draw it on a map. The researchers observed a loss of information on the sketched map compared to the real-world data; however, the experiments revealed that the drawn maps contained cognitively adequate, accurate, and reliable spatial information with 90% accuracy, including landmarks,

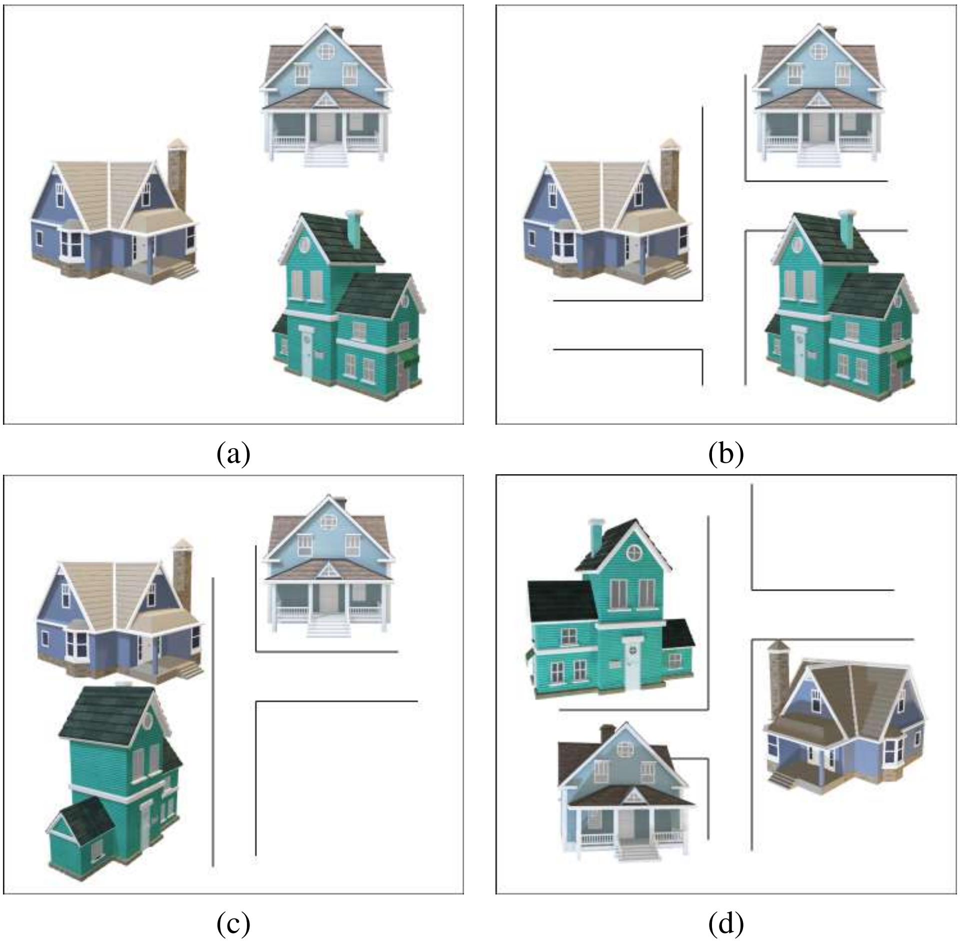


Figure 2.8: The use of spatial knowledge combined with the salient objects in the scene. Although the landmarks appearing in the scene are the same, the arrangement in (c) differs from (b), which indicates that it is a different location. If we observe carefully, the arrangement of landmarks in (d) is similar to (b) in the reverse direction. We can infer that spatial knowledge is important in the image-to-image matching task.

road segments, and junctions. The findings imply that humans have some ability to project their knowledge in the form of landmarks and spatial arrangement.

In the computer vision and robotics field, the use of the spatial distribution is conceptually applied in many research areas. For example, for feature-based descriptors like SIFT, the descriptor itself also consists of the orientation between detected keypoints' features. This concept of spatial integration has been demonstrated in several studies to improve the traditional feature techniques, such as in [154, 155, 156]; these stated methods integrated the spatial contextual information into BoW to strengthen the matching process. Another scenario is the way in which autonomous systems store their surroundings as a map; the typical forms of the robotic map are topological based and metric based (more details in Chapter 4), which include some degree of spatial knowledge.

Regarding robot navigation, [157] demonstrated the use of spatial knowledge (i) to make the robot learn to generate the cognitive map and (ii) to train the robot to make a decision by finding shortcuts and establishing new routes. Both applications are based on the landmark, route and survey (LRS) theory [21], which refers to the cognitive map construction of humans using landmarks and their spatial relationships (more details about LRS theory are discussed in Chapter 4). In addition, others [158, 159, 160] used this concept to create a map that contained abstraction between the level of the sketch map (human cognitive representation) and topographic map (spatial representation) for robot navigation; this map was shared between the robot and the human instructor. Conceptually, the robot received human instructions in natural language and performed the navigation task based on that command. To do that, the robot needed to understand the human context, including the spatial representation of the scene. These provisional experiments indicate some improvement in robot navigation using spatial knowledge. However, studies of the full use of spatial context and the impact on performing location-based activities still receive limited attention from the community.

2.6 Discussion

This chapter has presented several image-to-image-based place recognition systems, starting from traditional techniques using local-based and global-based features, then moving to the learning-based approach. Still, gaps and open questions remain. We chose semantic-based approaches because they are better for invariance, more compact, and more similar to human perception. For this, we propose a method of place recognition that enforces the spatial relations among salient objects.

We consider that applying spatial knowledge to the system impacts the performance of recognising the place. This is similar to [58, 59], which worked on environmental changes by identifying the static features appearing in the scene across seasons. In this respect, the spatial distribution is also static information that is highly resistant to any changes. For example, as demonstrated in Figure 2.1b, even across the seasons, the distribution of salient objects such as buildings, trees, and road junctions is still the same. This is also true for the other examples in Figure 2.1. Based on this, we propose a method that contains the following mechanisms:

(i) Detect salient landmark regions
(ii) Create landmarks feature vectors
(iii) Match landmark regions

(iv) Maintain consistency of spatial distribution of landmarks

Although we aim to observe the impact of spatial knowledge, selecting salient objects is also important. To do that, we seek a method to detect the regions of interest within a scene and convert each salient region into some form of representation; the place representation needs to be invariant to changes in the scene. Based on these conditions, we adopt the method of Sünderhauf et al. [62], who demonstrated improved vision-based place recognition using proposal regions from *Edge Boxes* [16] combined with vector features from CNNs [11] to match regions between scenes. By using Edge Boxes, the system is free from training a network for a specific environment. They evaluated the method on an urban environment using road sequences in Berlin, Germany. The results revealed some improvement over the earlier state-of-the-art techniques, SMART [114] and FAB-MAP [54], especially in terms of invariance. However, their process of region matching did not include any spatial knowledge; we later explain this limitation.

We then extend the approach by encoding the spatial distribution of the regions within a scene. The number of scene understanding applications is based on the relationships between object alignment or their hierarchical connections. As noted in [41], in CBIR problems, several proposals store the spatial location of each region of interest and use it for searching. An example of using this for the scene-related problem is a model called visual Memex (Memory and Index), which records objects and their spatial relationships [161]. The model is used to predict the lost object within a scene. For example, given a hidden object, such as a car, the Memex model makes an assumption based on the following characteristics: the object connects to a road, nearby is another car, and trees and windows are in a higher position. These spatial relationships are used to predict the missing object to be a car. This demonstrates the importance of spatial distribution within a scene.

For place recognition, the work in [162] presented scene matching using the histogram of forces (F-histogram) as a scene representation and created the object map; this work relates to our interest, as, for matching, they enforced the use of the spatial relationship between all object pairs in the scene. By using this representation, they coped well with the changes in viewpoints between scenes, but they were limited by the number of objects in the scene. If there are several objects, the computation costs can be loaded for real-world applications. Similar to this, the related work presented the semantic-based matching between scenes, which were widely disparate by making use of the relative positions between the detected traffic lights in each scene [163]; this method demonstrated the potential of recognising a place, even with the drastic viewpoint changes. Motivated by these studies, we adopt the same concepts by storing the order of landmarks. More details on this are provided in Chapter 3.

Chapter 3

Place recognition using landmark distribution descriptors

In the previous chapter, we reviewed several works on vision-based place recognition. The key element behind the task is to find a suitable representation for the view. The common approaches of feature extraction fall into two main categories: matching local features between views (local descriptors) and comparing whole-image characteristics (global descriptors). However, as highlighted in [91], none of these features are simultaneously invariant to viewing positions, directions, and changes in viewing conditions. As the problem we focus on is finding the match between different views of the same location taken at a different time, we sought the alternative.

In this work, we focus on single images for matching. Using a one-to-one pair of the test and reference views may lead to inaccuracy compared to the use of image sequences, but the latter would make the problem more challenging. In addition, there is a chance that a suitable algorithm for a single-image problem, can later be extended to work with sequences. Therefore, we demonstrate that linking spatial knowledge to the detected landmarks can improve the performance of single-image place recognition with no additional information provided. Figure 3.1 illustrates an example of image matching using the feature-based technique (top row) and the salient regions technique (bottom row). Upon careful observation, the matching features are not well aligned compared to the salient landmark regions that visibly maintain their spatial arrangement. However, matching between scenes is not a straightforward task. As depicted in Figure 3.2, similar landmarks appear in different places; though the individual landmark may match, their relative positions might be incorrect. Therefore, along with feature vector matching, the system should maintain the consistency of the spatial pattern of landmarks between views of a place.

To achieve this, we introduce a method for place representation called the *landmark distribution descriptor* (LDD), which consists of the horizontal stack of landmark feature

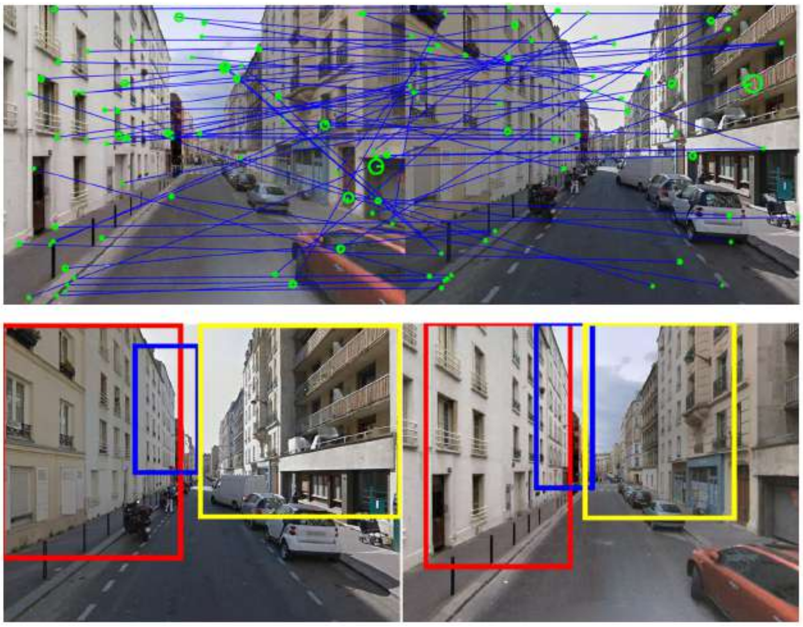


Figure 3.1: An example of an image pair captured from the same location but with a slight displacement. The top row displays matching using the features method, and the bottom row depicts matching using salient landmarks. The matched features are not well aligned compared to the salient landmark pairs.

vectors. In Section 3.1, we summarise the system overview of the LDD, which consists of the processes of Section 3.1.1 – landmark selection and construction of LDD, and Section 3.1.2 – a comparison between two LDDs, with spatial knowledge enforced. In Section 3.2, we compare our described method with the selection of image-to-image-based methods. Finally, in Section 3.3, we summarise the overall results, which lead to future research possibilities, discussed further in Chapter 7. Note that the arguments, figures, and results described in Chapter 3 have been published in [164] and, in this part of work, we use the terms 'landmark' and 'salient regions/objects' interchangeably, which differs slightly from how humans linguistically perceive actual landmarks.

3.1 Vision-based place recognition using landmark distribution descriptors

The general idea of the image-to-image matching processes is summarised in Figure 2.2. We designed our descriptor following this concept, with an additional mechanism for spatial integration, as illustrated in Figure 3.3. Based on this, in this section, we describe the processes of (i) constructing the landmark distribution descriptor (LDD) (Figure 3.3a-c) and (ii) comparing between two LDDs (Figure 3.3d). There are three steps to constructing

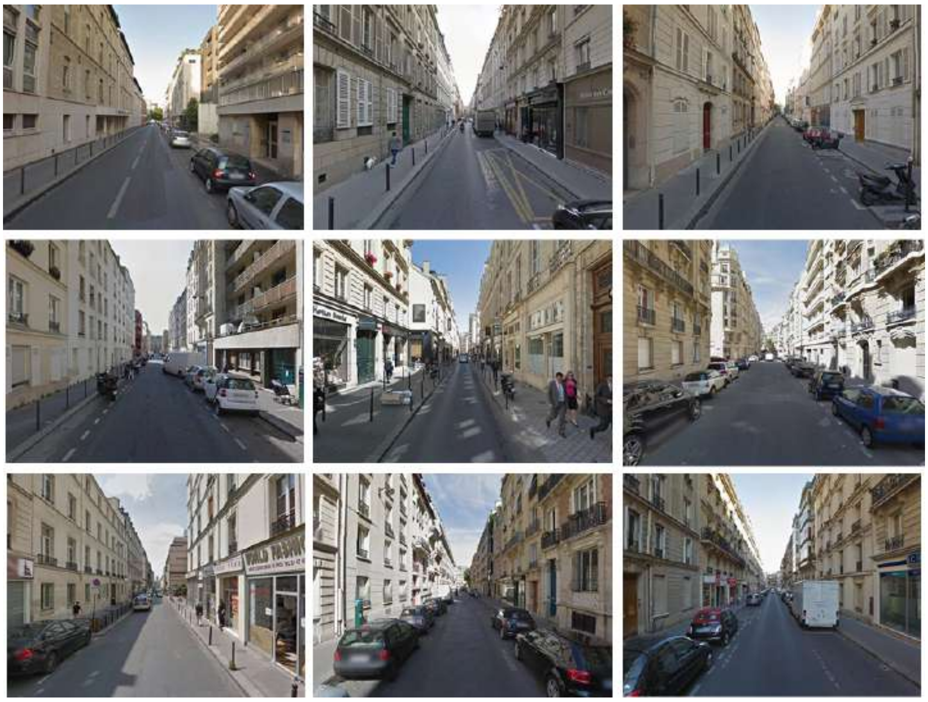


Figure 3.2: Examples of views captured in the same cities but positioned differently. The similarity in architecture is clear. This makes our problem more challenging.

an LDD. The first is to select a set of landmarks or salient regions (Figure 3.3a). The second is to identify the connection between each landmark to create a spatial relationship (Figure 3.3b). Finally, the linked landmarks are stacked to create an LDD (Figure 3.3c). This is discussed in greater detail in the following section.

3.1.1 Constructing landmark distribution descriptors

In this section, we discuss the processes of constructing LDDs: landmark selection and spatial integration and landmark feature vector construction. The processes are as follows.

Landmark selection and spatial integration

An urban scene typically contains various objects, as depicted in Figure 3.4. Thus, selecting a set of objects that represent a scene is a challenging task. Vision-based applications apply a variety of objects, such as traffic lights [44, 163] and road lanes [165]. We can relate this problem to object detection and visual saliency detection [166]. In general, the techniques can be grouped as non-learning based and learning based. For non-learning-based approaches, the standard method is proposal detection [167], such as in [16, 45, 145]. For learning-based approaches, the general technique is to train systems to detect the presence of instance objects, such as cars, trees, and buildings, as reviewed in [168]. In

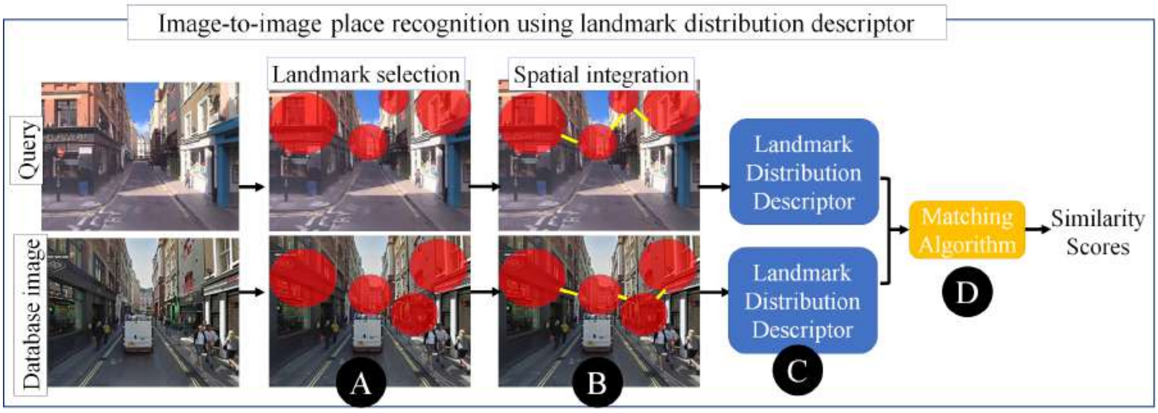


Image-to-image place recognition using landmark distribution descriptor
Landmark selection
Spatial integration
Query
Database image
Landmark Distribution Descriptor
Landmark Distribution Descriptor
Matching Algorithm
Similarity Scores
A
B
C
D

Figure 3.3: The concept of image-to-image-based place recognition using landmark distribution descriptors. The key steps are (a–c) to construct the descriptor by detecting salient landmarks and integrating the spatial relationship between detected landmarks, and (d) to compare between two descriptors to obtain the similarity score, which indicates how close they are.

recent years, a common technique is to apply CNN models for region detection, such as R-CNN [148] and Faster R-CNN [149].

Initially, we chose the simple non-learning method for proposal detection to reduce the complexity of our system. Among the stated techniques, we used Edge Boxes [16], a tool for object recognition, which later demonstrated effectiveness for our application. The process of Edge Boxes is based on the observation of the likelihood that something is an object of the contours contained within a bounding box. Those boxes form a set of proposal regions suitable for further processing. Similar to [16, 62], we made use of the Edge Boxes ranking to limit the number of landmark proposals and to speed up our computational time.

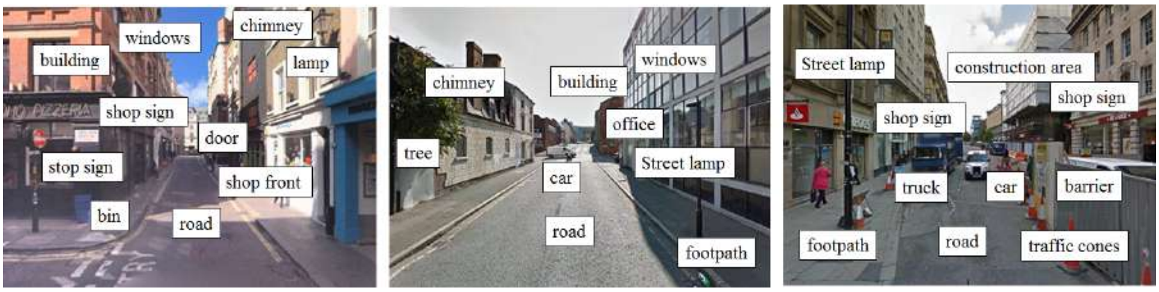


Figure 3.4: Examples of semantic objects appear in the scene; how can we detect all possible objects? For this, we sought techniques that provided proposal regions with those objects in the scene that are likely to be salient.

The next step was to integrate the detected regions with spatial knowledge. To construct the ordering relationship between salient regions, a number of works have used graph-based techniques to identify the connectivity or hierarchical structure between landmarks, such as

[161, 162]. However, for the same purpose as with proposal detection, we chose a simpler operation to specify the relationship of each salient region. To do that, we vertically partitioned the image and selected a fixed number of the highest-ranking landmark proposals in each section. We call these *panoramic sections*. In addition, we allowed partial overlapping between sections so that individual landmarks could belong to more than one adjacent section. This is important when matching landmarks, as we avoid discarding any boundary straddle proposals.

In general, we denote $L = \{l_1, l_2, \ldots, l_N\}$ as the set of landmark proposals in an image detected using the Edge Boxes algorithm. We selected a subset of landmarks $\hat{L}$ such that $\hat{L} \subset L$ and

$$\hat{L} = \bigcup_{s=1}^{S} \hat{L}_s \tag{3.1}$$

where $\hat{L}_s$ is a subset of top-ranking proposals in a panoramic section s, and S is the number of panoramic sections.

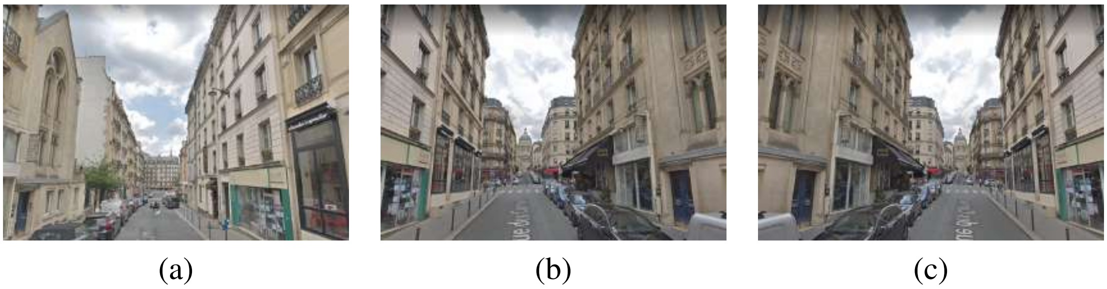

(a) (b) (c)

Figure 3.5: Examples of the reverse direction situation – set (a) and (b) is an image pair captured from different directions, but each of them points to the same location. In (c), we show the horizontal flip of (b). There are some landmarks missing, and some new landmarks appear, which makes the problem more challenging.

By using the panoramic sections, we gain spatial knowledge of the scene; however, there are situations that our method cannot cover, such as the reverse direction. A related work proposed a technique using regions of interest similar to ours, but they handled a reverse scene situation using the deep-learning method to link the salient regions [169]. However, this solution can partially solve the problem. An example of a more difficult situation is shown in Figure 3.5. The appearance of a scene in Figure 3.5b and Figure 3.5c (the flipped version) seems to differ from Figure 3.5a, especially in the middle areas. This is further discussed in Chapter 7.

Landmark feature vectors

Given a view, we constructed feature vectors for all of the landmark regions in the selected subset of proposals $\hat{L}$. The vectors corresponding to the section subsets $\hat{L}_s$ then formed the LDD for the view. To do that, we constructed feature vectors using CNN models. Such models are typically used in image matching (i) to generate similarity functions between image pairs [102, 103, 104] and (ii) to extract feature descriptors [12, 13, 14]. Followed [62], we used an off-the-shelf pre-trained AlexNet network [11] provided by MatConvNet [170] and extracted the feature vector from the third convolutional layer (*conv3*). To match the required network input size, each landmark region was resized to 227×227 pixels. We used the results from the conv3 layer to produce feature vectors of dimensions $13 \times 13 \times 384 = 64,896$. However, this number is quite large. To reduce the size of the feature vectors, there are several techniques for dimensional reduction, such as in [122, 123]. We followed [62] to project each vector onto a lower-dimensional space using a Gaussian distribution matrix. This is a simple but effective method for dimensional reduction, as feature vectors are projected onto a significantly smaller number of orthogonal random vectors. For the experiments, we applied the integer-based random projection matrix Gaussian random projection (GRP) stated in [171] to dimensionally reduce our feature vectors.

3.1.2 Comparison of landmark distribution descriptors

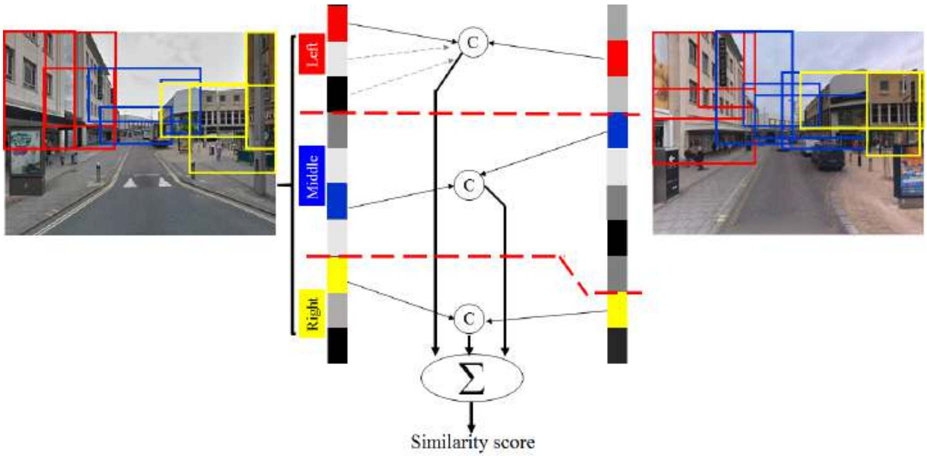


Figure 3.6: Concepts of LDD comparison using three panoramic sections ($S = 3$). We only compared proposal regions within the same panoramic section, which is the final step in enforcing the spatial distribution of landmarks in the scene.

Given two images, to perform image-to-image matching, we convert them into LDDs and indicate the similarity between the two feature vectors. There are several methods for measuring this. The most widely used is the L2-norm or Euclidean distance. However, we found that computing their cosine similarity yields some improvement over using a straight

Euclidean distance. Using cosine similarity to match between two CNN features has been reported in several works, such as [62, 172, 173, 174].

Therefore, in each of the corresponding panoramic sections, we determined the best matching feature vectors (the maximum cosine similarity) of each section and summed them up to reveal the similarity score between a pair, as illustrated in Figure 3.6. Note that the provisional experiment showed that it is more efficient to use only one best pair per section. In general, given two descriptors, LDD1 and LDD2, containing a set of landmarks

$$\{\hat{L}_1^k, \hat{L}_2^k, \ldots, \hat{L}_S^k\} \tag{3.2}$$

for $k = 1$ and 2, we look for the set of S pairs $(\hat{l}_i^1, \hat{l}_j^2)^s$, $1 \leq s \leq S$, such that

$$(\hat{l}_i^1, \hat{l}_j^2)^s = \underset{l_i^1 \in \hat{L}_s^1, l_j^2 \in \hat{L}_s^2}{\arg\max}\ c(\mathbf{v}_i^1, \mathbf{v}_j^2) \tag{3.3}$$

where $\mathbf{v}_i^1$ and $\mathbf{v}_j^2$ are the feature vectors associated with landmarks l_i^1 and l_j^2, respectively, and $c(\mathbf{u}, \mathbf{v}) = \mathbf{u}.\mathbf{v}/||\mathbf{u}||||\mathbf{v}||$ denotes the cosine similarity between two vectors $\mathbf{u}$ and $\mathbf{v}$, where '.' denotes the dot product and $||\mathbf{u}||$ is the length of $\mathbf{u}$.

To avoid duplicating matching landmarks, we set a constraint that no landmark is allowed to be in more than one matching pair. The overall similarity score between the two LDDs (the two views) is then given by the sum of the S cosine similarities:

$$sim_{12} = \sum_{\substack{(\hat{l}_i^1, \hat{l}_j^2)^s \\ 1 \leq s \leq S}} c(\hat{\mathbf{v}}_i^1, \hat{\mathbf{v}}_j^2) \tag{3.4}$$

The process of strictly matching each panoramic section is our final measure to enforce spatial knowledge. This is to ensure that the compared landmarks are aligned in the same order.

For place recognition using LDDs, we applied this matching process to test and reference views. Given an image, we searched for LDDs within the database that were closest to the LDD of the test view. Note that, because numbers of test data were acceptable, we used the nearest neighbour technique to search the database. In other words, the LDD of the test view was compared to the LDDs of all reference views in the database. However, in the long-run, the search strategy might need improvement when the size of the dataset increases.

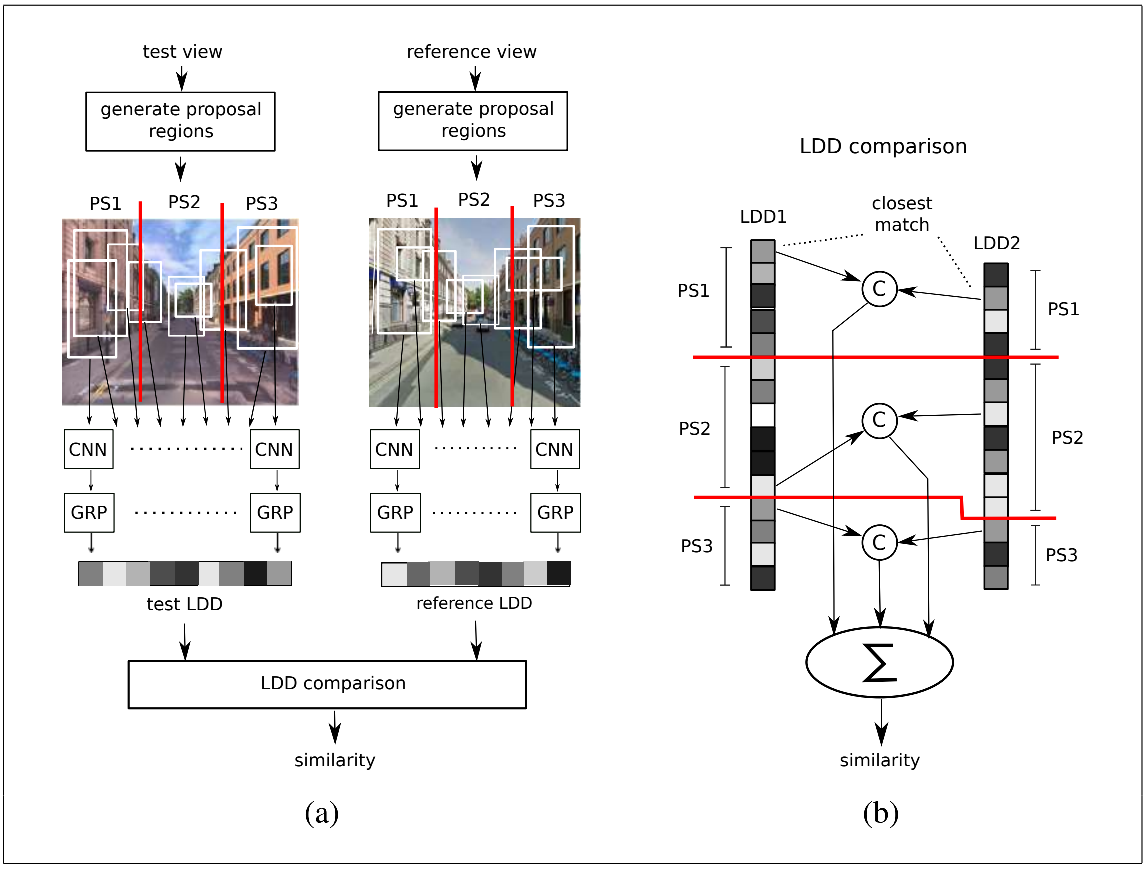


Figure 3.7: Construction and comparison of landmark distribution descriptors (LDDs): (a) landmark proposals are generated for the test and reference image using Edge Boxes [16] distributed within panoramic sections *(PS1-PS3)*. Landmark feature vectors derived from a CNN [11] followed by GRP [17] are then horizontally stacked in spatial order to form an LDD for each image, and (b) descriptors LDD1 and LDD2 are compared by identifying closest landmark feature vectors within each panoramic section and summing the (cosine similarity) distances between them to derive a similarity score.

3.1.3 Summary of the system

In sum, we have introduced a novel method of using proposal regions coupled with feature vectors from CNNs to match landmarks between views. By extending the approach of Sünderhauf et al. [62], we propose descriptors created from landmark features that encode the spatial distribution of the landmarks within a view. Descriptors are matched while enforcing the consistency of the relative positions of landmarks between views.

Figure 3.7 illustrates the overall concept of image-to-image based place recognition using LDDs. The technique of Edge Boxes is applied for landmark region detection. To maintain the spatial distribution between the landmark, the scene is vertically separated into panoramic sections. In this example, the numbers of the panoramic section have been set to three (as shown in Figure 3.7, PS1–PS3). The spatial-encoded landmarks within each section are converted into feature vectors using the CNN features, specifically AlexNet [11]. To make the descriptor more compact, GRP [17] is applied for dimensional reduction. The product from these processes is the LDD, as illustrated in Figure 3.7a.

For the matching process, we measure the similarity of two LDDs using cosine distances. Only landmark features from the same panoramic section can be compared. In particular, given two LDDs from image A and image B, the landmark features within PS1 of image A can compare only with landmark features within PS1 of image B, and cannot match across its section to PS2 and PS3 of image B. The same rule applies to other panoramic sections. This is to ensure the spatial distribution within the scene is maintained. The maximum cosine scores of each section are summed up to get the final similarity score to indicate the similarity between two LDDs.

3.2 Experiments and results

In this section, we evaluate the method using single-image pairs (one reference and one test) datasets taken from urban environments. Large datasets of random places were generated using online image provider services. The experiments demonstrate the comparison between our method and the selection of image-to-image-based methods.

3.2.1 Data providers

There are several image provider services, such as Google Street View[1], Bing Streetside[2], and Mapillary[3]. For this work, we used datasets obtained from Google Street View and Bing Streetside. There are three benefits of using more than one image provider. First, both Google Street View and Bing Streetside provide a large amount of data. Second, using more than one image providers allows for variation of data, which means that, even at the same location, the appearance of the captured image pair is slightly different in terms of shade, tone, and viewing angle. Finally, as displayed in Figure 3.8, with a pair of only Google Street View images, even with some displacement, the matching features seem to target the sky patterns, rather than other significant semantic patterns such as buildings or roads. By using Bing Streetside, we retrieved a more proper test environment.

Figure 3.9 illustrates an overview of the data gathering processes on Google Street View and Bing Streetside. Specifically, we set up an area for inspection and randomly selected a set of locations using latitude and longitude coordinates. For each location, we

[1] https://www.google.co.th/maps

[2] https://www.bing.com/maps

[3] https://www.mapillary.com/app

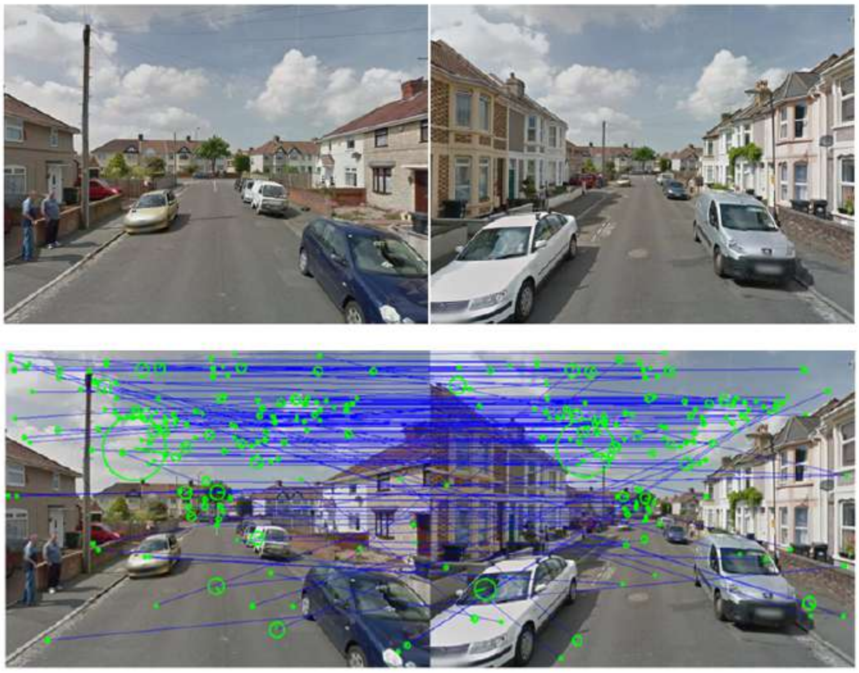


Figure 3.8: An example of image pairs captured using Google Street View with displacement, consisting of the original image pair (top row) and feature matching (bottom row). The alignment of feature pairs focused on sky patterns, rather than other features. Thus, we decided not to use the pair of Google Street View images in our experiments.

obtained image data by sending the corresponding coordinates to Google Street View and Bing Streetside APIs.

To add more variation to our dataset, we introduced a small displacement between the image pair. This range of displacement was selected under the condition that numerous similar landmark proposals appeared in the scenes, because if the images are too far apart, it is difficult to recognise the similarity, even for humans. Therefore, given the latitude and longitude randomly obtained from Google Street View, lat_G and lon_G, we applied:

$$[lat_B, lon_B] = [lat_G, lon_G] - \gamma \tag{3.5}$$

where lat_B and lon_B is the latitude and longitude set from Bing Streetside. In this work, we set the displacement $\gamma = 0.0001$ as it reflects the distance by 5–10 metres apart.

3.2.2 Data variation

To evaluate our system over invariance, aside from having two data sources and the aforementioned small displacement, we added more variation by using datasets from different cities. This enabled us to observe the performance of the method across various urban architectures. Specifically, we selected 200 random locations in six different cities: London,

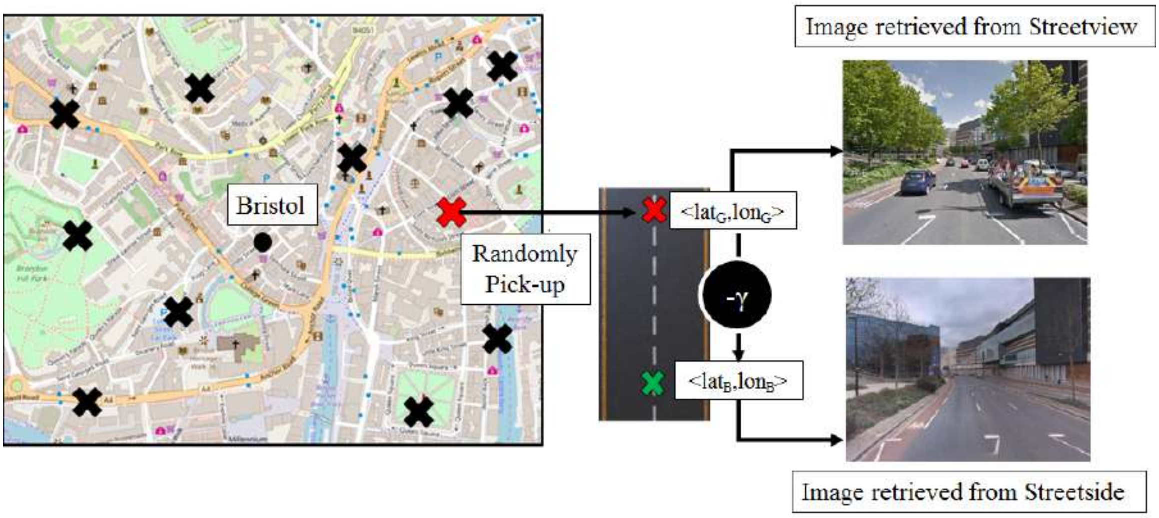


Image retrieved from Streetview
Bristol
Randomly Pick-up
$<lat_G,lon_G>$
$-\gamma$
$<lat_B,lon_B>$
Image retrieved from Streetside

Figure 3.9: Conceptualisation of the image pair gathered using Google Street View and Bing Streetside. We first selected the area for the experiment and randomly determined the locations (depicted by the cross signs). At each location, we requested the corresponding images by sending their coordinates (latitudes and longitudes) to each program's APIs. In this experiment, we initiated the displacement between the two as $\gamma = 0.0001$, which reflects a distance of 5–10 metres apart.

Bristol, Birmingham, Liverpool, Manchester, and Paris. In addition, the London and Bristol datasets were gathered three times (set1, set2, and set3). The difference between each set was the distance from the city centre, i.e., the test areas moved further away from an urban environment. Hence, there were ten test datasets with 2,000 different locations in total.

Figure 3.10 displays examples of image pairs from different cities. Although the physical distance between the viewing positions is not substantial, there are significant changes in structural appearance caused by varying lighting and visibility conditions, such as the presence of pedestrians and vehicles, as well as significant changes in the scale or focus of buildings. This makes the process of recognition more challenging. However, even with these environmental changes, the spatial patterns of the visible landmarks are still maintained; this is the characteristic of the urban scene that we aim to investigate through this experiment.

3.2.3 Comparison methods

To evaluate the performance of our method, we applied the same datasets and compared our results with four other methods: (i) the CNN landmark-matching method of [62], (ii) whole-image CNN matching [13], (iii) whole-image SIFT matching, and (iv) whole-image

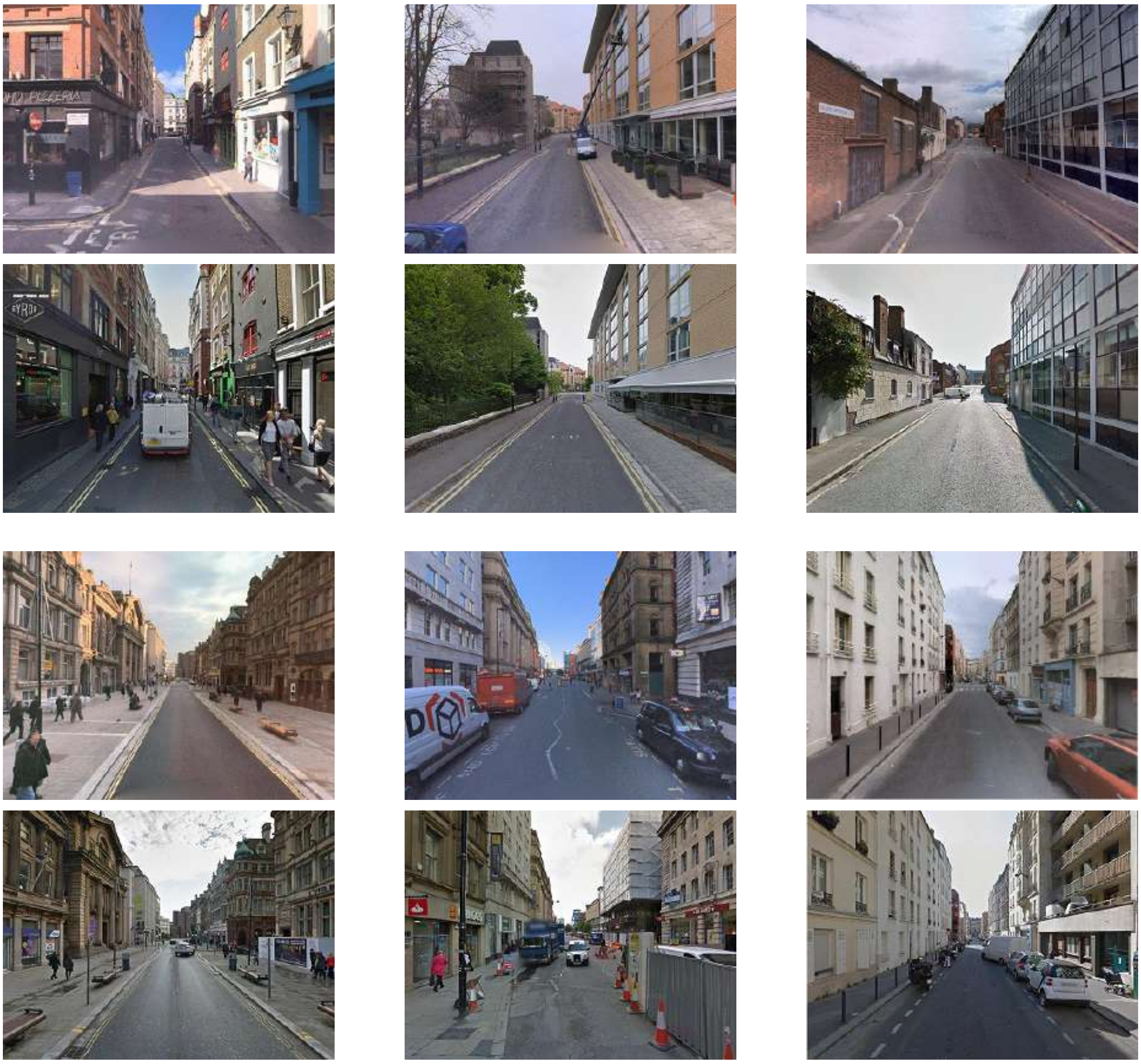


Figure 3.10: Examples of image pairs from each of the six cities in the ten datasets used in the experiments. Each pair is displayed vertically, and there are three pairs per row. From the top row to the bottom row and left to right are London, Bristol, Birmingham, Liverpool, Manchester, and Paris.

GIST matching. Relevant details for each method are discussed in this section.

CNN Landmark matching (CLM)

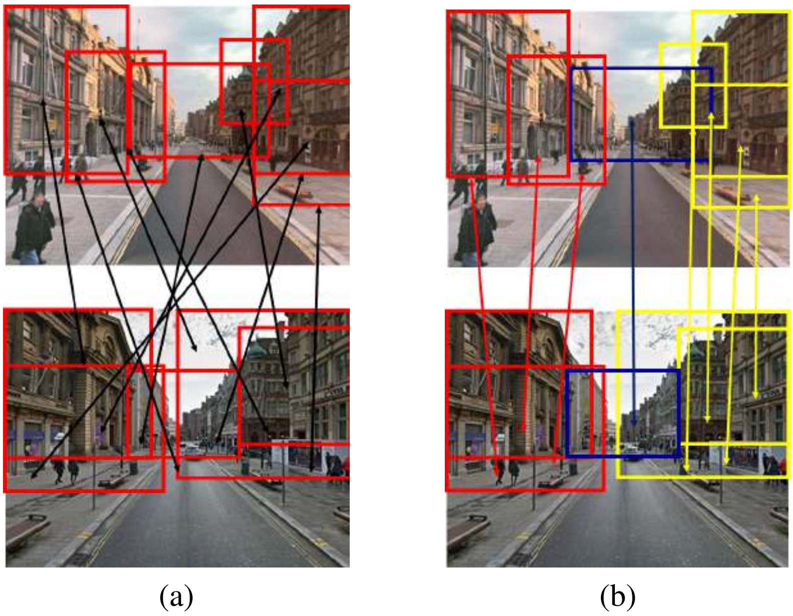


Figure 3.11: Conceptualisation of (a) CLM and (b) LDD landmark proposal matching. Though the basis is the same, LDD enforces the spatial distribution. Landmark proposals, in our method, can match within their section when CLM proposals contain no restriction for matching.

As illustrated in Figure 3.11, the primary difference between our method and that described in [62] is that matching in the latter is based on identifying similar landmarks across both views without a relative position. Specifically, best-matching pairs of CNN-GRP feature vectors are selected from Edge Boxes proposals, and the overall similarity between two views is the sum of the cosine similarities, weighted by a measure of similarity of the box size. In regard to clarity concerning our experiments, we found that the similarity metric provided in [62] did not yield sufficient performance. Therefore, we used a modified version that provided significantly better performance.

To find the similarity score S_{ab} between scene images I_a and I_b, we applied the modified version of equations (2) and (3) stated in [62] as:

$$S_{ab} = \frac{1}{n_a \cdot n_b} \sum_{ij} (d_{ij} \cdot s_{ij})) \tag{3.6}$$

where n_a and n_b denote the total number of detected Edge Boxes in I_a and I_b, and d_{ij} denotes the cosine similarity of the i^{th} and j^{th} Edge Boxes detected within scene images I_a and I_b, respectively. The variable s_{ij} denotes the shape similarity scores calculated from

$$s_{ij} = 1 - (\frac{1}{2}(\frac{|w_i - w_j|}{max(w_i, w_j)} + \frac{|h_i - h_j|}{max(h_i, h_j)})) \tag{3.7}$$

, where w_i and w_j are the width and h_i and h_j are the height of the i^{th} and j^{th} Edge boxes, respectively.

For comparison, we chose the same number of landmark proposals and the same GRP reduction size as ours. One used 25 proposals (CLM-25) and another used 50 proposals (CLM-50). Note that it is possible to improve the matching process of CLM using the technique of aligning the matching salient regions. However, this necessitates further investigation. In the present research, we only focus on evaluating the impact of spatial knowledge integration; using this method is an efficient way to represent the contrast of applying spatial knowledge.

CNN whole image matching (CWI)

Rather than finding landmark proposal regions for whole-image matching, we directly extracted the same CNN-GRP feature vectors from the whole image as used in [62]. Therefore, cosine similarity is again used as the comparison metric. We apply this method to observe the result using a CNN model in whole-image form comparing to patch-based form.

Dense SIFT matching (SIFT)

We chose a dense keypoint version of SIFT descriptors [80] as the representation of local feature-based methods. Specifically, we applied the implementation provided in the VLFeat library [175]. Note that by using dense keypoints, the accuracy of matching might be better than the normal SIFT; however, the processing time is slower, as there are a greater number of features. However, in this work, we focus on the precision of matching, rather than processing time.

GIST matching (GIST)

We selected the GIST descriptors as the representation of global feature-based methods. Therefore, we compared our method with whole-image GIST matching based on the implementation provided by Oliva and Torralba, as described in [83].

We then applied the above methods to our image pair datasets and compared the results with our method.

3.2.4 Place recognition using LDDs

We compared the performance of our method against the comparison methods for all ten datasets from six different cities. Each dataset contained 200 view pairs from various locations, one view taken from Google Street View and the other from Bing Streetside. Note that we used the Streetside images as test images and the Street View images as reference images. There are several methods to measure the performance of retrieval. At the initial state of work, we followed [62] to record precision and recall, indicating the overall accuracy of the system. This is also a standard evaluation method for place recognition systems, as stated in [91]. We discuss further improvements in Chapter 7.

Therefore, precision (P) and recall (R) are defined as:

$$P = \frac{tp}{tp + fp} \qquad\qquad R = \frac{tp}{tp + fn} \tag{3.8}$$

, where tp, fp and fn denote the number of true positives, false positives, and false negatives, respectively. We define each term as:

- True positive: the test image is matched with the reference image taken at the same location.
- False positive: the test image is matched with a reference image taken at a different location.
- False negative: the test image is not matched with any of the reference images based on a threshold of the ratio between the closest and second closest matches. Note that the variation of this threshold also enables us to create precision-recall curves.

Hence, in the experiment, we used the initial set-up as follows:

- **Size of the test images**

  We set the image sizes as 640×480 pixels for both Google Street View and Bing Streetside and used sections of size 320 pixels. Our provisional tests showed that using the larger number of proposals in the central panoramic section had a significant impact on performance.

- **Panoramic section**

  To gain a sense of spatial distribution, we applied the concept of vertically partitioning the image. In this work, we used three panoramic sections: left, middle, and right (in equation 3.1, $S = 3$), which allowed for 50% overlap with their adjacent sections, as illustrated in Figure 3.12. This overlap reduced the sensitivity to the positioning of section boundaries.

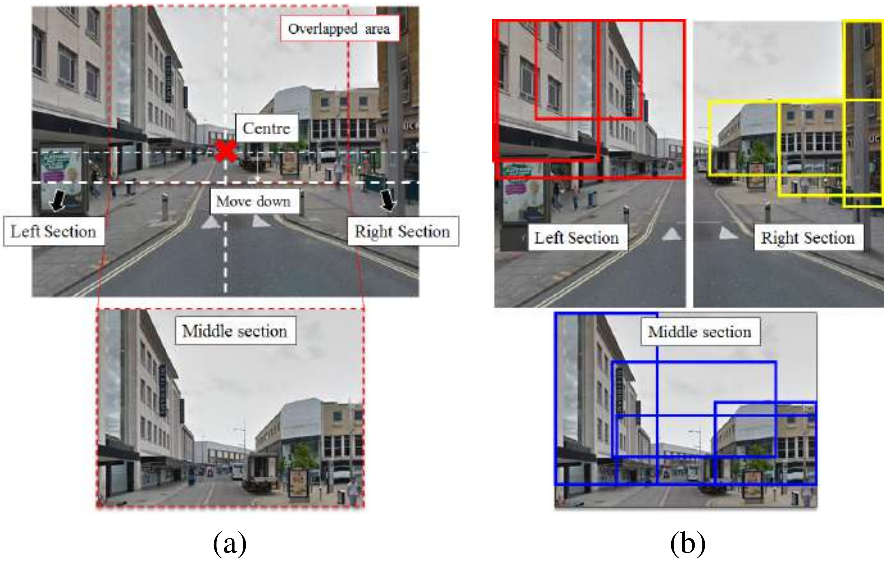


Figure 3.12: An example of segmenting the view into three panoramic sections ($S = 3$). In this work, we assume that the vanishing position is at the centre of the image, as depicted in (a), and (b) Edge Boxes detection applies to each section separately. Note that for the middle section, we used a smaller patch to avoid detecting elements on the road.

- **Centre of the partition**

  We assumed the images had been taken from a regular viewpoint, which is the camera pointing to the road. Therefore, we used the image centre as the centre of partitioning, as illustrated in Figure 3.12. However, the data might not always align in the regular viewpoint, and we propose later using the vanishing position (VP) as the centre. This is discussed in greater detail in Section 3.2.5.

- **Number of top-ranking proposals**

  We fix the number of top-ranking proposals according to the panoramic section. In this work, we limited the number of top proposals to 25 (10 left - 15 middle - 10 right) and 50 (15 left - 20 middle - 15 right) and labelled them as LDD-25 and LDD-50, respectively. A proposal was counted in a section only if each of its Edge Box was entirely within the section boundary, and the size of Edge Box region passed the threshold. Therefore, it is possible to have fewer proposals than stated. For example, at top proposals of 25, it is possible to have an LDD containing only 22 proposals (e.g. 7 Left - 15 middle - 10 right) as other proposals do not pass the constraint.

- **Size of the descriptor**

  As we apply GRP for dimensional reduction, we conducted some provisional tests, reducing the feature vectors from 64,896 down to 512, 1024, and 4096. After com-

Table 3.1: The record of precision values for 100% recall for all ten datasets using all seven comparison methods. It is clear that our LDDs, both in 25 and 50 proposals, generally performed better than other techniques.

	LDD-25	**CLM-25**	**LDD-50**	**CLM-50**	**SIFT**	**GIST**	**CWI**
London1	**84.5**	55	**88**	73.5	59.5	47	66
London2	83	68	**90**	**84**	58	44	74
London3	**72**	57	**83**	69	51	58	64
Bristol1	**66.5**	51.5	**68.5**	58	51.5	33	60.5
Bristol2	**63.5**	50.5	**65.5**	59.5	40.5	26	54.5
Bristol3	59.5	47	**67**	**64.5**	48	37	61
Birmingham	**62**	44	**71.5**	60	26.5	38	44
Manchester	**69**	50.5	**71.5**	63.5	33.5	33.5	63
Liverpool	**74**	46	**75**	62	52.5	40.5	53
Paris	**61**	35	**70.5**	49	40	35	38

paring the results and processing time, we chose to fix the dimension size down to 1024. Therefore, with a total of 25 and 50 proposals per view distributed over three panoramic sections, each descriptor was of size 25×1024 and 50×1024, respectively. By stacking feature vectors in this manner, we maintained the encoded spatial relationship between each landmark in the LDD.

Table 3.1 displays the precision values recorded for the various methods at 100% recall (all matches are treated as positives). Among all datasets, the LDD-50 method showed the best performance, and, with the exception of two datasets, the LDD-25 method was the second best. On the average, LDD-25 and LDD-50 got around 68% and 74% of precision, when CLM-25 and CLM-50 get 48% and 62%. As the bottleneck in this method is the CNN feature vector extraction, using 25 landmark proposals reduces the computational load by half. This finding shows the significant impact of using LDDs. By reducing the number from 50 to 25, our method still maintains a good performance compared to the CLM method. We believe this directly results from enforcing spatial distribution along with matching as our representation creates more distinctive views. However, the results of London datasets and Bristol datasets in Table 3.1 show the trend of gradually dropping precision. The precision of set2 and set3 of both cities are lower than set1. We believe that this causes by the different urban characteristic because data in set2 and set3 are gathered further away from the city centre area comparing to data in set1.

Analysing this deeper, it is also noticeable that the results from the London datasets were better than the others. Upon close inspection of the samples in Figure 3.13, we find that the predominant characteristics of places in London contained a highly distinctive appearance (low degrees of ubiquity). This contrasts with a mix of vegetation and buildings in the Bristol datasets and high degrees of ubiquitous architecture within the Paris dataset. Similar to London, the Liverpool dataset also contained characteristics of high distinction; however, like Bristol, there were changes in vegetation. For the other cities, the Birmingham and Manchester datasets contained characteristics similar to Liverpool, with lesser distinction and some vegetation. These urban characteristics are consistent with the results in Table 3.1.

We selected the London1 and Paris datasets for further investigation, as they exhibited highly contrasting characteristics in the ubiquitous architecture. Figure 3.14 demonstrates the variation in precision for the two; this variation in precision comes from reducing recall by increasing the number of false negatives via the threshold on the ratio of the closest and second closest matches. Both versions of our method, LDD-25 and LDD-50, outperformed the other methods. The difference in LDD-25 and CLM-25 was noticeable, with the former achieving a nearly 30% (60 locations) increase in precision using the same number of proposal landmarks. The results clearly illustrate the advantage of using landmark distribution to characterise views. Another useful result is displayed in Figure 3.15, which illustrates the similarity matrices for the same two datasets using methods LDD-50 and CLM-50. The matrices recorded the similarity scores between each test and reference view. There were high values down the main diagonal for the LDD-50 method, indicating the strong distinction of the correct places. In contrast, without spatial distribution, using the CLM-50 method, the closeness of the values was obtained, especially for the Paris dataset.

In addition, Figure 3.16 demonstrates examples of correctly matched views, and *none of these examples are correctly matched by the other methods*. In each case, the best matching landmarks found in each panoramic section are indicated by the colours red, blue, and yellow, from left to right, respectively. Each colour indicates corresponding landmarks in each view. These are challenging examples, as they contain differences in appearance and structure between the reference and test views, particularly with changes in vegetation and building structure. The similarity between the matched landmark pairs is clearly shown. Figure 3.17 illustrates examples in which our method failed to match the correct view. The top row shows the test images, the middle row shows the incorrectly matched views, and the bottom row shows the correct views. These are particularly challenging examples, as they are further complicated by a partial disappearance of landmarks. Dealing with cases such as these will be the subject of future research.

3.2.5 Vanishing position integration

As mentioned, in the previous experiments, we used the centre of the image as the centre for partition, implementing a slight pan (around 5°) from the viewing direction between

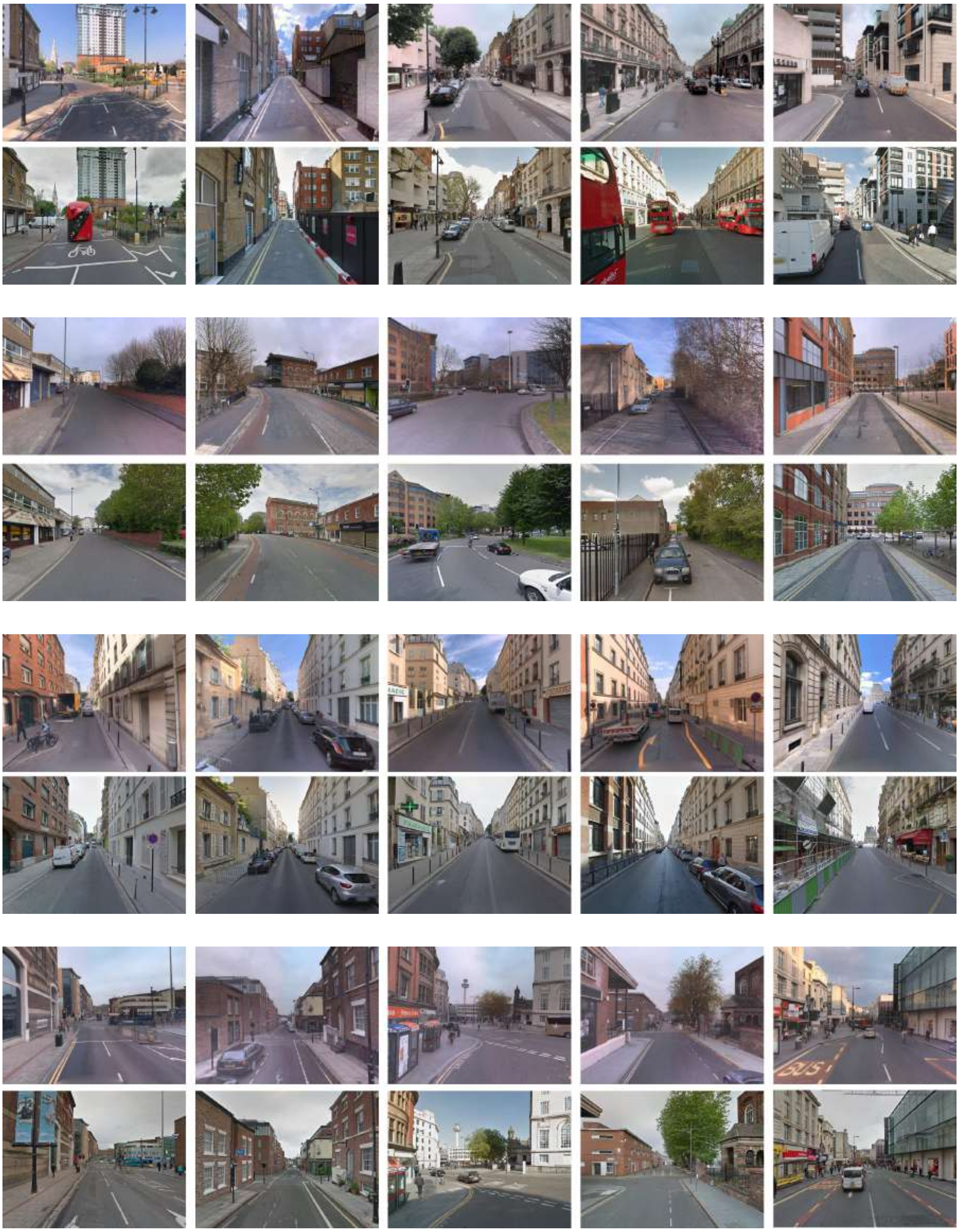
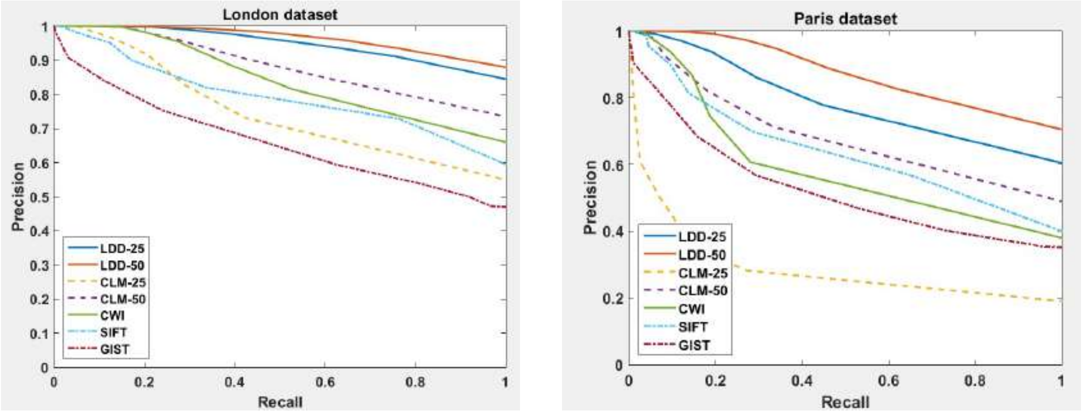


Figure 3.13: Examples of image pairs from four different cities. From top to bottom are London (high discrimination), Bristol (vegetation effect), Paris (high ubiquity), and Liverpool (high discrimination with some vegetation). The variation in each dataset is clearly depicted. This seems to reflect the high precision results of London in Table 3.1, as the city contains less vegetation and more ubiquity than the others in terms of buildings and architecture.



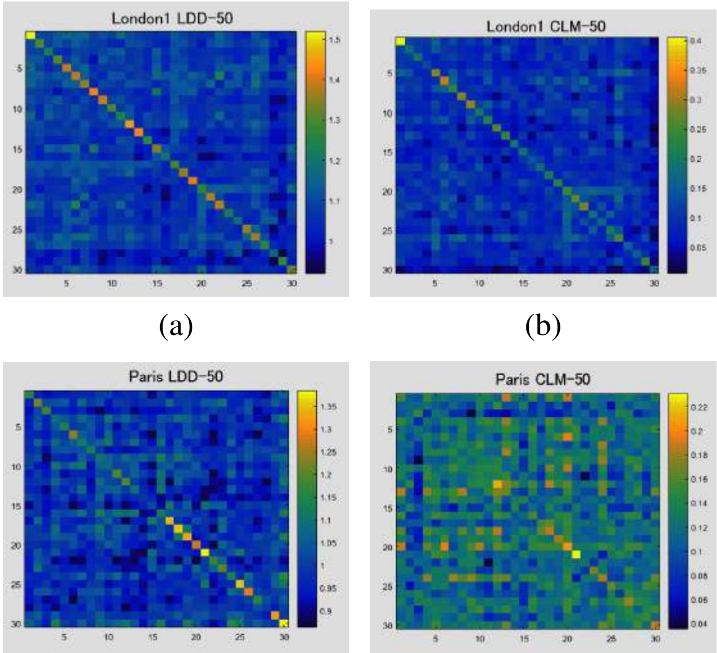
(a) (b)

Figure 3.14: Precision-recall curves obtained for all comparison methods for (a) the London1 dataset and (b) the Paris dataset. These figures support our belief that the spatial arrangement helps improve the performance of place recognition, especially in a city with less discrimination (as shown in Figure 3.13), like Paris.

image pairs. In this section, we investigate the case of a severe change in the viewing direction and provide a provisional solution.

In general, the centre position has little impact, since the vanishing position is often close to the image centre. However, in several cases, such as with a high degree of change in viewing direction, it makes a difference and corrects the matching of previously incorrect places. Therefore, we experimented with adapting the positioning of the panoramic sections according to view content. Rather than dividing the image evenly into three sections about the image centre, we computed the location of the vanishing point in each view. We applied the method described in [176] to centre the middle section, with the adaptation of the two outer sections.

Table 3.2 displays the provisional results of using the vanishing position. For the*dataset-vp1*, which only contains 10° changes from the centre of the viewing direction, there is a slight improvement. However, in the *dataset-vp2*, which contains 30° changes from the centre of the viewing direction, applying a pre-defined vanishing position showed more effectiveness. Figure 3.18 further illustrates examples from *dataset-vp2*. The top row displays a pair of views captured at the same place, with selected landmark regions derived using the image centre; this was the incorrect solution, as the detected landmarks in each test view did not correspond to the same landmarks in the reference views. In contrast, by shifting the centre to the detected vanishing position of each view, the detected landmark proposals in the test and the reference views successfully matched. However, these are only provisional results, and further investigation is necessary to determine the generality of using the vanishing position.



(a)

(b)

(c)

(d)

Figure 3.15: Matrices record similarity scores for 30 locations in the London1 and Paris datasets using (a) and (c) LDD-50, and (c) and (d) CLM-50. Note that we used 30 randomly selected location pairs, rather than all 200, for visualisation purposes.

3.2.6 Humans and place recognition

Our previous tests revealed the impact of using spatial knowledge for vision-based place recognition. However, there are some cases that both our algorithm and the state-of-the-art techniques could not solve. To reinforce the assertion that humans are better aligned with solving this problem, we conducted a small online experiment by selecting ten images from the London1 and Bristol1 datasets, 20 in total. The level of difficulty varied; 80% of these questions are unsolvable by the system. There were five choices of answers: four of them were images we manually selected, and another was a ‘no match’ choice. The latter was to ensure that the participants would carefully observe all choices.

In total, 112 participants joined this experiment; 54% of the participants were between 18 and 24 years old. All were Thai, were unfamiliar with Western architecture, and had never been to London and Bristol. The average processing time to complete the test was around 2-5 minutes per set, which was consistent with the psychology studies in [177, 178, 179] claiming that each person has varying levels of ability in location-based activities depending on their background knowledge.

The provisional results revealed that the average percentage of participants getting each dataset correct was 72% and 64% for the London and Bristol datasets, respectively. Figure

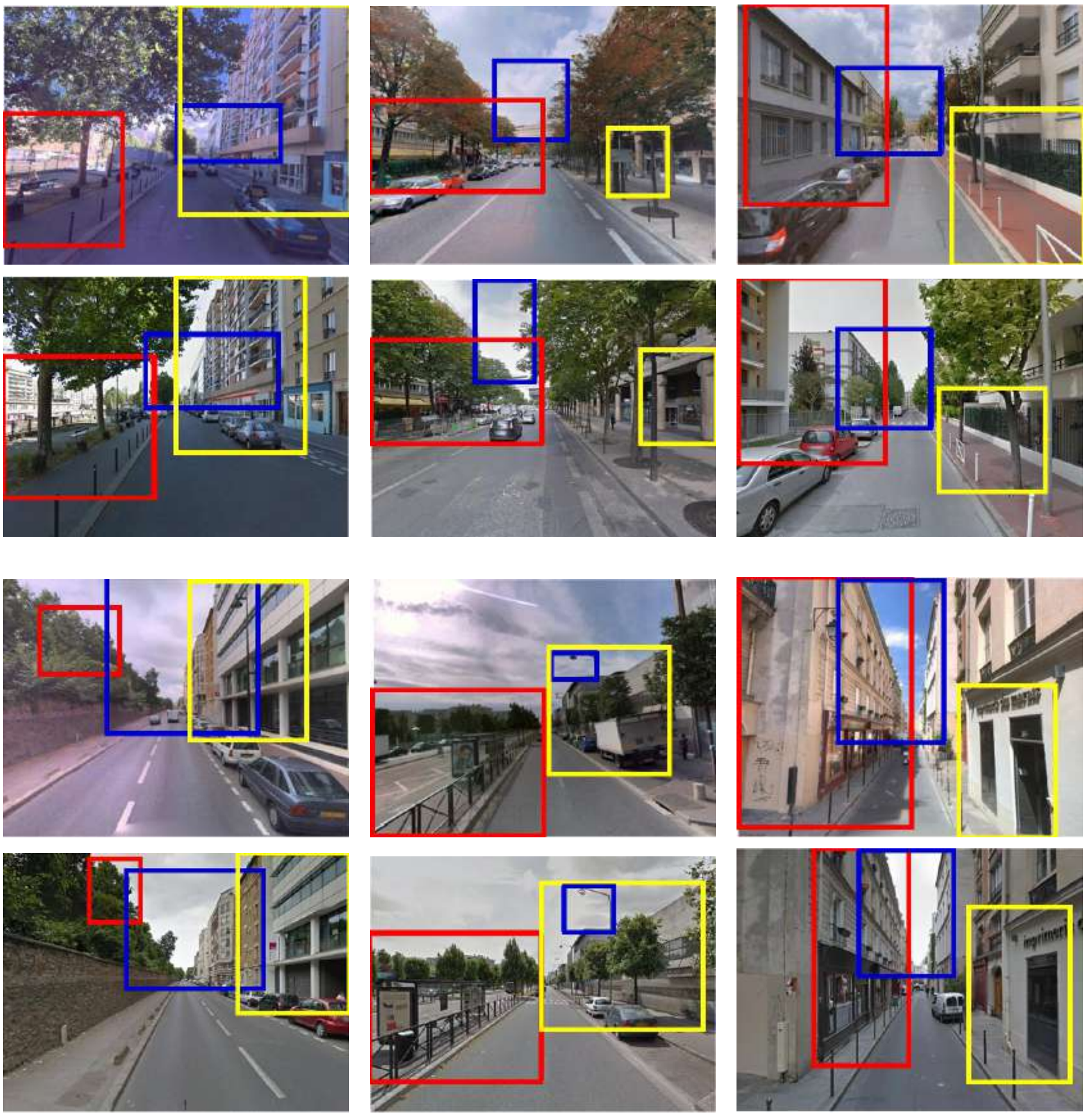


Figure 3.16: Examples of the correct view matches obtained using the LDD-50 method. Matches are displayed one above the other, and there are three matches per row; the match images contain the sensible match landmark pairs in all three panoramic sections.

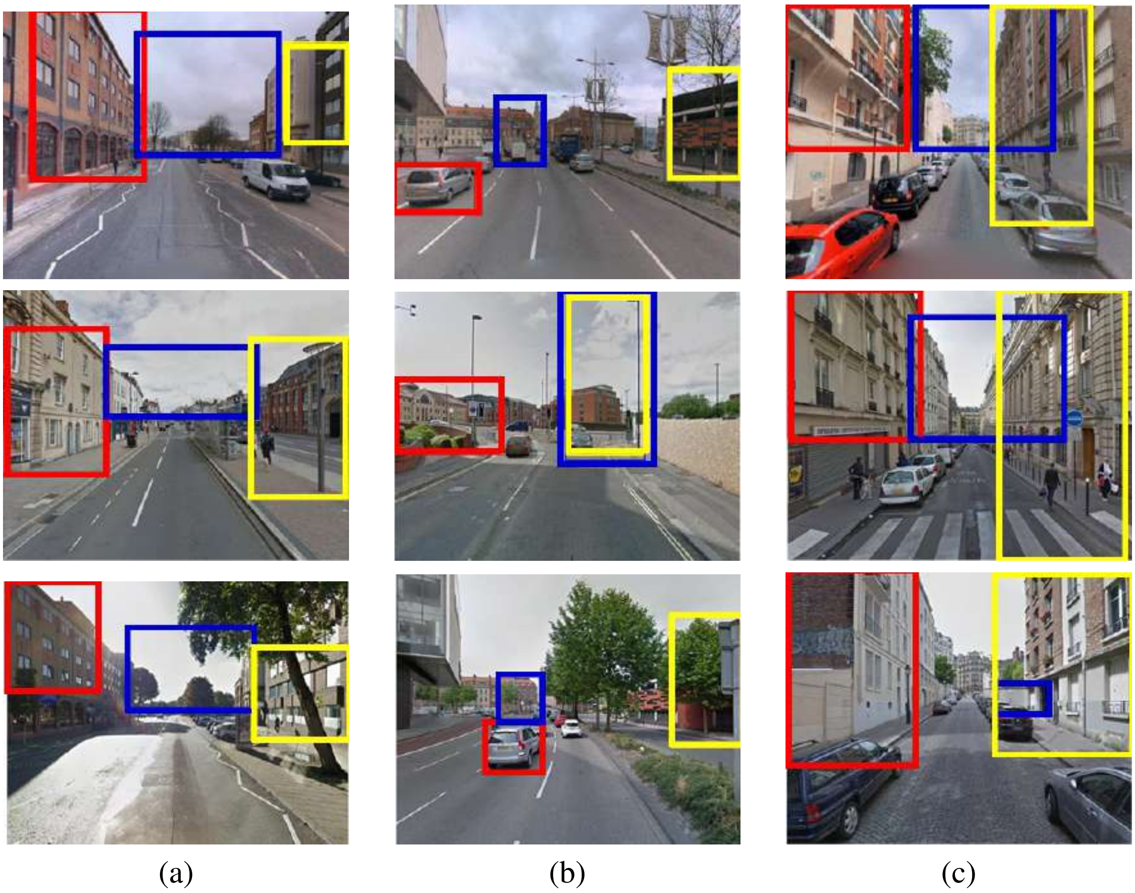


Figure 3.17: Examples of incorrectly matched views obtained using the LDD-50 method consisting of test images (top row), best match images (middle row), and the correct results (bottom row). The correct pairs shown in the top and bottom rows differ in terms of appearance due to the changes in occlusion, such as (a) shadows and light and (b) a tree, as well as (c) the partial disappearance of landmarks.

Table 3.2: Records of accuracy using centre partition at a centre of images and at a vanishing position. Dataset-vp1 contains image pairs with only a small change in the centre of the viewing direction, while *dataset-vp2* contains image pairs with higher degrees of changes. The cLDD method refers to LDDs with the image centre, and vLDD refers to LDDs with the vanishing position.

	CLM-25	**cLDD-25**	**vLDD-25**
Dataset-vp1	38%	60%	**66%**
Dataset-vp2	20%	38%	**62%**

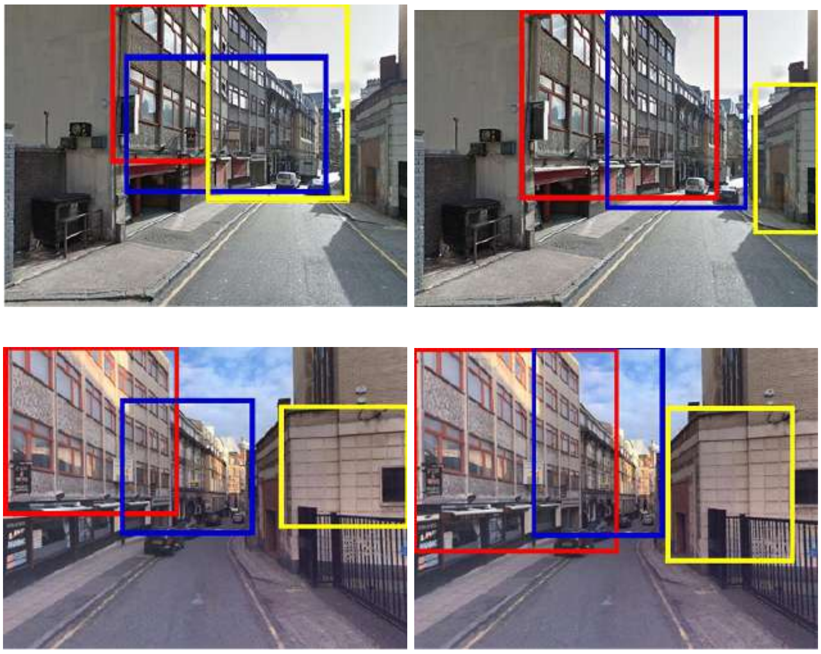


Figure 3.18: Using the view vanishing point to centre the panoramic sections improved the matching of landmarks (right) as compared to using the image centre (left).

3.19 presents an example of a test view given to the participants and the percentage of people that chose each answer. Figure 3.20 further illustrates examples used in the experiment. The results strongly suggest that, even when the degree of environmental change is high, humans can still recognise the similarity between images captured at the same location, even if they are not familiar with the area. This opens up further exploration directions.

3.3 Summary

In this chapter, we have presented a novel method for vision-based place recognition using landmark regions represented by CNN features. Although the method has aspects in common with the CLM method of Sünderhauf et al. [62], we have demonstrated that the use of LDDs has a major impact on performance. We gain significant precision not only over

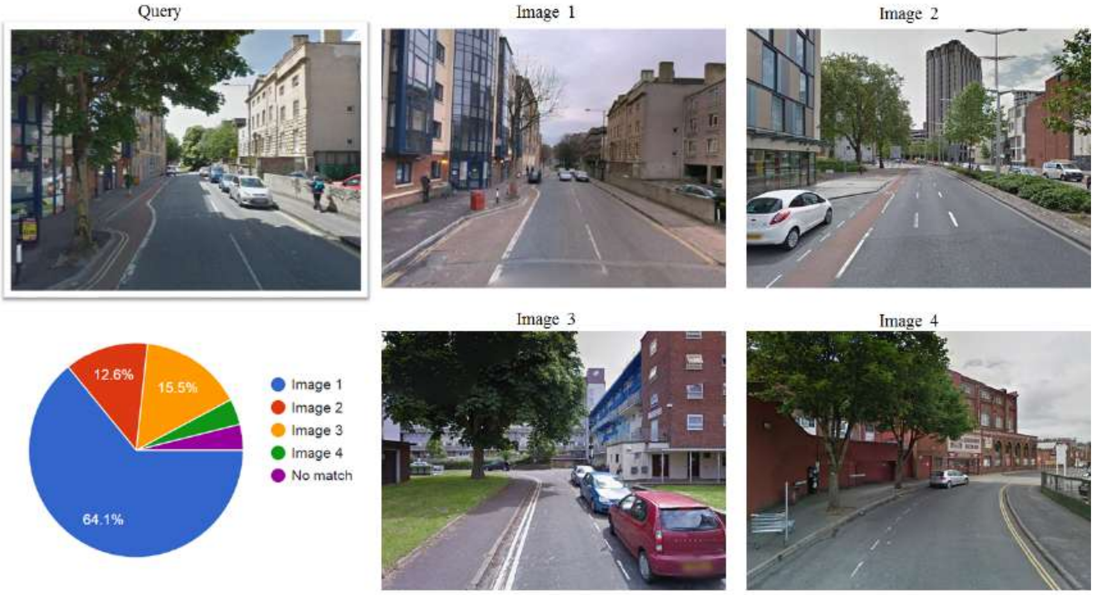


Figure 3.19: Example of a test we gave to the participants. The top left column contains the test image, and the middle and right columns display examples of the given choices. The pie chart represents the percentage of answers, revealing that up to 64% of the participants chose the correct answer (Image 1) even though there were obstructions such as trees and shadows in the query image.

CLM, but over the other whole-image techniques as well. For example, in experiments on the 200 image pair database, the performance of LDD-25 was approximately 20% greater (40 locations), on average, compared to CLM-25.

In addition, we demonstrated the impact of data variation on place recognition. For example, the London and Paris datasets contained various characteristics that affected the performance of the method. Our method using spatial knowledge exhibited a distinguishing power over the more ubiquitous areas like Paris. Moreover, for a more practical perspective, we demonstrated the effectiveness of using the vanishing position as the centre to improve the performance on image pairs that contained different pan angles. The results revealed that, by implementing the proper central location, the landmarks detected in each region were better arranged. Therefore, these experiments suggest the impact of combining spatial knowledge with landmark detection. This finding is consistent with the human spatial cognition studies reviewed in Chapter 2. Further, to reinforce the human power of place recognition, we conducted another provisional experiment using our data with participants. The results indicate that humans have some ability to solve this problem using no external information. We discuss further limitations and future possibilities of this work in the final conclusions in Chapter 7.

In the second part of this thesis, we shift to a more challenging problem: localisation

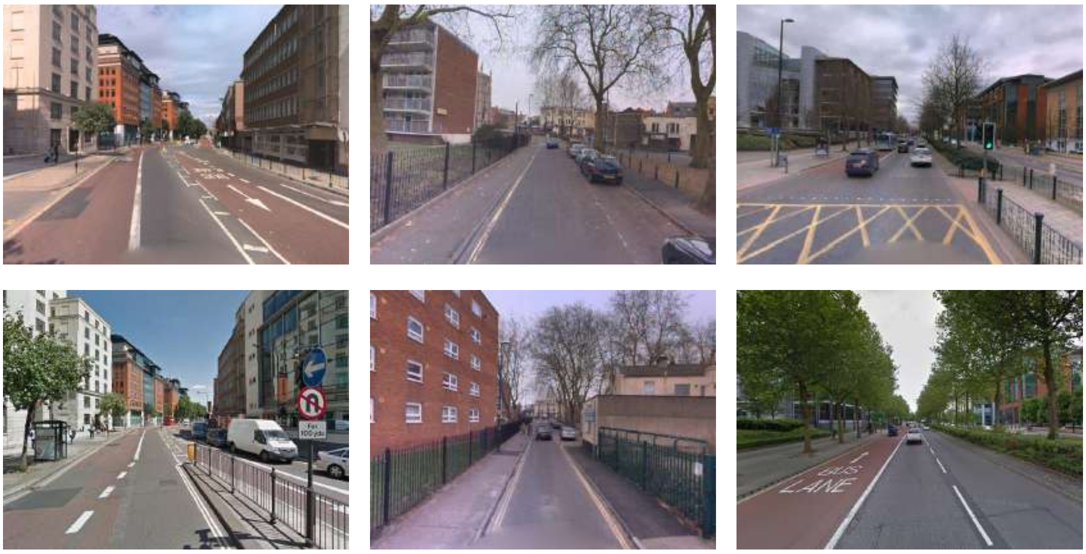


Figure 3.20: Examples of views used in the human experiment under three different change conditions: (a) lighting, (b) displacement, and (c) vegetation. The top row shows the test view, and the bottom row shows the correct reference. The rates of participants getting the answer correct were 73%, 44%, and 28%, respectively.

using image to 2-D map. Rather than using the image-to-image approach, we introduce the autonomous localisation system using image-to-2-D-map approaches. By changing the reference from images to the 2-D vector map, we aim to gain advantages in scalability, invariance, and human interaction. Further details are provided in Chapters 4–6.

Chapter 4

Vision-based cross-view localisation

In Chapters 2 and 3, we demonstrated the use of vision-based place recognition systems that provide an alternative to infrastructure-dependent sensing, such as GPS, especially when operating in urban environments. Most approaches adopt image-to-image database matching, in which environmental images are matched to a database of location-tagged images or image features [91]. However, as stated in Section 2.3, in recent years, one of the key concerns raised with such methods regards the invariance of representations to temporal changes (seasons, time of day, etc.) and spatial changes (viewpoints, occlusions, etc.) For example, the FAB-MAP algorithms [55] use image features with viewpoint invariance to provide large-scale matching over long routes of up to 1000 km. At the same time, other methods deal with changing appearance either through invariant representations [111], storing multiple representations [180], or learning models of appearance changes [110]. More recent works have utilised the power of deep-learning methods to gain improved matching [62, 164]. However, in all cases, large-scale localisation has large-scale memory requirements, in the order of hundreds of gigabytes [55], which leads to the problem of scalability.

To overcome these issues, we are motivated by how humans tackle this problem, which is to use semantic reasoning approaches. In computer vision, as reviewed in Chapter 2, a group of works [44, 62, 150, 151, 181] applied semantic information for recognising a place and demonstrated the potential of semantic features to provide invariance and reduced representation size. However, for large-scale localisation, these methods do not naturally scale. In this chapter, we consider the use of the 2-D map, which is how humans project their mental maps for localisation and navigation activities. In particular, in Section 4.1, we discuss the use of map and navigation. In Section 4.2, we review the development of autonomous navigation related to human wayfinding abilities. In Section 4.3, we explore the cross-view localisation approaches, which are similar to the human perception of location-based activities, and extend our focus to the image-to-2-D-map localisation approach, as well as discuss the related works. Finally, in Section 4.4, we summarise the reviewed techniques and address problems we intend to solve.

4.1 Maps and Navigation

In location-based subjects, a ‘map’ is a form of an image that depicts the selected features of physical geography. Symbols are generally used as representations of these features; for example, a square represents a building, or a circle represents a lake. The practice of creating a map is called cartography and the use of maps has a long history. For example, there is some evidence that prehistoric people may have used maps [39], supported by depictions of places that have been found on rocks and carved animal bones, as shown in Figure 4.1. Subsequently, ancient people expanded their interest in recording their surroundings [182]. For example, ancient Babylonians and Greeks created their version of the world map. Since that time, the art of cartography has spread across the world.

In general, the work of a cartographer is to project visual information onto a map that can be stored in various forms and materials. In ancient China, the territories and landscapes were recorded on bamboo, wooden blocks, and pieces of silk, and later shifted to paper. At the same time, to make the record more precise, people have made use of tools such as compasses and telescopes. In this respect, map development has occurred in parallel with printing technology and navigation instruments. Photography techniques have also improved information gathering. For example, we can retrieve a wider range of information from an aerial view (bird’s-eye view) using satellite and remote sensing.

Larger-scale operations require proper frames of reference, which has introduced the use of geographical coordinates: latitude and longitude. Each set of latitude and longitude numbers points to a specific location on Earth’s surface. The system designed to create, manipulate, store, and analyse the geographical data is called a geographic information system (GIS) [183]. For more than a century, people have used maps containing GIS data for administrative tasks such as city management, strategic plans, and facility infrastructure plans. For example, in London in 1854, Dr John Snow used the map to localise cholera victims and analysed the clusters to identify water areas that were sources of the disease [184]. With technology development, people started to record geographical data using digital GIS databases. Common tools for computer mapping are computer-aided design (CAD) and GIS software. Some works [185, 186] have demonstrated that using computer software improves the cartographic works. By converting to the digital format, one 2-D map can store several levels of semantic information used for administration, such as ownership of the land, address, and restriction, as illustrated in Figure Figure 4.2.

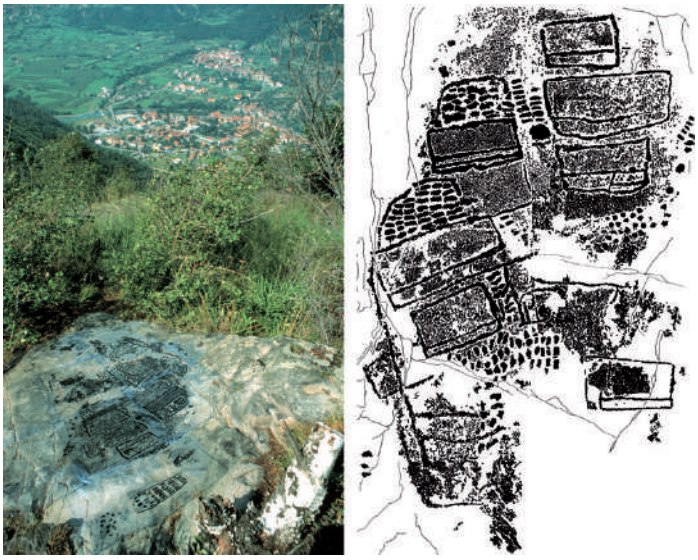


Figure 4.1: Valcamonica rock art, from the 4^{th} millennium BC, which appears to depict a map [18].

Today, people use digital applications to support their location-based activities, such as finding restaurants, tracking public transportation, and playing augmented reality games. To support human activities, there are several digital map providers, such as Google map[1], and OpenStreetMap[2]. Digital maps are typically created from annotating objects that appear on satellite images. The process of satellite image retrieval can be performed automatically, but the process of annotation still requires manpower. As reported in [187], Google Maps employs thousands of people, including map creators, GIS analysts, and AutoCAD designers. Another example is OpenStreetMap; as they are an open-source map provider, they ask volunteers to do tasks like inserting and annotating GIS data. As reported in [188], in mid-2019, there were around five million registered OpenStreetMap users.

Based on these examples, it is clear that the current state of cartography still involves numerous people. To automate the task relates to research in automatic image annotation and semantic segmentation [189, 190]. Some studies [191, 192] have applied the techniques to auto-annotate satellite images. However, these techniques are still being developed to operate in large-scale areas. In addition to this, several map providers do not only convert data from paper-based to digital-based, but they also offer digital administrative functions similar to the paper-based maps, such as natural disaster management using Google Crisis Map[3] and Humanitarian OpenStreetMap[4]. However, the main function of

[1] http://map.google.com

[2] https://www.openstreetmap.org/

[3] https://www.google.org/crisismap/weather_and_events

[4] https://www.hotosm.org/

the map is still for navigational purposes. Even with developing technology, people still maintain map-reading and navigation abilities. Therefore, in the next section, we discuss the process of human wayfinding and compare it to autonomous navigation.

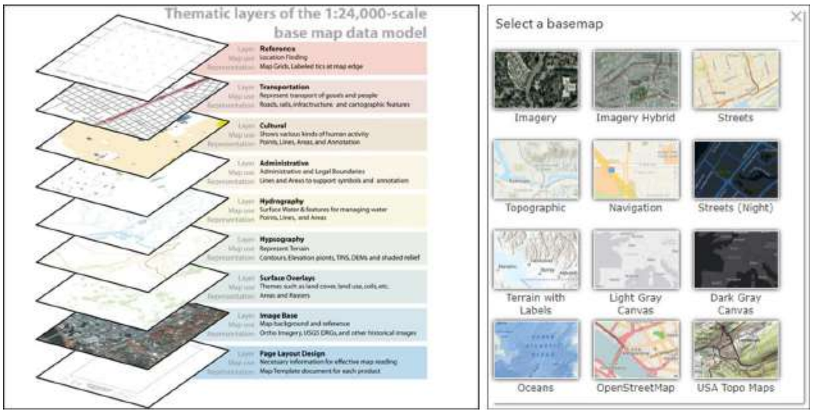


Figure 4.2: Examples of various levels of GIS information integrated on a 2-D map. Images were retrieved from [19].

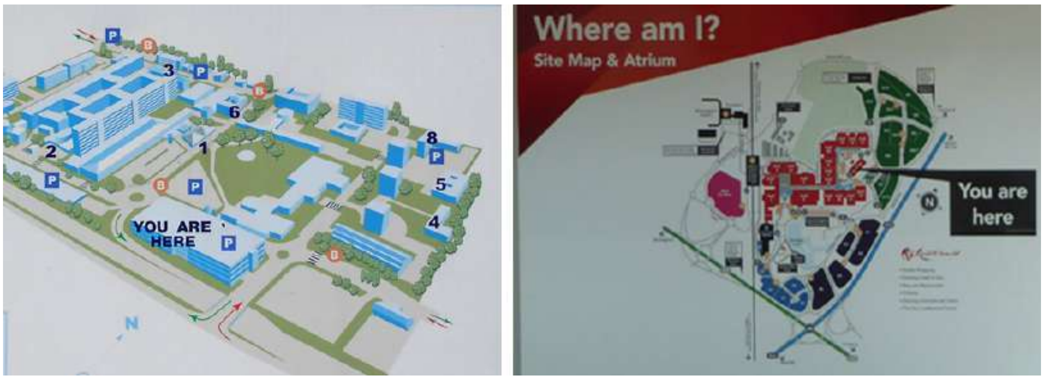


Figure 4.3: Examples of the ‘you-are-here (YAH)’ map, which is an ad hoc physical 2-D cartographic map to help humans to orient and navigate themselves in an unknown environment. Images were retrieved from [20].

4.2 Human wayfinding and autonomous navigation

As previously mentioned, the relationship between maps and navigation has existed since ancient times. Humans have used maps for centuries to cope with location-based activities such as localisation, path planning, and navigation. With the development of technology in recent years, systems have become more efficient in performing autonomy tasks, especially navigation. In this section, we review the concept of human wayfinding and the development of autonomous navigation imitating human behaviours.

4.2.1 Human wayfinding

Psychology research describes the purposes of human wayfinding as (i) to reach a familiar destination, (ii) to revisit an initial position, and (iii) to visit a new or unfamiliar destination [67]. To do that, people take cues from their surroundings to orient themselves within their *cognitive maps* (or mental maps) and navigate from place to place [67, 68, 152]. The cognitive map is a representation that the human brain constructs by decoding information about relative features and locations of the spatial environment. As highlighted in several spatial cognitive studies [71, 157, 193, 194], one of the longest established models for a large-scale space representation is the landmark route survey (LRS) framework [21], which explains the spatial knowledge representation using:

- **Landmark knowledge** - the characteristics of cues to be counted as landmarks are saliency, orientation, static, and dependent (Figure 4.4a).
- **Route knowledge** - once landmark knowledge has been retrieved, route knowledge is formed by combining those landmarks (Figure 4.4b).
- **Survey knowledge** - finally, the route knowledge is used to construct a graph of the environment, called a survey. Survey knowledge is used for decision making to select the most suitable route. This type of knowledge is only formed when people are familiar with the area (Figure 4.4c).

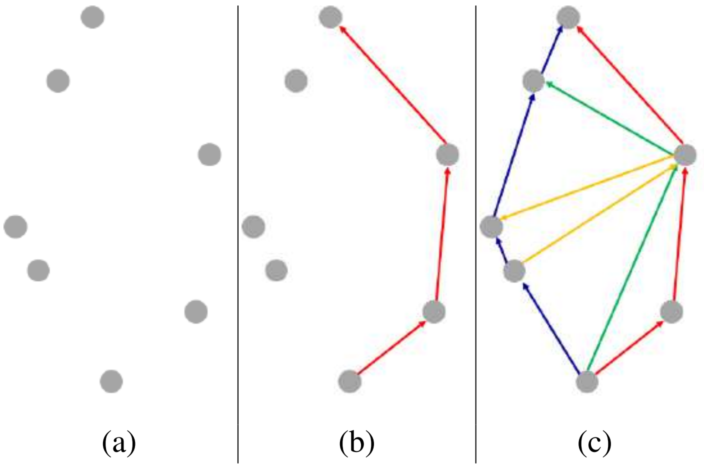


Figure 4.4: Conceptualisation of landmark, route, and survey framework: (a) landmark knowledge, (b) route knowledge, and (c) survey knowledge derived from Siegel and White [21]. The landmark refers to the objects of interest in the scene, and the spatial relationship between them forms the route knowledge. With many available routes, the survey knowledge is used to determine the one that is most suitable.

Thus far, due to the complexity of human brain functions, the LRS theory has remained unsolved. Some studies [195, 196] have made opposing statements regarding the definition of survey knowledge. However, the concept of using spatial relationships between landmarks for cognitive map construction is supported by a group of psychological studies [71, 197, 198], as well as a group of works in robot navigation [158, 159, 160]. Based on the stated theory, several studies [199, 200, 201] have noted that, conceptually, the human cognitive map is usually depicted as a bird's-eye view, or a 2-D map form, as this is the easiest way to illustrate the relationship between spatial components. This is consistent with Figure 4.1, which depicts the 2-D map used by prehistoric people.

For navigation in an unfamiliar environment, people usually rely on external information such as signs, posts, or verbal pieces of advice [202]. In the real world, people use some physical tools, such as paper maps, YAH maps (Figure 4.3), and mobile navigation devices [203]; these secondary sources typically provide the important landmarks, road segments, some specific information, and, more importantly, the spatial knowledge between elements in the map. People rely on these maps to localise themselves and perform path planning to reach their destination. Therefore, it can be inferred that the key elements of human wayfinding are landmarks and maps.

4.2.2 Autonomous navigation imitating human wayfinding

For the autonomous system, as described in [204, 205], the vision-based autonomous navigation systems are categorised into map-based, map-less, and map-building approaches. We follow these categories with some modification in terms of definitions to better fit them to modern applications. Therefore, in this thesis, we define the difference between the map-based and map-less approaches as the reliance on map creation. For the map-less approach, the autonomous systems do not require any prior knowledge, but they apply collision avoidance measures, such as object detection or motion-tracking techniques, so that the robot could move freely in the environment [206]. A common technique for the map-less approach is to use optical flow, which was initiated by a study that imitated the movement of insects [207] and was later extended into several applications, as in [208, 209, 210], specifically for micro air vehicles (MAVs) and bio-robotic studies.

For the map-based approach, the autonomous systems localise and navigate themselves based on the map or information provided either before or in the process of operation. This category is more influential for vision-based navigation applications, including some of those we have reviewed in Chapter 2, such as [54, 211, 212]. Types of map representation used in these systems generally fall into two categories: topological-based and metric-based. Figure 4.6 illustrates the structure of both map representation approaches;

the topological-based approach manipulates objects in terms of a relationship, while the metric-based approach works on a 2-D grid space. Regarding usage, the difference between these two is not well defined [213]. However, in practice, metric-based approaches are usually applied for tasks requiring high accuracy, such as obstacle detection, while the latter, topological-based approaches, are more suitable for simple purposes such as path planning. To gain advantages over both types, many applications have combined both techniques as a hybrid topological-metric approach [214, 215, 216]. Both types of maps in some sense replicate the way humans perceive the world; however, they still need more semantic information encoded.

The final category, which falls between the two aforementioned categories, is the map building approach. The systems in this category do not require prior knowledge, but they build the map while localising themselves, which is the concept of simultaneous localisation and mapping (SLAM) systems [37, 38]. The difficulty with this problem is that localising the robot necessitates the consistency of the map; however, to retrieve the map, an accurate estimation of the current location is also required. This is similar to the chicken-and-egg problem. The SLAM system has become one of the most challenging problems in robotic localisation and navigation, as discussed in [91, 121, 196]. This supports the use of the map for autonomous navigation.

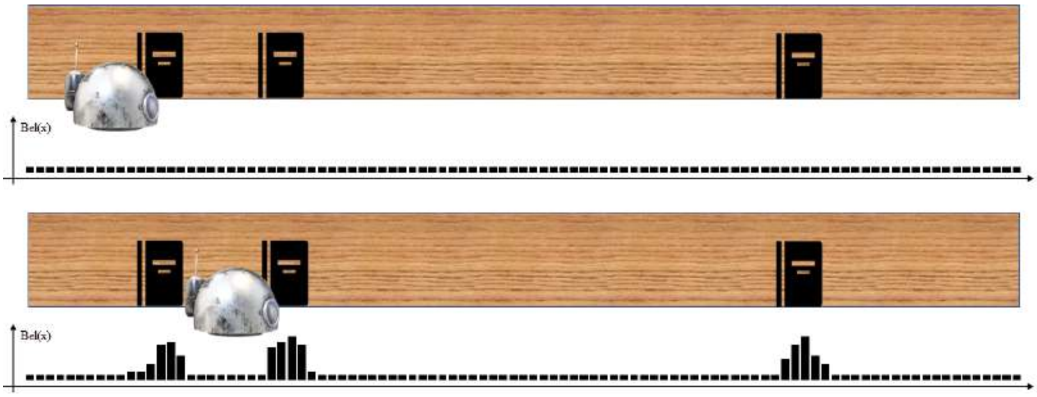


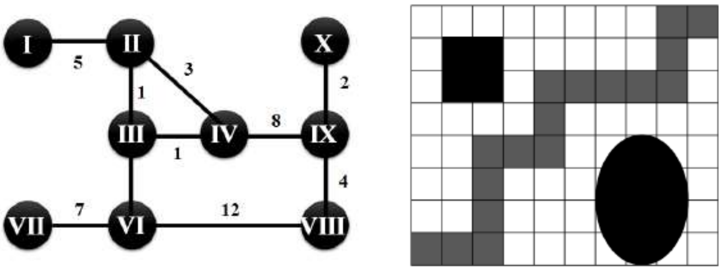
Figure 4.5: The concept of Monte Carlo localisation (MCL). At the initial stage (the top row), the probability of the robot being at each location is equal. Once the robot moves and gathers more information from its surroundings (in this case, the location of the doors), the belief changes. The robot is more certain that its location should be at one of the doors.

One of the most well-known algorithms integrated with the use of the map is the probabilistic approach [213]. The initial technique used in the earlier state of autonomous navigation was the Kalman filter [217, 218], which used the probability distribution over all possible positions to estimate the state of the system. However, originally, the Kalman filter was only applied for position tracking from the given initial position. Motivated by this, to localise at an unknown initial position, the Markov Model [219] and Monte Carlo Localisation (MCL) [220] proposed methods of pose estimation using the probability distribution based on the Bayes algorithms, which rely on the assumption that the future states depend on the current state. This concept has later been extended to the particle filter (or sequential MCL) technique [221]. Figure 4.5 represents the concept of using MLC, starting with initially setting all locations with equal possibility. Once the robot moves, the retrieved information is used for re-calculating the possibility. In the given example, the robot perceives the presence of a door. Therefore, the likelihood of it being where the door is located is higher than in other locations.

4.2.3 Semantic representations

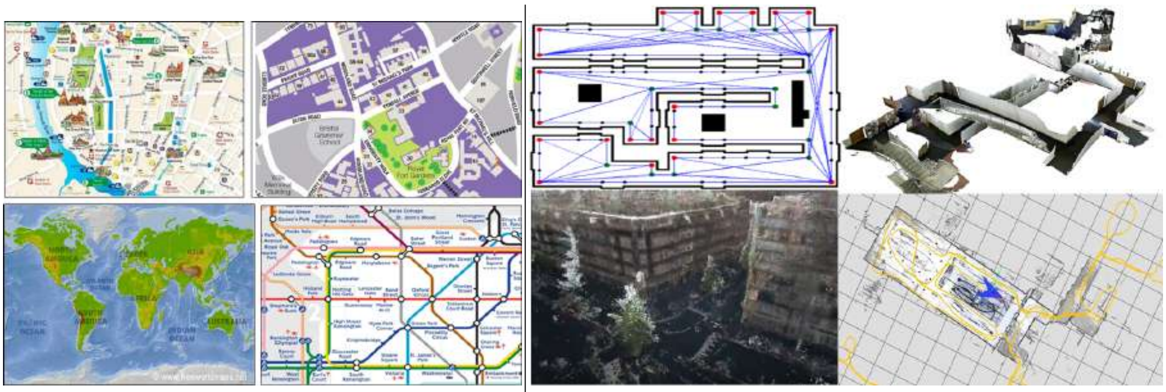
Based on the probabilistic model, a group of works [222, 223, 224] have performed localisation by making use of range sensor data to detect the presence of semantic information like doors, rooms, or corridors in the indoor environment. With the later introduction of visual sensors, autonomous navigation has become more similar to human behaviour and has made progress in the semantic area, as discussed in Chapter 2. The prominent techniques in the early state were to extract the salient image features using traditional feature extraction [80, 81] and construct a group of features as a set of visual vocabulary or BoF [82]; these sets of virtual vocabulary were treated as visual landmarks. However, these features differ from how humans perceive landmarks. The community later introduced the technique of extracting a higher sense of semantics than visual words, but lower than semantic objects, such as building façade segmentation [46, 132, 133] and cities' repetitive patterns [136, 137]. Additionally, with the introduction of learning-based techniques such as the SVM and the CNN, improvements in detecting semantic information have shown great progress, as demonstrated in [129, 130, 131]. To move closer to human perception, several applications have proposed methods of detecting higher-level semantic objects, such as landmarks [62], road lanes [63, 64], and texts [150, 151], or constructing the semantic segmented map [43], which demonstrated increased robustness of the technique.

To improve human-system interaction, the ideal situation is that the systems can understand the environment in the human context. Figure 4.7 illustrates examples of maps used by humans and the autonomous system. The maps humans rely on usually contain richer semantic information, especially in linguistic form. To bridge the gap between humans and autonomous systems, several applications have proposed methods based on those imitating the human wayfinding capability, such as the PhotoMap application [225, 226], which uses images of YAH public maps. The maps are manually geo-referenced with online maps to



(a) (b)

Figure 4.6: (a) topological-based and (b) metric-based (or grid-based) are common types of maps used in localisation and navigation problems. The topological map is constructed from nodes and links. This contrasts with the grid area of the metric-based map.



(a) (b)

Figure 4.7: Examples of maps commonly used by (a) humans, retrieved from [22, 23, 24, 25], and (b) autonomous systems, used for localisation and navigation tasks, as in [26, 27, 28, 29]. There are differences in terms of the provided semantic information. Humans tend to rely on images, symbols, and texts, which suggests how important semantic-reasoning is for human wayfinding.

provide specialised local data with navigation information on mobile devices and recognise the value of pictorial map data for human spatial cognition. Other examples, such as in [150, 151], applied techniques to extract the surrounding texts (such as from signposts, shopfronts, and the names of roads), which provided rich semantic information, and matched them with the information retrieved from the map. This is similar to the human ability to self-orient by reading physical maps.

However, this topic, in comparison to others, still has not set a solid foundation, especially when a 2-D map is involved. The basic requirement of the system is to match between two sources whose appearances and viewpoints are significantly different.

4.3 Image to 2-D map localisation

Image-to-2-D-map localisation is the process of matching between images and 2-D maps that correspond with one another. Given a street-level image, the system should be able to identify its location compared to the references containing extreme viewpoint changes. This is similar to how humans use their vision sensors (eyes) to perceive their surroundings and try to orient their current location (localise) in relation to any forms of cartographic 2-D maps. The task of matching between images and a 2-D map is a part of the cross-view localisation problem.

4.3.1 Cross-view localisation

In general, vision-based cross-view localisation approaches are operations that perform by matching two visual sources in a different domain or with a drastic change in viewpoints. Figure 4.8 illustrates examples of key challenges in cross-view image matching. The given examples represent the same place recorded in different domains. Therefore, the major challenge is for the system to recognise the similarity between them. In addition to this, with drastic viewpoint changes, traditional hand-crafted feature descriptors such as SIFT or SURF fail to detect and match, as demonstrated in Figure 4.9.

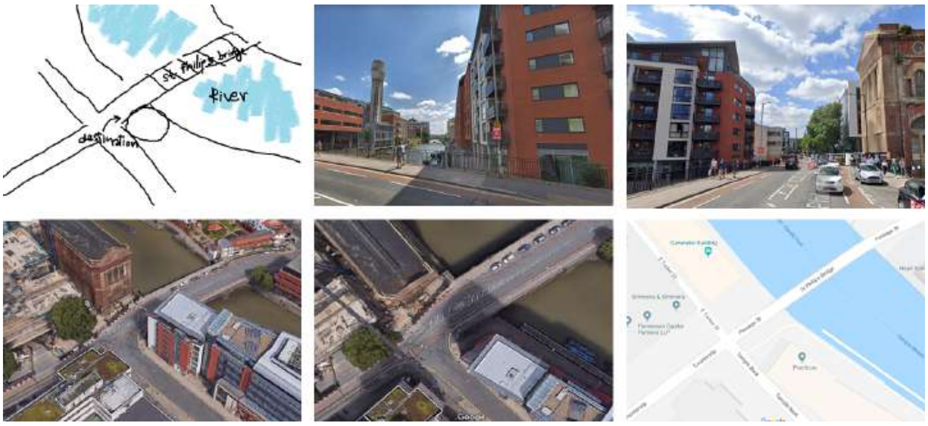


Figure 4.8: Examples of a scene captured at the same location, but with differences in appearance, viewpoint, and domain. This demonstrates the major problem of cross-view localisation, which is to find the similarity between these images.

In computer vision, cross-view localisation approaches are grouped by types of databases as (i) 3-D databases and (ii) 2-D or 2.5-D databases. The former is based on 2-D-to-3-D matching. Given a query 2-D image, the systems return its pose estimation on the corresponding 3-D point cloud maps. For instance, the system in [227] made use of 2-D map

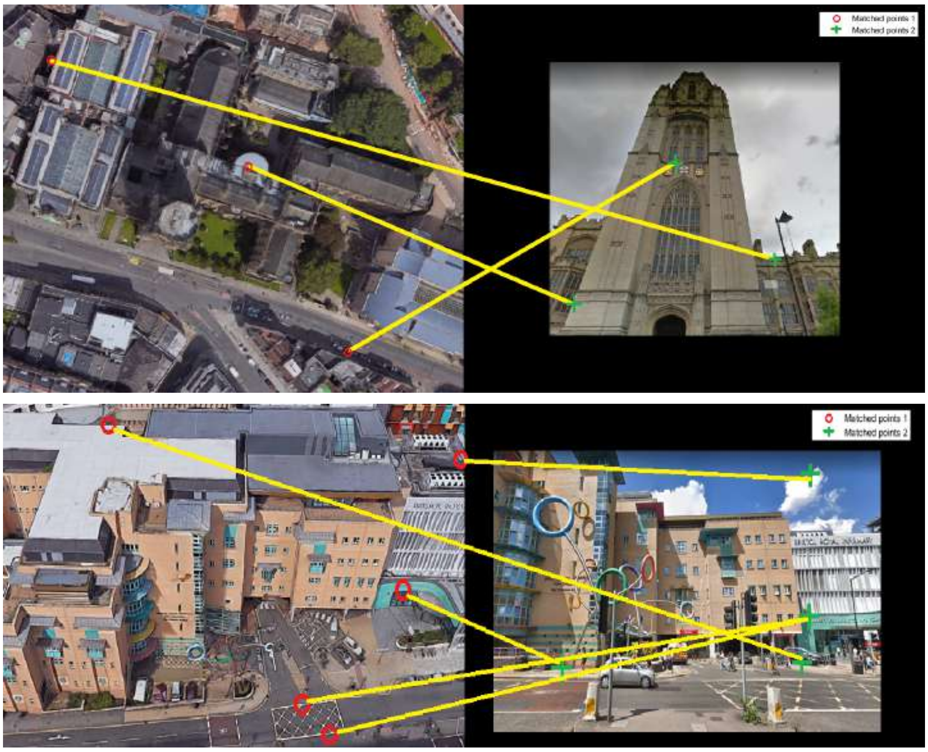


Figure 4.9: An example of cross-view matching between 2-D aerial view and street-level image using SURF features. This illustrates the difficulty of solving this problem using feature-based approaches, which makes the problem highly challenging.

planes retrieved from Ordnance Survey MasterMap[5] as the prior information for 3-D graph-based SLAM. For the latter, the task is to match between the 2-D image and another 2-D or 2.5-D database, which is stored in a different domain or with a drastic change in viewpoint; this approach is sometimes called ultra-wide baseline matching due to those characteristics. As this is closer to our interest, it is discussed in greater detail in the next section.

In practice, the cross-view matching approaches are usually paired with autonomous aircrafts, such as micro aerial vehicles (MAV) and unmanned aerial vehicles (UAV), as depicted in Figure 4.10 for air-ground image matching from a household level to military operations [36]. Another usage for location-based cross-view matching is to search for a scene across the world to find the closest geo-location [228], which demonstrates the potential of using the cross-view approach for large-scale location searching.

4.3.2 Ultra-wide baseline matching

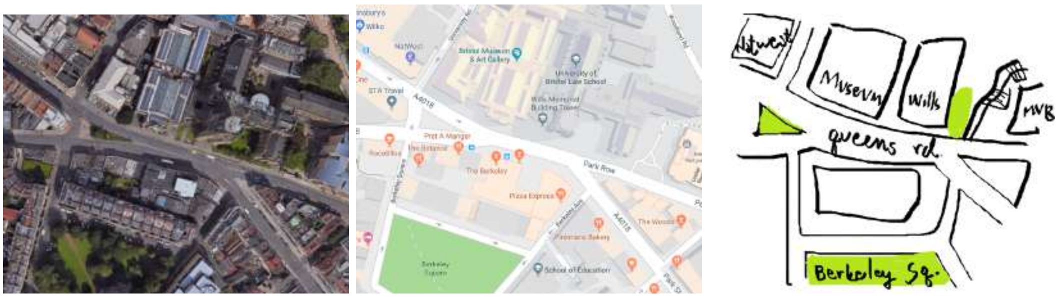
In this work, we focus on imitating human map-reading activity, which is the process of orienting oneself by matching visual information with a 2-D physical map. This description

[5] https://www.ordnancesurvey.co.uk/

(a) (b)

Figure 4.10: Examples of autonomous aircraft: (a) micro aerial vehicle (MAV) and (b) unmanned aerial vehicle (UAV). As their operations are aerial-based, cross-view localisation is needed for air-ground matching. Images were retrieved from [30, 31].

fits the cross-view localisation problem, as the visual information and the 2-D map are in different domains and viewpoints. Figure 4.11 illustrates the 2-D maps generally used in location-based applications: orthophoto map, base map, and sketch directional map. These are discussed in detail in this section.



(a) (b) (c)

Figure 4.11: Examples of GIS map discussed in this work: (a) an orthophoto or satellite map, (b) a base map or 2-D vector map, and (c) a sketch directional map. These are the 2-D maps that commonly appear in location-based applications. Each contains a different level of visual and semantic information.

Orthophoto map

An orthophoto map (Figure 4.11a) is a raster image map representing geo-data taken by a satellite. Applications of orthophoto providers include Google Earth[6] and USGS Earth Explorer[7]. The standard method for cross-view localisation using the orthophoto map is to use learning-based techniques, such as a Siamese model [101], to identify the similarity between street-level images and the orthophoto patches, as in

[6] https://earth.google.com/

[7] https://earthexplorer.usgs.gov/

[229, 230, 231]. Aside from the learning-based models, the alternative is to transform the domain of one view to another view, as in [232], which proposed a method of converting a query panoramic image into a top-view image and used that to match with the aerial view. However, these stated methods only use CNN to pair aerial-ground images; they do not interpret any semantic information from images or maps. Another group of works [233, 234, 235, 236] used 2-D map data to aid the estimation of a 6-DoF camera pose using a combination of positioning systems and images; building edges and planar façades extracted from images were utilised to align with 2-D and 2.5-D maps and were geo-localised using GPS. This gave an improved estimation of camera position and orientation. However, these techniques focused on obtaining precise metric pose estimates for applications. For example, in [234], they relied on the 2.5-D map to integrate the system with augmented reality, and there was a reliance on having clear views of building façades. Therefore, with different purposes, these approaches would be difficult to extend to general localisation.

Base map

The base map (Figure 4.11b) is a general term used in GIS applications. This map contains records of geographic data such as road lanes, buildings or properties, site addresses, uses, and restriction. The base map is usually constructed in the form of vector graphics. Similar to the semantic approaches discussed in Chapter 2, the basic operations of this type of map are segmentation and classification. For instance, in [237], the street-level images were semantically segmented and matched with the geo-information retrieved from the 2-D base map; they ran experiments using a combination of features and those with the highest performance were buildings, lampposts, trees, and traffic signs. Various works [238, 239, 240, 241] have used deep-learning models to recognise semantic features in images, such as junctions, numbers of lanes, and sun direction. These stated works applied deep-learning networks for annotating images using the semantic data retrieved from the 2-D map. The outputs were used to validate GPS map locations for self-driving car applications. However, most of these works only used the semantic information to narrow down the likely areas of the given image query; they did not directly use semantic approaches for localisation and navigation tasks. As these applications are similar to our focus, they are discussed in greater detail in Section 4.4.

Sketch directional map

> Compared to the others, a sketch directional map (Figure 4.11c) contains a coarser level of information, which is similar to the way humans provide details of directions. To make it more practical, some additional mechanisms are required to convert a sketch image into a digital image. Numerous works [72, 153, 242] have proposed techniques to segment a drawn map and convert these segments into a digital map. However, these concepts are still under development and need further investigation to yield full use of autonomy for navigation. A small number of robotic applications have made use of the directional map. The schematic map [158], which is a structural representation encoded with spatial knowledge of the environment, was created from this concept for guiding robot navigation. The schematic map is defined as the abstraction level between the sketch map (cognitive representation) and topographic map (spatial representation). Followed by [159, 243], these applications have made use of extracted information for robot navigation using human guidance. However, the stated techniques still require partial human guidance in their operations, which reduces the degree of autonomy.

Table 4.1 summarises the key elements of three types of 2-D maps. Among the three, based on our focus, we chose 2-D base maps combined with semantic approaches because they are more similar to human activity. It is simpler to extract the semantic features directly from a base map, unlike the orthophoto map or sketch directional map, which require additional information extraction processes. The number of systems in the image to-2-D map localisation is limited compared to the image-to-image-based methods. However, this field of study is gradually expanding.

4.4 Discussion

In sum, although vision-based localisation using image-to-image matching systems has demonstrated impressive performance, it is still limited in three key respects: scalability, invariance, and human-system interaction. For humans, when they are in places that they have never been before, they usually rely on the 2-D map and orient themselves using visual cues to match with semantic information provided in the map (localisation). Once they know their current location, humans use that information to navigate themselves from one place to another. The process of navigation is considered to be a large-scale spatial task; one of the well-known established models in the spatial knowledge field is the landmark route and survey model (LRS). The mechanism is that humans first identify landmarks in their surroundings; route knowledge is formed using the aligned landmark knowledge

Table 4.1: Summary of semantic levels and limitations of three different 2-D maps: the orthophoto map, the base map, and the sketch map.

Method	Semantic level	Limitation
Orthophoto map	- rich visual information - closer to human perception	- difficult to extract the semantic information - need large storage
Base map	- contain a variety of semantic information - more ready to use in digital-based - less data storage	- accuracy and coverage depends on the map provider
Sketch map	- closer to the human cognitive map - provide distinct landmarks (in human perception level)	- need to be pre-processed in digital format - contain some distortion of information

gathered while they are moving in the environment. Once humans become familiar with the environment, survey knowledge is formed to gain the full capability to traverse in the area. This knowledge is usually projected in the form of a 2-D map, which is most familiar to humans.

Motivated by this, we consider an alternative to vision-based localisation using *image-to-2-D-map matching*. In other words, we link images to semantic features on a 2-D map of an environment to provide localisation; this is similar to how humans relate the visual appearance of their surroundings to the semantic information they perceive on a map. More importantly, the 2-D map itself encodes with the sense of spatial knowledge, which is the main focus of this thesis. Furthermore, we believe that this is better suited to human-system interaction. As discussed in Chapter 2, the use of semantic descriptions gains some advantages over invariance, and the compact representations also offer the potential for scalability, as our semantic descriptors are many orders of magnitude smaller than images or sets of image features. Therefore, we present preliminary investigations into the approach by seeking a method that contains the following properties:

- Locations are characterised by a small number of semantic features and form a compact representation
- The stated representation appears in both the image and 2-D map
- The stated representation is highly distinctive in the degree that localisation is possible

To achieve this, we characterised locations by a small number of simple semantic features relating to road junctions and buildings. Each location is represented by a 4-bit binary semantic descriptor (BSD), with each bit indicating the presence or lack of a given feature in a given viewing direction. In addition to the advantage over invariance, using BSD provides a highly compact representation, which also helps to increase scalability. To recognise the features in the images, we designed classifiers that allowed us to estimate the descriptors. We performed localisation through comparison with a database of location-tagged descriptors derived directly from the 2-D map. On their own, these descriptors are not distinctive enough, but once they are concatenated sequentially over routes as the route descriptors, they become highly distinctive. This is to the extent that localisation is possible even with imperfect classifiers; the pattern of semantic features observed along a route becomes unique when the route is sufficiently long. Moreover, when the direction of travel between locations along a route is also taken into account, such as left and right turns, the performance is further improved.

In Chapters 5 and 6, we present an implementation using Google Street View and OpenStreetMap data, with the latter providing vector maps and the former giving 360-degree images at regular locations along roads. In this work, we used road junctions and gaps between buildings as our semantic features, assuming the former to be present or not in front- and back-facing views, and the latter to be present or not in left- and right-facing views. This gave us 4-bit descriptors for each location. We could extract these features directly from OpenStreetMap, but, for the image, we trained the CNN classifiers to recognise them.

The closest work to that presented in this thesis is [240]. They applied a CNN approach to recognise semantic features in images such as junctions, numbers of lanes, bike lanes, and one-way versus two-way. Their network training was based on labels obtained from OpenStreetMap and images from Google Street View, which is similar to our approach. However, the purpose of their classifier outputs was to validate GPS map locations for self-driving car applications, rather than for general localisation. Another similar application is [238]. In that work, locations were considered in isolation, in contrast to our use of route information. The concept of route has been used in a number of localisation techniques based on map matching, as in [239, 244], where visual odometry based on feature matching was used to generate route trajectories; these were then matched with a base map based on road patterns. Similar techniques have been used to localise noisy GPS data, as detailed in [245].

The extended version of [238] was provided in [241] by including semantic features extracted from forward-facing images, including sun direction, road type, and junction presence, and integrating with vehicle speed and odometry to estimate the location and heading corresponding to the 2-D map. They trained classifiers for road types and the presence of junctions using image labels derived from OpenStreetMap. This work has clear similarities with ours, but differs in that the semantic features derived from the map, such as road types and junctions, were used for street identification to narrow down the set of possibilities; this contrasts with our descriptor-based approach, in which the presence of semantic features is used to encode specific locations and use the odometry (in this case, turn patterns) sensing from the system to constrain the map matching. This likely reflects the differences in the application.

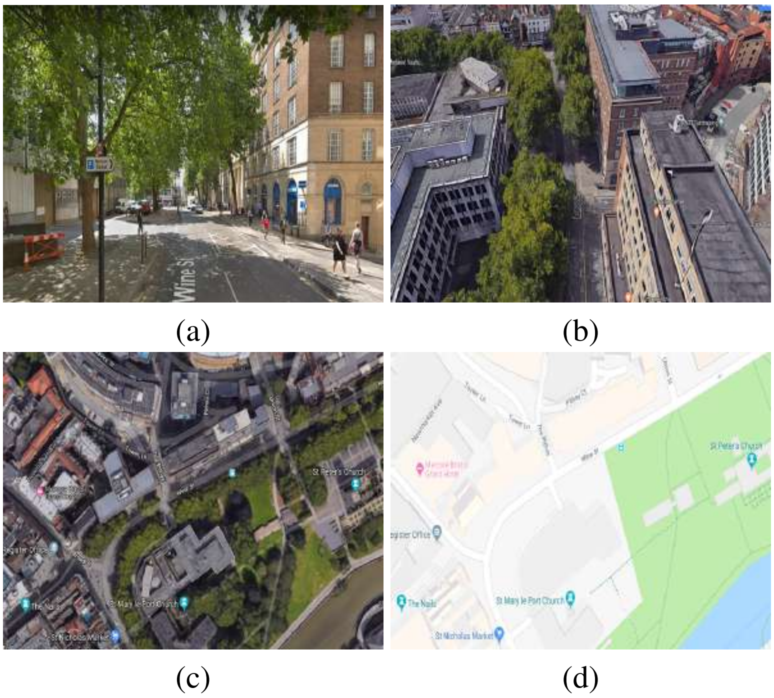
Compared to the stated applications, we are more interested in further adapting our work for slow-moving pedestrians or robots, i.e., for more human-friendly applications. As opposed to with moving road vehicles, semantic descriptors directly related to map locations are a natural choice. This also aligns with using representations that better reflect human map reading. Moreover, this makes comparing the two methods difficult, not least because the results presented in [238, 241] were obtained using a front-facing camera, whereas we require 360-degree views at each location. However, our later experiments demonstrate consistency with their methods. More details are provided in Chapters 5 and 6.

Chapter 5

Localisation in 2-D maps using binary semantic descriptors

In the previous chapter, we addressed the three key limitations of image-to-image based matching: invariance, scalability, and human-system interaction. To overcome these issues, we consider the alternative using *image-to-2-D-map matching* to link images to semantic features on a 2-D map of an environment to provide localisation. This is similar to human map reading, as a person relates the surrounding visual appearance of an environment to the semantic information they can perceive on a map, such as buildings and road layout. However, to match between images and the 2-D map is not as straightforward as it is for humans. Figure 5.1 illustrates the difficulty of this problem, including the drastic viewpoint changes (aerial-ground comparison) and domain changes (an image and a 2-D vector). As demonstrated in Chapters 2 and 3, the use of semantic information increases degrees of invariance and scalability. Applied to the use of the 2-D map, our image-to-2-D-map matching scheme can improve the human-system interaction. Given a situation in which people walk down the street with their robot, when they discuss the direction, it is more natural to use the terms (semantic information) and sources (the 2-D map) with which humans feel more familiar. As mentioned in the previous chapter, several proposals [239, 240, 241] have introduced methods to face this 2-D map localisation using semantic information. However, the outputs were generally used for validating GPS map locations for self-driving car applications.

Motivated by this, we introduce a novel approach to image-based localisation in urban environments using semantic matching between images and a 2-D map. Figure 5.2 illustrates the overall process of our system. Our central idea is to characterise locations using a small number of simple semantic features relating to road junctions and buildings and represent each location using a BSD, with each bit indicating the presence or lack of a given feature in a given viewing direction. By using highly compact binary descriptors



(a) (b)

(c) (d)

Figure 5.1: The difficulty of image-to-2-D-map matching problems; (a) an image taken from the location pictured in the 2-D maps from (b) and (c) – the orthophoto map in a different level of changes, and (d) the 2-D vector base map. In (b), we can still see some semantic objects related to the objects in (a), as well as in (c) with some partial information loss. However, in (d), all objects in the scene have been projected in the vector shape form. This makes the match more difficult and adds more challenges to the problem.

to represent spatial semantic features at locations, our method significantly increases scalability compared to existing methods and has the potential for greater invariance to various imaging conditions.

In Section 5.1, we discuss the properties of semantic representations required for our method. In Section 5.2, we apply the selected representation and construct a novel binary descriptor. In Sections 5.3, 5.4 and 5.5, we explain the processes of constructing the binary descriptors as route descriptors incorporating with turn patterns for localisation and optimisation. In Section 5.6, we demonstrate localisation using estimated binary descriptors. Finally, in Section 5.7, we summarise the overall ideas of this chapter. Note that some of the arguments, figures, and results described in Chapters 5 and 6 have been published in [246]. In addition, a video demonstrating the process of vision-based localisation using BSDs can be found at *www.youtube.com/watch?v=fwZWrWXCRw*.

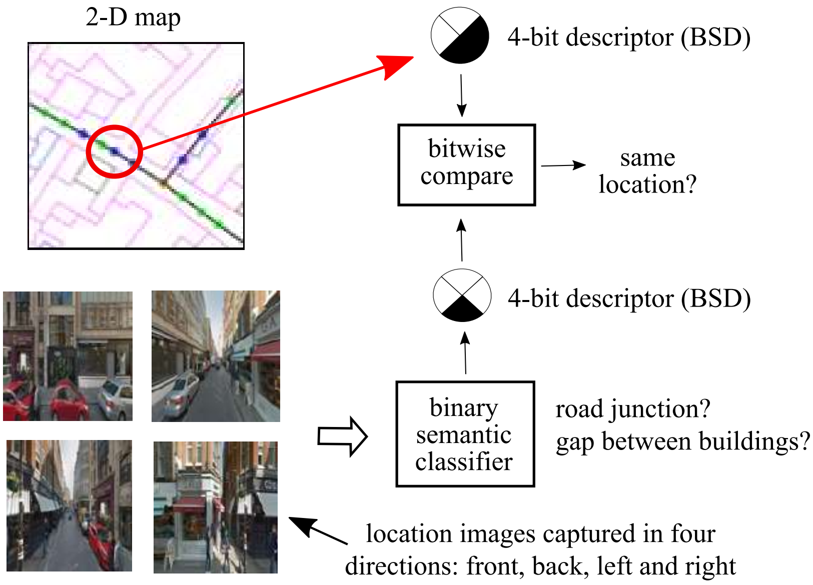


Figure 5.2: A conceptualisation of BSDs. Four-bit binary descriptors are used to represent locations indicating the presence or lack of semantic features in four directions (front and back facing – junctions; left and right facing – gaps between buildings). These were derived from a 2-D map and compared bitwise with descriptors estimated via classifiers from images captured in the same direction to establish localisation corresponding to the map. Using 4-bit descriptors gives a highly compact representation, thus increasing scalability.

5.1 Semantic Representations

Having robust place representations is the key to localisation using visual information. The term ‘robust’ in vision-based localisation refers to the ability to localise to the degree that, if the system revisits a place under different conditions, it can distinguish the place from others. In Chapter 3, to recognise a place, we made use of salient regions combined with spatial knowledge to represent a scene. The findings demonstrated the impact of using semantic information. However, unlike the previous work, our sources here are in different domains. We cannot apply either feature-based techniques or LDDs. Therefore, to automatically identify where the images were taken on a 2-D map, a robust representation for both images and 2-D maps is required.

Figure 5.3 illustrates the different levels of perception between humans and autonomous systems. For humans, to compare between street-level images and the 2-D map is a map-reading task and is a relatively straightforward process, albeit with varying degrees of difficulty. However, automating the process contains some complexity. This contrasts with how an autonomous system projects the given visual information. To solve this, it requires sophisticated analysis, such as reasoning about the scene. As illustrated in Figure 5.4,

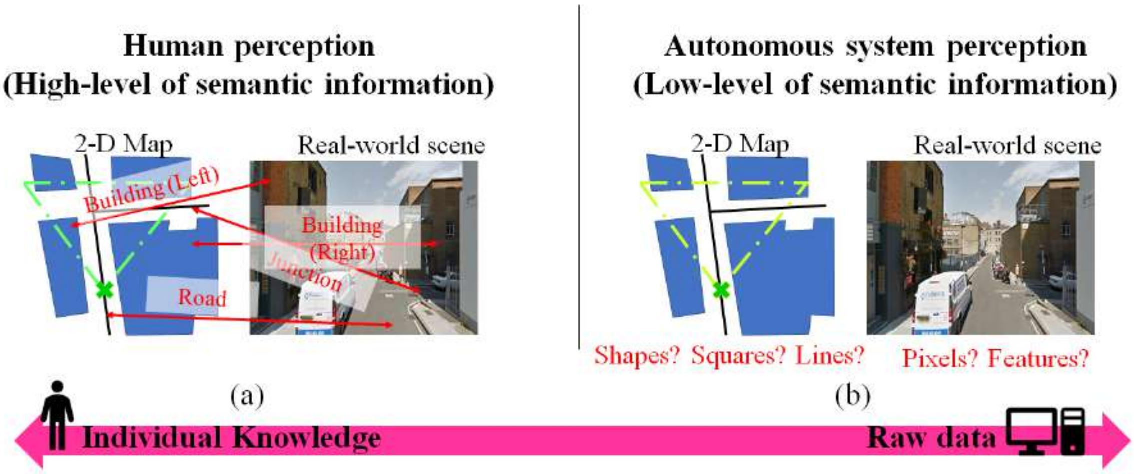


Figure 5.3: Conceptualisation of (a) human 'map-reading' ability, which makes use of the rational ability to interpret and compare objects in the scene, and (b) how an autonomous system treats a 2-D map and its corresponding image. This leads to the following question: given street images, can the system automatically identify where the images are taken on a 2-D map?

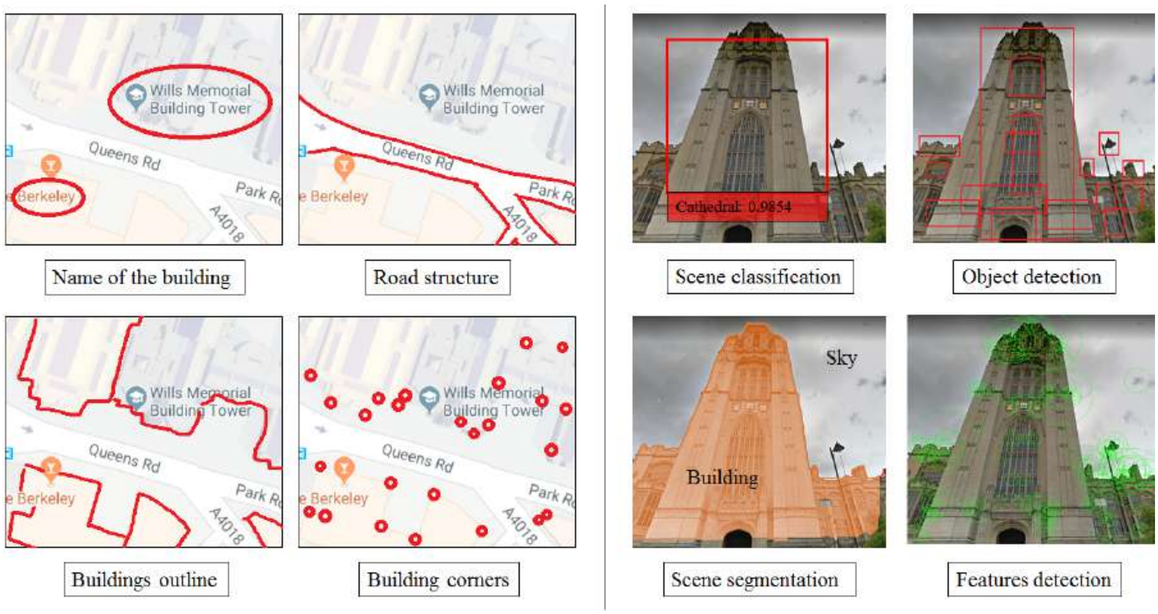
both the image and the 2-D map contain a number of options for semantic interpretation. Therefore, in this section, we discuss some constraints used for selecting the semantic representations. We focus on five issues: availability, reliability, clarity, ubiquity, distinction, and locality.

Availability

This term refers to the presence of semantic features in both sources, an image and a 2-D map. For example, the name of the building or an address might be useful semantic information retrieved from a 2-D map, but it cannot be obtained directly from the image (excluding the use of meta-data). This is similar to the lower-level semantic information of an image, such as shape, colour, or texture; they are clearly shown on the image, but nearly impossible to obtain from a 2-D map. We label this property as our major concern. In addition, as we use the 2-D map, rather than the orthophoto map (as shown in Figure 5.5), the projected information is generally more limited. For example, a scene contains a line of trees in the background, but this information might not appear on the corresponding 2-D base map.

Reliability

Reliability, in this work, mainly relates to data. First, as we have obtained ready-to-use digital 2-D maps, they contain a degree of incomplete and unreliable data. Therefore, the features should be robust enough so that they are least affected by



Name of the building

Road structure

Buildings outline

Building corners

Scene classification

Object detection

Scene segmentation

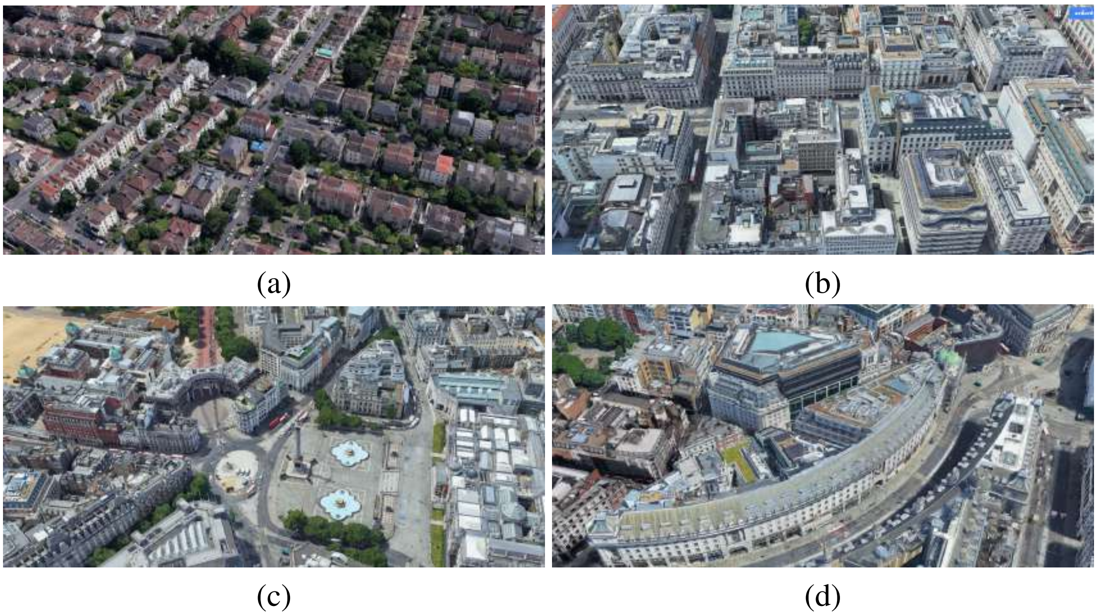
Features detection

Figure 5.4: Examples of semantic information perceived from a 2-D map and an image scene. As they are in the different domains, the choices of information we can obtain are also different. Our aim is to find the most suitable representation that can link images to a 2-D map.

unreliability. Second, an image is sensitive to environmental changes, as discussed in Chapter 2. Thus, the selected semantic features should be tolerant to the unreliable characteristic of both 2-D maps and images. For example, if we select trees and vegetation as objects of interest, on the 2-D map, we might not see any seasonal changes. This contrasts with how these objects are visualised on the image.

Clarity

This term is, in some sense, similar to reliability, but clarity is more concerned with specifying the solid threshold to distinguish the features from others. For example, if we use the density of buildings as our feature, it is possible to set a threshold using numbers of buildings; for example, a scene with less than four buildings could be labelled as 'not dense', and vice versa. However, this number does not provide enough clarification of the data; we may have three buildings packed in the scene that seem denser than four buildings spread all over the scene. Figures 5.5a and 5.5b were taken from areas of a similar size and seem to be cluttered, but the numbers of buildings appearing in the scenes are different. To avoid this, we consider using binary semantic information to indicate the presence of features in the scene.



(a)

(b)

(c)

(d)

Figure 5.5: The orthophoto map retrieved from various areas. The purpose is to illustrate characteristics of layouts in the city of London: (a) a residential area and (b) a city area, in which the road structures are well defined and there is no appearance of highly distinct landmarks. In contrast, (c) and (d) illustrate city areas with the presence of distinct landmarks and road layouts. Therefore, how can we find the common representation for these areas? Note that by projecting these maps into 2-D base maps (or vector maps), there is less retrieved information, which makes this problem more challenging.

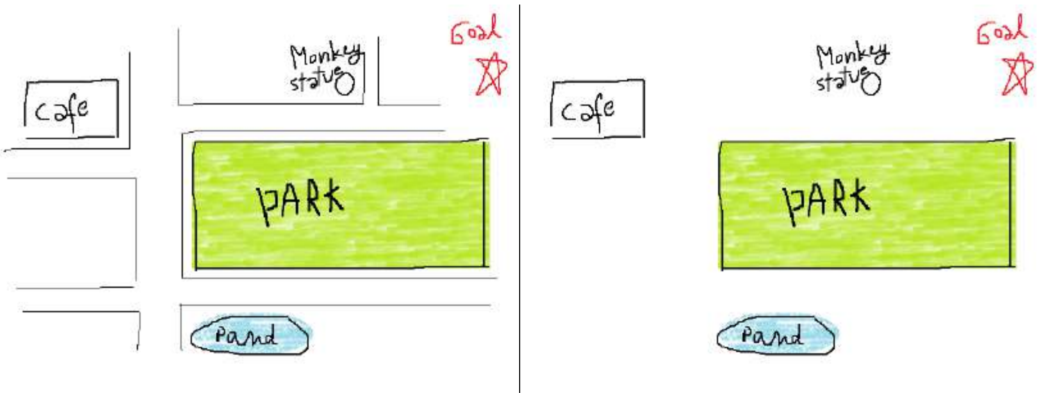
Generalisation

Generalisation in this work refers to the balance between ubiquity and distinction. The proper representation of a scene should be general enough to be found everywhere in the area; however, at the same time, it should not be too general without any distinction from others. For example, given that trees can be found all over, as illustrated in Figure 5.5a, they contain a low level of distinction, as they are too general to be used. Although the vegetation can be separated by their types or families, this information is too specific and requires more specific 2-D maps, special knowledge, and experts. In the other aspect, given the Eiffel Tower, the object contains high degrees of distinction. However, the object is too distinct, so it cannot be found at other locations. Therefore, the selected features should be balanced between distinct and ubiquitous characteristics.

Locality

In this work, locality refers to a sense of features having a known location. For instance, if we look at a line on a 2-D map, it is difficult to find its locality, as they are everywhere, with no sense of start and end. This is similar to image pixel or points. We can extract the colour information from an image, but the feature contains no sense of locality. Based on this, the low level of semantic information is not suitable.

In the end, we chose the presence of junctions and gaps between buildings as our representations. Both features fulfil all previously stated conditions. First, we can retrieve them from both sources. They are appearance-based properties that can be observed or detected. Second, regarding reliability, both features are included in the main infrastructure and are hardly affected by environmental changes. Moreover, as parts of the infrastructure, both features contain some degree of ubiquity. At the same time, these properties are sufficiently distinguishable. Finally, they contain permanent positions that give a sense of locality. These features demonstrated effectiveness for urban localisation both in our experiments and works described in [240, 241]. To emphasise this, Figure 5.6 depicts a sketch map, which is a type of map that people usually rely on for giving directions. Comparing between having and not having road segments, the former increases the spatial sense of the area. Hence, we can infer the importance of road structure for location-based activities.



(a) (b)

Figure 5.6: An example of a human-generated sketch map for giving directions: (a) with road segments (normal version) and (b) without road segments. By removing the road segments, spatial knowledge of the scene is discarded. This reveals the importance of the road structure in the scene.

5.2 Binary semantic descriptors

Having the semantic features for representing both images and 2-D maps, we constructed a binary description for four directions: front, back, left, and right. Therefore, we denote the finite set of locations in an area of interest by

$$\mathcal{L} = \{l_1, l_2, \ldots, l_N\} \tag{5.1}$$

where N is the total number of locations in the area. Associated with each location l_i is a BSD that is in the form of the binary string d_i. We define the set of all descriptors within the area $\mathcal{L}$ as

$$\mathcal{D} = \{d_1, d_2, \ldots, d_N\} \tag{5.2}$$

Each BSD was constructed in terms of d_{ij}. The j^{th} bit corresponded to the number of directions from which we extracted semantic information; in particular, in this work, we set $j \in \{1, 2, 3, 4\}$ and each bit of a BSD d_{ij} denoting the presence of junctions or gaps between buildings in one of four viewing directions centred on location i. These are derived from the vector map as follows:

$$d_{ij} = \begin{cases} JUNC(V_{ij}) & \text{if } j \in \{1, 2\} \\ BGAP(V_{ij}) & \text{if } j \in \{3, 4\} \end{cases} \tag{5.3}$$

where (V_{i1},V_{i2}) and (V_{i3},V_{i4}) denote the (front, back) and (left, right) viewing directions at location i, respectively. The functions $JUNC(V_{ij})$ and $BGAP(V_{ij})$ return 1 if there exists a junction or a gap between buildings, respectively, in direction V_{ij}, and 0 otherwise. For instance, given a location l_i with one junction at the front and no gaps between buildings for both viewing directions, the corresponding BSD d_i are '1000'.

Figure 5.7 depicts the construction of a BSD; the circular discs represent the BSDs, while the black and white segments indicate individual bits. A feature is counted as present in a viewing direction if it is within the relevant quadrant of a given area centred on the location of interest, where the front and back viewing directions are aligned with the road location. In the experiments, we set the viewing radius to 30 metres, similar to [240]. By making use of fixed directions (front, back, left, and right), we ensured the spatial relationship between detected semantic features in a scene.

5.3 Route descriptors and turn patterns

Because of their simplicity, on their own, the binary descriptors are not sufficiently discriminative to identify a location uniquely and allow for localisation. This is true even if we are able to design perfect classifiers for extracting the descriptors. For example, given an area $\mathcal{L}$ with the total numbers of locations $N = 1000$ and a 4-bit BSD representing each location, having only $2^4 = 16$ possible BSD patterns means 63 locations may have the same BSD pattern. Note that this is the case in which all patterns are equally distributed, which is not always true. To make the BSD more robust, we used a technique similar to the probabilistic approaches (such as the Kalman filter [217, 218] and particle filter [221])

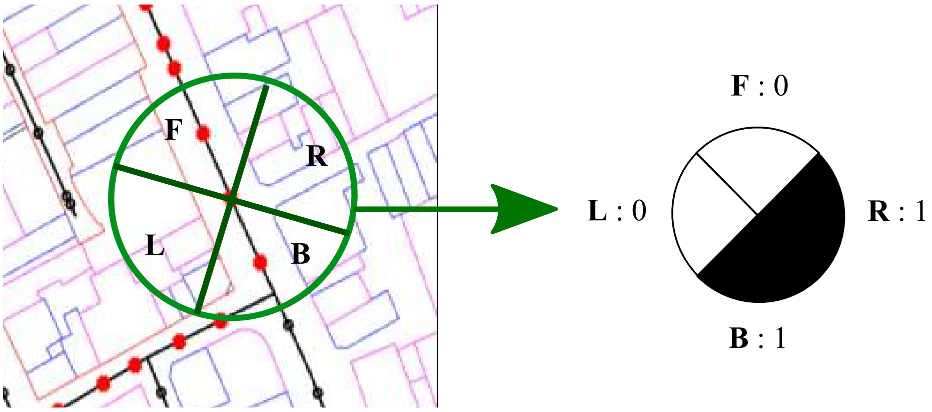


Figure 5.7: The illustration of a BSD generating from the vector map. We set the area of interest and partitioned them into 4 sides (front, back, left, and right) and extracted the semantic information that appeared within each section. In this work, the presence of junctions (front and back) and the presence of gaps between buildings (left and right) were applied for constructing the BSDs.

reviewed in Section 4.2.2, which work under the concept of sequentially predicting and updating the likelihood of the current location using the previous data. In other words, the more system moves for gathering data, the more confident they are. In practice, the concept of probabilistic methods has been used in indoor tracking and navigation applications, such as in [247, 248, 249]. Motivated by this, we addressed the problem in two ways: route descriptors and turn patterns. This is described in greater detail as follows.

Route descriptors

For locations in the area $\mathcal{L}$, with N road locations, we constructed a route from connecting adjacent locations. As each location is represented by a BSD, we obtained the *route descriptors* by stacking the BSDs, which yielded high discrimination once the routes reached a certain length. Therefore, as we used a 4-bit representation, each route descriptor was of length $4N_r$ bits, where N_r is the number of locations in the route. Figure 5.8 illustrates a route descriptor at $N_r = 5$ formed by stacking five 4-bit BSDs.

Thus, let A be an $N \times N$ adjacency matrix, such that $A_{ab} = 1$ if locations l_a and l_b are adjacent, and $A_{ab} = 0$, otherwise. Locations are regarded as adjacent if on the 2-D map they are connected by a road and there are no other locations between them. A *route* is then defined as a finite sequence of adjacent locations, i.e., the route of length N_r is written as

$$r = (l_{\gamma(1)}, l_{\gamma(2)}, \dots, l_{\gamma(N_r)}) \tag{5.4}$$

where $\gamma(a)$ defines a sequence of adjacent locations such that $A_{\gamma(a)\gamma(a+1)} = 1$, and $\forall\ 1 \leq a < N_r$. Note that for simplicity, we restricted ourselves to routes that did not loop or turn back on themselves, i.e., $\gamma(a) \neq \gamma(a)$, $\forall\ a \neq b$, $1 \leq a, b \leq N_r$. However, the method can be readily extended to deal with these issues.

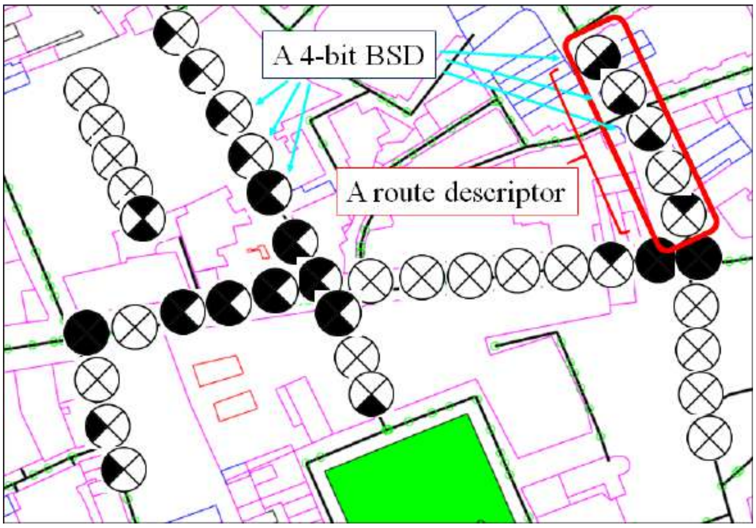


Figure 5.8: The conceptualisation of using BSD to construct a route descriptor. Within the area, we converted every road location to 4-bit BSDs. The given example illustrates the route descriptor with a length of 5 (N_r = 5).

Each route is a *route descriptor* which consists of the sequence of the BSDs corresponding to the locations along the route. In other words,

$$s = (d_{\gamma(1)}, d_{\gamma(2)}, \ldots, d_{\gamma(M)}) \tag{5.5}$$

Within the area $\mathcal{L}$, we define $\mathcal{R}_M$ as the set of all such routes up to the maximum length M defined amongst all locations in $\mathcal{L}$, and $1 \leq M \leq N$. Note that we set the maximum length of M as a constraint to limit the size of the database. Hence, $\mathcal{S}_M$ was defined as the set of all route descriptors corresponding to the routes in $\mathcal{R}_M$. A database of location-tagged route descriptors $\mathcal{S}_M$ is created by computing all possible routes within the area of interest up to a certain length in terms of the number of adjacent locations and concatenating the set of associated BSDs. Having this, localisation could proceed by matching the test route with a database of all possible route descriptors constructed offline.

Although the number of possible routes can be quite large, the route database has a small memory footprint. For example, in the following experiments, for an area of approximately a 2 km^2 range, the number of routes containing 40 locations was just under 40×10^6. Note that this is approximately equal to 400 metres long represented by a 160-bit route descriptor. The route descriptor database was then around 800 MB in raw form (before any compression), which would be made possible by significant overlap between routes. This contrasts with the use of the image-to-image database matching discussed in Section 2.3. Given the same example, in [55], a single 400-metre route required 71 MB space to store image features. Compared to our 160 bits per 400-metre route, we saved space by 3.5×10^6 times, which makes our representation superior in terms of handling scalability.

Turn patterns

To make the route descriptors more robust, we incorporated them with the turn patterns, which are the stamps of directional changes. This idea is consistent with [238, 241]; they worked on the odometry of the vehicle combined with semantic features such as sun direction, road type, and presence of junctions, the latter being similar to our BSDs. This add-on is also based on the fact that autonomous systems usually record their directional changes. Therefore, we incorporated the route descriptors with the direction of travel between locations or turn patterns observed along a route into a sequence of no turn, turn left, and turn right at each location. We used these to identify the most likely match within the database. To incorporate turn information into the representation, we define a turn pattern t associated with a route r at the location N_r as

$$t = (t_{\gamma(1)}, t_{\gamma(2)}, \ldots, t_{\gamma(N_r-1)}) \tag{5.6}$$

Each i^{th} bit of t indicates whether no turn, turn left, and turn right actions are presented between locations $l_{\gamma(i)}$ and $l_{\gamma(i+1)}$. In other words:

$$t_{\gamma(i)} = TURN(V_{\gamma(i)}, V_{\gamma(i+1)}) \tag{5.7}$$

, where $V_{\gamma(i)}$ denotes the 2-D unit vector of front-facing direction contains the degree of angle $\theta_{\gamma(i)}$ at location $l_{\gamma(i)}$, and

$$TURN(V_i, V_j) = \begin{cases} 2 & \text{if } \mathit{TDIR}(V_i, V_j) \geq +\tau \\ 1 & \text{if } \mathit{TDIR}(V_i, V_j) \leq -\tau \\ 0 & \text{otherwise} \end{cases} \tag{5.8}$$

where *TDIR* is the function giving a normalised turning angle between $V_{\gamma(i)}$ and $V_{\gamma(i+1)}$ (or moving from $l_{\gamma(i)}$ to $l_{\gamma(i+1)}$) resulting from

$$\mathit{TDIR}(V_i, V_j) = atan2(V_j.y, V_j.x) - atan2(V_i.y, V_i.x) \tag{5.9}$$

, and τ is an angle threshold, which we set to be 60° to ensure that we only included significant turns. The results of 1 and 2 represent the right and left turning directions, respectively. Therefore, t represents the sequence of turns that take place along a route. We define $\mathcal{T}_M$ to be the set of such turn patterns corresponding to the routes in $\mathcal{R}_M$. Using the example given in Figure 5.8, for a given a route descriptor at $N_r = 5$, the turn patterns should be a set of four consecutive zeros or a straight road pattern with no turn for five consecutive locations.

5.4 Localisation and bootstrapping

Once the route patterns are incorporated with the turn, we use them to perform the localisation. Given a situation that an autonomous system makes its way through an urban environment, moving between locations in an area $\mathcal{L}$ along a specific route of length $\leq M$ (maximum locations). At any given location N_r, our goal is to identify its current location by recognising the route taken to date, consisting of the current location plus the previous $N_r - 1$ locations. As our purpose is to find the current location, we can apply a simple search technique. Note that the searching process presented in this section is a simple nearest neighbour method. The advanced version is further discussed in Section 5.5.

Localisation

At each location, we first concatenated the estimated BSDs to obtain the route descriptor $\hat{s}$ and compared it with those in $\mathcal{S}_{N_r}$ and its turn pattern $\hat{t}$ with those in $\mathcal{T}_{N_r}$. We obtained the most likely route from those in $\mathcal{R}_{N_r}$. It is important to note that, in the main experiment, we assume a one-to-one correspondence between the locations in our 2-D map and the locations in the environment. This enables us to perform a direct comparison between the estimated route descriptors and those in the database. We define the most likely route $r^* \in \mathcal{R}_{N_r}$ as the route for which the route descriptor s is closest to $\hat{s}$ and for which the turn pattern t^* matches $\hat{t}$, such that

$$s^* = \underset{s \in \mathcal{S}_{N_r}}{\arg\min}\, DIST(s, \hat{s}) \tag{5.10}$$

and

$$DIST(t^*, \hat{t}) = 0 \tag{5.11}$$

where $DIST(x, y)$ denotes the Hamming distance between two binary strings x and y. This concept is similar to our enforcement of the spatial distribution of semantic features discussed in Chapter 3. However, for long routes, the number of elements in

$\mathcal{S}_{N_r}$ becomes quite large. Further details regarding the optimisation are discussed in the next section.

Given this, we assume that the turn pattern for the query route is correct, but we allow errors in estimating the route descriptor. Our motivation for the former is that, in practice, detecting significant left or right turns using an autonomous system can be achieved reliably; thus, requiring an exact match is reasonable. Note that, as we later demonstrate, turn patterns alone take a long run to achieve localisation, and it is their combination with route descriptors that provides the greater level of distinctiveness. More details are discussed in Chapter 6.

Bootstrapping

As the localisation process only provided us with an indication of the most likely location given the current route, it does not indicate the confidence in the estimate, i.e., during a test, we can check the likely returned route s^* against the ground truth. However, in practice, we cannot rely on that information. To ensure the correctness of an estimation, more mechanisms are required. There are several possibilities for this, including basing it on the distance between s^* and $\hat{s}$ and the distance of s^* from the second-best matching route descriptor. We found that a consistency metric yields the greatest effectiveness. Therefore, a route is localised if there is sufficient overlap between the most likely routes r^* for a number of successive locations. In this work, we set the overlap to 80% of the locations being the same, and we required this to occur for five successive locations. In essence, if successive query routes are matched with routes that have significant overlap, this indicates that successful localisation has been achieved.

We additionally demonstrate that, once the above consistency criterion is met, the query route length can be fixed and localisation proceeded by successively updating the query route by appending the latest BSD onto the end and removing the first descriptor. Therefore, the phase of the query route growth is regarded as a *bootstrapping process*; from this, the route descriptor continues extending until it becomes sufficiently distinct to allow for localisation. Once achieved, *continuous tracking* can then take place using the fixed-length query at the same rate as the BSDs created at successive locations. An example of bootstrapping and tracking can be found at *www.youtube.com/watch?v=fwZWr_WXCRw* and further discussed in Chapter 6.

5.5 Search Optimisation

In the process of localisation and bootstrapping, if the routes keep extending, at some point, the number of elements in the database $\mathcal{S}_{N_r}$ would become quite large. For long routes, or $N_r > 20$, the number of elements in $\mathcal{S}_{N_r}$ are more than 500×10^3, rising to near 40×10^6 for $N_r = M = 40$ in a 2 km^2 range.

As one of our major concerns is the scalability of the system, we applied search optimisation techniques to both data structures and searching algorithms. In this work, we made use of the Burkhard Keller tree (BK-tree) [250] and combined it with the technique of dynamically constructing a route descriptor database. These are described in greater detail as follows.

BK-tree database

There numerous methods for nearest neighbour searching, as reviewed in [251]. We chose the BK-tree [250], which is one of the data structures originally used in spell check or string-related problems. Initially, this method applied the Levenshtein distance (or editing distance) [252], which utilises the triangle inequality to filter child nodes when searching. However, to enforce the spatial relationship between the semantic feature sequences, we replaced the Levenshtein distance with Hamming distance. By using Hamming distance, we gained advantages over speed, as well as fixed the comparison direction (i.e. front-bit to front-bit). We changed the code provided by [253] and converted our offline route descriptors database $\mathcal{S}_M$ into BK-trees. We searched for the closest route descriptor following Equation 5.10. By using the BK-tree, the time to process the millions of data in $\mathcal{S}_M$ was significantly reduced from eight hours using the naive comparison method to less than 1.5 minutes for the 40-location routes.

Dynamic route descriptors database

Although we applied a BK-tree structure to reduce the size of the database, for a larger area, the size of the offline database could affect the speed of searching. To enhance this, we made use of the dynamic database technique, as depicted in Figure 5.9. First, we performed route descriptor searching on the static database (or offline database generated from the 2-D map), which is the same process as previously mentioned. We continued searching and increasing the numbers of N_r (from 2, 3, $\ldots$) until it reached the threshold. The static BSD database $\mathcal{S}_{N_r}$ was then switched to the dynamic database $\mathcal{S}^*_{N_r}$ where $\mathcal{S}^*_{N_r} \subset \mathcal{S}_{N_r}$ and the size of $\mathcal{S}^*_{N_r}$ is much smaller than $\mathcal{S}_{N_r}$.

To construct the dynamic database $\mathcal{S}^*_{N_r}$, at the threshold position N_r, we made use of the candidate list (the routes that are likely to be the correct location) from the previous step. For example, in the case of Figure 5.9, at $N_r = 3$, we retrieved the ranked results from the previous step $N_r = 2$, and selected only top-n routes (e.g., for this work, we chose top-20 routes). Each candidate route retrieved from $\mathcal{S}_2$ was then extended along with the adjacent locations. The extended list was used as the new database $\mathcal{S}^*_3$ and we performed the search over this database instead of the static database $\mathcal{S}_3$. The ranked results from this step were extended to create the database for the next step $N_r = 4$. The processes were repeated until the current location was localised or the system reached the maximum route lengths M. Note that, in the main experiment, we used $N_r = 15$ as a threshold. The number came from the observation that it is sufficient for the system so that the relevant routes would not be discarded.

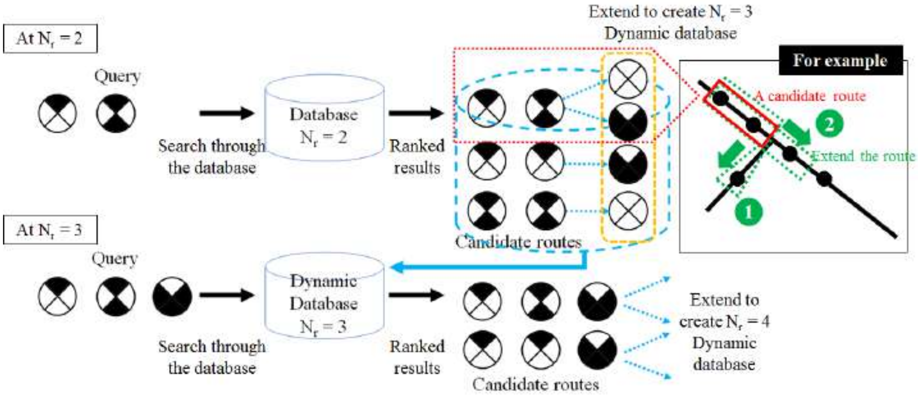


Figure 5.9: The concept of using a dynamic database. Note that, for visualisation, we used the example of the 2 BSDs route, or $N_r = 2$. (in the real process, we set the threshold $N_r = 15$). We generated the dynamic database at $N_r = 3$ from the candidate routes retrieved from searching the query through an offline database $N_r = 2$. Each candidate route extended itself to its adjacently connected node(s). The products from searching in the dynamic database at $N_r = 3$ were further used for generating a dynamic database at $N_r = 4$.

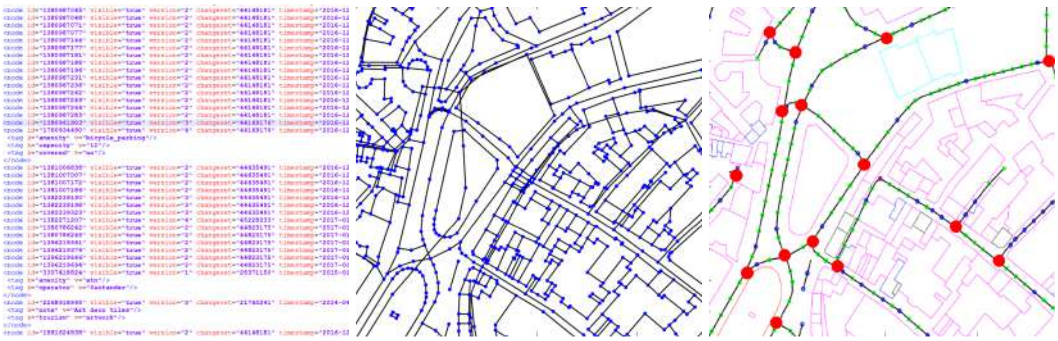
By using BK-trees and dynamic database, we could reduce the time of searching to less than one second with the extra 15–25 seconds for generating $\mathcal{S}^*_{N_r}$. Note that the time is different because the size of the extended list is not a static number. In addition, note that in this work we ran the experiment based on MATLAB. The process might be faster using more light-weight programming tools.

5.6 Localisation on 2-D maps using BSDs

In this section, we first explain the processes of semantic extraction from a 2-D map and BSDs offline database construction. Given a set of test data, we conducted some experiments using estimated BSDs to localise on 2-D maps. The aims of these experiments are (i) to evaluate the performance of the BSDs with variations in accuracy and (ii) to observe the impact of incorporating turn patterns.

5.6.1 Obtaining 2-D maps data

There numerous options for retrieving 2-D digital maps, as reported in [254]. In this work, we chose OpenStreetMap, which is an open-source digital 2-D map provider. The geo-data that can be extracted from OpenStreetMap consists of the locations of roads, buildings, and natural resources, such as mountains and rivers, as well as some properties related to the roads and buildings, such as name, address, and types of places. All 2-D data were stored in the form of geographical coordinates (latitude and longitude).



(a) (b) (c)

Figure 5.10: The pre-process of OpenStreetMap extraction starting from (a) XML-liked data of the selected area in a file exported from the website (b) raw clusters contain geographical coordinates and (c) the filtered objects and locations of junctions.

Figure 5.10 illustrates the pre-processes of semantic extraction from OpenStreetMap. Starting from the raw OpenStreetMap file (Figure 5.10a), we obtained raw data in the form of coordinate clusters (Figure 5.10b) using codes from [255]. We then labelled each cluster and calculated the locations of the junctions (Figure 5.10c). Finally, at each OpenStreetMap location, we extracted the semantic information and converted them into 4-bit BSDs. Note that the information in OpenStreetMap is user-generated, so this source sometimes lacks reliability and could affect the process of semantic extraction.

To observe the potential for localisation on the 2-D map using BSDs, we obtained test sets from five different cities: Birmingham (0.5 km^2), Glasgow (1 km^2), Manchester (1.5

km^2), Bristol (1.7 km^2), and London (2 km^2). A set of test routes was randomly selected from each city. The largest test set in this work, London city, consisted of 6656 locations related to the number of captured images, which we discuss in Chapter 6. All locations were captured approximately 10–15 metres away from each other. In total, the five cities together contain around 115 km of explorable roads.

5.6.2 Offline database construction

Given an area retrieved from OpenStreetMap encoded with semantic information as shown in Figure 5.11, to generate 4-bit BSD, we extracted the simple semantic information from each road location. To do that, the first step was to specify the area of interest around the given point. The radius used in this experiment was 30 metres. However, the given data were in latitude and longitude format. To find the distance between two coordinates, we used the Haversine formula [256], given (lat_1, lon_1) and (lat_2, lon_2) are the latitude and longitude for location l_1 and l_2 in radians. Therefore,

$$dist_m = 2r_e sin^{-1}\sqrt{sin^2\left(\frac{lat_2 - lat_1}{2}\right) + cos(lat_1)cos(lat_2)sin^2\left(\frac{lon_2 - lon_1}{2}\right)} \tag{5.12}$$

where r_e is the earth radius, and, in this experiment, we used the fixed number 6,378,137 metres. This number came from the assumption that the earth is perfectly spherical, which contrasts with the reality that it is in an ellipsoidal form. Therefore, this might have caused some inaccuracy for the computation. Based on the experiments in [257], the error of the Haversine formula is around 0.4% if the distances between locations are within hundreds of metres, which was sufficient for our application.

Figure 5.12 illustrates the processes of extracting semantic features from the given 2-D area; inside the circle, the area is divided into four quadrants: front, right, back, and left, corresponding to each bit in the BSD (Figure 5.12a). In each quadrant, we detected interest features: the presence of junctions and gaps between buildings. To obtain the former features, we checked for the presence of junctions within the quadrant (Figure 5.12b). For the latter, at the centre, we casted the virtual ray to the surrounding buildings within the quadrant; this process is similar to the component of ray casting in the ray-tracing technique [258, 259] using in computer graphics. The idea was to check whether the virtual rays hit the objects around the road location (Figure 5.12c). In this work, we equally spread the ray every 2° to scan for the presence of buildings.

To provide a better understanding, Figure 5.13 illustrates the histogram of 4-bit ground truth descriptors obtained from OpenStreetMap, which shows the distribution of descriptors

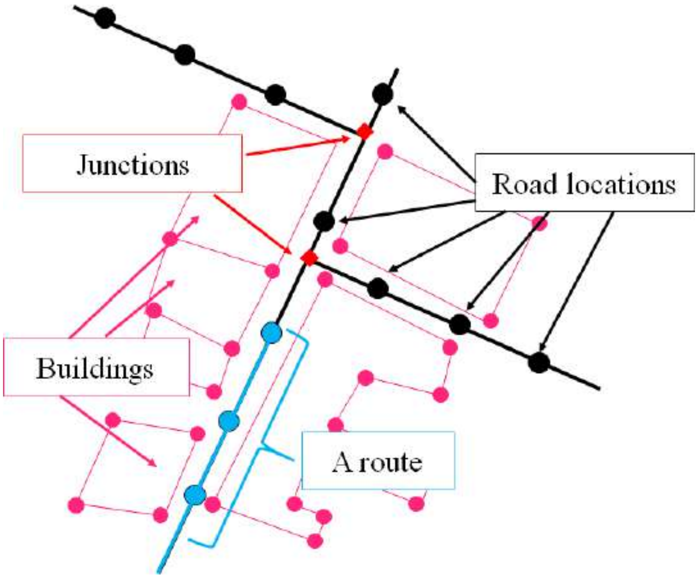


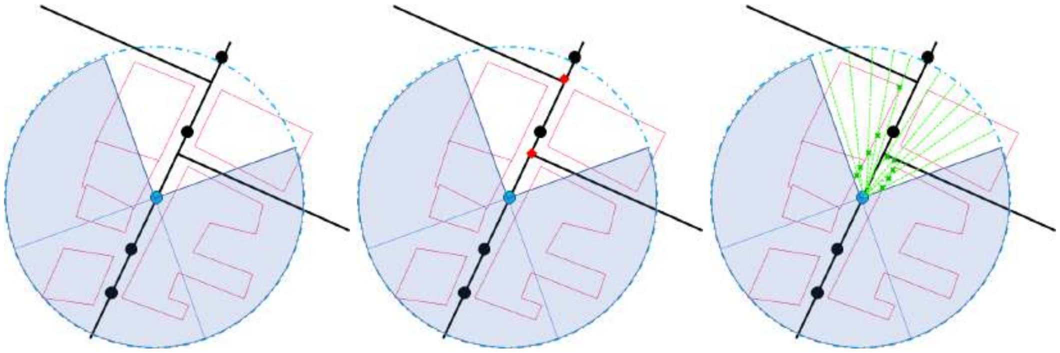
Figure 5.11: The key elements in the extracted OpenStreetMap area. Each point represents the geo-coordinate in latitude and longitude format. In this work, we only focus on where road segments and buildings are located; however, in the raw OSM data (Figure 5.10a), there are more details, such as types of buildings (e.g., a shop, office, and residents), and types of roads (such as the highway and the bicycle lane).

across each region. From this example, the predominance of BSDs is with a pattern '0000'. This corresponds to locations that contain neither gaps between buildings to the left or right, nor junctions towards the front or back. By comparing the test map in Figure 5.14 to the histogram in Figure 5.13, we can see that it was reasonable to obtain these patterns due to the nature of the layout. Furthermore, the colours indicating sixteen possible patterns within the area also reveal a high possibility of routes being distinct in the long run.

5.6.3 Simulation using estimated BSD

We can obtain BSDs directly from the 2-D map by identifying semantic features as described above. For localisation, we need to estimate BSDs from sets of four images taken in the forward, backward, left and right directions at each location. To do so, we can use semantic binary classifiers as described in the next chapter. Here we want to simulate that process and so investigate the performance of the proposed technique, particularly in relation to the performance of the semantic classifiers, i.e. to investigate the level of pdf classifier accuracy required to achieve a given level of localisation performance. We describe this investigation and the results obtained in this section.

The semantic classifiers are binary classifier, indicating the presence or not of the semantic feature (junction/gaps). Hence we are interested in the true positive rate (TPR) and



(a) (b) (c)

Figure 5.12: The process of BSD extraction: (a) we circled the area of interest and divided it into four partitions; for each, we (b) counted the appeared road intersection points, and (c) used the virtual ray to identify the gaps between buildings. The process of (b) was only applied to the front and back areas, and the process of (c) was only applied to the left and right areas.

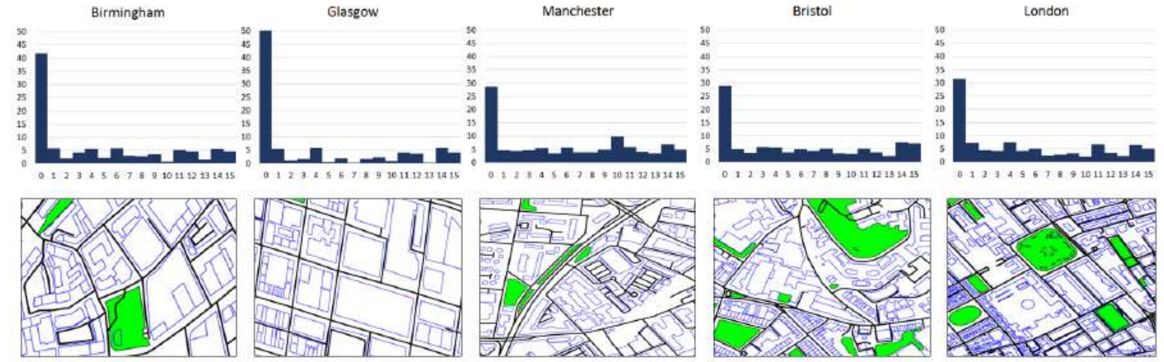


Figure 5.13: The histograms of ground-truth BSD and their corresponding maps captured from five different cities in different ranges, from 0.5 to 2 km^2.

true negative rate (TNR) defined as:

$$TPR = \frac{tp}{tp + fn} \qquad TNR = \frac{tn}{tn + fp} \tag{5.13}$$

, where tp,tn,fp and fn denote the number of true positive, true negative, false positive and false negative, respectively. We define each term as:

- True positive: the presence of the semantic feature is correctly classified
- True negative: the absence of the semantic feature is correctly classified
- False positive: the presence of the semantic feature is incorrectly classified
- False negative: the absence of the semantic feature is incorrectly classified

In our experiments we assumed that the TPR and TNR for the classifiers are all equal, which is consistent with our findings in Chapter 6 in the which the TPR and TNR of our

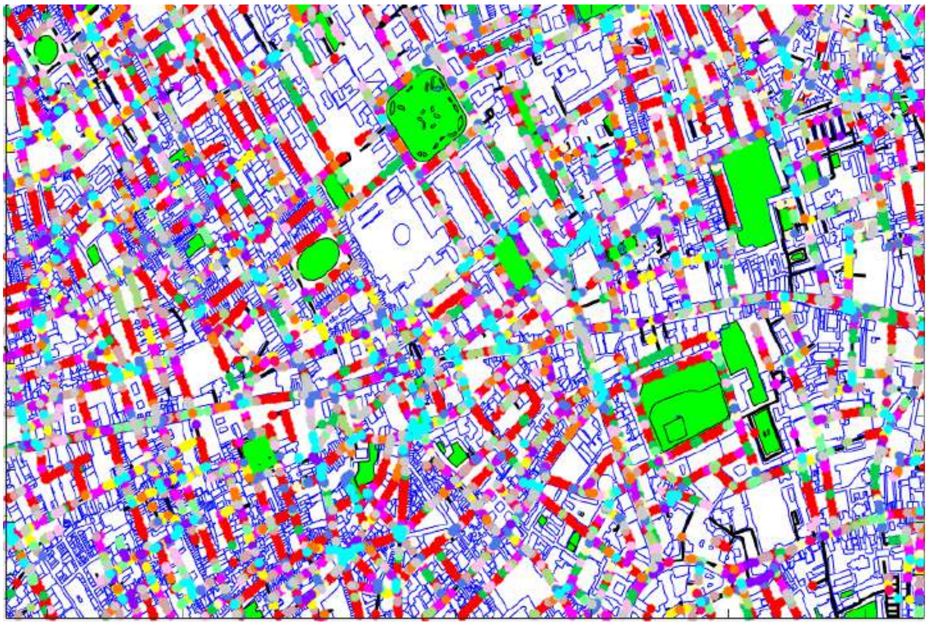


Figure 5.14: The 2 km^2 range of the London map labelled by a BSD pattern (from '0000' to '1111') in various colours. Each colour indicates sixteen possible patterns of BSDs. The stacked colours indicate the likelihood of routes being distinctive.

neural network binary classifiers were very similar. To reflect this, we henceforth characterise the classifier performance in terms of their (balanced) accuracy defined as:

$$Accuracy = \frac{TPR + TNR}{2} = \alpha \tag{5.14}$$

where for all classifiers we assume $TPR = TNR = \alpha$. We used this to simulate the estimation of BSDs at each location.

In the experiments, we selected 50–150 random test routes from each city (depending on the size) and simulating the estimated test routes using the variation of the accuracy from 50% (random chance of semantic features to be correctly detected – as our BSD is a binary descriptor) to 100% (perfect classification). Note that we assumed the same accuracy for detecting the presence of both junctions and gaps between buildings. We applied the processes described in Section 5.2–5.5. The correctness of localisation was verified by comparing the last location of the best candidate route with the location of the ground-truth. If they are exactly the same, we counted as a correct localisation.

Figure 5.15 illustrates the localisation performance of the London set (as the biggest test set we have). We observed the change in classifier accuracy from 50% to 100%; when classifier accuracy increased to over 80%, 80–90% of routes were correctly localised using less than 15 locations. This illustrates the potential of the binary semantic descriptor approach. To gain a better understanding, Figure 5.16a displays a comparison of the percentage of correctly localised routes versus the classifier (s) accuracy for four different

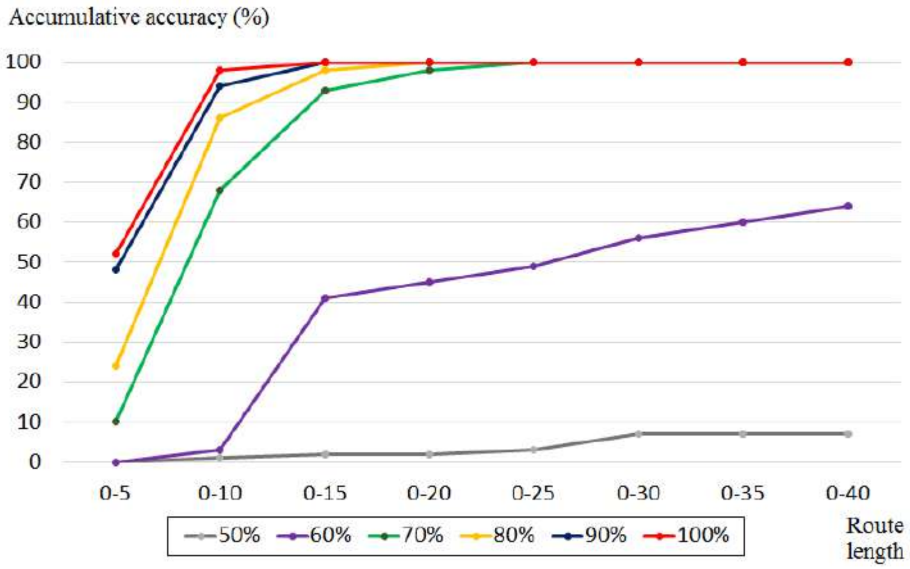
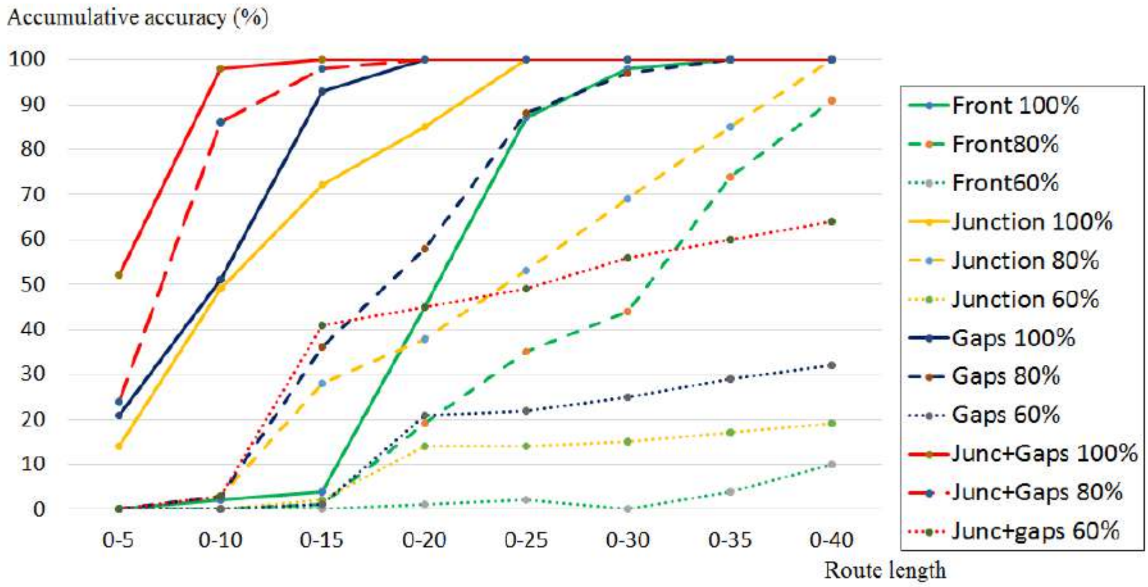


Figure 5.15: Accumulative accuracy of localisation (% of correctly identified routes) versus classifier accuracy for different ranges of route length using the presence of junctions and gaps between buildings within 2 km^2 range of the London test map.

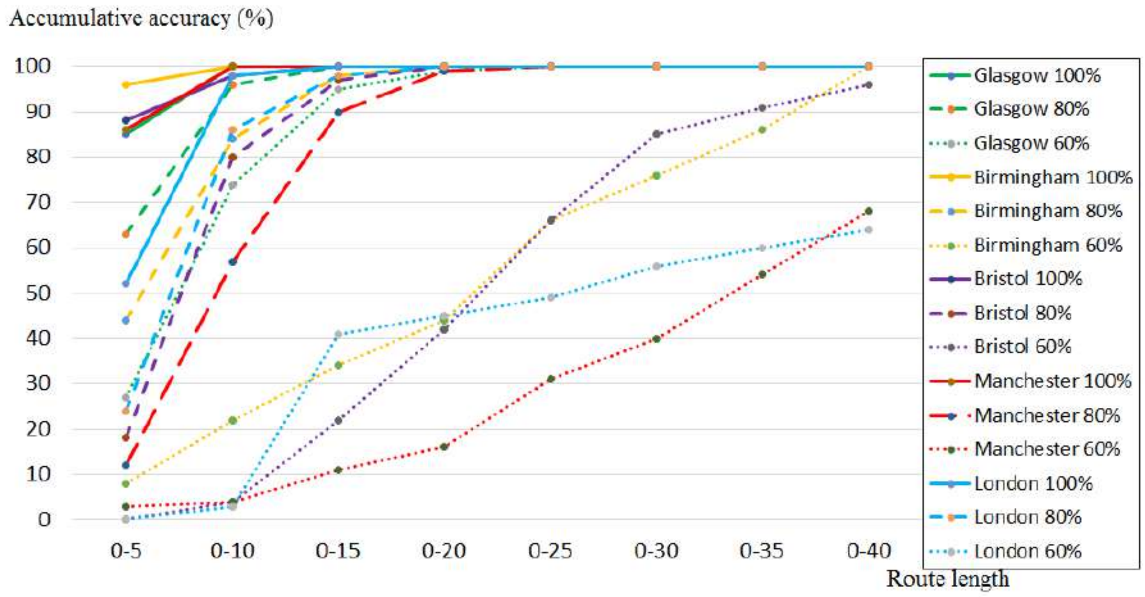
methods: only front (detect junctions at the front), only junctions (detect junctions at the front and back), only gaps (detect gaps between buildings on the left and right), and both junctions and gaps. At 100% accuracy, using both junctions and gaps between buildings outperformed the others. With lower accuracy, Figure 5.16b shows that the results of the small area maps (less than 1.5 km^2), such as Birmingham and Glasgow, contained higher tolerance to the low classifier accuracy comparing to the others. Moreover, for the larger areas, at 60% classifier accuracy, the localisation rates significantly decreased. We believe these were direct effects of the area size, because the larger the area means more locations and less unique route patterns. Based on the two sets of results in Figures 5.15 and 5.16, we set 70% as a minimum threshold for accuracy of estimation.

We conducted an additional experiment to observe the impact of incorporating turn patterns. The results of using only the presence of junctions (either only front or front and back) in Figure 5.16 were consistent with the outcome in [238, 241]. As stated in the papers, by obtaining visual odometry from the front view, their system could localise within 200–350 metres (depending on whether monocular or stereo-based methods were used). However, their method using only front views was limited by straight route patterns (route sequences with no turn). We handled this issue more efficiently using semantic cues from 360-degree views.

Figure 5.17 further illustrates the distribution of Hamming distances from descriptors in the database for a given test (query) route at lengths of 15 (left) and 30 (right) locations,



(a)



(b)

Figure 5.16: Accumulative accuracy of localisation (% of correctly identified routes) versus classifier accuracy for various ranges of route length using (a) different semantic features at 100%, 80%, and 60% of accuracy in London and (b) the presence of junctions and gaps between buildings over five different cities at 100%, 80%, and 60% of accuracy.

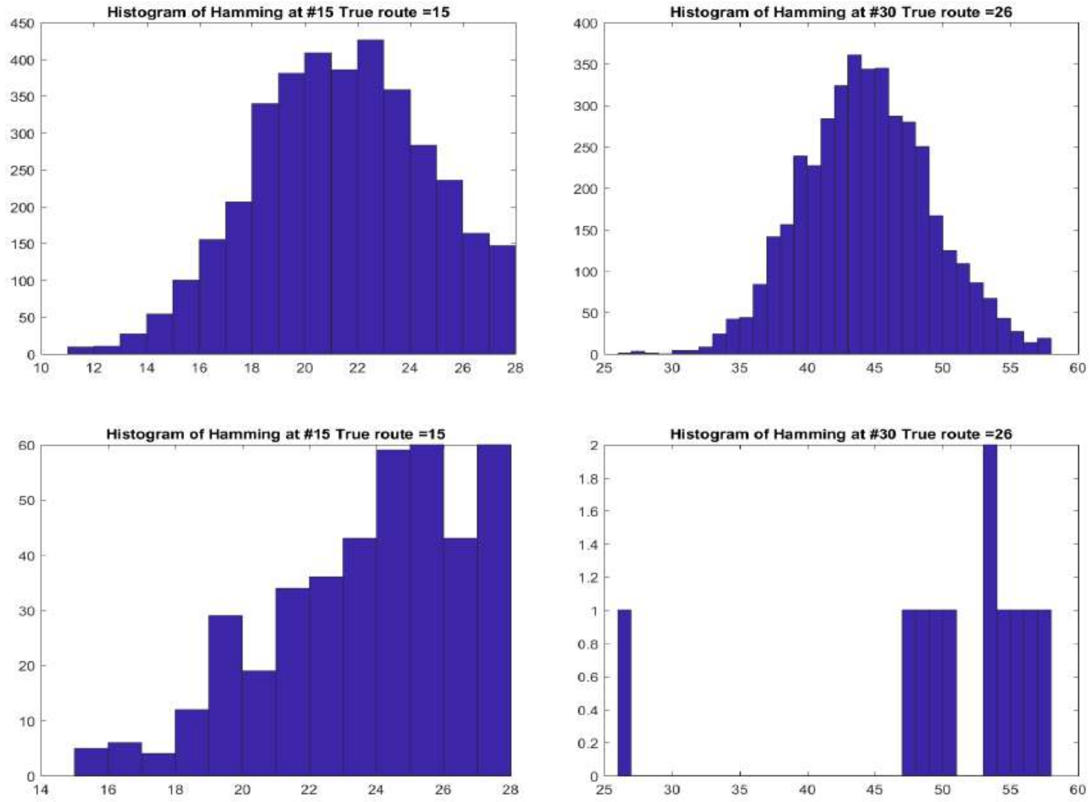


Figure 5.17: Histograms of Hamming distances between a test route descriptor and those in the database for route lengths of 15 (left) and 30 (right) locations, with (bottom) and without (top) using turn patterns. This demonstrates the improvement of localisation using turn patterns.

with and without using turn patterns (bottom and top, respectively). The correct matches for lengths 15 and 30 had Hamming distances of 15 and 26, respectively. When the test route length was 15 locations, the correct route was not the closest (there were other Hamming distances with values < 15), although using turns (bottom) significantly reduced the number of routes close to the query route. With 30 locations and without using turns, the correct route becomes equal closest with 18 others and there was a significant number of others close by. In contrast, using turn patterns with 30 locations drastically reduced the number of candidate routes, and the correct route became the closest, with a Hamming distance margin of over 20. This indicates that the effect of turn patterns narrows down the unwanted results.

In sum, the findings presented in this section suggest that using 4-bit BSDs yields a greater impact than others, and the larger area requires higher classifier accuracy. A suitable classifier accuracy that allows for using 4-bit BSDs for localisation in a 2 km^2 range should be more than 70%. We apply this information in the next chapter to train the classifiers.

5.7 Summary

In this chapter, we have conducted preliminary investigations into the approach. We aimed to characterise locations by a small number of semantic features relating to road junctions and buildings. We represented each location by a binary descriptor, with each bit indicating the presence or lack of a given feature in a given viewing direction and increasing scalability. We sought suitable semantic representations that contained characteristics of availability, reliability, clarity, ubiquity, distinction, and locality. Finally, we identified the presence of junctions and gaps between buildings and constructed a 4-bit descriptor, known as BSD.

However, due to their simplicity, the BSDs were not sufficiently discriminative on their own. We introduced route descriptors by sequentially concatenating the BSDs. Once a sufficiently long route was established, the pattern of semantic features observed along a route became unique. Moreover, when the direction of travel between locations along a route was taken into account as turn patterns, the performance was further improved.

By using a small number of bits per location, we demonstrated the effectiveness of this method over the 2 km^2 range test map, which is difficult to achieve using the comparatively large representations. The next step is to integrate the system with the images by making use of classifiers to predict the presence of semantic features. As demonstrated in Section 5.6.3, the minimum requirement for the classifier accuracy to enable localisation is 70%. Hence, in the next chapter, we demonstrate the process of image data gathering, training, and testing the selected classifiers.

Chapter 6

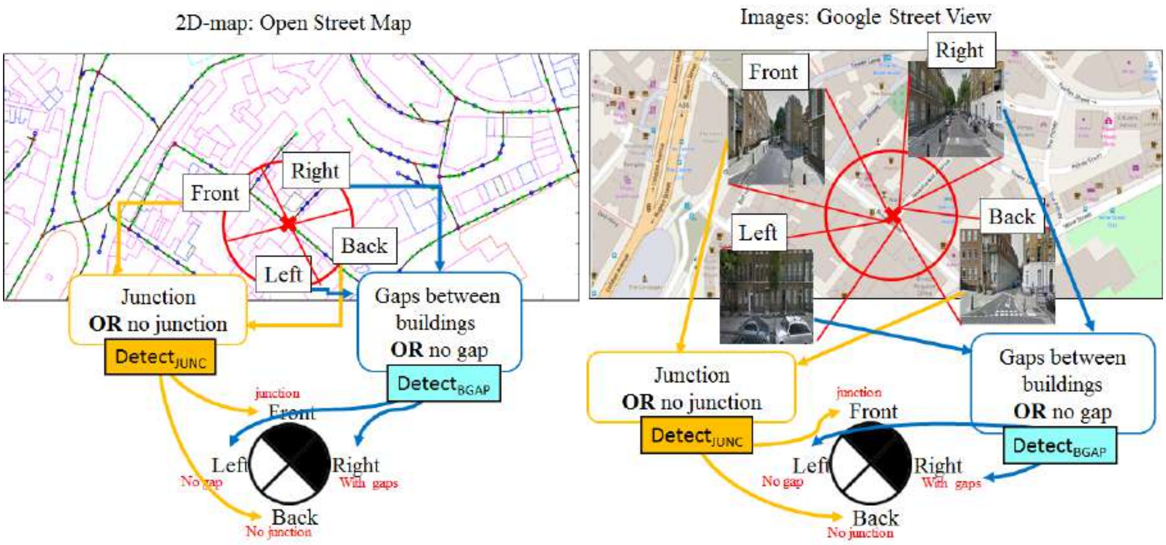
Image-to-2-D map matching using binary semantic descriptors

In the previous chapter, we presented an approach to position localisation in urban areas and conducted preliminary investigations into the approach. We aim to characterise locations by a small number of semantic features. To do that, a common representation between images and a 2-D map is required. Therefore, we defined the characteristics of our representation as availability, reliability, clarity, ubiquity, distinction, and locality. In this respect, we chose the presence of junctions and gaps between buildings. We constructed a 4-bit binary semantic descriptor (BSD), with each bit indicating the presence or lack of a given feature in a given viewing direction. On a 2-D map, semantic features can be extracted directly. However, without using metadata, a digital image only contains a low level of semantic information, such as colours. To extract the presence of given features, we need a method to convert images to semantic information.

In Section 6.1, we discuss the process of converting images to a BSD using the learning-based method to classify semantic features. In Section 6.2, we then combine all of the components and explain the whole process of image-to-2-D-map localisation using BSD. In Section 6.3, we demonstrate the use of BSD for localisation and provide some provisional results for the real-world integration. Finally, in Section 6.4, we summarise the overall results, which lead to future possibilities for this work, further discussed in Chapter 7.

6.1 Image to Binary Semantic Descriptor

In the previous chapter, we chose the presence of junctions and gaps between buildings to represent both 2-D maps and images. By obtaining digital base maps from OpenStreetMap, the given features could be directly extracted to construct a 4-bit BSD. However, it is more difficult for images, as we cannot apply the same process to retrieve the semantic features.



(a) (b)

Figure 6.1: Processes of converting (a) 2-D map to a BSD and (b) a set of four directional images to a BSD. Both apply the presence of junctions (front and back images) and the presence of gaps between buildings (left and right images).

Based on this, we aim to identify suitable classification techniques to convert images into an estimated BSD. Given an image I_{ij} at location i in viewing direction V_{ij}, the estimated BSD $\hat{s}$ is given by

$$\hat{d}_{ij} = \begin{cases} DETECT_{JUNC}(I_{ij}) & \text{if } j \in \{1, 2\} \\ DETECT_{BGAP}(I_{ij}) & \text{if } j \in \{3, 4\} \end{cases} \tag{6.1}$$

where $DETECT_{JUNC}(I_{ij})$ and $DETECT_{BGAP}(I_{ij})$ return 1 if junctions or gaps between buildings, respectively, are detected in image I_{ij}, and 0 otherwise. This process mirrors the BSD generation functions in Equation 5.3.

6.1.1 Image features classification

There are several algorithms available for semantic classification; some have been discussed in Chapters 2 and 3, such as [16, 45, 260, 261]. The first technique is to use edges and contours to detect the locations of objects, such as using line detection for finding road lanes [262]. The second technique is to use learning-based models, such as SVM [263, 264] and CNN [240].

Among the learning-based techniques, we chose CNN to design the binary classifiers $DETECT_{JUNC}$ and $DETECT_{BGAP}$ due to its high effectiveness over image classification problems. Similar to [240], we applied the pre-trained Places205-AlexNet model [265] derived from [11]. This model is specifically designed for scene classification in urban environments, which is well aligned with our application. Note that it is possible to train a classifier for multiple labels, such as in [266, 267], but we separate both detectors for simplicity. More details of improvements are discussed in Chapter 7.

6.1.2 Dataset and training

As using CNN models required a large set of labelled data, we chose Google Street View as the image provider. However, the service only provides images without semantic labelling. With loads of images, a method for auto labelling is required. Hence, similar to [240, 241], we made use of semantic information from OpenStreetMap. To do that, Google Street View images were linked to the OpenStreetMap locations using the corresponding geographical coordinates (latitude and longitude). For each feature type, we collect positive samples by identifying the locations of the relevant features and storing the images and viewing directions of the locations. In addition, to gain data variation, we obtained a uniform mix of viewing scenarios. For example, in the case of junctions, we used front- and back-facing images aligned with the road and ensured that we had examples that covered the range of distances from the junction up to the viewing radius. We completed the training set by collecting the same number of negative samples in the corresponding viewing directions that did not contain the feature of interest. This was similarly done for gaps between buildings, but we used the left- and right-facing images instead.

As for training and testing the classifiers, we used colour images cropped from Google Street View panoramas in the required viewing direction corresponding to a 90° horizontal field of view and resized to 227×227 pixels, which is the required size of AlexNet model. The latter resulted in some distortion; however, given that we used the same process for both training and testing, this is not considered to be an issue. In the experiments, we used training sets consisting of 440,000 images per classifier taken from 220,000 locations in 23 different cities in the UK[1]. The performance of each classifier was evaluated using a test set of 8000 images taken from the same 23 cities but at locations not within the training set and with an equal number of positive and negative samples. Figure 6.2 illustrates examples of the training data with variation in environmental conditions and architectural styles.

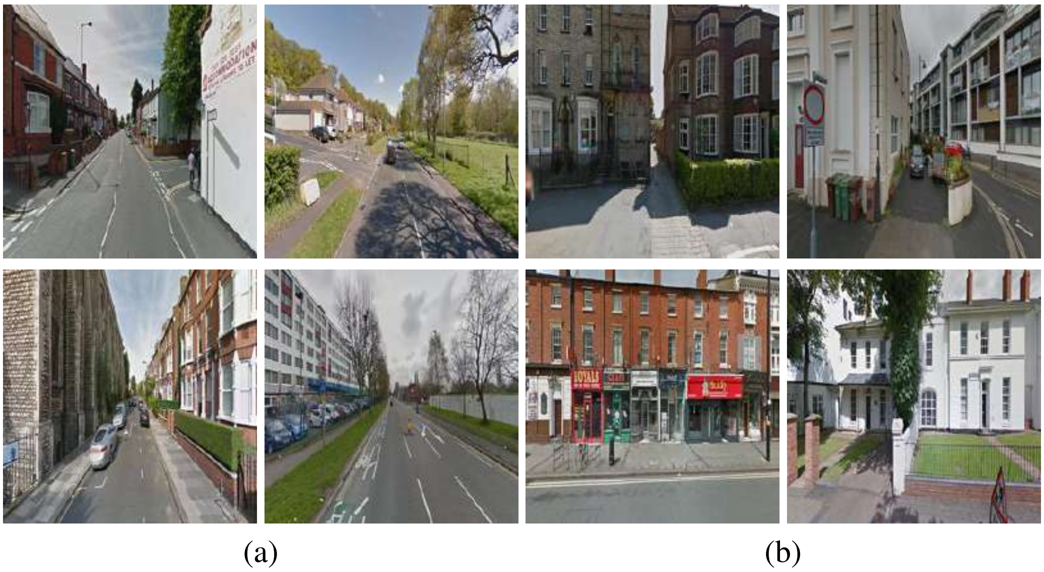
[1] The twenty-three cities contained Bath, Bristol, Cambridge, Cheltenham, Coventry, Derby, Edinburgh, Glasgow, Leeds, Liverpool, Livingston, London, Manchester, Newcastle Upon Tyne, Norwich, Sheffield, Southampton, Plymouth, Preston, Wakefield, Walsall, Wolverhampton, and York.

Table 6.1: Recorded percentage of TPR, TNR and accuracy of each classifier using different trained image features those are: the distance to the closest building, the density of buildings appear in the scene, the average size of buildings appear in the scene, the presence of junctions, and the presence of gaps between buildings, respectively.

	TPR	**TNR**	**Accuracy**
Distance	61	41.65	51.32
Density	57.3	59.4	58.35
Size	53.2	59	56.1
Junction	73.78	76.92	75.35
Gap	74.65	77.83	76.24

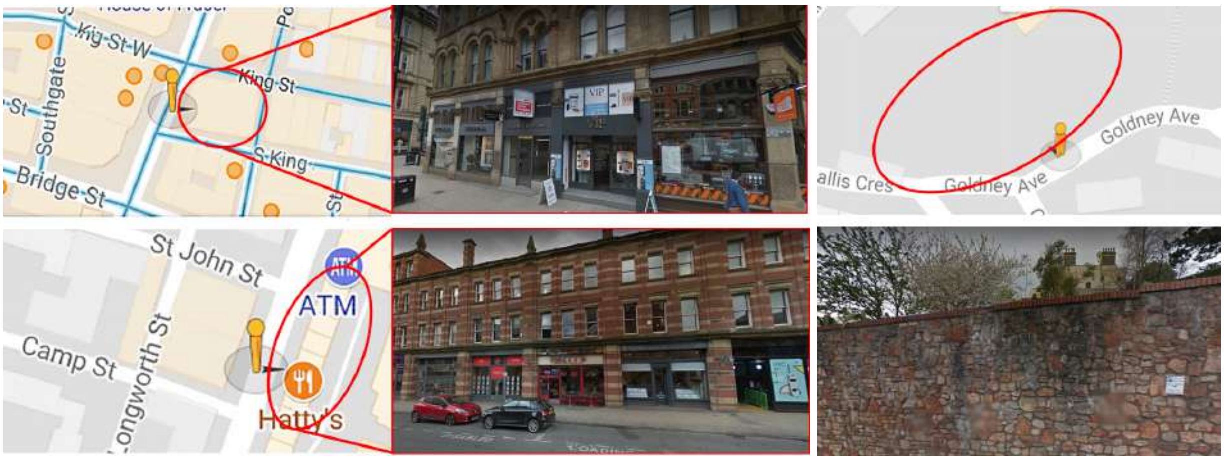
6.1.3 Model evaluation

We evaluated the performance using the stated test set and recorded the accuracy of each classifier. In addition to junctions and gaps between buildings, we included other semantic features: distance, density, and size. The distance was the binary classification for indicating that the given location was close to a building. The density was for indicating the density of buildings in the scene. The size was for indicating whether the scene contained a large building. Note that these selected features did not pass our constraints in Section 5.1, but we selected them to emphasise the importance of those properties. This is discussed in further detail below.



(a) (b)

Figure 6.2: Examples of positive (features present) and negative (features not present) images from the training datasets used for the semantic classifiers: (a) junction (top) and no junction (bottom); (b) gap (top) and no gap (bottom).

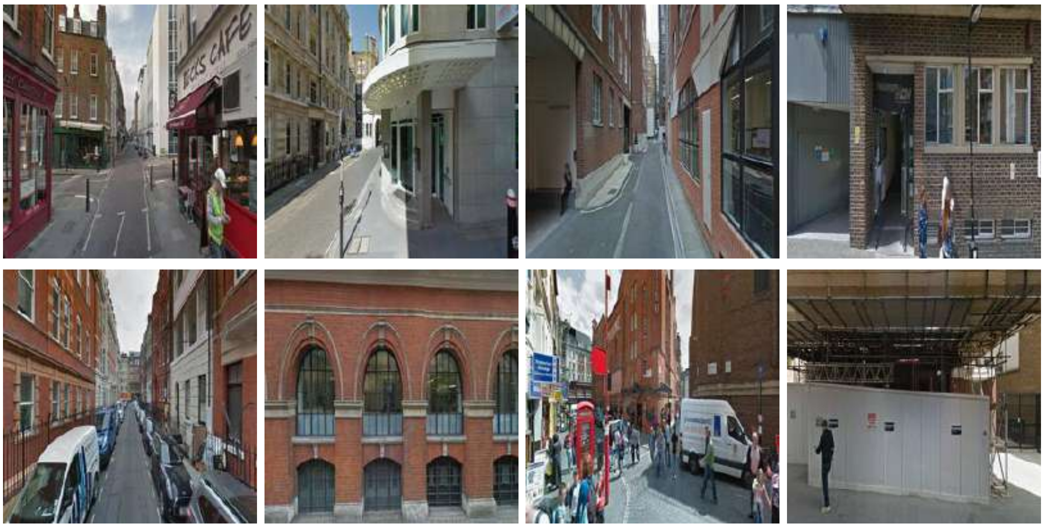


(a) (b)

Figure 6.3: Examples of ambiguity in training image features. In (a), the top row shows an image and 2-D map pair; a solid rectangle shape in the 2-D map represents the rows of shopfronts in their corresponding image; in contrast, while the bottom row also depicts an image of a shopfront, there is a cluster of rectangles in the 2-D map. Therefore, a scene in the top row might be labelled differently from the bottom row, even though the corresponding images are the same. In (b), the top row shows a 2-D map area containing a blank space, in contrast to its corresponding Google Street View image, which displays a wall. This example demonstrates the unreliability of the map, which may affect the classifier accuracy.

Table 6.1 represents the percentage of TPR, TNR and accuracy using the test features. The results indicate that both junctions and gaps between buildings can fulfil the minimum constraint of 70% accuracy stated in Section 5.6.3. Figure 6.3 further illustrates the problems behind other test features: distance, density, and size. The problem is a lack of clarification, as these features require a finite number for the threshold. For example, in the case of distance, we set 15 metres as a threshold for indicating there was a building near the scene; if there was a building within 15 metres, the image was labelled as ‘near’; otherwise, it was labelled as ‘not near’. However, in the real-world environment, there is a possibility that scenes with 15-metre and 17-metre ranges would appear exactly the same. The same problem exists for size and density. Coupled with a high probability of having more than one building in a scene, it is difficult to justify the concept of these features. Moreover, clarification is not the only issue behind inaccuracy. Figure 6.3a displays a pair of images and 2-D maps captured at the same geographical location. There is a row of shops in the image, but the 2-D map shows only one building. This example demonstrates a high density of buildings with a ‘not dense’ label.

Compared to these features, the presence of junctions and gaps between buildings is



(a) (b)

Figure 6.4: Examples of semantic classifications: (a) true positives (top) and true negatives (bottom); (b) false positives (top) and false negatives (bottom). In both (a) and (b), examples are arranged as: junction (top-left); gap (top-right); no junction (bottom-left); no gap (bottom-right).

more solid. Both classifiers demonstrated well-balanced performance in detecting the presence and non-presence of junctions and gaps between buildings, with precision and recall values around 0.75 on the test sets. However, these features are not entirely reliable; the noisy data are presented in Figure 6.3b. A 2-D map contains only a blank space, but its corresponding Google Street View images display high walls. Hence, the given example is labelled as ‘contains a gap’ with a corresponding ‘no gap’ image. This is one of the factors that affects the accuracy of our classifiers. Figure 6.2 illustrates examples of positive and negative images from the training dataset.

Furthermore, Figure 6.4 illustrates examples of correct classifications (true positives and true negatives) and incorrect classifications (false positives and false negatives). Note that the latter illustrates the difficulty of the task. For example, the bottom left view in Figure 6.4b contains a junction that is significantly obscured and is incorrectly classified as containing no junction. Another example is the bottom right view; its corresponding 2-D map indicates that it should contain a gap, but the site appears to be under redevelopment and has been incorrectly classified as not containing a gap. This is an example of the unreliable nature of the data providers, which could affect our performance. Nonetheless, the junctions and gaps between buildings are still suitable features for our system. It is also important to note that these selected features contained a sense of hand-crafted selection. In other words, we chose them from the experiments, and evaluating their effectiveness is difficult. For a more practical system, automated selection of semantic features might be a better solution. More details are discussed in Chapter 7.

6.2 Image-to-2-D map localisation using BSD

In this section, we summarise the overall findings, combining the components discussed in Chapters 5 and 6 to gain a complete picture of the system of the image-to-2-D-map localisation using BSDs. Figure 6.5 illustrates three main components of the approach: (i) database creation (Figure 6.5d), (ii) image sequence generation (Figure 6.5a–c), and (iii) search process (Figure 6.5e–g). More details are provided below.

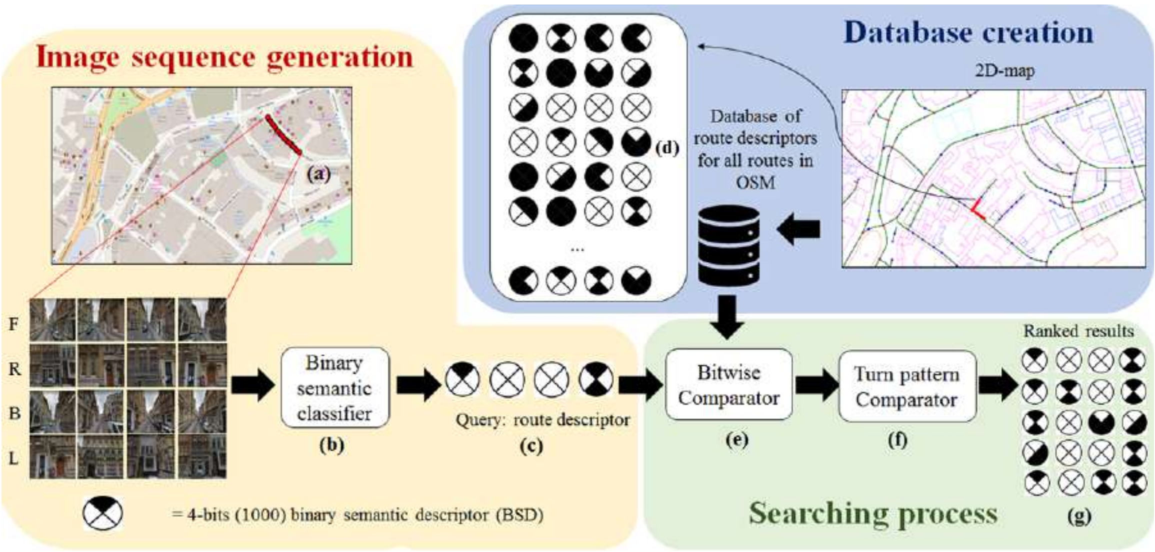


Image sequence generation
(a)
F
R
B
L
Binary semantic classifier
(b)
Query: route descriptor
(c)
= 4-bits (1000) binary semantic descriptor (BSD)
Database creation
2D-map
(d)
Database of route descriptors for all routes in OSM
...
Bitwise Comparator
(e)
Turn pattern Comparator
(f)
Ranked results
(g)
Searching process

Figure 6.5: Processes of route-based localisation – (a) images captured in four directions (front, back, left, and right facing) at locations along a route. They are converted to BSDs using (b) binary classifiers and concatenated to produce (c) route descriptors. These are (e) bit-wise compared with (d) a database of ground-truth BSDs to determine the closest matching route. Routes are then compared in terms of (f) turn patterns to give (g) a final ranking of possible locations of the images corresponding to the 2-D map.

Database creation

First, from a 2-D vector map obtained using OpenStreetMap, we generated BSDs for locations spaced at regular intervals along roads in an urban environment. Each descriptor consisted of four bits, with each bit indicating the presence or lack of a semantic feature in each given viewing direction: front, back, left, and right. We extracted the presence of two semantic features: junctions from the front and back views, and gaps between buildings from the left and right views.

An offline database of location-tagged route descriptors was generated by computing all possible routes within the area of interest up to a certain length in terms of the number of adjacent locations and concatenated the set of associated BSDs. As indicated in Figure 6.5d, the circular discs represent the BSDs, and the black and white segments indicate individual bits. Each route descriptor is then of length $4N_r$ bits, where N_r is the number of locations in the route where $1 \leq N_r \leq M$, with a maximum route length in this work of $M = 40$. To improve the performance of the searching algorithm, we applied the BK-tree data structure and, once the length of routes reached the given threshold, the database (Figure 6.5d) was dynamically generated at each turn resulting from the previously visited locations.

Image sequence generation

A 'virtual user' was generated to project the movement along a route in the environment to mirror the way a real user walks in the street. Google Street View images in the four viewing directions were captured at successive locations corresponding to each road location in a one-to-one relationship. Each image was fed to a binary classifier, which detected the presence or lack of junctions for the front- and back-facing views and gaps between buildings for the left- and right-facing views. The sequences of the estimated BSDs then formed a route descriptor (Figure 6.5a–c) to further be used in the searching process.

Searching process

Given sequences of the estimated 4-bit BSDs, localisation then proceeded by comparing the estimated BSD $\hat{s}$ and those for all locations in the 2-D map to give localisation with Hamming distances used to provide a ranked list of likely locations (Figure 6.5e–g). To add further discrimination, we compared the turn patterns associated with the query and database routes, requiring that these were identical for a valid match. The motivation here is that direction changes of, for example, an autonomous vehicle can be detected reliably and can thus eliminate spurious matches between route descriptors, similar to human wayfinding. The database route with the lowest Hamming distance corresponding to the query route and the same turn pattern then provided the location estimate.

The state of searching continued by increasing numbers of route length N_r until there was sufficient overlap between the most likely routes for a number of successive locations or the virtual user walked a certain distance M; we then confirmed localisation and maintained it using a fixed number of route length or bootstrapping.

6.3 Experiments and results

In this section, we demonstrate the use of BSDs in three experiments: (i) image-to-2-D-map localisation using BSDs, (ii) impact of urban patterns over BSDs, and (iii) BSDs to real-world integration. The aim of the first is to investigate the use of BSDs for image-to-2-D-map localisation. In other words, we performed the same experiment as in Section 5.6, with the real image classifiers applied. The second is to observe the impact of urban patterns by testing on different cities than those used in the first experiment. Finally, the last is to test our system using real-world data.

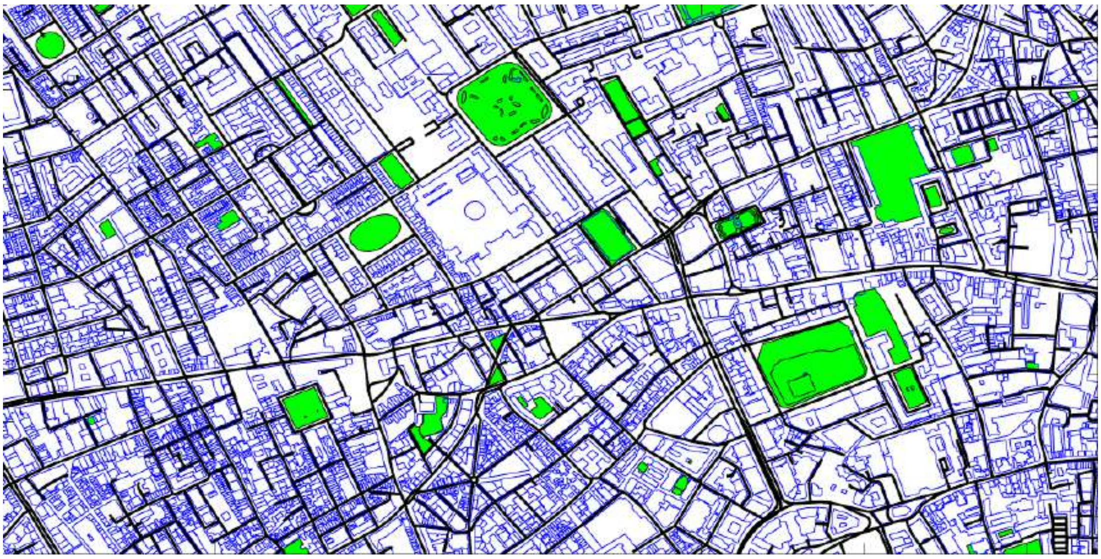


Figure 6.6: OpenStreetMap data for a 2 km^2 range region of London we used for testing. This is the same map that was used in the experiment in Section 5.6. There are 6656 road locations, or approximately 66 km of explorable road.

6.3.1 Localisation using BSDs

In this section, we evaluate the performance of the method using Google Street View and OpenStreetMap data for a 2 km^2 region in London. There are 6656 road locations ($N = 6656$), or approximately 66 km of explorable road in total, as illustrated in Figure 6.6. Note that this is the same test set we used in Section 5.6. From each location, we gathered images corresponding to the four viewing directions, as well as the estimated 4-bit BSDs using the classifiers. Figure 6.7 displays the histogram of 4-bit ground-truth descriptors (obtained from OpenStreetMap, shown in blue) and estimated descriptors (predicted from Google Street View images, shown in red); the horizontal axis corresponds to the sixteen possible 4-bit BSD patterns (or ‘0000’ to ‘1111’ in binary terms). This illustrates the distribution of descriptors across the region and the performance of the classifiers. The distribution of the estimated descriptors is close to that of the ground truth due to the classifier accuracy tested in Section 6.1.3, which was approximately 75% in both categories. Figure 6.8 displays examples of the estimated BSDs, their corresponding images in the four viewing directions, and the ground-truth BSDs. The deviation of the BSDs estimated from the ground truth was caused by the inaccuracy of the classifiers. However, this also increases the challenging nature of the detection task and confirms the utility of concatenating BSDs along a route to gain uniqueness, thus enabling localisation.

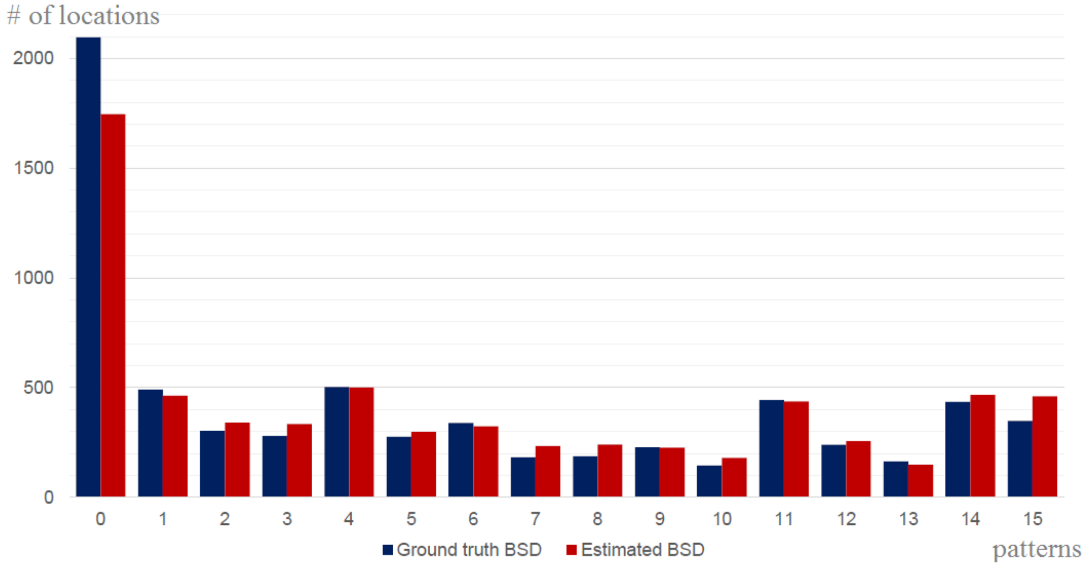


Figure 6.7: The histogram showing the distribution of 4-bit ground-truth (blue) and estimated (red) BSDs obtained from OpenStreetMap and Google Street View images, respectively. The distribution of the estimated descriptors is close to the ground truth due to the classifier accuracy.

We considered route lengths up to a maximum of $M = 40$ locations (approximately 400 metres) and tested the method using 150 randomly selected test routes. Each of them contained a mixture of route sequential patterns, as illustrated in Figure 6.9. For each, we recorded the route length at which localisation was achieved according to the consistency measurement; there were five successive consistent localisations. Figure 6.10 presents the percentage of routes that were correctly localised within route lengths of 0–5, 0–10, . . . , to 0–40 locations. We display the results for five methods of matching routes: using only binary turn patterns (light blue), using only left and right turn patterns (grey), using only route BSDs (dark blue), using both BSDs and binary turn patterns (yellow), and using both BSDs and left-right turn patterns (dark green). The latter significantly outperformed the others, and these results of 75% of classifier accuracy are consistent with the plot of the

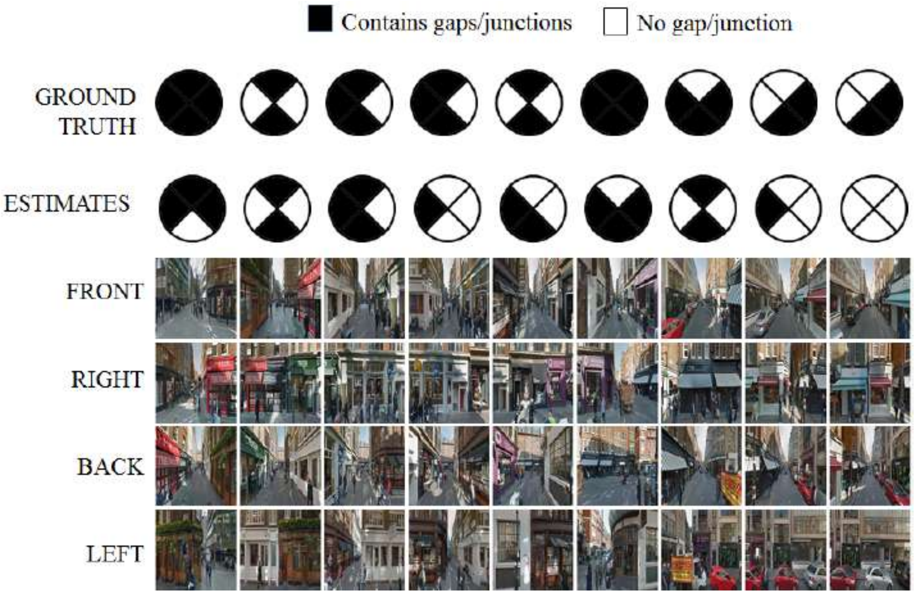


Figure 6.8: Examples of ground-truth BSDs from OpenStreetMap, and BSD estimates, from the classification of the Google Streetview images in four directions. There are some errors in the estimated BSD because of the classifier accuracy. However, the results show that these errors do not affect the performance of the system.

percentage of correctly localised routes versus the classifier accuracy simulated in Section 5.6.

Moreover, with a short moving distance, BSDs alone were better than using only turn patterns, while, in the longer run (in this case, up to 300–350 metres), the performance of using left and right turn patterns increased. These results are consistent with the experiment using visual odometry presented in [238, 241].

There are reasons behind the dissimilar results obtained between binary and non-binary turn patterns. First, adding more bits means more distinct patterns, which can boost accuracy. Second, as displayed in Figure 6.6, the road structure of our test set contained several four-way junctions. Given a more precise turning direction, this boosted the speed of decision making. However, it is unavoidable that adding more bits might affect the computation time; if we compare the results between two types of turn patterns, the processing time is compensable, as the performance extremely improves. Specifically, over 95% of the test routes were correctly localised using 20 locations or approximately 200 metres in physical distance. The average distance for all routes to be localised was 97.2 metres. We also gained more benefits over [238, 241] by dissolving their problem of failure over straight route patterns. In addition, with a variety of data in temporal changes, spatial changes, and domain changes, our system demonstrated effectiveness over invariance. At the same time, the compact size of the descriptor also increased the sense of scalability. To strengthen this

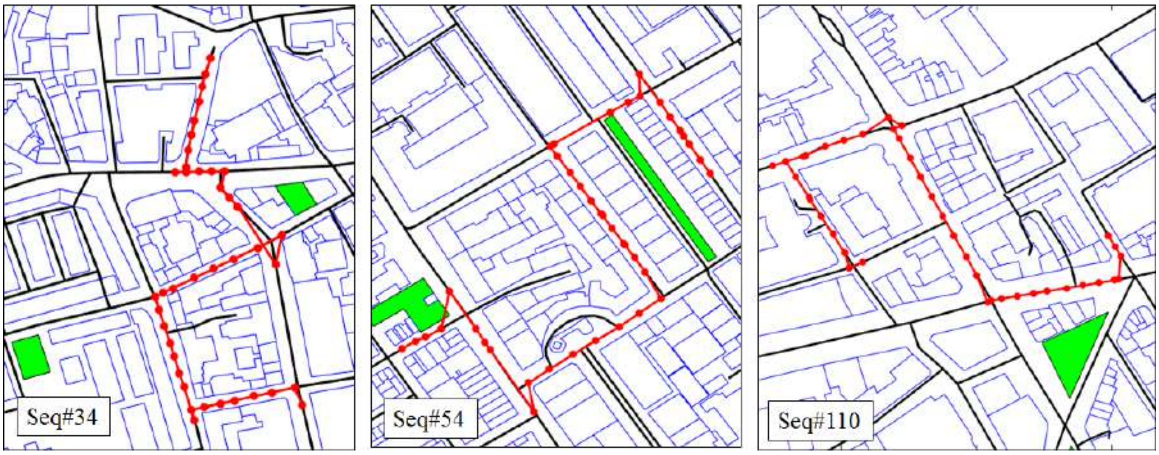


Figure 6.9: Examples of random test route sequences used in the experiment. The red line represents the moving path of the virtual user within the maximum of route lengths is 40 ($M = 40$) locations, which equals approximately 400 metres.

statement, we extended our experiment to demonstrate the scalability of our BSDs. We randomly selected 100–500 test routes. Figure 6.11 illustrates the results, displaying the maintenance of the percentage of routes that were correctly localised, which reflected the scalable characteristics of our system.

To illustrate the localisation process, Figure Figure 6.12 illustrates snapshots of the localisation of a test route at route lengths of 2, 24, and 48 locations. It displays the 2-D map, with the locations indicated by the coloured square markers along roads. The colour indicates the likelihood of the location being corrected calculated from a probabilistic formulation (as stated in Appendix A). Note that we used the probabilistic method just to illustrate an alternative way to indicate the likelihood of the possible locations at any given point. The difference from the Hamming distance is that this method also accounts for classifier accuracy. We used it in the hot spot images to represent the closeness between each route descriptor and the test route from the highest (dark red) to the lowest (dark blue). However, it has no impact on the ranking of the matching with the database. The latest location along the test route is indicated by an orange or red circle. Orange indicates that the route has yet to be correctly and consistently localised. Red indicates that localisation has been achieved. The BSDs, both estimated and ground truth, along with their corresponding images, are shown below the 2-D maps. Note that the bottom row of images shows the views at the closest (best) match locations, but we do not use them in the matching process. A video illustrating the complete process is available at *www.youtube.com/watch?v=fwZWr_WXCRw*.

At the top row of Figure 6.12, with a route length of 2, the majority of locations have

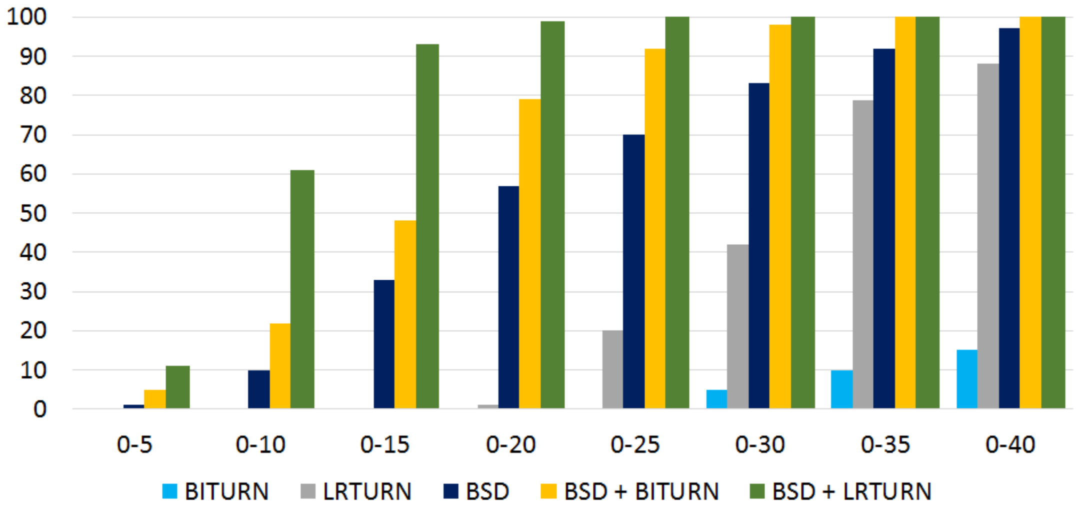


Figure 6.10: Accumulative accuracy of localisation (% of correctly identified routes) versus route length using binary turn patterns (light blue), left and right turn patterns (grey), route descriptors (dark blue), route descriptors with binary turn patterns (yellow), and route descriptors with left and right turn patterns (dark green).

a low likelihood of being correct (dark blue); at the same time, the figure shows a small number of disparate locations that have a high likelihood (dark red). This reflects the lack of distinctiveness of using 8 bits of BSD (or two 4-bit BSDs), which is equal to moving around 20 metres. In contrast, as displayed in the middle row of Figure 6.12, once 24 locations were reached, the route had been successfully localised; this time, the vast majority of other locations (or routes) have been discarded (their markers are not shown). Once localised, we continued tracking by moving only one location per time and used the fixed-length query; in this case, is 24 as equal to where the localisation confirmed. The bottom row of Figure 6.12 illustrates that, at 48 locations reached, or approximately 480 metres, the test route was still localised. This reflects the confidence of the localisation using BSDs.

6.3.2 Impact of urban patterns over BSDs

This experiment aimed to strengthen the general use of BSDs for localisation. As in the previous experiments (Sections 5.6 and 6.3.1), the test data we used were from areas within the United Kingdom. It can be argued that the trained network might not only focus on the simple elements, such as junctions and gaps, but also includes other factors, such as the architectural styles. Therefore, we measured the impact of the selected semantic features over the variation of urban areas using data outside the training sets. Specifically, we randomly obtained 100 routes in five different cities outside the United Kingdom: New York, Washington DC, Paris, Madrid, and Rome. The size of each map ranged from approximately

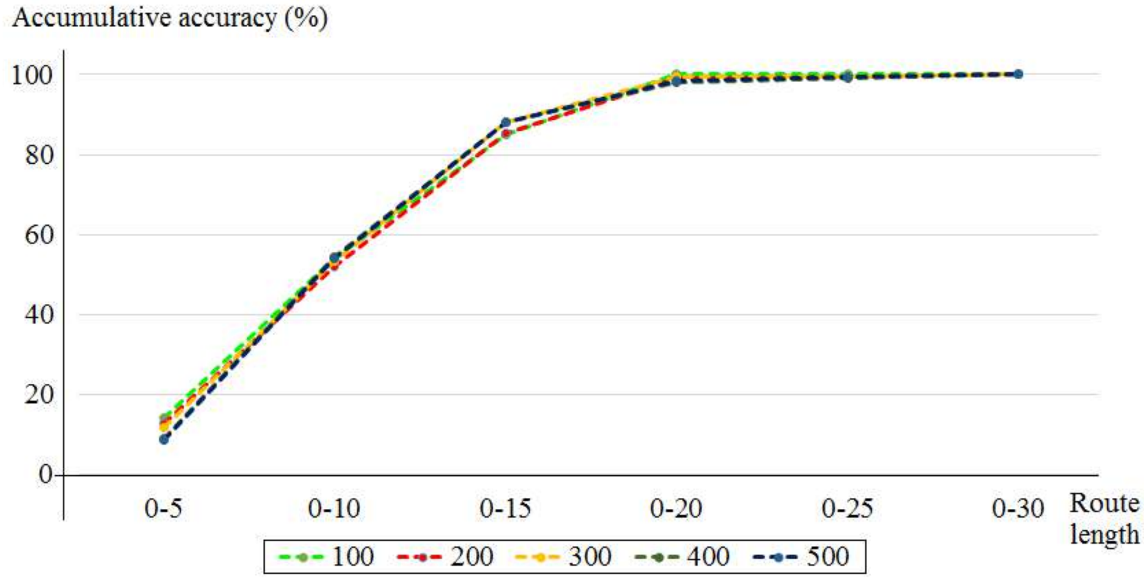


Figure 6.11: Accumulative accuracy of localisation (% of correctly identified routes) versus route length for 100, 200, . . . , 500 test routes. This shows the stability in our performance, as increasing the number of test routes did not affect the accuracy.

1–2 km^2. The other processes were the same as in the previous experiments.

Figure 6.14 and Table 6.2 present the results of applying BSDs for localisation in the stated cities. Over 75% of test routes could be localised within 200 metres, and the average distance required before being successfully localised in a 1–2 km^2 map was around 100 metres. We intend to observe the 2-D map using the histogram of sixteen 4-bit BSD patterns, as depicted in Figure 6.15. The results present some distinctive patterns; however, they are not sufficient for further analysis.

We therefore inspected the images in Figure 6.13, along with the data in Table 6.2. It is as expected that cities in Europe, such as Paris, Rome, and especially Madrid, yielded better accuracy for detecting gaps between buildings than cities in the United States. These scenes in Europe contained architectural styles similar to those in our training sets. Compared to the other two, the Rome set demonstrated lower accuracy due to the high number of transient objects, in this case vehicles. At the same time, the Madrid set showed the most accuracy for detecting gaps between buildings. We believe that this was due to the greater similarity with our training data regarding architectural style. For cities in the United States, the New York set yielded high accuracy over junction detection. This might have been caused by the well-defined road structure, as shown in Figure 6.13; the result is also consistent with the work in [240], which used the dataset from the same city. For the Washington DC set, the selected area contained several grade separations, which caused

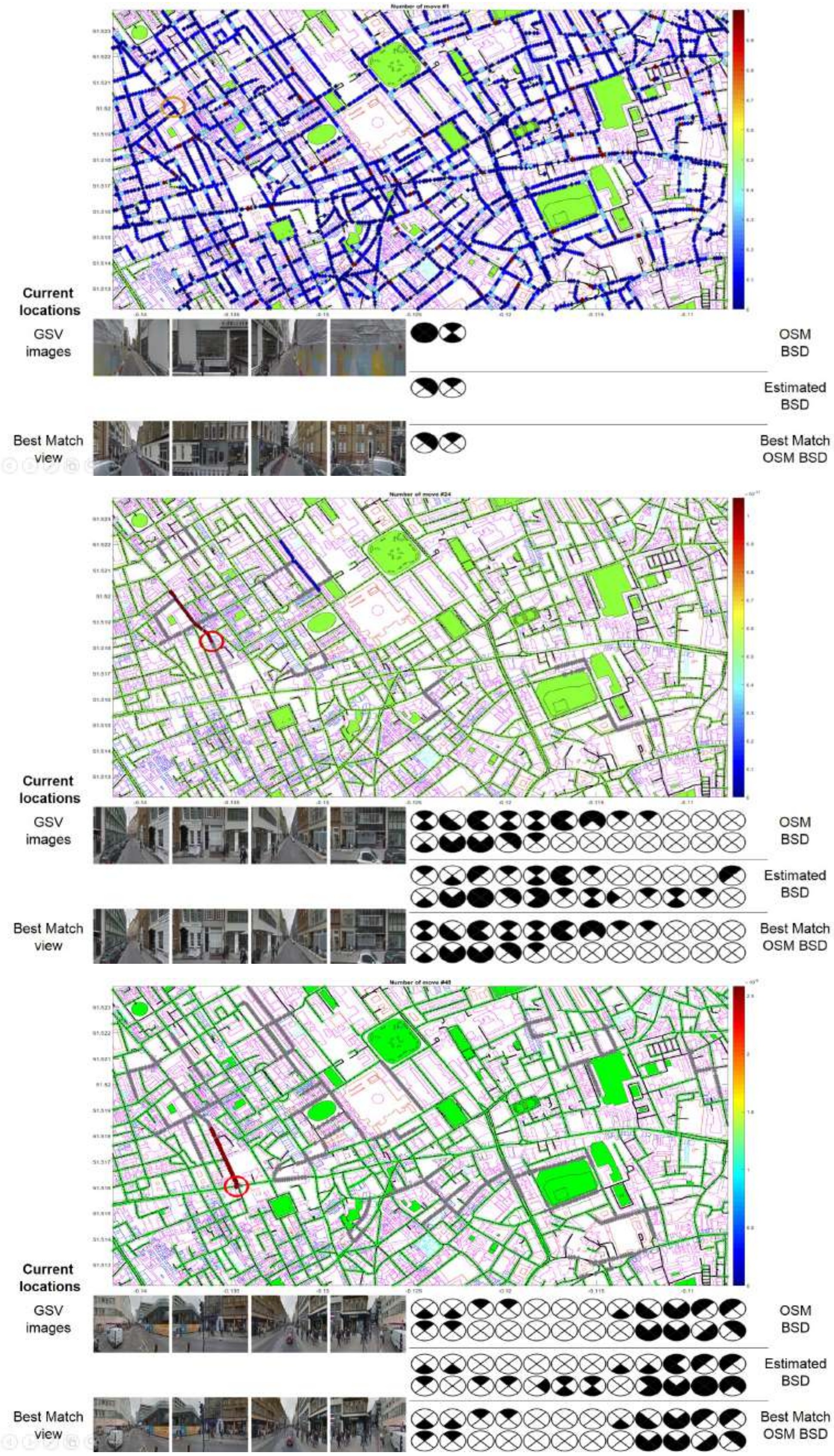


Figure 6.12: Snapshots of the localisation process for test route lengths of 2 (top), 24 (middle), and 48 locations (bottom). More details are provided in the text.

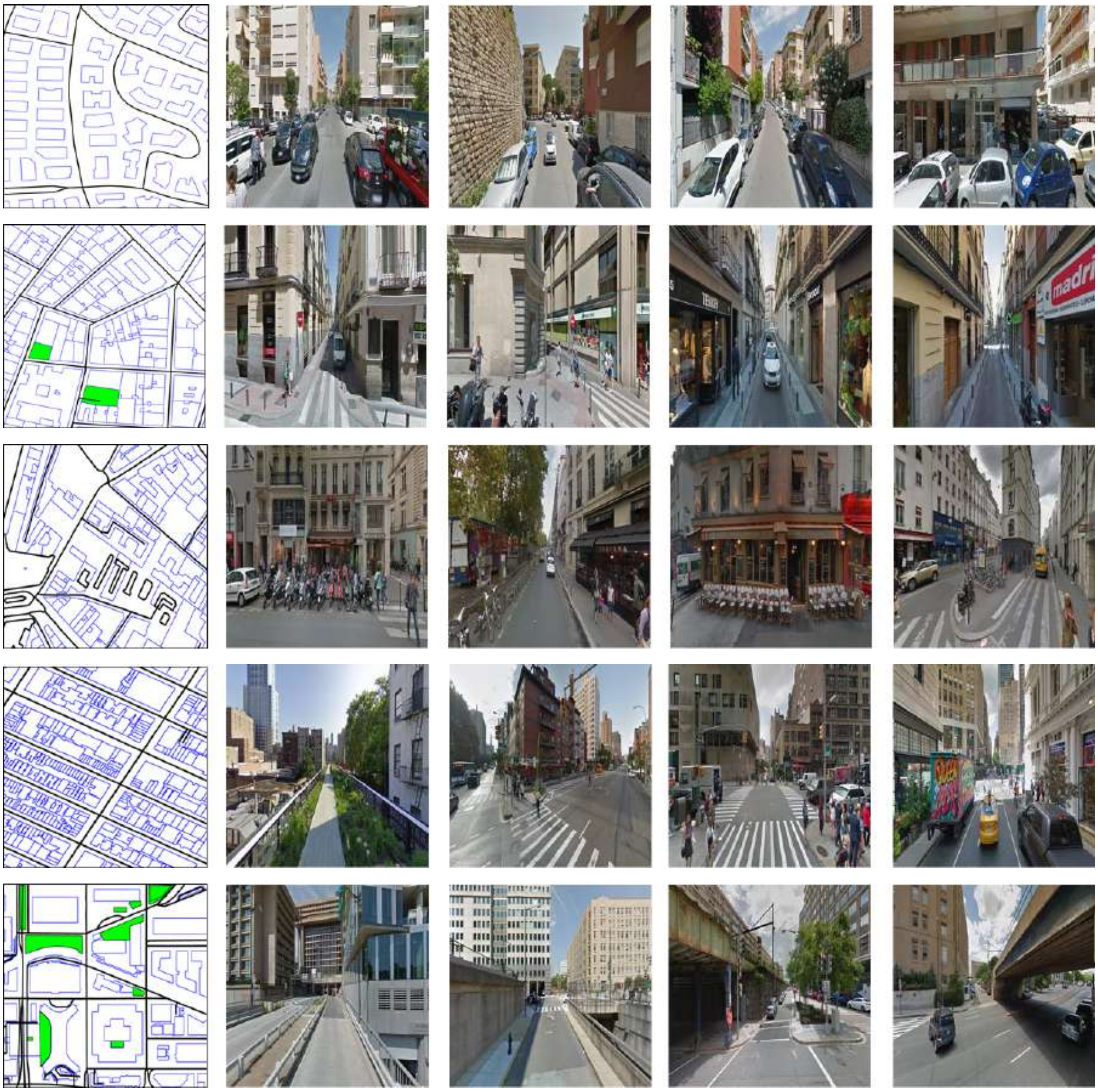


Figure 6.13: Examples of map patches and scenes captured from five different cities – from top to bottom are Rome, Madrid, Paris, New York, and Washington DC. Upon observation, each city contained different variations. For example, New York contained a more well-defined road structure than others. In Rome, there were various transient objects, such as vehicles. The selected area in Washington DC contained several grade separations. We believe that these properties affected the results presented in Table 6.2 and Figure 6.14.

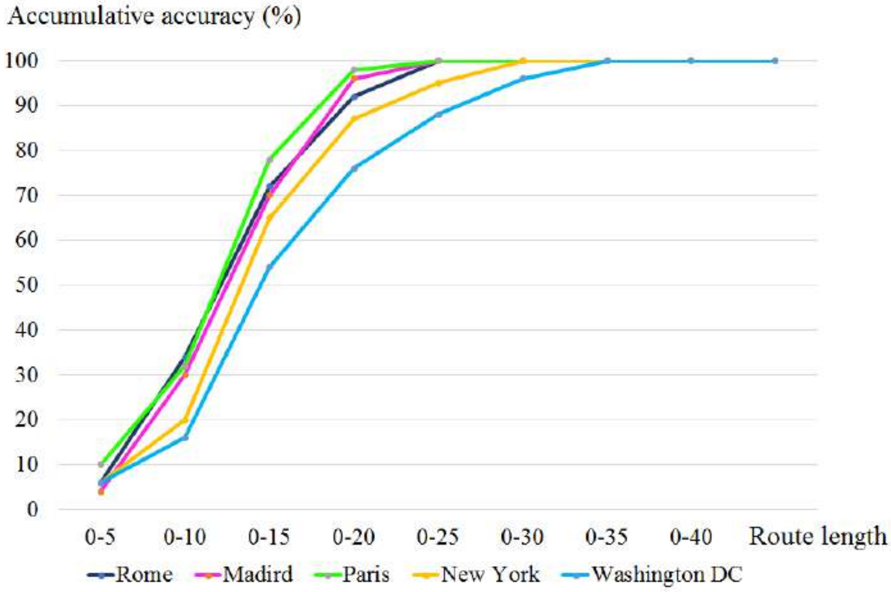


Figure 6.14: Accumulative accuracy of localisation (% of correctly identified routes) versus route length of five different cities, which indicates the consistency of our system in terms of performance.

detection error, as our training datasets did not contain this type of data.

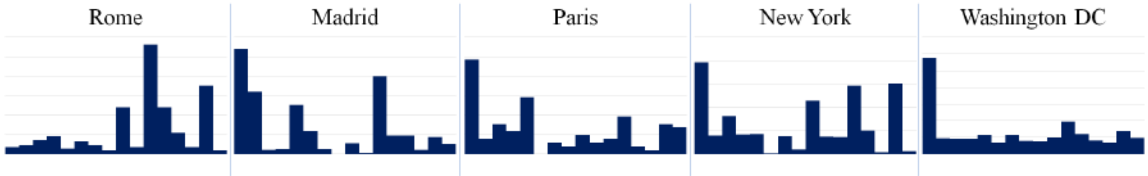


Figure 6.15: The histograms of sixteen patterns of ground-truth BSD of five different cities outside the UK. There is some variation in the patterns; however, it is not enough to warrant further analysis.

Therefore, this experiment initially presents the impact of variation in urban patterns. The findings revealed that the architectural styles had more impact on the gaps between buildings, while the road segments had more impact on the junctions. It can also be implied that if we have a well-trained network specifically for global urban patterns, BSDs can generally be used anywhere. Results supporting this statement were observed in the Paris set. If we compare these results to the those in Chapter 3, even in a city with a high level of recognition difficulty such as Paris, using BSDs appeared to be effective. We believe this reflects improvement due to using the 2-D map, as it is more tolerable to changes. This assumption will be investigated in future work.

Table 6.2: The percentage of accuracy of junctions and gaps between buildings classifiers applied to six different cities and their average distances required before being localised; five of these were taken outside the United Kingdom. Note that the London set is the same dataset we used in the previous tests and is incorporated in this table only for comparison purposes.

	Junction (%)	**Gaps (%)**	**AVG Distance (Metres)**
London	75.35	76.24	97.2
Rome	70.35	72.58	81.2
Madrid	72.14	**84.97**	83.6
Paris	**78.27**	69.44	**70.6**
New York	**78.24**	53.33	161.7
Washington DC	71.29	57.87	117.4

6.3.3 Image-to-2-D-map localisation using BSD in real-world environment

The experiment in this section aims to test our system using real-world data. In the previous experiments, we assumed the one-to-one relationship between images and road locations on the 2-D map. However, we cannot apply the same to a real-world environment without using positioning systems. Therefore, for this experiment, we initially deployed the use of a dense 2-D map that created the one-to-many relationships between images and road locations.

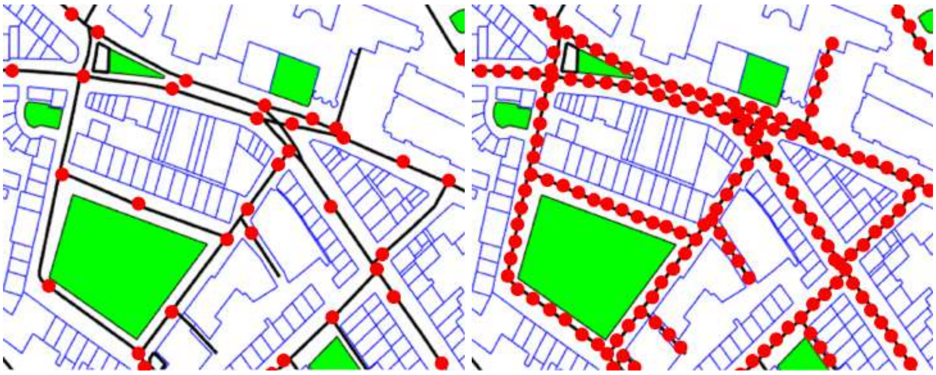

(a) (b)

Figure 6.16: Concept of (a) normal space interval (10–15 metres) and (b) dense space interval (1–2 metres). By applying the dense version, the relationship between road locations and images is not one-to-one, which makes the problem more challenging.

To generate a densely plotted 2-D map, we changed the space of the regular interval between locations from a 10–15 metre range to a 1–2 metre range; these road locations were not paired with any captured Google Street View images, so the relationship between road locations and images was no longer a one-to-one relationship. The other processes, such as feature extraction or turn pattern generation, remained the same. Figure 6.16 illustrates a conceptualisation of the difference between normal and dense versions. We generated the database of the dense version by constructing sets of routes in a mixed range of distances from 8–16 metres. Note that this method may not be entirely practical for the real-world system, especially in terms of scalability. Further investigation is needed in future works.

We performed the experiment in a 2 km^2-range map of Bristol, and twelve 40-location routes on the 2-D map were randomly selected. For data gathering, we used a smartphone camera with a 720×1280 pixel resolution to capture four directional images at the same frontal angles as the Google Street View images. To add more variation, we included a Bing Streetside dataset in our test. Images in the dataset were gathered using a technique similar to that mentioned in Section 3.2.1. Figure 6.17 presents examples of images gathered from Google Street View, Bing Streetside, and the real-world environment. These scenes contain some degree of environmental change, such as occlusions, as well as the presence of transient objects, such as vehicles and pedestrians. We then followed the same processes as the previous experiments. However, as the relationship between images and road locations was not one-to-one, we could not apply the previous measurement. To evaluate the performance, we retrieved the latitude and longitude from the last location in the test image sequence[2] and compared them to the coordinates from a predicted current location given by our algorithm. We applied the Haversine formula [256] to calculate the distance between the two. If it was within a 5-metre range, we counted it as correctly localised.

The results reveal that, for each dataset — Google Street View, Bing Streetside, and the real-world data – all of the test routes could be localised within a 400-metre range, and 80% of them successfully localised at around 13.91 metres, 16.91 metres, and 24.4 metres, respectively. On inspection, as shown in Figure 6.17, despite these scenes containing some environmental changes, our classifier was still able to detect the trained characteristics. In addition, for the real-world dataset, the average classifier accuracy for detection of both junctions and gaps between building categories was around 70%. This number indicates that our classifiers can work on real-world data. These results are still consistent with that in Section 5.6; however, they are only provisional results, and further work is needed.

[2] We did not use this geo-location information in the experiment

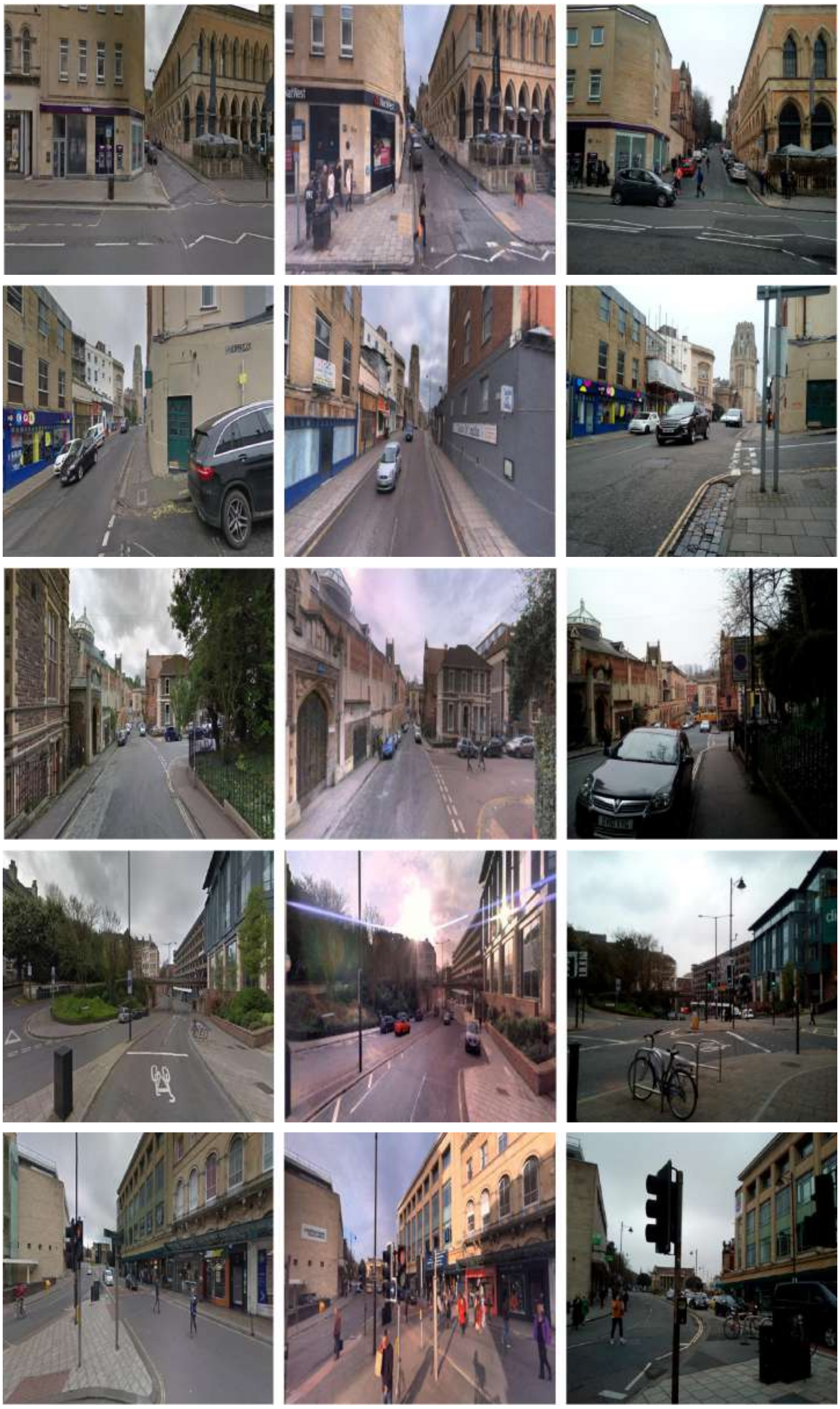


Figure 6.17: Examples of data taken from Google Street View, Bing Streetside and the real-world environment (left to right). The variations in the data, such as lighting condition, shadow, and transient objects, are clearly shown.

6.4 Summary

In sum, we have presented a novel method for localisation using BSD, and experimental results obtained using images from Google Street View and 2-D maps from OpenStreetMap indicate the considerable potential of using this approach. Specifically, we achieved a localisation accuracy of over 80% when using routes consisting of 20 or more locations (approximately 200 meters) on an area within a 2 km^2 range. Although the process of localisation was delayed at the initial state due to the expansion of the route, once bootstrapped to the correct location, our method successfully tracked the route at the same rate as location images were captured. It achieved this using a significantly smaller database than what is required in image-to-image database matching. The results suggest that the method has considerable potential. We extended the tests over different urban patterns and real-world data, the findings of which seem promising. We discuss further limitations and future possibilities for this work in Chapter 7.

Chapter 7

Conclusions and discussion

In this thesis, we separated our work into two parts. First, we investigated the effect of enforcing spatial knowledge combined with salient regions for vision-based place recognition (Chapters 2–3). Second, we explored the use of a spatial organisation of junctions and gaps between buildings consistent with the 2-D map for localisation (Chapters 4–6). The following section summarises the overall ideas of our works and findings, while Section 7.2 discusses the limitations and future directions of this thesis.

7.1 Conclusions and findings

In this section, we discuss each contribution and conclude our findings.

- We investigated the impact of incorporating spatial distribution in place recognition. We proposed the use of image descriptors that encode the spatial distribution of objects in the scene. By applying semantic approaches, we aimed to handle the invariance (temporal changes, spatial changes) and scalability of the system.

First, in Chapter 2, we reviewed the earlier and recent aspects of place recognition. The traditional feature-based approaches dominated the earlier state of image-to-image matching studies; however, the techniques were limited by problems of invariance and scalability. Therefore, we shifted to semantic approaches by choosing the salient landmarks that enforce spatial knowledge. In Chapter 3, we introduced image descriptors that record regions of interest in the scene and their spatial order, called landmark distribution descriptors (LDDs). To evaluate the performance, we compared our proposed method to state-of-the-art methods. In the experiments on ten image-pair datasets, we recorded the average precision of around 70% at 100% recall. Compared with 54% obtained using whole-image CNN features and the method in [62], we believe that this improvement directly resulted from our method. Furthermore, as the datasets contained environmental changes, our system revealed some robust degrees over invariance. Using LDDs partially addressed the issue of scalability, as the compressed descriptor is smaller than the state-of-the-art methods.

However, to operate in the real-world, more compact and robust descriptors were necessary, and we were interested in making the system more human friendly. Indeed, humans are more adept at perceiving visual cues from their surroundings and orienting themselves on a 2-D map. As a result, we shifted from an image-to-image matching database to localisation by matching between images and data in a 2-D map, which leads to our second contribution:

- We investigate the possibility of localisation on a 2-D map using visual information. We introduce the novel technique of using semantic representations. Therefore, we aim for a representation that is robust to invariance (temporal changes, spatial changes, and domain changes), compact for scalability, and closer to how humans perceive the problem.

Second, in Chapter 4, we reviewed the use of the map and the current works in cross-view localisation. This problem was challenging in nature due to the drastic changes in test and reference views. In Chapter 5, we proposed the use of 4-bit binary semantic descriptors (BSDs) and demonstrated the possibility of using BSDs for localisation on the 2-D map. The findings revealed that it was possible to localise on a 2-D map within a 2 km^2 range if the accuracy of our estimated BSDs was greater than 70%. Therefore, in Chapter 6, we integrated the image classifiers to convert four directional images to a BSD. Using junctions and gaps between buildings as our BSD features, we yielded around 75% accuracy, which was sufficient for localisation to proceed. The results indicate that our system can localise around 80% of the test routes within 200 metres, even in the real-world environment. In this respect, we have advantages in three key aspects. First, based on our dataset, our system copes well with several types of invariance: temporal changes, spatial changes, and domain changes. Second, as we use a 4-bit representation, we gain robustness over scalability; our BSD is more compact and requires less storage than image-to-image approaches. Finally, this business over the 2-D map is closer to how humans perceive the localisation problem.

7.2 Limitations and future work

Although the two topics in this thesis share the same concept of investigating the impact of spatial knowledge and semantic features, based on the contents, there are various differences. Therefore, this discussion is separated into two sub-parts.

The following are issues of concern and indicate the future direction of using landmark distribution descriptors:

- **Improve processing time** - based on the current performance, to make our method more practical, we need to improve the processing time. One solution is to convert our code from MATLAB to more lightweight programming tools.

- **Increase the displacement between the image pair** - initially, we specified the displacement between an image pair (Google Street View and Bing Streetside) as 5–10 metres apart, because adding more displacement than that might destroy the salient region pairs, as shown in Figure 7.1. In future work, we will extend our investigation by increasing the degree of displacement to observe the tolerance over lost landmark pairs.

- **Add more datasets** - numerous datasets (including videos) have been provided for evaluating place recognition system, such as the Tokyo 24-7 dataset [51], Nordland dataset [112], and Pittsburgh Street View dataset [268]. In the future, we can apply our method to see how it copes with these different areas and environmental changes.

- **Remove non-static landmarks** - regarding the failure of test results in Section 3.2.4, one cause of this was the presence of non-static objects in the scene, such as cars and trees. This limitation resulted from the proposal detection technique returning all regions likely to be objects in the scene, including non-static ones. Therefore, a more specific proposal detector is necessary. It could be possible to replace the process of proposal detection with learning-based models, such as R-CNN [148], and train them to detect only static objects.

- **Change the viewing direction** - our method may have limitations with viewing direction changes. In Section 3.2.5, we demonstrated the use of a vanishing position to better align the panoramic sections; it helped increase the invariance of small degrees of viewing angle changes (around 30°). However, the larger changes in viewing direction (over 180°) or reverse direction are of greater concern. To solve this, it could be possible to apply the learning-based model, as in [169]; as discussed in Section 3.1.1, some situations might not cope well with this technique. The other solution is to record a pair of images taken from the same location in the reverse direction, but more data means more space required. In the long run, it is more practical to have a system that can auto-generate a scene descriptor to record landmarks and spatial relationships in all directions.

- **Add more evaluation methods** - previously, we only used precision and recall to evaluate the performance of our method. For a better analysis, we might add more evaluation matrices for image retrieval, such as the receiver operating characteristic (ROC) and the mean average precision (mAP). In addition to the method of evaluation, we might also add a more recent method of place recognition for comparison, such as the NetVLAD network [52], which demonstrates high potential with severe changes in viewpoints and environmental conditions.

- **Add degree of belief** - in our experiments, there were no false-positive results, as we assumed that all test images had matches. However, to make the system more practical, it should give the degree of belief indicating the relevance of the query. In addition, in [91], researchers proposed the future concept of vision-based place recognition that systems should be able to correct the results, instead of avoiding false matches. This could be an additional aim for future research.

- **Replace with learning-based models** - rather than detecting salient regions and enforcing their spatial arrangement, it could be possible to re-implement the whole system in the form of the learning-based model. For example, in [269], researchers applied the CNN model to identify and extract the spatial relationship between salient objects in the scene using the labels above, below, and beside. It could be possible to apply the same concept to our work.

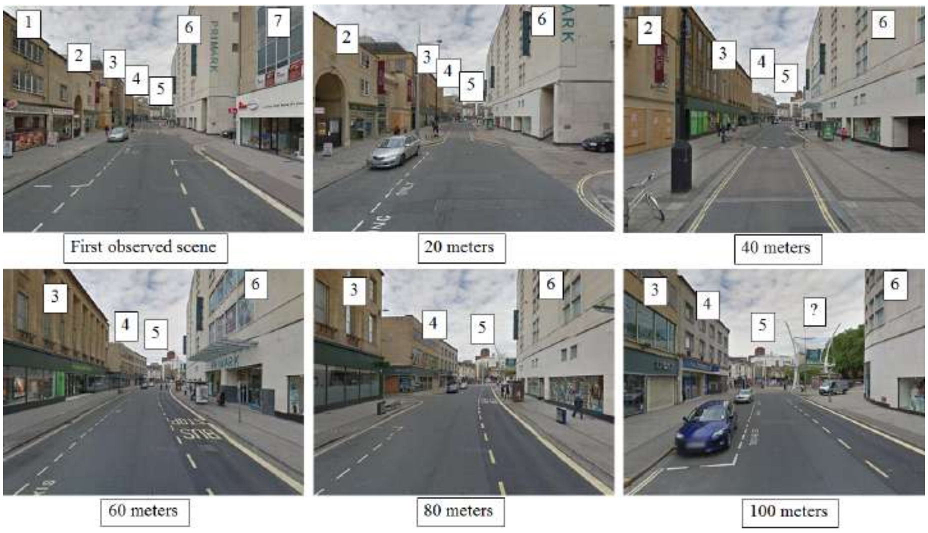


Figure 7.1: Examples of scenes captured at the same location with varying distances from 20–100 metres. Landmark loss is visible in the scene. How far can our system maintain full functioning?

Next, we discuss the future direction of localisation using binary semantic descriptors. Aside from code-level optimisation, the following are issues of concern:

- **Improve feature classifiers** - there are several possible areas of improvement regarding the performance of our classifiers. For example, as we initiate converting images to the BSD using two separate classifiers, we can apply the multi-labels classifier instead. Other possibilities involve boosting operations, such as noise removing, parameter tuning, and further data gathering. Additionally, as we assume the relationship between images and the 2-D map as one-to-one, it is also possible to extend the method to perform sequence-to-sequence classification. One method for this would be to integrate the CNN model with other sequential models, such as recurrent networks [270, 271].

- **Improve data structure** - in the main experiment, we assumed one-to-one correspondence between images and a 2-D map (Google Street View and OpenStreetMap). Then, in Section 6.3.3, we demonstrated the use of the one-to-many approach, the results of which suggest the potential to maintain performance even when extending the method. However, the extension means more road locations on the 2-D map; even with the dynamic database, this can still affect memory consumption. Therefore, future work should consider how the route databases are constructed, stored, and accessed.

- **Improve features selection** - our selected features, the presence of junctions and gaps between buildings, were hand selected. They derived from experiments with numerous features. Therefore, it was difficult to evaluate the effectiveness of these selections. To make the system more robust, mechanisms to auto-generate features are suggested. One way to do this is to apply a learning-based model to identify global features. For example, in one study, a set of image patches was trained to define the characteristics of Paris (inside and outside the city) [136]. In this respect, our system should be able to define salient features without human guidance.

- **Add data with more variation** - the degrees of invariance in our experiment did not cover some environmental changes, such as drastic changes in time of day (day-night) and some urban properties, such as grade separations, partially demonstrated in Section 6.3.2. It is likely that our classifiers may not have fully performed in the stated situations because we have never trained them using those data before. Thus, we might need to add more variation in the training sets. For example, we can retrieve the day-night time data from the Tokyo 24-7 dataset [51].

In addition, for 2-D maps, it could be possible to obtain more map data from other map service providers, such as Mapillary[1], and OpenStreetCam[2]. For images, in the city variation experiment in Section 6.3.2, we chose cities in the United States and Europe, which still contained some architectural styles in common with our training data. Therefore, we might expand our investigation to different geographical areas.

- **Incorporate with a formal framework** - at the initial state of work, we used the simple searching method and assumed movement in fixed locations, moving from one location to the adjacent location. This leads to the main limitation of our method, which is a lack of formal framework. To improve this, we might incorporate a probabilistic filter, such as the Kalman filter [217, 218] and particle filter [221] reviewed in Chapter 4. Based on our current setting, which works with a discrete and fixed number of locations, the filter would be simple. However, in a real-world environment full of uncertainty, the probabilistic approaches would provide more robustness.

- **Integration in real-world applications** - This work has demonstrated that the use of semantic features encoded in a simple location descriptor (BSD) leads to impressive localisation results when the descriptors are concatenated over time. We have demonstrated this in a proof-of-principle system and presented results using GSV data. The next step would be to take these ideas and develop them into real-world applications. Key examples include navigation and path planning and in these cases it would make sense to embed the ideas into a formal decision making framework based on Gauss-Markov modelling, for example [272]. It would also be interesting to look at how they may also be embedded into a learning framework, in which decisions are made based past experience, particularly those employing recent neural network techniques such as Long short-term memory (LSTM) or similar [273].

  In addition, for a more human-friendly system, we propose replacing the 2-D vector maps with the corresponding sketch version, as partially proposed in [153, 158, 159, 160, 274]. One solution is to add more mechanisms to convert a sketch map into a digital base map, such as converting a raster image floor plan into a vector-graphics representation, as in [275]. However, as mentioned in [153], this problem is more challenging because of the information provided in the sketch map; i.e., people usually discard some information that they feel is unrelated to the individual perception.

[1] https://www.mapillary.com/app

[2] https://openstreetcam.org

Another real-world integration may be to combine our method with traditional map-making processes. As discussed in Section 4.1, constructing digital 2-D maps requires human involvement. Rather than performing localisation, we are looking for 2-D map creation based on the semantic features of given images. This also partially relates to our development of the SLAM system in terms of map building.

In sum, though there were limitations to this work, the investigation regarding spatial knowledge in vision-based place recognition and localisation presented in this thesis has achieved the aims of this research, as well as opened up future research directions.

Appendix A

Probabilistic formulation

Base on the binary semantic descriptor we presented in Chapters 5 and 6, it was possible to convert the Hamming distance metric into the probabilistic form, which was applied in the visual demonstration in Section 6.3.1. We applied the probabilistic formulation to indicate the likelihood of possible locations. Thus, given an estimated BSD $\hat{d}$ obtained at a single location l, the conditional probability that l corresponds to $l_i \in \mathcal{L}$ can be written as

$$P(l_i|\hat{d}) = P(l_i|d_i)P(d_i|\hat{d}) \propto P(l_i|d_i)P(\hat{d}|d_i) \tag{A.1}$$

where we assume that all descriptors d_i are equally likely. Note that the term $P(l_i|d_i)$ expresses the uniqueness of the ground-truth descriptor d_i derived from the 2-D map. Since our descriptors were only 4 bits long, for a large number of locations, such as with our largest test set of approximately 6,000 locations, $P(l_i|d_i) << 1$, indicating that many locations had the same descriptors and localisation was therefore not possible.

Given that we had an estimate of the accuracy of our classifiers and hence the detectors $DETECT_{JUNC}$ and $DETECT_{BGAP}$, we could approximate the likelihood $P(\hat{d}|d_i)$ in terms of the Hamming distance h between d_i and $\hat{d}$, i.e.

$$P(\hat{d}|d_i) \propto q^{4-h}(1-q)^h \tag{A.2}$$

where q is the probability of correctly detecting the presence or not of both junctions and gaps. In this case, we assumed the same value for both probabilities for simplicity and we also observed similar values in practice of ≈ 0.75.

Extending the above to routes, we obtained the following conditional probability that the route descriptor estimate $\hat{s} = (\hat{d}_1, \hat{d}_2, \ldots, \hat{d}_{N_r})$ corresponds to route $r \in \mathcal{R}_{N_r}$

$$P(r|\hat{s}) = P(r|s)P(s|\hat{s}) = P(r|s)P(\hat{s}|s) \tag{A.3}$$

Hence, from Equations (A.1) and (A.2) and assuming independence between descriptors

$$\begin{aligned} P(r|\hat{s}) &\propto P(r|s)\prod_{i=1}^{N_r} P(\hat{d}_i|d_{\gamma(i)}) && \text{(A.4)} \\ &\propto P(r|s)q^{4N_r-H}(1-q)^H && \text{(A.5)} \end{aligned}$$

where H denotes the Hamming distance between s and $\hat{s}$. Here, $P(r|s)$ expresses the uniqueness of the route descriptor s, which, as we demonstrated, is high for a sufficiently long routes and thus $P(r|s) \to 1$, giving

$$P(r|\hat{s}) \propto q^{4N_r-H}(1-q)^H \tag{A.6}$$

Using this expression, we could obtain an estimate of the likelihood ratio of one route r_i over another r_j for a given $\hat{s}$

$$\frac{P(r_i|\hat{s})}{P(r_j|\hat{s})} = \frac{(1-q)^{H_i-H_j}}{q^{H_i-H_j}} \tag{A.7}$$

where H_i is the Hamming distance between s_i and $\hat{s}$.

Hence, for $q = 0.75$, this yielded a likelihood ratio of $1/3^\delta$ for a difference of δ in Hamming distance from the estimated route descriptor, which is significant. For example, a route whose descriptor is δ bits closer in Hamming distance to the estimated descriptor is 3^δ times more likely to be the correct route. This further validates our simulation in Section 5.6.3; even with a detector accuracy of only around 75% for individual BSDs, the concatenation of descriptors along routes can lead to a high degree of distinctiveness for long enough routes.

Bibliography

[1] logotire. Image of apple watch (white). https://www.logotire.com/stock-graphic/45146-apple-watch-psd.html, N/A. Online; accessed March 2019.

[2] Hyjiyastore. Image of apple watch (black). https://www.hyjiyastore.com/product/apple-watch-series-2-mp062-42mm-aluminum-case-black-sport-band/, N/A. Online; accessed March 2019.

[3] Nicetoknow.wiki. Image of drone. http://nicetoknow.wiki/2015/05/13/this-camera-drone-flies-itself/, 2015. Online; accessed March 2019.

[4] Glass. Image of smart glass. https://www.x.company/glass/, N/A. Online; accessed March 2019.

[5] GoogleSelfDrivingCarProject. Image of google self-driving car. https://www.youtube.com/channel/UCCLyNDhxwpqNe3UeEmGHl8g, 2017. Online; accessed March 2019.

[6] Robotics.com. Image of sandot, humanoid robot from lg. https://www.robotics.com.hk/product/sanbot-max/, N/A. Online; accessed March 2019.

[7] Javier Gonzalez-Jimenez, JR Ruiz-Sarmiento, and Cipriano Galindo. Improving 2d reactive navigators with kinect. In *10th International Conference on Informatics in Control, Automation and Robotics (ICINCO)*, 2013.

[8] Scale. Human-powered pixel-level image segmentation and annotation by api. https://scale.ai/semantic-segmentation, 2019. Online; accessed May 2019.

[9] kurzweilai. Image of google self-driving car's map. http://www.kurzweilai.net/googles-self-driving-car-gathers-nearly-1-gbsec, 2013. Online; accessed March 2019.

[10] Wikipedia. Image of london tower bridge. https://en.wikipedia.org/wiki/Tower_Bridge, 2015. Online; accessed March 2019.

[11] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. Imagenet classification with deep convolutional neural networks. In *Advances in neural information processing systems*, pages 1097–1105, 2012.

[12] Yi Hou, Hong Zhang, and Shilin Zhou. Convolutional neural network-based image representation for visual loop closure detection. In *Information and Automation, 2015 IEEE International Conference on*, pages 2238–2245. IEEE, 2015.

[13] Niko Sünderhauf, Sareh Shirazi, Feras Dayoub, Ben Upcroft, and Michael Milford. On the performance of convnet features for place recognition. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 4297–4304. IEEE, 2015.

[14] Joe Yue-Hei Ng, Fan Yang, and Larry S Davis. Exploiting local features from deep networks for image retrieval. In *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*, pages 53–61, 2015.

[15] Qiang Liu, Ruihao Li, Huosheng Hu, and Dongbing Gu. Extracting semantic information from visual data: A survey. *Robotics*, 5(1):8, 2016.

[16] C Lawrence Zitnick and Piotr Dollár. Edge boxes: Locating object proposals from edges. In *European conference on computer vision*, pages 391–405. Springer, 2014.

[17] Ella Bingham and Heikki Mannila. Random projection in dimensionality reduction: applications to image and text data. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 245–250. ACM, 2001.

[18] Andrea Arcá. Le raffigurazioni topografiche, colture e culture preistoriche nella prima fase dell'arte rupestre di paspardo. *AE Fossati (a cura di), La Castagna della Vallecamonica. Paspardo, arte rupestre e castanicoltura*, pages 35–56, 2007.

[19] Esri. Image of basemap. https://www.esri.com/en-us/home, N/A. Online; accessed June 2019.

[20] Travel Wayfinding. Image of you are here map. https://www.travelwayfinding.com/sales/you-are-here-maps/, N/A. Online; accessed March 2019.

[21] Alexander W Siegel and Sheldon H White. The development of spatial representations of large-scale environments. In *Advances in child development and behavior*, volume 10, pages 9–55. Elsevier, 1975.

[22] Artphototravel. Image of bangkok floating market map. http://www.artphototravel.net/portfolio/floating-markets-bangkok/, N/A. Online; accessed March 2019.

[23] Mapofworld. Image of map of world. http://www.freeworldmaps.net/, N/A. Online; accessed March 2019.

[24] Tubemaplondon. Image of tube map london. http://www.tubemaplondon.org/, 2018. Online; accessed March 2019.

[25] UoB. Image of university of bristol map. https://people.maths.bris.ac.uk/m̃atyd/BMC/UoB-map.pdf, N/A. Online; accessed March 2019.

[26] Wesley H Huang and Kristopher R Beevers. Topological mapping with sensing-limited robots. In *Algorithmic Foundations of Robotics VI*, pages 235–250. Springer, 2004.

[27] Thomas Whelan, Michael Kaess, John J Leonard, and John McDonald. Deformation-based loop closure for large scale dense rgb-d slam. In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 548–555. IEEE, 2013.

[28] Wolfgang Hess, Damon Kohler, Holger Rapp, and Daniel Andor. Real-time loop closure in 2d lidar slam. In *2016 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1271–1278. IEEE, 2016.

[29] Pierre Moulon, Pascal Monasse, Romuald Perrot, and Renaud Marlet. Openmvg: Open multiple view geometry. In *International Workshop on Reproducible Research in Pattern Recognition*, pages 60–74. Springer, 2016.

[30] Wikipedia. Image of the rq-16 t-hawk, a micro air vehicle. https://en.wikipedia.org/wiki/Micro_air_vehicle, 2006. Online; accessed March 2019.

[31] Wikipedia. Image of a mq-9 reaper us military unmanned aerial vehicle. https://en.wikipedia.org/wiki/Unmanned_aerial_vehicle, 2007. Online; accessed March 2019.

[32] Yingfeng Chen, Feng Wu, Wei Shuai, Ningyang Wang, Rongya Chen, and Xiaoping Chen. Kejia robot–an attractive shopping mall guider. In *International Conference on Social Robotics*, pages 145–154. Springer, 2015.

[33] Jorge Sales, Jose V Martí, Raúl Marín, Enric Cervera, and Pedro J Sanz. Comparob: the shopping cart assistance robot. *International Journal of Distributed Sensor Networks*, 12(2):4781280, 2016.

[34] Madoka Nakajima and Shinichiro Haruyama. New indoor navigation system for visually impaired people using visible light communication. *EURASIP Journal on Wireless Communications and Networking*, 2013(1):37, 2013.

[35] Jinqiang Bai, Zhaoxiang Liu, Yimin Lin, Ye Li, Shiguo Lian, and Dijun Liu. Wearable travel aid for environment perception and navigation of visually impaired people. *Electronics*, 8(6):697, 2019.

[36] G Balamurugan, J Valarmathi, and VPS Naidu. Survey on uav navigation in gps denied environments. In *2016 International Conference on Signal Processing, Communication, Power and Embedded System (SCOPES)*, pages 198–204. IEEE, 2016.

[37] Randall C Smith and Peter Cheeseman. On the representation and estimation of spatial uncertainty. *The international journal of Robotics Research*, 5(4):56–68, 1986.

[38] Hugh F Durrant-Whyte. Uncertain geometry in robotics. *IEEE Journal on Robotics and Automation*, 4(1):23–31, 1988.

[39] Leo Bagrow. *History of cartography*. Routledge, 2017.

[40] James Ash, Rob Kitchin, and Agnieszka Leszczynski. Digital turn, digital geographies? *Progress in Human Geography*, 42(1):25–43, 2018.

[41] Ahmad Alzu'bi, Abbes Amira, and Naeem Ramzan. Semantic content-based image retrieval: A comprehensive study. *Journal of Visual Communication and Image Representation*, 32:20–54, 2015.

[42] Ioannis Kostavelis and Antonios Gasteratos. Semantic mapping for mobile robotics tasks: A survey. *Robotics and Autonomous Systems*, 66:86–103, 2015.

[43] Arsalan Mousavian, Jana Košecká, and Jyh-Ming Lien. Semantically guided location recognition for outdoors scenes. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 4882–4889. IEEE, 2015.

[44] Shervin Ardeshir, Amir Roshan Zamir, Alejandro Torroella, and Mubarak Shah. Gis-assisted object detection and geospatial localization. In *European Conference on Computer Vision*, pages 602–617. Springer, 2014.

[45] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, and Philip Torr. Bing: Binarized normed gradients for objectness estimation at 300fps. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3286–3293, 2014.

[46] Jianxiong Xiao, Tian Fang, Ping Tan, Peng Zhao, Eyal Ofek, and Long Quan. Image-based façade modeling. In *ACM transactions on graphics (TOG)*, volume 27, page 161. ACM, 2008.

[47] Akihiko Torii, Josef Sivic, Tomas Pajdla, and Masatoshi Okutomi. Visual place recognition with repetitive structures. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 883–890, 2013.

[48] Arnold Irschara, Christopher Zach, Jan-Michael Frahm, and Horst Bischof. From structure-from-motion point clouds to fast location recognition. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 2599–2606. IEEE, 2009.

[49] Titus Cieslewski, Elena Stumm, Abel Gawel, Mike Bosse, Simon Lynen, and Roland Siegwart. Point cloud descriptors for place recognition using sparse visual information. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 4830–4836. IEEE, 2016.

[50] Colin McManus, Ben Upcroft, and Paul Newmann. Scene signatures: Localised and point-less features for localisation. *Robotics: Science and Systems*, 2014.

[51] Akihiko Torii, Relja Arandjelovic, Josef Sivic, Masatoshi Okutomi, and Tomas Pajdla. 24/7 place recognition by view synthesis. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1808–1817, 2015.

[52] Relja Arandjelovic, Petr Gronat, Akihiko Torii, Tomas Pajdla, and Josef Sivic. Netvlad: Cnn architecture for weakly supervised place recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 5297–5307, 2016.

[53] Jorge Fuentes-Pacheco, José Ruiz-Ascencio, and Juan Manuel Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*, 43(1):55–81, 2015.

[54] Mark Cummins and Paul Newman. Fab-map: Probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.

[55] Mark Cummins and Paul Newman. Appearance-only slam at large scale with fab-map 2.0. *The International Journal of Robotics Research*, 30(9):1100–1123, 2011.

[56] Winston Churchill and Paul Newman. Practice makes perfect? managing and leveraging visual experiences for lifelong navigation. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 4525–4532. IEEE, 2012.

[57] Winston Churchill and Paul Newman. Experience-based navigation for long-term localisation. *The International Journal of Robotics Research*, 32(14):1645–1661, 2013.

[58] Edward Johns and Guang-Zhong Yang. Feature co-occurrence maps: Appearance-based localisation throughout the day. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 3212–3218. IEEE, 2013.

[59] Edward Johns and Guang-Zhong Yang. Dynamic scene models for incremental, long-term, appearance-based localisation. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 2731–2736. IEEE, 2013.

[60] Colin McManus, Winston Churchill, Will Maddern, Alexander D Stewart, and Paul Newman. Shady dealings: Robust, long-term visual localisation using illumination invariance. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 901–906. IEEE, 2014.

[61] Roberto Arroyo, Pablo F Alcantarilla, Luis M Bergasa, and Eduardo Romera. Towards life-long visual localization using an efficient matching of binary sequences from images. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 6328–6335. IEEE, 2015.

[62] Niko Sünderhauf, Sareh Shirazi, Adam Jacobson, Feras Dayoub, Edward Pepperell, Ben Upcroft, and Michael Milford. Place recognition with convnet landmarks: Viewpoint-robust, condition-robust, training-free. *Proceedings of Robotics: Science and Systems XII*, 2015.

[63] Aharon Bar Hillel, Ronen Lerner, Dan Levi, and Guy Raz. Recent progress in road and lane detection: a survey. *Machine vision and applications*, 25(3):727–745, 2014.

[64] Davy Neven, Bert De Brabandere, Stamatios Georgoulis, Marc Proesmans, and Luc Van Gool. Towards end-to-end lane detection: an instance segmentation approach. *arXiv preprint arXiv:1802.05591*, 2018.

[65] Jianxiong Xiao and Long Quan. Multiple view semantic segmentation for street view images. In *Computer Vision, 2009 IEEE 12th International Conference on*, pages 686–693. IEEE, 2009.

[66] Rashmi Tonge, Subhransu Maji, and CV Jawahar. Parsing world's skylines using shape-constrained mrfs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3174–3181, 2014.

[67] Gary L Allen. Spatial abilities, cognitive maps, and wayfinding. *Wayfinding behavior: Cognitive mapping and other spatial processes*, 4680, 1999.

[68] Reginald G Golledge et al. *Wayfinding behavior: Cognitive mapping and other spatial processes*. JHU press, 1999.

[69] M Jeanne Sholl. Landmarks, places, environments: Multiple mind–brain systems for spatial orientation. *Geoforum*, 23(2):151–164, 1992.

[70] R Biegler and RGM Morris. Landmark stability is a prerequisite for spatial but not discrimination learning. *Nature*, 361(6413):631, 1993.

[71] Rudolph P. Darken, Barry Peterson, and B. Spatial Orientation. Spatial orientation, wayfinding, and representation. In *In K. M. Stanney (Ed.), Handbook of Virtual Environments: Design, Implementation, and Applications*, pages 493–518. Erlbaum, 2001.

[72] Jia Wang and Michael Worboys. Ontologies and representation spaces for sketch map interpretation. *International Journal of Geographical Information Science*, 31(9):1697–1721, 2017.

[73] Chiara Meneghetti, Francesca Pazzaglia, and Rossana De Beni. Spatial mental representations derived from survey and route descriptions: When individuals prefer extrinsic frame of reference. *Learning and Individual Differences*, 21(2):150–157, 2011.

[74] Michael Milford. Vision-based place recognition: how low can you go? *The International Journal of Robotics Research*, 32(7):766–789, 2013.

[75] Rohan Paul and Paul Newman. Fab-map 3d: Topological mapping with spatial and visual appearance. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 2649–2656. IEEE, 2010.

[76] Felix Endres, Jürgen Hess, Nikolas Engelhard, Jürgen Sturm, Daniel Cremers, and Wolfram Burgard. An evaluation of the rgb-d slam system. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1691–1696. IEEE, 2012.

[77] Mathieu Labbe and François Michaud. Online global loop closure detection for large-scale multi-session graph-based slam. In *Intelligent Robots and Systems (IROS 2014), 2014 IEEE/RSJ International Conference on*, pages 2661–2666. IEEE, 2014.

[78] Andrew J Davison and David W Murray. Simultaneous localization and map-building using active vision. *IEEE transactions on pattern analysis and machine intelligence*, 24(7):865–880, 2002.

[79] Kurt Konolige and James Bowman. Towards lifelong visual maps. In *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, pages 1156–1163. IEEE, 2009.

[80] David G Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2):91–110, 2004.

[81] Herbert Bay, Tinne Tuytelaars, and Luc Van Gool. Surf: Speeded up robust features. In *European conference on computer vision*, pages 404–417. Springer, 2006.

[82] Josef Sivic, Bryan C Russell, Alexei A Efros, Andrew Zisserman, and William T Freeman. Discovering objects and their location in images. In *Computer Vision, 2005. ICCV 2005. Tenth IEEE International Conference on*, volume 1, pages 370–377. IEEE, 2005.

[83] Aude Oliva and Antonio Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *International journal of computer vision*, 42(3):145–175, 2001.

[84] Matthijs Douze, Hervé Jégou, Harsimrat Sandhawalia, Laurent Amsaleg, and Cordelia Schmid. Evaluation of gist descriptors for web-scale image search. In *Proceedings of the ACM International Conference on Image and Video Retrieval*, page 19. ACM, 2009.

[85] Ana Cris Murillo and Jana Kosecka. Experiments in place recognition using gist panoramas. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 2196–2203. IEEE, 2009.

[86] Ana Cristina Murillo, Gautam Singh, Jana Kosecká, and José Jesús Guerrero. Localization in urban environments using a panoramic gist descriptor. *IEEE Trans. Robotics*, 29(1):146–160, 2013.

[87] Christian Weiss, Andreas Masselli, Hashem Tamimi, and Andreas Zell. Fast outdoor robot localization using integral invariants. In *Proceedings of the 5th International Conference on Computer Vision Systems (ICVS). Bielefeld, Germany*, 2007.

[88] Luis Payá, Lorenzo Fernández, Arturo Gil, and Oscar Reinoso. Map building and monte carlo localization using global appearance of omnidirectional images. *Sensors*, 10(12):11468–11497, 2010.

[89] Gautam Singh and J Kosecka. Visual loop closing using gist descriptors in manhattan world. In *ICRA Omnidirectional Vision Workshop*, 2010.

[90] Alejandro Rituerto, AC Murillo, and JJ Guerrero. Semantic labeling for indoor topological mapping using a wearable catadioptric system. *Robotics and Autonomous Systems*, 62(5):685–695, 2014.

[91] Stephanie Lowry, Niko Sünderhauf, Paul Newman, John J Leonard, David Cox, Peter Corke, and Michael J Milford. Visual place recognition: A survey. *IEEE Transactions on Robotics*, 32(1):1–19, 2016.

[92] Abhinav Shrivastava, Tomasz Malisiewicz, Abhinav Gupta, and Alexei A Efros. Data-driven visual similarity for cross-domain image matching. *ACM Transactions on Graphics (ToG)*, 30(6):154, 2011.

[93] James Hays and Alexei A Efros. Im2gps: estimating geographic information from a single image. In *Computer Vision and Pattern Recognition, 2008. CVPR 2008. IEEE Conference on*, pages 1–8. IEEE, 2008.

[94] Charbel Azzi, Daniel C Asmar, Adel H Fakih, and John S Zelek. Filtering 3d keypoints using gist for accurate image-based localization. In *he British Machine Vision Conference*, 2016.

[95] Andrzej Pronobis, O Martinez Mozos, and Barbara Caputo. Svm-based discriminative accumulation scheme for place recognition. In *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pages 522–529. IEEE, 2008.

[96] Andrzej Pronobis, Luo Jie, and Barbara Caputo. The more you learn, the less you store: Memory-controlled incremental svm for visual place recognition. *Image and Vision Computing*, 28(7):1080–1097, 2010.

[97] Chris Linegar, Winston Churchill, and Paul Newman. Made to measure: Bespoke landmarks for 24-hour, all-weather localisation with a camera. In *Robotics and Automation (ICRA), 2016 IEEE International Conference on*, pages 787–794. IEEE, 2016.

[98] Christian Szegedy, Wei Liu, Yangqing Jia, Pierre Sermanet, Scott Reed, Dragomir Anguelov, Dumitru Erhan, Vincent Vanhoucke, and Andrew Rabinovich. Going deeper with convolutions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015.

[99] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[100] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[101] Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. Signature verification using a ”siamese” time delay neural network. In *Advances in neural information processing systems*, pages 737–744, 1994.

[102] Hani Altwaijry, Eduard Trulls, James Hays, Pascal Fua, and Serge Belongie. Learning to match aerial images with deep attentive architectures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3539–3547, 2016.

[103] Iaroslav Melekhov, Juho Kannala, and Esa Rahtu. Siamese network features for image matching. In *Pattern Recognition (ICPR), 2016 23rd International Conference on*, pages 378–383. IEEE, 2016.

[104] Kwang Moo Yi, Eduard Trulls, Vincent Lepetit, and Pascal Fua. Lift: Learned invariant feature transform. In *European Conference on Computer Vision*, pages 467–483. Springer, 2016.

[105] Manuel Lopez-Antequera, Ruben Gomez-Ojeda, Nicolai Petkov, and Javier Gonzalez-Jimenez. Appearance-invariant place recognition by discriminatively training a convolutional neural network. *Pattern Recognition Letters*, 92:89–95, 2017.

[106] Boris Ivanovic. Visual place recognition in changing environments with time-invariant image patch descriptors. Representation Learning in Computer Vision course, Standford university, 2016.

[107] Yan Ke and Rahul Sukthankar. Pca-sift: A more distinctive representation for local image descriptors. In *Computer Vision and Pattern Recognition, 2004. CVPR 2004. Proceedings of the 2004 IEEE Computer Society Conference on*, volume 2, pages II–II. IEEE, 2004.

[108] David G Lowe. Object recognition from local scale-invariant features. In *Computer vision, 1999. The proceedings of the seventh IEEE international conference on*, volume 2, pages 1150–1157. Ieee, 1999.

[109] Will Maddern, Alex Stewart, Colin McManus, Ben Upcroft, Winston Churchill, and Paul Newman. Illumination invariant imaging: Applications in robust vision-based localisation, mapping and classification for autonomous vehicles. In *Proceedings of the Visual Place Recognition in Changing Environments Workshop, IEEE International Conference on Robotics and Automation (ICRA), Hong Kong, China*, volume 2, page 3, 2014.

[110] Stephanie Lowry, Michael Milford, and Gordon Wyeth. Transforming morning to afternoon using linear regression techniques. In *Proceedings of the 2014 IEEE International Conference on Robotics and Automation (ICRA 2014)*, pages 3950–3955. IEEE, 2014.

[111] Michael J Milford and Gordon F Wyeth. Seqslam: Visual route-based navigation for sunny summer days and stormy winter nights. In *Robotics and Automation (ICRA), 2012 IEEE International Conference on*, pages 1643–1649. IEEE, 2012.

[112] Niko Sünderhauf, Peer Neubert, and Peter Protzel. Are we there yet? challenging seqslam on a 3000 km journey across all four seasons. In *Proc. of Workshop on Long-Term Autonomy, IEEE International Conference on Robotics and Automation (ICRA)*, page 2013. Citeseer, 2013.

[113] Edward Pepperell, Peter Corke, and Michael Milford. Towards persistent visual navigation using smart. In *Proceedings of Australasian Conference on Robotics and Automation*. ARAA, 2013.

[114] Edward Pepperell, Peter I Corke, and Michael J Milford. All-environment visual place recognition with smart. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 1612–1618. IEEE, 2014.

[115] S Lowry and MJ Milford. Change removal: Robust online learning for changing appearance and changing viewpoint. *ICRA15 WS VPRiCE*, 2015.

[116] Roberto Arroyo, Pablo F Alcantarilla, Luis M Bergasa, and Eduardo Romera. Fusion and binarization of cnn features for robust topological localization across seasons. In *Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on*, pages 4656–4663. IEEE, 2016.

[117] Ahmet Iscen, Giorgos Tolias, Yannis Avrithis, Teddy Furon, and Ondřej Chum. Panorama to panorama matching for location recognition. In *ACM International Conference on Multimedia Retrieval (ICMR) 2017*, 2017.

[118] Hyo Jin Kim, Enrique Dunn, and Jan-Michael Frahm. Learned contextual feature reweighting for image geo-localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 5, page 8, 2017.

[119] Mao Wang, En Zhu, Qiang Liu, Yongkai Ye, Yuewei Ming, and Jianping Yin. Intensity filtering and group fusion for accurate mobile place recognition. *IEEE Access*, 6:31088–31098, 2018.

[120] DfT Statistical Release. Road lengths in great britain 2016. https://www.gov.uk/government/statistics/road-lengths-in-great-britain-2016, April 2017.

[121] Cesar Cadena, Luca Carlone, Henry Carrillo, Yasir Latif, Davide Scaramuzza, José Neira, Ian Reid, and John J Leonard. Past, present, and future of simultaneous localization and mapping: Toward the robust-perception age. *IEEE Transactions on Robotics*, 32(6):1309–1332, 2016.

[122] Karl Pearson. Liii. on lines and planes of closest fit to systems of points in space. *The London, Edinburgh, and Dublin Philosophical Magazine and Journal of Science*, 2(11):559–572, 1901.

[123] Pierre Comon. Independent component analysis, a new concept? *Signal processing*, 36(3):287–314, 1994.

[124] Panu Turcot and David G Lowe. Better matching with fewer features: The selection of useful features in large database recognition problems. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 2109–2116. IEEE, 2009.

[125] Fatima Binta Adamu, Adib Habbal, Suhaidi Hassan, U Utara Malaysia, R Les Cottrell, Bebo White, Ibrahim Abdullah, U Utara Malaysia, et al. A survey on big data indexing strategies. Technical report, SLAC National Accelerator Lab., Menlo Park, CA (United States), 2016.

[126] Hervé Jégou, Matthijs Douze, Cordelia Schmid, and Patrick Pérez. Aggregating local descriptors into a compact image representation. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 3304–3311. IEEE, 2010.

[127] Alexandra Millonig and Katja Schechtner. Developing landmark-based pedestrian-navigation systems. *IEEE Transactions on Intelligent Transportation Systems*, 8(1):43–49, 2007.

[128] Yu A Rozanov. Markov random fields. In *Markov Random Fields*, pages 55–102. Springer, 1982.

[129] Vijay Badrinarayanan, Alex Kendall, and Roberto Cipolla. Segnet: A deep convolutional encoder-decoder architecture for image segmentation. *arXiv preprint arXiv:1511.00561*, 2015.

[130] Marius Cordts, Mohamed Omran, Sebastian Ramos, Timo Rehfeld, Markus Enzweiler, Rodrigo Benenson, Uwe Franke, Stefan Roth, and Bernt Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3213–3223, 2016.

[131] Kujtim Rahmani and Helmut Mayer. High quality facade segmentation based on structured random forest, region proposal network and rectangular fitting. *ISPRS Annals of Photogrammetry, Remote Sensing & Spatial Information Sciences*, 4(2), 2018.

[132] Olivier Teboul, Loic Simon, Panagiotis Koutsourakis, and Nikos Paragios. Segmentation of building facades using procedural shape priors. *IEEE Conference on Computer Vision and Pattern Recognition*, 2010.

[133] Anand Mishra, Karteek Alahari, and CV Jawahar. Top-down and bottom-up cues for scene text recognition. In *CVPR-IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2012.

[134] Saurabh Singh, Abhinav Gupta, and Alexei A Efros. Unsupervised discovery of mid-level discriminative patches. In *Computer Vision–ECCV 2012*, pages 73–86. Springer, 2012.

[135] Carl Doersch, Abhinav Gupta, and Alexei A Efros. Mid-level visual element discovery as discriminative mode seeking. In *Advances in neural information processing systems*, pages 494–502, 2013.

[136] Carl Doersch, Saurabh Singh, Abhinav Gupta, Josef Sivic, and Alexei Efros. What makes paris look like paris? *ACM Transactions on Graphics*, 31(4), 2012.

[137] Sean M Arietta, Alexei A Efros, Ravi Ramamoorthi, and Maneesh Agrawala. City forensics: Using visual elements to predict non-visual city attributes. *IEEE transactions on visualization and computer graphics*, 20(12):2624–2633, 2014.

[138] Quan Fang, Jitao Sang, and Changsheng Xu. Giant: Geo-informative attributes for location recognition and exploration. In *Proceedings of the 21st ACM international conference on Multimedia*, pages 13–22. ACM, 2013.

[139] Bolei Zhou, Liu Liu, Aude Oliva, and Antonio Torralba. Recognizing city identity via attribute analysis of geo-tagged images. In *European conference on computer vision*, pages 519–534. Springer, 2014.

[140] Stefan Lee, Nicolas Maisonneuve, David J. Crandall, Alexei A. Efros, and Josef Sivic. Linking past to present: Discovering style in two centuries of architecture. *IEEE International Conference on Computational Photography (ICCP)*, pages 1–10, 2015.

[141] Mayank Bansal, Kostas Daniilidis, and Harpreet Sawhney. Ultrawide baseline facade matching for geo-localization. In *Large-Scale Visual Geo-Localization*, pages 77–98. Springer, 2016.

[142] Christian Siagian and Laurent Itti. Biologically-inspired robotics vision monte-carlo localization in the outdoor environment. In *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, pages 1723–1730. IEEE, 2007.

[143] Radhakrishna Achanta, Francisco Estrada, Patricia Wils, and Sabine Süsstrunk. Salient region detection and segmentation. In *International conference on computer vision systems*, pages 66–75. Springer, 2008.

[144] Pedro F Felzenszwalb, Ross B Girshick, David McAllester, and Deva Ramanan. Object detection with discriminatively trained part-based models. *IEEE transactions on pattern analysis and machine intelligence*, 32(9):1627–1645, 2010.

[145] Jasper RR Uijlings, Koen EA Van De Sande, Theo Gevers, and Arnold WM Smeulders. Selective search for object recognition. *International journal of computer vision*, 104(2):154–171, 2013.

[146] Lukáš Neumann and Jiří Matas. Real-time scene text localization and recognition. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3538–3545. IEEE, 2012.

[147] Alvaro Gonzalez, Luis M Bergasa, and J Javier Yebes. Text detection and recognition on traffic panels from street-level imagery using visual appearance. *IEEE Transactions on Intelligent Transportation Systems*, 15(1):228–238, 2014.

[148] Ross Girshick, Jeff Donahue, Trevor Darrell, and Jitendra Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 580–587, 2014.

[149] Ross Girshick. Fast r-cnn. In *Proceedings of the IEEE international conference on computer vision*, pages 1440–1448, 2015.

[150] Ingmar Posner, Peter Corke, and Paul Newman. Using text-spotting to query the world. In *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems 2010*, pages 3181–3186. IEEE, 2010.

[151] Muhammad Sami, Yasar Ayaz, Mohsin Jamil, Syed Omer Gilani, and Muhammad Naveed. Text detection and recognition for semantic mapping in indoor navigation. In *IT Convergence and Security (ICITCS), 2015 5th International Conference on*, pages 1–4. IEEE, 2015.

[152] Daniel R Montello. *Navigation.* Cambridge University Press, 2005.

[153] Klaus Broelemann, Xiaoyi Jiang, and Angela Schwering. Automatic understanding of sketch maps using context-aware classification. *Expert systems with applications*, 45:195–207, 2016.

[154] Shiliang Zhang, Qingming Huang, Gang Hua, Shuqiang Jiang, Wen Gao, and Qi Tian. Building contextual visual vocabulary for large-scale image applications. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 501–510. ACM, 2010.

[155] Xiaoyu Wang, Ming Yang, Timothee Cour, Shenghuo Zhu, Kai Yu, and Tony X Han. Contextual weighting for vocabulary tree based image retrieval. In *Computer vision (ICCV), 2011 IEEE international conference on*, pages 209–216. IEEE, 2011.

[156] Hongtao Xie, Ke Gao, Yongdong Zhang, Sheng Tang, Jintao Li, and Yizhi Liu. Efficient feature detection and effective post-verification for large scale near-duplicate image search. *IEEE TRANSACTIONS on multimedia*, 13(6):1319–1332, 2011.

[157] Steffen Werner, Bernd Krieg-Brückner, Hanspeter A Mallot, Karin Schweizer, and Christian Freksa. Spatial cognition: The role of landmark, route, and survey knowledge in human and robot navigation. In *Informatik'97 Informatik als Innovationsmotor*, pages 41–50. Springer, 1997.

[158] Christian Freksa, Reinhard Moratz, and Thomas Barkowsky. Schematic maps for robot navigation. In *Spatial Cognition II*, pages 100–114. Springer, 2000.

[159] George Chronis and Marjorie Skubic. Sketch-based navigation for mobile robots. In *Fuzzy Systems, 2003. FUZZ'03. The 12th IEEE International Conference on*, volume 1, pages 284–289. IEEE, 2003.

[160] George Chronis and Marjorie Skubic. Robot navigation using qualitative landmark states from sketched route maps. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 2, pages 1530–1535. IEEE, 2004.

[161] Tomasz Malisiewicz and Alyosha Efros. Beyond categories: The visual memex model for reasoning about object relationships. In *Advances in neural information processing systems*, pages 1222–1230, 2009.

[162] Q Sjahputera, James M Keller, and Pascal Matsakis. Scene matching by spatial relationships. In *Fuzzy Information Processing Society, 2003. NAFIPS 2003. 22nd International Conference of the North American*, pages 149–154. IEEE, 2003.

[163] Rob Frampton and Andrew Calway. Place recognition from disparate views. *Journal of Robotics Research*, 27(6):647–665, 2008.

[164] Pilailuck Panphattarasap and Andrew Calway. Visual place recognition using landmark distribution descriptors. In *Asian Conference on Computer Vision*, pages 487–502. Springer, 2016.

[165] Kwangyong Lim, Yongwon Hong, Minsong Ki, Yeongwoo Choi, and Hyeran Byun. Vision-based recognition of road regulation for intelligent vehicle. In *2018 IEEE Intelligent Vehicles Symposium (IV)*, pages 1418–1425. IEEE, 2018.

[166] Junwei Han, Dingwen Zhang, Gong Cheng, Nian Liu, and Dong Xu. Advanced deep-learning techniques for salient and category-specific object detection: a survey. *IEEE Signal Processing Magazine*, 35(1):84–100, 2018.

[167] Jan Hosang, Rodrigo Benenson, Piotr Dollár, and Bernt Schiele. What makes for effective detection proposals? *IEEE transactions on pattern analysis and machine intelligence*, 38(4):814–830, 2016.

[168] Li Liu, Wanli Ouyang, Xiaogang Wang, Paul Fieguth, Jie Chen, Xinwang Liu, and Matti Pietikäinen. Deep learning for generic object detection: A survey. *arXiv preprint arXiv:1809.02165*, 2018.

[169] Sourav Garg, Niko Suenderhauf, and Michael Milford. Lost? appearance-invariant place recognition for opposite viewpoints using visual semantics. *arXiv preprint arXiv:1804.05526*, 2018.

[170] Andrea Vedaldi and Karel Lenc. Matconvnet: Convolutional neural networks for matlab. In *Proceedings of the 23rd ACM international conference on Multimedia*, pages 689–692. ACM, 2015.

[171] Dimitris Achlioptas. Database-friendly random projections. In *Proceedings of the twentieth ACM SIGMOD-SIGACT-SIGART symposium on Principles of database systems*, pages 274–281. ACM, 2001.

[172] Sean Bell and Kavita Bala. Learning visual similarity for product design with convolutional neural networks. *ACM Transactions on Graphics (TOG)*, 34(4):98, 2015.

[173] Giorgos Tolias, Ronan Sicre, and Hervé Jégou. Particular object retrieval with integral max-pooling of cnn activations. *arXiv preprint arXiv:1511.05879*, 2015.

[174] Noa Garcia and George Vogiatzis. Learning non-metric visual similarity for image retrieval. *Image and Vision Computing*, 2019.

[175] Andrea Vedaldi and Brian Fulkerson. Vlfeat: An open and portable library of computer vision algorithms. In *Proceedings of the 18th ACM international conference on Multimedia*, pages 1469–1472. ACM, 2010.

[176] Hui Kong, Jean-Yves Audibert, and Jean Ponce. Vanishing point detection for road detection. In *Computer Vision and Pattern Recognition, 2009. CVPR 2009. IEEE Conference on*, pages 96–103. IEEE, 2009.

[177] Marie Sjölinder. Individual differences in spatial cognition and hypermedia navigation. *Exploring navigation: Towards a framework for design and evaluation of navigation in electronic spaces*, pages 61–72, 1998.

[178] Reginald G Golledge, R Daniel Jacobson, Robert Kitchin, and Mark Blades. Cognitive maps, spatial abilities, and human wayfinding. *Geographical review of Japan, Series B.*, 73(2):93–104, 2000.

[179] Thomas Wolbers and Mary Hegarty. What determines our navigational abilities? *Trends in cognitive sciences*, 14(3):138–146, 2010.

[180] P. Biber and T. Duckett. Experimental analysis of sample-based maps for long-term slam. *Int. J. on Robotics Research*, 28(1), 2009.

[181] Shenlong Wang, Sanja Fidler, and Raquel Urtasun. Lost shopping! monocular localization in large indoor spaces. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2695–2703, 2015.

[182] Monica L Smith. Networks, territories, and the cartography of ancient states. *Annals of the Association of American Geographers*, 95(4):832–849, 2005.

[183] Peter A Burrough, Rachael McDonnell, Rachael A McDonnell, and Christopher D Lloyd. *Principles of geographical information systems*. Oxford university press, 2015.

[184] Howard Brody, Michael Russell Rip, Peter Vinten-Johansen, Nigel Paneth, and Stephen Rachman. Map-making and myth-making in broad street: the london cholera epidemic, 1854. *The Lancet*, 356(9223):64–68, 2000.

[185] Larianne Collins. The impact of paper versus digital map technology on students' spatial thinking skill acquisition. *Journal of Geography*, 117(4):137–152, 2018.

[186] N Sentürk, M Akgül, T Öztürk, AO Akay, et al. Comparison of topographical map based traditional method and computer-assisted method in calculation of cut-fill volumes in forest roads. *Bartın Orman Fakültesi Dergisi*, 20(3):618–626, 2018.

[187] Businessinsider Australia Nicholas Carlson. To do what google does in maps, apple would have to hire 7,000 people. https://www.businessinsider.com.au/to-do-what-google-does-in-maps-apple-would-have-to-hire-7000-people-2012-6, June 2012. Online; accessed June 2019.

[188] OpenStreetMap. Openstreetmap stats report. https://www.openstreetmap.org/stats/data_stats.html, 2019. Online; accessed June 2019.

[189] Alberto Garcia-Garcia, Sergio Orts-Escolano, Sergiu Oprea, Victor Villena-Martinez, and Jose Garcia-Rodriguez. A review on deep learning techniques applied to semantic segmentation. *arXiv preprint arXiv:1704.06857*, 2017.

[190] Qimin Cheng, Qian Zhang, Peng Fu, Conghuan Tu, and Sen Li. A survey and analysis on automatic image annotation. *Pattern Recognition*, 79:242–259, 2018.

[191] Xiwen Yao, Junwei Han, Gong Cheng, Xueming Qian, and Lei Guo. Semantic annotation of high-resolution satellite images via weakly supervised learning. *IEEE Transactions on Geoscience and Remote Sensing*, 54(6):3660–3671, 2016.

[192] Randy S Roberts, John W Goforth, George F Weinert, Will R Ray, Charles W Grant, and Art G Jolly. A visual wikipedia for satellite imagery. In *Optics and Photonics for Information Processing XII*, volume 10751, page 1075103. International Society for Optics and Photonics, 2018.

[193] Chaomei Chen. Bridging the gap: The use of pathfinder networks in visual navigation. *Journal of Visual Languages & Computing*, 9(3):267–286, 1998.

[194] Hui Zhang, Ksenia Zherdeva, and Arne D Ekstrom. Different "routes" to a cognitive map: dissociable forms of spatial knowledge derived from route and cartographic map learning. *Memory & cognition*, 42(7):1106–1117, 2014.

[195] Jimmy Y Zhong and Maria Kozhevnikov. Relating allocentric and egocentric survey-based representations to the self-reported use of a navigation strategy of egocentric spatial updating. *Journal of Environmental Psychology*, 46:154–175, 2016.

[196] Nathan Piasco, Désiré Sidibé, Cédric Demonceaux, and Valérie Gouet-Brunet. A survey on visual-based localization: On the benefit of heterogeneous data. *Pattern Recognition*, 74:90–109, 2018.

[197] Edgar Chan, Oliver Baumann, Mark A Bellgrove, and Jason B Mattingley. From objects to landmarks: the function of visual location information in spatial navigation. *Frontiers in psychology*, 3:304, 2012.

[198] Arne D Ekstrom, Hugo J Spiers, Véronique D Bohbot, and R Shayna Rosenbaum. *Human Spatial Navigation*. Princeton University Press, 2018.

[199] Geoffrey Edwards. Spatial knowledge for image understanding. In *Cognitive and linguistic aspects of geographic space*, pages 295–307. Springer, 1991.

[200] Rob Kitchin. Cognitive maps. *International Encyclopaedia of Social and Behavioural Sciences*, 3:2120–2124, 2001.

[201] Teriitutea Quesnot and Stéphane Roche. Measure of landmark semantic salience through geosocial data streams. *ISPRS International Journal of Geo-Information*, 4(1):1–31, 2014.

[202] Martin Raubal. Human wayfinding in unfamiliar buildings: a simulation with a cognizing agent. *Cognitive Processing*, 2(3):363–388, 2001.

[203] Grant McKenzie and Alexander Klippel. The interaction of landmarks and map alignment in you-are-here maps. *The Cartographic Journal*, 53(1):43–54, 2016.

[204] Guilherme N DeSouza and Avinash C Kak. Vision for mobile robot navigation: A survey. *IEEE transactions on pattern analysis and machine intelligence*, 24(2):237–267, 2002.

[205] Francisco Bonin-Font, Alberto Ortiz, and Gabriel Oliver. Visual navigation for mobile robots: A survey. *Journal of intelligent and robotic systems*, 53(3):263–296, 2008.

[206] Mehmet Serdar Guzel and Robert Bicker. A behaviour-based architecture for mapless navigation using vision. *International Journal of Advanced Robotic Systems*, 9(1):18, 2012.

[207] José Santos-Victor, Giulio Sandini, Francesca Curotto, and Stefano Garibaldi. Divergent stereo in autonomous navigation: From bees to robots. *International Journal of Computer Vision*, 14(2):159–177, 1995.

[208] Matthias O Franz and Hanspeter A Mallot. Biomimetic robot navigation. *Robotics and autonomous Systems*, 30(1-2):133–153, 2000.

[209] Joseph Conroy, Gregory Gremillion, Badri Ranganathan, and J Sean Humbert. Implementation of wide-field integration of optic flow for autonomous quadrotor navigation. *Autonomous robots*, 27(3):189, 2009.

[210] Julien R Serres and Stéphane Viollet. Insect-inspired vision for autonomous vehicles. *Current opinion in insect science*, 2018.

[211] Howie Choset and Keiji Nagatani. Topological simultaneous localization and mapping (slam): toward exact localization without explicit localization. *IEEE Transactions on robotics and automation*, 17(2):125–137, 2001.

[212] David M Bradley, Rashmi Patel, Nicolas Vandapel, and Scott M Thayer. Real-time image-based topological localization in large outdoor environments. In *Intelligent Robots and Systems, 2005.(IROS 2005). 2005 IEEE/RSJ International Conference on*, pages 3670–3677. IEEE, 2005.

[213] Sebastian Thrun et al. Robotic mapping: A survey. *Exploring artificial intelligence in the new millennium*, 1(1-35):1, 2002.

[214] AC Murillo, Carlos Sagüés, José Jesús Guerrero, Toon Goedemé, Tinne Tuytelaars, and Luc Van Gool. From omnidirectional images to hierarchical localization. *Robotics and Autonomous Systems*, 55(5):372–382, 2007.

[215] Jose-Luis Blanco, Juan-Antonio Fernández-Madrigal, and Javier Gonzalez. Toward a unified bayesian approach to hybrid metric–topological slam. *IEEE Transactions on Robotics*, 24(2):259–270, 2008.

[216] Kurt Konolige, Eitan Marder-Eppstein, and Bhaskara Marthi. Navigation in hybrid metric-topological maps. In *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, pages 3041–3047. IEEE, 2011.

[217] Arthur Gelb. *Applied optimal estimation*. MIT press, 1974.

[218] Randall Smith, Matthew Self, and Peter Cheeseman. Estimating uncertain spatial relationships in robotics. In *Autonomous robot vehicles*, pages 167–193. Springer, 1990.

[219] Illah Nourbakhsh, Rob Powers, and Stan Birchfield. Dervish an office-navigating robot. *AI magazine*, 16(2):53, 1995.

[220] Frank Dellaert, Dieter Fox, Wolfram Burgard, and Sebastian Thrun. Monte carlo localization for mobile robots. In *Robotics and Automation, 1999. Proceedings. 1999 IEEE International Conference on*, volume 2, pages 1322–1328. IEEE, 1999.

[221] Jun S Liu and Rong Chen. Sequential monte carlo methods for dynamic systems. *Journal of the American statistical association*, 93(443):1032–1044, 1998.

[222] Sven Koenig and Reid Simmons. Xavier: A robot navigation architecture based on partially observable markov decision process models. *Artificial Intelligence Based Mobile Robotics: Case Studies of Successful Robot Systems*, pages 91–122, 1998.

[223] Pär Buschka and Alessandro Saffiotti. A virtual sensor for room detection. In *Intelligent Robots and Systems, 2002. IEEE/RSJ International Conference on*, volume 1, pages 637–642. IEEE, 2002.

[224] Philipp Althaus and Henrik I Christensen. Behavior coordination in structured environments. *Advanced Robotics*, 17(7):657–674, 2003.

[225] Keith Cheverst, Johannes Schöning, Antonio Krüger, and Michael Rohs. Photomap: Snap, grab and walk away with a ”you are here” map. In *MIRW*, pages 73–82, 2008.

[226] Johannes Schöning, Antonio Krüger, Keith Cheverst, Michael Rohs, Markus Löchtefeld, and Faisal Taher. Photomap: using spontaneously taken images of public maps for pedestrian navigation tasks on mobile devices. In *Proceedings of the 11th international Conference on Human-Computer interaction with Mobile Devices and Services*, page 14. ACM, 2009.

[227] Martin P Parsley and Simon J Julier. Towards the exploitation of prior information in slam. In *Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on*, pages 2991–2996. IEEE, 2010.

[228] Jan Brejcha and Martin Čadík. State-of-the-art in visual geo-localization. *Pattern Analysis and Applications*, 20(3):613–637, 2017.

[229] Tsung-Yi Lin, Yin Cui, Serge Belongie, and James Hays. Learning deep representations for ground-to-aerial geolocalization. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5007–5015, 2015.

[230] Scott Workman, Richard Souvenir, and Nathan Jacobs. Wide-area image geolocalization with aerial reference imagery. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 3961–3969, 2015.

[231] Yicong Tian, Chen Chen, and Mubarak Shah. Cross-view image matching for geo-localization in urban environments. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3608–3616, 2017.

[232] Sébastien Lefévre, Devis Tuia, Jan Dirk Wegner, Timothée Produit, and Ahmed Samy Nassaar. Toward seamless multiview scene analysis from satellite to street level. *Proceedings of the IEEE*, 105(10):1884–1899, 2017.

[233] Tat-Jen Cham, Arridhana Ciptadi, Wei-Chian Tan, Minh-Tri Pham, and Liang-Tien Chia. Estimating camera pose from a single urban ground-view omnidirectional image and a 2d building outline map. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 366–373. IEEE, 2010.

[234] Clemens Arth, Christian Pirchheim, Jonathan Ventura, Dieter Schmalstieg, and Vincent Lepetit. Instant outdoor localization and slam initialization from 2.5 d maps. *IEEE Trans. Vis. Comput. Graph.*, 21(11):1309–1318, 2015.

[235] Arsalan Mousavian and Jana Kosecka. Semantic image based geolocation given a map. *arXiv preprint arXiv:1609.00278*, 2016.

[236] Anil Armagan, Martin Hirzer, Peter M Roth, and Vincent Lepetit. Accurate camera registration in urban environments using high-level feature matching. In *Proceedings of the British Machine Vision Conference*, 2017.

[237] Francesco Castaldo, Amir Zamir, Roland Angst, Francesco Palmieri, and Silvio Savarese. Semantic cross-view matching. In *Proceedings of the IEEE International Conference on Computer Vision Workshops*, pages 9–17, 2015.

[238] Marcus A Brubaker, Andreas Geiger, and Raquel Urtasun. Lost! leveraging the crowd for probabilistic visual self-localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3057–3064, 2013.

[239] Georgios Floros, Benito van der Zander, and Bastian Leibe. Openstreetslam: Global vehicle localization using openstreetmaps. In *IEEE International Conference on Robotics and Automation*, volume 13, pages 1054–1059, 2013.

[240] Ari Seff and Jianxiong Xiao. Learning from maps: Visual common sense for autonomous driving. *arXiv preprint arXiv:1611.08583*, 2016.

[241] Wei-Chiu Ma, Shenlong Wang, Marcus A Brubaker, Sanja Fidler, and Raquel Urtasun. Find your way by observing the sun and other semantic cues. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 6292–6299. IEEE, 2017.

[242] Klaus Broelemann and Xiaoyi Jiang. A region-based method for sketch map segmentation. In *Graphics recognition. new trends and challenges*, pages 1–14. Springer, 2013.

[243] Marjorie Skubic, Pascal Matsakis, Benjamin Forrester, and George Chronis. Extracting navigation states from a hand-drawn map. In *Robotics and Automation, 2001. Proceedings 2001 ICRA. IEEE International Conference on*, volume 1, pages 259–264. IEEE, 2001.

[244] Marcus A Brubaker, Andreas Geiger, and Raquel Urtasun. Map-based probabilistic visual self-localization. *IEEE transactions on pattern analysis and machine intelligence*, 38(4):652–665, 2016.

[245] Adel Javanmard, Maya Haridasan, and Li Zhang. Multi-track map matching. In *Proceedings of the 20th International Conference on Advances in Geographic Information Systems*, pages 394–397. ACM, 2012.

[246] Pilailuck Panphattarasap and Andrew Calway. Automated map reading: Image based localisation in 2-d maps using binary semantic descriptors. In *International Conference on Intelligent Robots and Systems*, 2018.

[247] Dieter Fox, Jeffrey Hightower, Lin Liao, Dirk Schulz, and Gaetano Borriello. Bayesian filtering for location estimation. *IEEE pervasive computing*, pages 24–33, 2003.

[248] Zhuoling Xiao, Hongkai Wen, Andrew Markham, and Niki Trigoni. Lightweight map matching for indoor localisation using conditional random fields. In *Proceedings of the 13th international symposium on Information processing in sensor networks*, pages 131–142. IEEE Press, 2014.

[249] Zhenghua Chen, Han Zou, Hao Jiang, Qingchang Zhu, Yeng Soh, and Lihua Xie. Fusion of wifi, smartphone sensors and landmarks using the kalman filter for indoor localization. *Sensors*, 15(1):715–732, 2015.

[250] Walter A. Burkhard and Robert M. Keller. Some approaches to best-match file searching. *Communications of the ACM*, 16(4):230–236, 1973.

[251] Nitin Bhatia et al. Survey of nearest neighbor techniques. *arXiv preprint arXiv:1007.0085*, 2010.

[252] Vladimir I Levenshtein. Binary codes capable of correcting deletions, insertions, and reversals. In *Soviet physics doklady*, volume 10, pages 707–710, 1966.

[253] Dietrich Epp. Metric-tree-demo. https://github.com/depp/metric-tree-demo, 2011.

[254] Stefan Steiniger and Andrew JS Hunter. The 2012 free and open source gis software map–a guide to facilitate research, development, and adoption. *Computers, environment and urban systems*, 39:136–150, 2013.

[255] Ioannis Filippidis. Openstreetmap functions. https://www.mathworks.com/matlabcentral/fileexchange/35819-openstreetmap-functions, 2011.

[256] Roger W Sinnott. Virtues of the haversine. *Sky Telesc.*, 68:159, 1984.

[257] Hagar Mahmoud and Nadine Akkari. Shortest path calculation: a comparative study for location-based recommender system. In *2016 World Symposium on Computer Applications & Research (WSCAR)*, pages 1–5. IEEE, 2016.

[258] Scott D Roth. Ray casting for modeling solids. *Computer graphics and image processing*, 18(2):109–144, 1982.

[259] Arthur Appel. Some techniques for shading machine renderings of solids. In *Proceedings of the April 30–May 2, 1968, spring joint computer conference*, pages 37–45. ACM, 1968.

[260] Hao Zhang, Alexander C Berg, Michael Maire, and Jitendra Malik. Svm-knn: Discriminative nearest neighbor classification for visual category recognition. In *Computer Vision and Pattern Recognition, 2006 IEEE Computer Society Conference on*, volume 2, pages 2126–2136. IEEE, 2006.

[261] Jianxiong Xiao, James Hays, Krista A Ehinger, Aude Oliva, and Antonio Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *Computer vision and pattern recognition (CVPR), 2010 IEEE conference on*, pages 3485–3492. IEEE, 2010.

[262] Mohamed Aly. Real time detection of lane markers in urban streets. In *Intelligent Vehicles Symposium, 2008 IEEE*, pages 7–12. IEEE, 2008.

[263] Olivier Chapelle, Patrick Haffner, and Vladimir N Vapnik. Support vector machines for histogram-based image classification. *IEEE transactions on Neural Networks*, 10(5):1055–1064, 1999.

[264] Yuanqing Lin, Fengjun Lv, Shenghuo Zhu, Ming Yang, Timothee Cour, Kai Yu, Liangliang Cao, and Thomas Huang. Large-scale image classification: fast feature extraction and svm training. *IEEE Conference on Computer Vision and Pattern Recognition*, 2011.

[265] Bolei Zhou, Agata Lapedriza, Jianxiong Xiao, Antonio Torralba, and Aude Oliva. Learning deep features for scene recognition using places database. In *Advances in neural information processing systems*, pages 487–495, 2014.

[266] Matthew R Boutell, Jiebo Luo, Xipeng Shen, and Christopher M Brown. Learning multi-label scene classification. *Pattern recognition*, 37(9):1757–1771, 2004.

[267] Jiang Wang, Yi Yang, Junhua Mao, Zhiheng Huang, Chang Huang, and Wei Xu. Cnn-rnn: A unified framework for multi-label image classification. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2285–2294, 2016.

[268] Amir Roshan Zamir and Mubarak Shah. Image geo-localization based on multiplenearest neighbor feature matching usinggeneralized graphs. *IEEE transactions on pattern analysis and machine intelligence*, 36(8):1546–1558, 2014.

[269] Mandar Haldekar, Ashwinkumar Ganesan, and Tim Oates. Identifying spatial relations in images using convolutional neural networks. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 3593–3600. IEEE, 2017.

[270] Christoph Goller and Andreas Kuchler. Learning task-dependent distributed representations by backpropagation through structure. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, volume 1, pages 347–352. IEEE, 1996.

[271] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[272] Mahdi Fakoor, Amirreza Kosari, and Mohsen Jafarzadeh. Humanoid robot path planning with fuzzy markov decision processes. *Journal of applied research and technology*, 14(5):300–310, 2016.

[273] Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2616–2625, 2017.

[274] Angela Schwering, Jia Wang, Malumbo Chipofya, Sahib Jan, Rui Li, and Klaus Broelemann. Sketchmapia: Qualitative representations for the alignment of sketch and metric maps. *Spatial cognition & computation*, 14(3):220–254, 2014.

[275] Chen Liu, Jiajun Wu, Pushmeet Kohli, and Yasutaka Furukawa. Raster-to-vector: revisiting floorplan transformation. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 2195–2203, 2017.