PIRF-Nav: An Online Incremental Appearance-based Localization and Mapping in Dynamic Environments

実環境における画像を用いたオンラインかつ追加的な自己位置推定・地図作成手法

By

Aram Kawewong

Supervisor:

Dr. Osamu Hasegawa

Department of Computational Intelligence and Systems Science

Interdisciplinary Graduate School of Science and Engineering

Tokyo Institute of Technology

September 2010

Thesis Inspectors:

Osamu Hasegawa, Hiroshi Nagahashi, Makoto Sato, Katsumi Nitta, Sumio Watanabe

ACKNOWLEDGEMENT

First and foremost, thanks are given to my supervisor Osamu Hasegawa. With a number of times for our discussion, I learned uncountable things from him. Under his supervision, I got not only the wise words at the right time, but also the freedom to explore any interesting things.

I also want to give my gratitude to many people who have given me a hand in doing my researches. To Sirinart Tangruamsub for the data collection at the very beginning. To Noppharit Tongprasit for some interesting ideas and discussions. And of course, thanks to all the members of Hasegawa research groups for friendship and good times, so that I can concentrate on my research without needs to worry about social life in Japan.

Also, I am certain I have to acknowledge the support from NEDO (New Energy and Industrial Technology Development Organization) of Japan and Toyota company in term of funding.

Finally, but most importantly, to my family and my girlfriend have been there even when I had some hard times. Especially to my dearest, dearest father who passed away during my study at Tokyo Institute of Technology. He always stood steadfastly and lovingly by my side at all times. He has been everything a good father should be. From now on, he will be my greatest inspiration in pursuing my dream to become one of the best professor and researcher in Thailand.

Table of Contents

Index of Figures

CHAPTER 1

INTRODUCTION

"Where are we" has been always a very important question for human. Before doing any tasks, it is always the case that we need to first know about our own current position. Some may argue that, in order to finish some specific tasks such as moving or controlling objects, knowing an exact position is not a concern. Given that the main objective is to move or control objects, such task can be done in the same as in any places. Nevertheless, this argument will be inapplicable to human as long as human is not stationary. That is, as human, one needs to know where he is before completing any specific tasks. In artificial intelligence and robotics research field, this capability is known as localization.

Same as human, the capability of localization is indispensable for mobile robots. Distinct from task-specific robots that have been used in manufacturing process in factories, mobile robot can explore an unknown area and plan the path to reach the destination with minimum effort. The topic of exploration and path planning has been re-defined into new topics to make it easier for robotics researchers; Localization, Mapping and Path Planning.

Traditionally, path planning was often done in toy-world or grid world. The problem was mostly solved in the way of optimization problem. Given that the map is a priori known, by specifying the start position and goal position, shortest path is mathematically guaranteed to be found. One of the most famous and popular solutions of this problem is the use of reinforcement learning (RL) [1], [6]. Afterwards, researchers shifted their attention to path-planning in real-world robots [2], [3], [4], [5]. The problem thus becomes more difficult in the sense that the robot does not know the map a priori. To

create the mobile robots that can function in any places like human, it needs to know how to incrementally create the map (mapping), localize itself relatively to the map, and then calculating the shortest path or route to navigate from the current position to the destination. Furthermore, these all function must be done simultaneously. This is the origin of the recently popular problem of SLAM (Simultaneous Localization and Mapping) in robotics community.

1.1 Mobile Robot Localization

Localization in mobile robot is the problem of determining the pose of a robot relative to a map of the environment. It is often called *position estimation*. Mobile robot localization is a variance of the general localization problem, which is the most fundamental perceptual robotics problem. Most robotics tasks require knowledge of the location of objects that are being controlled.

Mobile robot localization can also be viewed as a problem of coordinate transformation. Maps are usually described in a global coordinate system, which does not depend of a robot's pose. Localization is the process of finding correspondence between the map coordinate system and the robot's local coordinate system. Knowing this coordinate transformation enables the robot to realize the location of target objects within its own coordinate frame. As it can be easily verified, knowing the pose of the robot is sufficient to determine this coordinate transformation, assuming that the pose is expressed in the same coordinate frame as the map. Unfortunately, the pose cannot always be sensed directly. To put it in a different way, it is very unlikely that robots always possess a noise-free sensor for measuring its pose. The pose thus has to be inferred from data. A key difficulty arises from the fact that measurement from a single sensor is usually insufficient to determine the accurate pose. Instead, the robot should integrate data over time to determine its pose. To see why this is necessary, just imagine a robot located inside a building where many corridors look similar. In this case, a single sensor measurement (e.g., a laser range scanner) might be insufficient to identify the specific corridor.

Localization techniques have been developed for a various set of map representations as shown in Figure 1.1. This figure portrays a hand-drawn metric 2-D map, a graph-like topological map, an occupancy grid map, and an image mosaic of a ceiling.

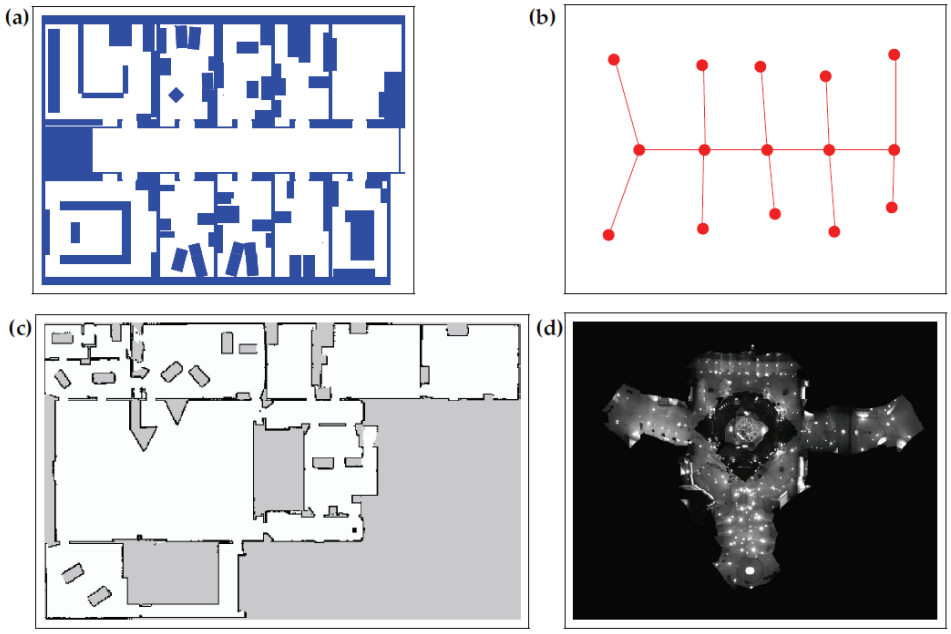


Figure 1.1 Example map representations used for robot localization: (a) a manually constructed 2-D metric layout, (b) a graph-like topological map, (c) an occupancy grid map, and (d) an image mosaic of a ceiling. The figure is taken from [7].

Not every localization problems is equally difficult. To get familiar with the localization problem, the taxonomy of localization problems are hereby discussed. The taxonomy divides the problem of localization along a number of important dimensions based on the nature of the environment and the initial knowledge that a robot may possess *a priori* relative to the localization problem.

1.1.1 Local VS Global Localization

Localization problems are characterized by the type of knowledge provided both in advance and at run-time. We distinguish three types of localization problems with different degree of difficulty.

Position Tracking: The initially known pose of the robot is assumed. Localization is achieved by accommodating the noise in robot's motion. The effect of such noise is usually not too big. Thus, it is always the case that methods for position tracking rely on the assumption that the error in pose's measurement is small. The pose uncertainty is frequently approximated by a uniform distribution (e.g., a Gaussian). The position tracking problem is

regarded as a *local* problem, since the uncertainty is local and confined to region near the robot's actual pose.

Global Localization: The initial pose of the robot is not known. At the very beginning, the robot is placed somewhere in its environment, lacking of knowledge of its whereabouts. Global localization approaches cannot assume boundedness of the error in pose measurement. The uniform probability distributions are usually inappropriate. This makes the global Localization more difficult than position tracking; in fact, it subsumes the position tracking problem.

Kidnapped Robot Problem: This problem is a modified from global localization problem, but one that is even more difficult. While operating, the robot can get kidnapped and teleported to some other location. The kidnapped robot problem is more difficult than the global localization problems, because the robot might believe it knows where it is while it does not actually know. One may argue that it is unlikely that robots would be kidnapped in practice. Nevertheless, the practical importance of this problem arises from the observation that most state-of-the-art localization methods cannot be guaranteed never to fail. The ability to recover from failures is indispensable for truly autonomous mobile robots. Testing a localization algorithm by kidnapping it measures its ability to recover from global localization failures.

1.1.2 Static VS Dynamic Environment

A second dimension that has a substantial impact on the difficulty of localization is the environment as it can be static or dynamic.

Static Environments are the environments where the only variable quantity (state) is the pose of robot. In other words, only the robot can move in static environment; all other objects remain at the same location forever (static). From mathematical viewpoint, static environments are amenable to efficient probabilistic estimation.

Dynamic Environments possess objects other than the robot whose location or configuration may change over time, especially the changes that persist over time and may affect more than one sensor. Certainly, changes that are not measurable are of no relevance to localization, and those that affect only a single measurement are best treated as noise. Examples of more persistent changes are: people, daylight, movable objects such as car and empty big boxes.

Apparently, most real environments are dynamic, with state changes occurring at a range of different speeds.

1.1.3 Passive VS Active Localization

A third dimension that characterizes different localization problems bases on the fact whether or not the localization algorithm controls the motion of the robot. This can be divided into two cases.

Passive Localization — In this localization, the localization module only observes the robot's operating. The robot is controlled via some other means, and its motion is not aimed at facilitating localization. For instance, the robot can move randomly and perform its everyday tasks.

Active Localization — The algorithms control the robot in order to minimize the localization error and/or the costs of moving a poorly localized robot into a complex environment.

Active approaches usually yield better localization results than passive ones. Figure 1.2 shows the sample situation. The robot is located in a symmetrical corridor, and its belief after traveling along the corridor for awhile is centered at two symmetric poses. Because of the local symmetry of the environment, it is impossible for the robot to perform localization while being in the corridor. It will be able to eliminate ambiguity and to determine its pose only if it moves into a room will. In this situation, active localization gives much better results: instead of just waiting until the robot accidentally moves into a room, active approach force to recognize the impasse and escape from it.

Nonetheless, active approaches also pose some limitations in which they require control over the robot, making an active localization technique alone practically insufficient. The robot must be able to localize itself not only when performing localization but also some other tasks. Some active localization techniques are built on top of passive technique. Others combine task performance goals with localization goals when controlling a robot.

1.1.4 Single-Robot VS Multi-Robot Localization

A fourth dimension of the localization problem is related to the number of robots operated in an environment.

Single-robot Localization is the most commonly studied approach. This approach deals with only a single robot. It offers the convenience that all data

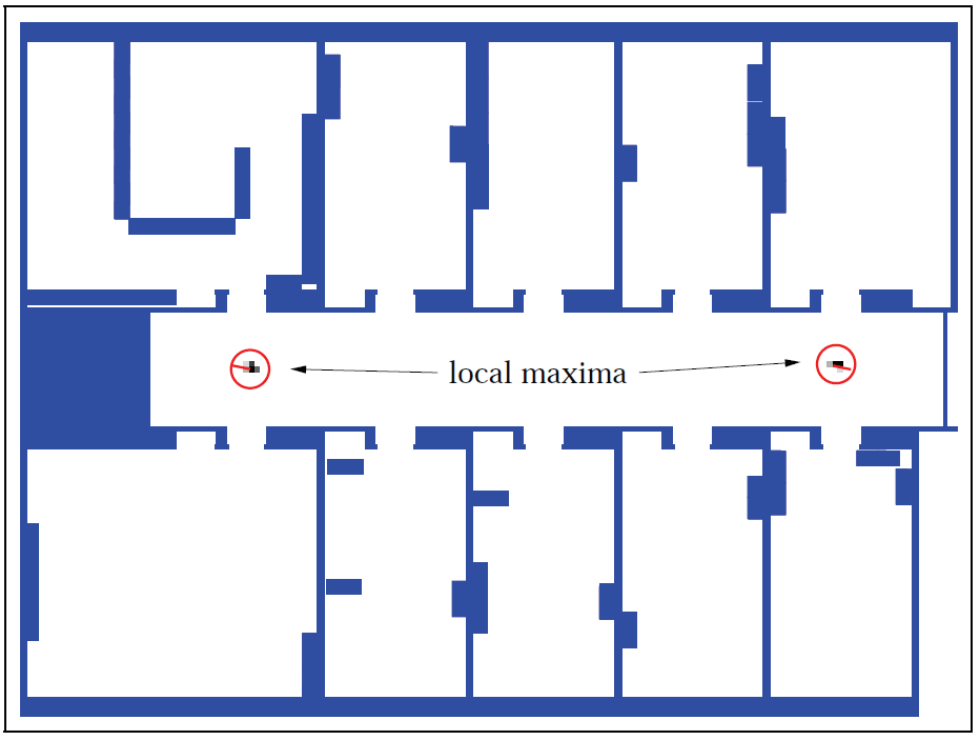


Figure 1.2 Example situation that shows a typical belief during global localization in a locally symmetric environment. The root has to move into some room to determine its location. (Figure is referred from [7]).

is collected at a single robot platform, and there is no concern about communication among robots.

The *multi-robot localization* problem arises in team robotics. By this problem, each robot could localize itself individually. Hence, the multi-robot localization problem can be solved through single-robot localization. If robots are able to detect each other, however, there is the opportunity to do better; a belief of one robot can be used to bias a belief of another robot if knowledge of the relative location of both robots is available. The issue of multi-robot localization raises interesting, non-trivial issues on the representation of beliefs and the nature of the communication between them.

These four dimensions represent the four most important characteristics of the mobile robot localization problem. There exist a number of other characterizations that impact the difficulty of the problem, such as the information provided by robot measurements and the information lost through motion. Also, due to the higher degree of ambiguity, symmetric environments usually are more difficult than asymmetric ones,.

1.2 Simultaneous Localization and Mapping (SLAM)

The simultaneous localization and mapping (SLAM) in robotics is the problem of how can a mobile robot incrementally build a consistent map of an environment while simultaneously determining its location within the map even when it is placed at an unknown location in an unknown environment. Some even said that a solution to the SLAM problem could be seen as a "holy grail" for the mobile robotics community because it provides the means to make a robot truly autonomous.

The "solution" of the SLAM problem is one of the notable successes of the robotics community over the past decade. SLAM has been formulated and solved as theoretical problem in various forms. It has also been implemented in various domains from indoors to outdoors, underwater, and airborne systems. At a theoretical and conceptual level, SLAM can now be considered an already "solved problem". However, substantial issues still remain in practically realizing more general SLAM solutions and notably in building and using perceptually rich maps as part of a SLAM algorithm.

1.2.1 Brief History of the SLAM

Briefly on SLAM's history, the topic occurred at the 1986 IEEE Robotics and Automation Conference held in San Francisco, California. This was a time when probabilistic approaches were only just beginning to be applied into both robotics and artificial intelligence (AI) field of researches. A number of researchers had been interested in applying estimation-theoretic methods to mapping and localization problems, including Peter Cheeseman, Jim Crowley, and Hugh Durrant-Whyte [8], [9]. At the conference, many paper table cloths and napkins were filled with long discussions about robotics consistent mapping. Along the way, Raja Chatila, Oliver Faugeras, Randal Smith, and others also made useful contributions to the conversation. This becomes the first important step in robotics SLAM.

The result of this conversation leads to the conclusion that consistent probabilistic mapping was a fundamental problem in robotics with major conceptual and computational issues that needed to be addressed. Over the next few years, many key papers were published. Work by Smith and Cheeseman [8] and Durrant-Whyte [9] proposed a statistical basis for

describing relationships between landmarks and manipulating geometric uncertainty. A key feature of this work was to show that there must be a high degree of correlation between estimates of the location of different landmarks in a map and that, indeed, these correlations would grow with successive observations.

At the same time Ayache and Faugeras [10] were working on visual navigation, Crowley [11] and Chalita and Laumond [12] were undertaking a sonar-based navigation of mobile robots using Kalman-filter-type algorithms. These two researches had much in common and resulted soon after in the landmark paper by Smith *et al.* [13]. This paper showed that while a mobile robot moves through an unknown environment taking relative observations of landmarks, the estimates of these landmarks are all necessarily correlated with each other due to the common error in estimated vehicle location [14]. The implication of this was important in the sense that a consistent full solution to the combined localization and mapping problem would require a joint state composed of the vehicle pose and every landmark position, to be updated according to each landmark observation. In turn, this would require the estimator to employ a huge state vector (depending on the order of the number of landmarks maintained in the map) with computation scaling as the square of the number of landmarks.

Critically, this work neither looked at the convergence properties of the map nor its steady-state behavior. Actually, tt was widely assumed at the time that the estimated map errors would not converge and instead exhibit a random-walk behavior with unbounded error growth. Thus, given the computational complexity of the mapping problem without knowledge of the convergence behavior of the map, many researchers instead focused on a series of approximations to the consistent mapping problem, which assumed the correlations between landmarks to be minimized or eliminated, so reducing the full filter to a series of decouples landmark to vehicle filters ([15] and [16] for example). Also for these reasons, theoretical work on the combined localization and mapping problem was halt for the time being, with work often focused on either mapping or localization as separate problems.

The conceptual breakthrough is the realization that formulating the combined mapping and localization problem as a single estimation problem makes the problem become convergent. Most importantly, it was found that the correlations between landmarks, which most researchers had tried to

minimize, were actually the critical part of the problem and that, on the contrary, the more these correlations grew, the better the solution. The structure of the SLAM problem, the convergence result and the coining of the acronym SALM was first introduced in a survey paper of mobile robots presented at the 1995 International Symposium of Robotics Research (ISRR) [17]. The important theory on convergence and many of the initial results were developed by Csorba [18]. Several groups already working on mapping and localization, notably at the Massachusetts Institute of Technology (MIT) [19], Zaragoza [20], the ACFR at Sydney [21], [22], and others [23], [24], began working in earnest on SLAM in indoor, outdoor, and subsea environments.

1.2.2 SLAM Paradigms

Until now, the SLAM's solution could be divided into three main paradigms: Metric-, Topological-, and Hybrid-SLAM. The latter one has just become popular in last decades since it combines those favorable features of Metric- and Topological-SLAM. In fact, the work presented in this thesis is exactly one of the topological-SLAM methods. Nonetheless, it would be more appropriated to briefly describe about other two paradigms, metric and hybrid, so that the readers can know well what are the differences among these paradigms.

1.2.2.1 Metric SLAM (Traditional)

As stated previously, even with a lot of difficulties described about the SLAM, the final goal of solving this SLAM problem is to achieve the autonomous navigation system for mobile robot. At the very beginning, SLAM is done only for small indoor environments. By the definition of SLAM, the robot does not have access to map of the environment, nor does it know its own pose. Instead, all it is given are measurements $z_{1:t}$ and controls$u_{1:t}$. In SLAM, the robot acquires a map of its environment while simultaneously localizing itself relative to this map. SLAM is significantly more difficult than localization in that the map is unknown and has to be estimated along the way. It is more difficult than mapping with known poses, since the poses are unknown and have to be estimated along the way.

From a probabilistic point of view, there are two main forms of the SLAM problem, which are both of equal practical importance. One is known as the *online SLAM problem*; it involves estimating the posterior over the momentary

pose along with the map:

$$p(x_t, m|z_{1:t}, u_{1:t}) \tag{1.1}$$

where x_t is the pose at time *t, m* is the map, and $z_{1:t}$ and $u_{1:t}$ are the measurements and controls, respectively. This problem is called the online SLAM problem because it merely involves the estimation of variables that persist at time *t*. Many algorithms for the online SLAM problem are incremental in the sense that they discard past measurements and controls after being processed.

The second SLAM problem type is the *full SLAM problem*. By this problem, we seek to calculate a posterior over the entire path $x_{1:t}$ along with the map, instead of just the current pose x_t:

$$p(x_{1:t}, m|z_{1:t}, u_{1:t}) \tag{1.2}$$

This slight difference between online and full SLAM has ramifications in the type of algorithms that can be brought to bear. That is, the online SLAM problem is the result of integrating out past poses from the full SLAM problem as follows:-

$$p(x_t, m|z_{1:t}, u_{1:t}) = \iint \cdots \int p(x_{1:t}, m|z_{1:t}, u_{1:t}) dx_1 dx_2 \ldots dx_{t-1} \tag{1.3}$$

In online SLAM, these integrations are typically performed one-at-a-time. They cause interesting changes of the dependency structures in SLAM.

A second feature of the SLAM problem is related to the nature of the estimation problem. SLAM problems possess a continuous and a discrete component. The continuous estimation problem pertains to the location of the objects in the map and the robot's own pose variables. Objects may be landmarks in feature-based representation, or they might be object patches detected by range finders. When an object is detected, a SLAM algorithm must reason about the relation of this object to previously detected objects. This reasoning is typically discrete; either the object is the same as a previously detected one, or it is not.

At any point in time *t*, the robot gets to observe a vector of ranges and bearings to nearby features: $z_t = \{z_t^1, z_t^2, \ldots\}$. It is reasonable to assume that all features are uniquely identifiable. For example, the Tokyo Tower in Tokyo is a landmark that is rarely confused with other landmarks, and it is widely visible through Tokyo. The identity of a feature is expressed by a set of

correspondence variables, denoted c_t^i, one for each feature vector z_t^i.

At times, it will be useful to make the correspondence variables explicit. The online SLAM problem posterior is then given by

$$p(x_t, m, c_t | z_{1:t}, u_{1:t}) \tag{1.4}$$

and the full SLAM posterior by

$$p(x_{1:t}, m, c_{1:t} | z_{1:t}, u_{1:t}) \tag{1.5}$$

The online posterior is derived from the full posterior by integrating out past robot poses and summing over all past correspondences:

$$p(x_t, m, c_t | z_{1:t}, u_{1:t})$$

$$= \iint \cdots \int \sum_{c_1} \sum_{c_2} \cdots \sum_{c_{t-1}} p(x_{1:t}, m | z_{1:t}, u_{1:t}) dx_1 dx_2 \ldots dx_{t-1} \tag{1.6}$$

In both versions of the SLAM problems (online and full) estimating the full posterior (1.4) or (1.5) is the standard of SLAM. The full posterior captures all useful information about the map and the pose of the path.

Practically, calculating a full posterior is usually infeasible. Imporant problems may occur from two sources: (i) the high dimensionality of the continuous parameter space, and (ii) the large number of discrete correspondence variables. Many state-of-the-art SLAM algorithms construct maps with tens of thousands of features, or more. Even under known correspondence, the posterior over those maps alone involves probability distributions over spaces with 10^5 or more dimensions. This contradicts to localization problems, in which posteriors were estimated over three-dimensional continuous spaces. Additionally, it is always the case that the correspondences are unknown. The number of possible assignments to the vector of all correspondence variables $c_{1:t}$ grows exponentially in time *t*. Hence, practical SLAM algorithms that can deal with the correspondence problem must rely on approximations.

Apparently, metric SLAM poses at least two main problems which remain unsolved. Firstly, although resultant metric map from metric SLAM is very accurate and detailed, it cannot be used in long-term. In life-long robotics, the longer the robot is in action, the higher the dimensionality of probability distribution it needs to calculate. The problem is also well known as *curse of*

dimensionality. This becomes the main obstacle that makes it difficult to use metric SLAM in long-term robots. Secondly, for large-scale outdoor environment, it becomes the case that metric map sometimes is over-detailed. For example, given that the robot needs to walk around the city with size 3x3 kilometers, metric map with detailed coordinate would become inapplicable. Additionally, metric maps are always not easy to interpret. While these important drawbacks had been gradually realized, many researchers started thinking of another type of map representation that is suitable to be used in long-term in large-scale environment and easy to be interpreted; a topological map.

1.2.2.2 Topological SLAM

The curse of dimensionality makes it difficult to use metric map in a large-scale environment in long term, because of the metric map itself does not suit to the scale of the environment. Kuipers et al. [25] developed a robust qualitative map representation method for robot exploration, mapping and navigation in large-scale spatial environments. They define the term "large-scale" as the environment where its spatial structure is at significantly larger scale than the sensory horizon of the observer.

Later, among works on topological mapping, Choset and Nagatani use the topological map to efficiently solve the SLAM problem [26] by the adoption of Voronoi graph. It is believed that the topological and geometric structure of free space induce a natural hierarchy of symbols and connections between them. For Kuipers, the symbols are distinct place, which are local maxima of the distance to nearby obstacles, and the connections are the graph edges that link distinct places. For an indoor office-like environments, junctions and termination points of hallways represent symbols while the hallway themselves are the connections. For the generalized Voronoi graph (GVG) [27], [28], the voronoi vertices (meet points) are the symbols while the edges form connections (see Figure 1.3).

The connections of a high level representation can be discretized into a sequence of symbols with their corresponding set of connections. Leonard and Durrant-Whyte's approach to SLAM is an excellent example [14], [29]. Their method uses sensor information to define "features" and the relation among them to direct a robot experiencing positioning error. Using a Kalman filter

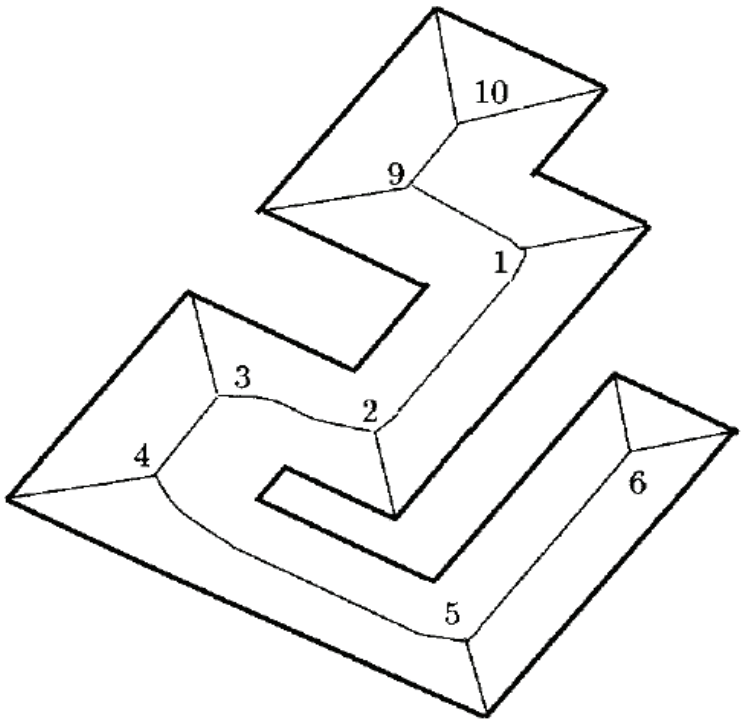


Figure 1.3 The GVG where the symbols (nodes) are labeled 1-10.

approach to determine the "best" correlation of features, the robot navigates similarly to a sailor using stars to navigate a ship at night. Through the proper understanding of the robot's relationship to the features, the robot can maintain an accurate estimate of its position while moving through the environment using the features as low-level symbols that guide the robot from one high-level symbol to the next.

There are many later works on topological mapping and SLAM that are based on Voronoi graph [30], [31]. The GVG is a map embedded in robot's free space and captures the topologically salient features of the free space. With the GVG the robot can plan a path between any two points in a static environment by first planning a path onto the GVG, then along the GVG, and finally from the GVG to the goal. Therefore, knowing the GVG is equivalent to knowing the free space and constructing the GVG is akin to exploring the free space.

Nevertheless, defining the symbols and their connections is not enough. Stable well-defined control laws must ensure that the robot can identify (and converge onto if necessary) a symbol location while at the same time move from symbol to symbol, i.e., traverse an edge. The control laws here are

essential for topological mapping. As described by Kuipers, the location-specific control laws are dynamically selected to control the robot's interaction with environment. These laws define *distinctive places and paths*, which are linked to form a topological network description. Therefore, unlike traditional metric SLAM where the topological structure is derived from the geometric description by analyzing the sensor input:

$$sensors \rightarrow geometry \rightarrow topology,$$

topological SLAM derive geometric knowledge from topology of map created by location-specific control algorithms:

$$[sensorimotor \leftrightarrow control] \rightarrow topology \rightarrow geometry.$$

Clearly, the topological SLAM (T-SLAM) is localization and mapping approach which consumes considerably less memory than that of metric map and is very suitable to solving the loop-closing problem especially in large-scale environment. The T-SLAM basically requires junctions, or places, to be distinctive and that they must be detected correctly whenever the robot passes close to them. Nevertheless, the pure T-SLAM still poses some serious drawbacks. Because of its fast computation and its ease of high-level interpretation, T-SLAM basically ignores most of the sensory input. This makes T-SLAM fall into the victim of the perceptual aliasing problem—a problem where different places look very similar. In indoor, the problem of perceptual aliasing is very common, since a number of places are artificial. By ignoring most sensors, the robot lacks of information for distinguishing the places. However, using traditional metric SLAM may solve this, but the problem of high cost of memory and computation still inevitably occur. As a result, the idea of combining these two fundamental SLAM approaches becomes reasonable and popular.

1.2.2.3 Hybrid Metric-Topological SLAM

As hinted previously, the idea of hybrid metric-topological SLAM (HMT-SLAM) is to find a new way of SLAM approach which take into consideration the observation in details when necessary, and ignore some sensory input that is considered as non-beneficial information. The key success of this solution is to answer the question: when should the robot

represent the map metrically, and when should it represent the map topologically.

Thrun and Bucken [103] have been known to be the first who integrate metric and topological paradigm and form a new paradigm hybrid metric-topological SLAM for mobile robots. In order to clearly understand the motivation of HMT-SLAM which is considered as the state-of-the-art SLAM's paradigm, the comparison between metric and topological SLAM should be made explicitly.

Both approaches M- and T-SLAM exhibit orthogonal strengths and weaknesses. Occupancy grids of M-SLAM are considerably easy to construct and to maintain even in large-scale environment [32]. Since the intrinsic geometry of a grid corresponds directly to the geometry of the environment, the robot's position within its model can be determined by its position and orientation in the real world—which can be determined sufficiently accurately using only sonar sensors, in environments of moderate size. As a pleasing consequence, different position positions for which sensors measure the same values (i.e., situations that look alike) are naturally disambiguated in M-SLAM. This is not the case of T-SLAM approaches, which determine the position of the robot relative to the model based on landmarks or distinct sensory features. For example, if the robot traverses two places that look alike, topological approaches often have difficulty determining if these places are the same or not (particularly if these places have been reached via different paths). Also, since sensory input usually depends strongly on the view-point of the robot, topological approaches may fail to recognize geometrically nearby places.

On the other hand, M-SLAM approaches suffer from their enormous space and time complexity. This is because the resolution of a grid must be fine enough to capture every important detail of the world. Compactness is a key advantage of topological representations. Topological maps are usually more compact, since their solution is determined by the complexity of the environment. Consequently, they permit fast planning, facilitate interfacing to symbolic planners and problem-solvers, and provide more natural interfaces for human instructions. Since topological approaches usually do not require the exact determination of the geometric position of the robot, they often recover better from drift and slippage (from use of odemeter)—phenomena that must constantly be monitored and compensated in grid-based metric map.

Metric SLAM	Topological SLAM
+ easy to build, represent, and maintain + recognition of places (based on geometry) is non-ambiguous and view-point independent + facilitates computation of shortest paths	+ Permits efficient planning, low space complexity (resolution depends on the complexity of the environment) + Does not require accurate determination of the robot's position + Convenient representation for symbolic planners, problem solvers, natural language interfaces
− Planning inefficient, space consuming (resolution does not depend on the complexity of the environment) − Require accurate determination of the robot's position − Poor interface for most symbolic problem solvers	− Difficult to construct and maintain in larger environments − Recognition of places (based on landmarks) often ambiguous, sensitive to the point of view − May yield suboptimal paths

Table 1.1 Comparison of metric and topological SLAM approaches to map building [103].

To summarize, both paradigms have orthogonal strengths and weaknesses, which are summarized in Table 1.1.

Hybrid metric-topological SLAM is thus proposed to integrate both existing paradigms, to gain the best of both worlds. It combines both grid-based (metric) and topological representation. To construct a grid-based model of the environment, sensor values are interpreted by an artificial neural network and mapped into probabilities for occupancy. Multiple interpretations are usually integrated over time using Bayes' rule. On top of the grid representation, more compact topological maps are generated by splitting the metric map into coherent regions, separated through critical lines. Critical

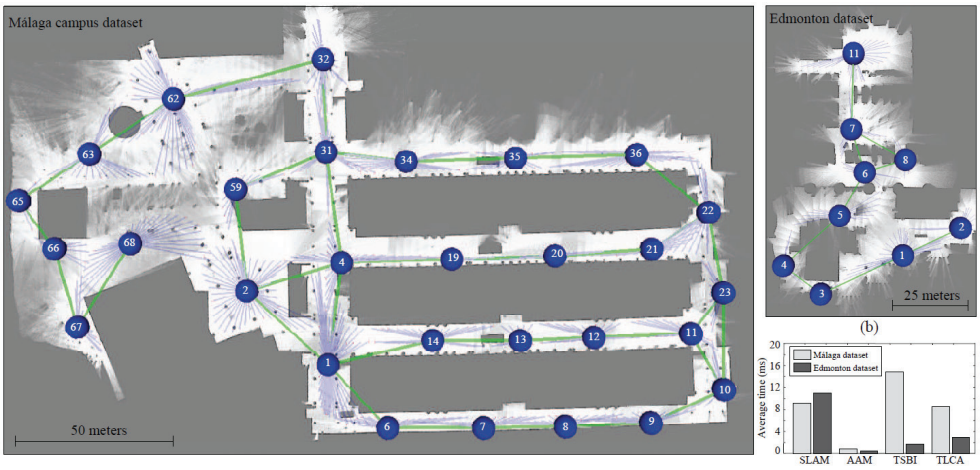


Figure 1.4 The sample of HMT resulting map from the work of [34]. The maps were generated from two different campus and shown as global maps relative to the first topological area (labeled as '1').

lines correspond to narrow passages such as doorways. By partitioning the metric map into a small number of regions, the number of topological entities is several orders of magnitude smaller than the number of celles in the gird representation. Therefore, the integration of both representations has unique advantages that cannot be found for either approach in isolation: the grid-based representation, which is considerably easy to construct and maintain in environments of moderate complexity (e.g., 20 by 30 meters), models the world consistently and disambiguates different positions. The topological representation, which is grounded in the metric representation, facilitates fast planning and problem solving.

Based on this concept of hybrid metric-topological SLAM, many premium recent researches address the problem of SLAM in large-scale dynamic environments very efficiently [33], [34]. For the current state-of-the-arts in SLAM, most approaches lay emphasis on trying to improve the efficiency of metric SLAM based on the concept of local metric global topological of hybrid SLAM. The method can successfully navigate the robot in both large-scale outdoor environment and small-scale indoor environment. However, the problem still exists when the environments is very dynamic and opened. Traditional sensors used for SLAM are proximity sensors and sonar. These sensors are useful for creating detailed metric map but capture insufficient information for recognizing some complex scenes. This motivates robotics researcher to include the most powerful sensors in robotic, a visual sensor.

Figure 1.4 portrays a typical generated hybrid metric-topological map in outdoor. The figure is taken from [34].

1.2.3 Appearance-based SLAM

Appearance-based localization and mapping have recently become hot topics of discussion in robotics because of the difficulty of detecting loop closure in metric SLAM [35], [36], [37]. Advanced computing technologies and the low prices of cameras have helped to supplement traditional metric SLAM methods with appearance-based visual information. Consequently, commonly used sensors such as laser scanners, radars, and sonar tend to be associated with, or replaced by, a mono or stereo camera. With the popularity of this topic, numerous approaches have been reported for fast and accurate localization and mapping.

Appearance-based SLAM is still considered one of the SLAM's paradigms. The use of word "appearance" or "vision" means that the sensors are not limited only to laser scanner or proximity sensors. Also, there is many works that focus on only improving in visual part so that the topic would be called "vision-only-based SLAM" or "appearance-only-based SLAM". Because visual sensor captures much more information than that of other sensors, the information extraction process becomes much more complex. By focusing on only visual sensors, one can quickly develop the most efficient method to extract salient information from abundant information contained in pictures.

The work presented in this thesis addresses the problem of appearance-based only topological SLAM. Only visual information has been used in order to find the most efficient way to extract most useful visual information from scenes captured in highly dynamic environments. By the term topological mapping, our presented work focuses on loop-closure detection rather than creating a whole coordinate-based map. That is, our work can either localize the input scene to the previously visited scene or augment the input scene as the new place into the map with 100% precision. As the work has already been successful, it can be easily connect to the research branches of SLAM such as metric localization and mapping approach to finally realize the practical and efficient SLAM system for mobile robots.

1.3 Chapter Summary

The chapter introduces the significance of SLAM problems towards researches and development in robotics. As described so far, localization, mapping, are loop-closure are also the essential components for navigation, and the navigation is a fundamental requirement for any mobile robots before any operations. Nevertheless, despite the long researches over decades, many problems still remain unsolved in SLAM. One of them is the loop-closure detection in dynamic environments. One of the efficient solutions is to integrate the visual data into proximity sensors and utilize both of them to accurately identify the place. Additionally, to focus on improvement of visual part, many concentrate on the topic vision-based-only SLAM.

This thesis aims to solve exactly the same task with the vision-based-only SLAM researches, but in more difficult form. A state-of-the-art approach [35] can perform loop-closure detection with the needs of batch processing of dictionary generation. Although some can run incrementally [65], it needs to trade off the incremental ability with the decreasing rate of accuracy. Therefore, we propose a new appearance-only-based SLAM which can run fully incrementally and offer significantly even higher accuracy than that of the offline methods. The computation time is sufficiently fast for real-time application. The system can be run anywhere without needs for specifically generated dictionary.

CHAPTER 2

IMAGE FEATURE

"A picture is worth a 1000 words" is one of the well known quotes in human society. Among all of the sensing organs of human, eyes seem to be the most important one considering those who suffer from being a blind man. Human possess the ability to capture and interpret most of the useful information in the picture while ignoring noises. For a simple idea, machine would become like human if it possess the same ability. Unfortunately, we have not yet clearly figured out how such ability functions.

One thing that has been known for many decades in computer vision community is that picture contains too large information and, in order to process picture (i.e., recognize, understand, classify), a compact representation is indispensable.

In computer vision and image processing research field, image feature is always one of the most important components in development. It is primary step in capturing useful information from pictures. Considering the same learning and testing frameworks, the accuracy of the recognition system might be drastically improved if the image feature was sufficiently good and vice versa. Even though the learning mechanism is very efficient, the overall performance will be incredibly low if the image feature is not appropriate. In the long history of computer vision and image processing, a number of image feature method has been proposed and gradually developed, each specifically appropriate to specific problems. For example, Histogram of Oriented Gradients (HOG) [38] is specifically powerful in human recognition, while GIST [39] feature of Olivia and Torralba is good for scene categorization.

Among many image features, only a few of them would be briefly reviewed

in this thesis. Because the main objective of the work, as stated in the previous section, is to address the problem of scene localization in dynamic environments, we consider only the features that are generally used in scene localization or recognition.

Image features used for localizing or recognizing scenes are divided into two main types: global and local feature. The global feature is a single feature used to globally represent a whole image, whereas the local feature only represent some interesting points in image. In other words, a single global feature is sufficient for representing an image but a local feature does not. Actually, the number of local features required to represent a single image can be and always be various, since the number of features directly depends on the number of interesting point detected in such image. If the scene is complex, a lot of local features are needed and vice versa. Currently, one of the popular and powerful global features are the GIST [39] and sPACT [40], while one of the well-known local features are SIFT [41] and SURF [42].

2.1 Global Image Feature

As already stated by its name, global feature considers an image as a whole and capture the most useful information. One of advantages of this type of feature is its compact size in image representation and its tractability in broad range of recognition system. Representing one image (data) with a single fixed-dimension feature vector has always been easy to use with any kind of machine learning approach such as support vectors machine (SVM).

Among many global image features proposed so far, there are recently two of them which have been reported successfully in robotics localization with an impressive accuracy: GIST and sPACT.

2.1.1 GIST [39]

The GIST feature is built based on the studies in brain science about the human's perception. In the studies, it has been shown that observers recognize a real-world scene at a single glance. He can comprehend a variety of perceptual and semantic information. The phenomenal experience of understanding everything at once, regardless of the visual complexity of the

scene, can be experienced while watching television and flipping rapidly through the channels: with a mere glimpse of each picture, observers can grasp each one's meaning (a politician, a car chase, the news, cartoon, etc.) independently of the clutter and the variety of details. This refers to the *gist* of a scene [43], [44].

Behavioral studies have shown that observers can recognize the basic-level category of the scene (e.g., a street [44]), its spatial layout (e.g. a street with tall vertical blocks on both sides), as well as other global structural information (e.g., a large volume in perspective) in less than 100 msec. Observers may also remember a few objects (e.g., a red car and green car), the context in which they appear (e.g., parked on the side) and other low-level characteristics of regions that are particularly salient.

Because gist includes all levels of visual information—ranging from low-level features (e.g., color, contours) to intermediate (e.g., shapes, texture regions) and high-level information (e.g., activation of semantic knowledge)—it can be represented at both perceptual and conceptual levels. Perceptual gist refers to the structural representation of a scene built during perception. Conceptual gist includes the semantic information that is inferred while viewing a scene or shortly after the scene has disappeared from view. Conceptual gist is enriched and modified as the perceptual information bubbles up from early stages of visual processing.

This idea of gist was realized in scene categorization with impressive results by Oliva and Torralba [39]. The first feature was created by the defining the spatial envelope properties of scenes: Degree of naturalness, degree of openness, degree of roughness, degree of expansion, degree of ruggedness. By considering these property and create the feature based on these properties, the feature generally represent an image as a whole, so that the result for categorizing scene (i.e., indoor/outdoor) is impressive.

Although the GIST was reports as being successful in scene categorization, Torralba et al. [39] use GIST feature to solve the robotics localization problem. The method is full batch learning algorithm using Hidden-Markov Model.

2.1.2 sPACT

Spatial Principal Component Analysis of Census Transform Histograms (sPACT) is a new representation for recognizing instances and categories of

places or scenes proposed by Wu and Rehg [40]. The Census Transform (CT) summarized local shape information, while the strong constraints among neighboring CT values and the PCA operation compactly encode the global shape in an image patch. Also, the spatial property of PACT encodes rough global spatial arrangement of sub-blocks in an image, and finds the tradeoff between discriminative power and invariance for place recognition tasks. The claimed advantages of sPACT over other features are:

- Superior recognition performance on multiple standard datasets;
- Significantly fewer parameters to tune;
- Extremely fast evaluation speed (>50 fps)
- Very easy to implement.

The evaluation of sPACT has been done with three main datasets. One of them is the KTH-IDOL [45] which is a dataset specific for robotics localization in indoor. Interestingly, sPACT clearly outperforms the method presented in [45].

The key success of sPACT is because it efficiently captures image structures in the 3 x 3 local area. The direct and indirect constraints existing among neighboring CT values propagate to pixels far apart, which enables PACT to implicitly capture the global shape in an image. PACT also handles the strong correlation among pairs of CT values using PCA. PACT is then combined with the spatial pyramid matching [46] idea. The spatial PACT was used to recognize both place instances and categories. Comparing with other representations, sPACT not only exhibits superior performance. It has nearly no parameter to tune and is easy to implement.

Despite of its success, there are some limitations of PACT. One of them is its variance against rotation. In dynamic environments, if some important landmarks were rotated (even with the scene upright), sPACT would simply lost information corresponding to such landmarks. This renders sPACT non-suitable to localization in dynamic environments.

2.2 Local Image Feature

For any computer vision system, the notion of a “feature” is important. Computer vision systems extract features from their input data and produce

useful information from those features. Castleman [47] defines a feature as follows.

> "A feature is a function of one or more measurements, computed so that it quantifies some significant characteristic of object."

This very general definition does not specify how exactly a feature is computed not if it describes the whole object to be measured or a part of it. This distinction is made in a further separation into *global features* and *local features*. Global features quantify characteristics of the whole image to be measured. An example would be a color histogram of an image which gives important information about the whole image. Local features quantify characteristics of a particular region of the object to be measured. For the case of images, a local feature may be limited to a small spatial area.

Why would one prefer one over other? The answer depends very much on the problem to be solved. For the problem of object classification and scene localization, it is advantageous to choose local image features with the following reasons.

- **Object to feature relationship**

  A global feature us a function of the entire image. As we are concerned with only a few objects that are considered as "good landmarks" in an individual place, it would be optimal to ignore any information not related to the landmark of our concern, such as the image background or other objects. In general, this is not possible, as the problem of foreground-background segmentation is non-trivial. However, by using local image features we can define image features in such a way that a single feature is unlikely to cover an area of the image that belongs to both the foreground object we are interested in *and* the background we want to ignore. Given a set of such extracted features, we do not know which features belong to the object and which do not, but we can be quite certain that each feature is either a feature from the foreground object or the background. This partitioning of primitive features can be exploited to our advantage in our problem setup.

- **Invariance against spatial transformations**

  The scene recognition and localization task we are trying to solve

should naturally be solved in a way such that moving the object within the image or rotating it should not affect the recognition result. In general, good localization system should be invariant against spatial transformations. Using some global features this can be achieved rather easily for some variances, such as translation and rotation in the view plane. But for other variances, such as enlargement of the object (because robot's observing positions are usually uncertain), it can be difficult to design a global feature to produce invariant values. For local image features techniques have been developed to extract features invariant of translation, rotation and changes in scale, while also being robust to general affine transformations.

- **Relationship between features**

  While a global feature condenses characteristics of an object to be measured into one piece of information, for local features not only the single feature itself is important but it may be the case that the *relationship between features* gives more useful information about the object than any single feature alone.

If local features or any local structures are analyzed within digital images, the notion of *scale* is crucial. It is the relative scale of structure present in the image that separates it from being a detail or a fundamental structure in the image. Additionally, the scale can imply hierarchy of the features, that is, it is structuring the primitive features. How this hierarchy relates to the computer vision problem at hand depends very much on the problem itself, but such a natural hierarchy almost always contains helpful information.

Among recent local image features, we discuss in detail the most popular features that have been used in robotics localization and mapping, the scale-invariant feature transformation (SIFT) [41] and speed-up robust feature (SURF) [42].

2.2.1 SIFT [41]

One of the most popular local image feature for general natural images is SIFT—the Scale-Invariant Feature Transformation—which was developed in 1999 by David Lowe [48], and later refined and extensively described in [41]. In this section, we examine SIFT in detail as our proposed feature is developed

based on this SIFT.

The SIFT algorithm is based directly on the scale-space framework given by Lindeberg [49], but extends the idea of locating interest points in scale-space to also describe their characteristics in a way so the resulting descriptor is invariant or robust against changes in scale, rotation and affine transformations. SIFT comprises four main steps.

1. **Detection of scale-space extreme**: The first stage of computation searches over all scales and image locations. It is implemented efficiently by using a difference-of-Gaussian function to identify potential interest points that are invariant to scale and orientation.
2. **Keypoint Localization**: At each candidate location, a detailed model is fit to determine location and scale. Keypoints are selected based on measures of their stability.
3. **Orientation Assignment**: One or more orientations are assigned to each keypoint location based on local image gradient directions. All future operations are performed on image data that has been transformed relative to the assigned orientation, scale, and location for each feature, thereby providing invariance to these transformations.
4. **Keypoint descriptor**: The local image gradients are measured at the selected scale in the region around each keypoint. These are transformed into a representation that allows for significant levels of local shape distortion and change in illumination.

The first stage used difference-of-Gaussian (DoG) function to identify potential interest points, which were invariant to scale and orientation. DoG was used instead of Gaussian to improve computation speed.

$$D(x, y, \sigma) = \big(G(x, y, k\sigma) - G(x, y, \sigma)\big) * I(x, y) = L(x, y, \sigma k) - L(x, y, \sigma)$$

In the keypoint localization step, they rejected the low contrast points and eliminated the edge response. Hessian matrix was used to compute the principal curvatures and eliminate the keypoints that have a ratio between the principal curvatures greater than the ratio. An orientation histogram was formed from the gradient orientations of sample points within a region around the keypoint in order to get an orientation assignment as shown in figure 2.1. According to the experiments, the best results were achieved with a 4×4 array of histograms with 8 orientation bins in each. So the descriptor of SIFT that

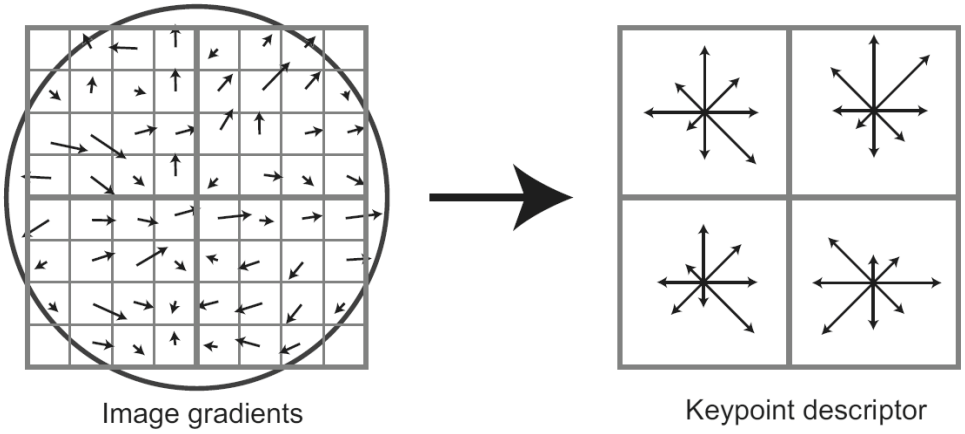


Figure 2.1 A keypoint descriptor is created by computing the gradient magnitude and orientation at each image sample point in a region around the keypoint location, as shown on the left. These samples are then accumulated into orientation histograms summarizing the contents over 4×4 sobregions, as shown on the right, with the length of each arrow corresponding to the sum of the gradient magnitudes near that direction within the region. The figure shows a 2×2 descriptor array computed from an 8×8 set of samples.

was used is 4×4×8 = 128 dimensions.

2.2.2 SURF [42]

Speeded Up Robust Feature (SURF) proposed by [42] employ slightly different ways of detecting features comparing to SIFT. SIFT builds an image pyramids, filtering each layer with Gaussians of increasing sigma values and taking difference. SURF, however, creates a "stack" without 2:1 down sampling for higher levels in the pyramid resulting in images of the same resolution. Due to the use of integral images, SURF filters the stack using a box filter approximation of second-order Gaussian partial derivatives, since integral images allow the computation of rectangular box filters in near constant time.

In keypoint matching step, the nearest neighbor is defined as the keypoint with minimum Euclidean distance for the invariant descriptor vector. Low used a more effective measurement that obtained by comparing the distance of the closest neighbor to that second-closest neighbor so the author of SURF decided to choose 0.5 as distance ratio like Lowe did in SIFT.

Nevertheless, one of the main advantages of SURF that make it very popular in robotics application is the high-speed extraction and matching time

comparing to SIFT. For the specific problem of scene recognition and localization in robotics SLAM, SURF offer approximately same accuracy as that of SIFT while its computation time is considerably faster. Many works in vision-based SLAM use SURF as the main feature [35], [36], [45], [50].

2.3 Chapter Summary

It is clear that recently local image feature seems to outperform global feature in appearance-based localization in robotics. However, even with the bag-of-word representation, recall rate at precision-1 is still considerably low [35], [65]. Although SIFT and SURF are very powerful in its discriminative power, a number of noisy features also exist and sometimes greatly reduce the performance of the system. In order to deal with the highly dynamic environment, a new local feature is required. We need the feature which captures only landmarks that is likely to appear permanently in the place while ignoring other movable objects. This motivates us to propose our novel local feature Position-Invariant Robust Feature, coined as PIRF which would be described in detail in the next chapter.

CHAPTER 3

POSITION-INVARIANT ROBUST FEATURE

By what we have discussed so far, it becomes clear that the local image features offer better performance in robotics localization and loop-closing than the use of global feature. Nevertheless, even with the currently most popular local feature like SIFT and SURF, the problem of dynamical changes in scene still exists. SIFT and SURF captures only interesting points in the scene which, in most cases, belongs to objects or landmarks. This enables the automatic rough image segmentation that separate foreground image from back ground image. Nevertheless, it is very common that SIFT always contains some noisy SIFTs—a SIFT which seems to appear only one time and is unlikely to appear again even with the exactly same object under the same lighting condition. This concern becomes more significant when it comes to the dynamic environment. For example, using SIFT will captures building, dogs, car, people as landmarks for place A. However, one the robot re-visit the place, dogs, car and people all disappear because they are not the stationary objects. With three in four landmarks being lost, robot is not confident in localization. Therefore, with this motivation, we present our original position-invariant robust feature, coined as PIRF, to use for efficient place localization in outdoor dynamic environments. This is the first contribution of this thesis.

3.1 Introduction to PIRF

Localization is an indispensable capability for both humans and machines. Knowing “Where we are” has always been an important topic in robotics and

computer vision communities. Especially for mobile robots, knowing its position in the world is a common fundamental requirement for navigation systems. The topic has been studied for more than two decades using several methods: metrical, topological, and hybrid (see [51], [52] for reviews). Although sonar and laser scanners have traditionally been the sensory modalities of choice [53], current advances in visual tools––video cameras, omnidirectional lenses––have made visual approaches more attractive, providing richer information at a lower price.

From the perspective of computer vision, an efficient robot vision system might need to overcome three difficulties: dynamical changes, viewpoint changes, and scene categorization. In a highly dynamic environment (i), places might look very different over time because of illumination changes (daytime, nighttime) and because of moved objects: parking lots are empty on holidays. These changes are dynamic because their appearances are stable only for only some period of time. Regarding the second sub-problem (ii), different viewpoints often make a scene look different. This problem also includes changes in weather and lighting conditions. An object's appearance might be very different when observed from different camera positions, even if viewed at exactly same time. Scene categorization (ii) describes how the robot understands the scene so that it can categorize new unseen places along with those it has seen previously. Inspired by biology, this ability further reduces the gap separating robots and humans. Recent scene recognition approaches might be divided into three main types: *Object-Based*, *Region-Based*, and *Context-Based*.

To date, most approaches to scene recognition have been *object-based* [54], [55], [56]. Using such approaches, a scene location is recognized by identifying a set of landmarks known to be included in a scene. These approaches are prone to carrying over and amplifying low-level errors along the stream of processing. For instance, upstream identification of small objects (pixel-wise) is hindered by downstream noise inherent to camera sensors and by varied lighting conditions. This is problematic in spacious environments where landmarks are more dispersed and more distant from the agent. This approach must be environment-specific to ensure the simplicity of selecting a small set of anchor objects as landmarks in an open problem.

For *region-based* scene recognition, the segmented image regions and their configurational relations are used to form a signature of a location. The major

problem hindering this approach is reliable region-based segmentation, in which individual regions must be characterized robustly and associated. Naïve template matching involving a rigid relation is often insufficiently flexible in the face of under-segmentation or over-segmentation, which is often true with unconstrained environments, such as outdoors. Some techniques such as normalized-cut [57], [58] are useful to improve segmentation quality. Nevertheless, the computation time for image segmentation might be problematic for real time applications.

Context-based approaches, unlike both previously described approaches, bypass traditional processing steps. Context-based approaches examine the input image as a whole and extract a low-dimensional signature that compactly summarizes the image's statistics and semantics. The challenge of discovering a compact and holistic representation for unconstrained images has therefore prompted considerable research effort recently. Renniger and Malik [59] use a set of texture descriptors and a histogram to create an overall profile of an image. Ulrich and Nourbakhsh [60] build color histograms and perform matching using a voting classifier. Oliva and Torralba [39] encode some spatial information by performing 2D Fourier Transform analyses in individual image subregions on a regularly spaced grid. The resulting spatially arranged set of signatures, one per grid region, is further reduced using principal component analysis (PCA) to yield a unique low-dimensional image classification. In more recent implementations, Torralba *et al.* [61] used steerable wavelet pyramids instead of the Fourier Transform to solve the robotic localization.

Among the three approaches described above, PIRF is most related to an *object-based* approach, in the sense that natural landmarks would be used as a signature of place. Our selection is motivated by the observation that outdoor scenes generally include distant objects such as distant buildings or walls. These distant objects seem to appear constantly in scenes irrespective of the camera position. Even in a highly dynamic scene where major components of scenes are changed, these small distant objects still appear. In this case, global representation of whole scenes might be problematic; such scenes include many unstable nearby objects that later fail the recognition. Therefore, we address this highly dynamic scene recognition problem as:

- how to detect objects *autonomously* which are visible to almost every

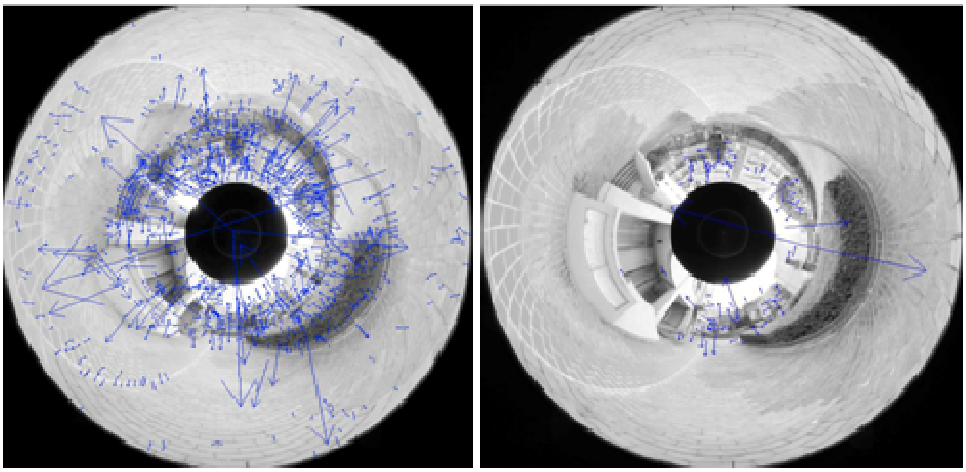


Figure 3.1 Sample omnidirectional outdoor image extracted with original SIFT (left) and the proposed PIRF (right). The distant objects' appearances are invariant to position changes.

position in such place;

- how to detect objects autonomously which are unique to a single place; and
- how to describe such objects precisely despite their distance.

These distant objects can be a good signature of each place; a group of these objects can be used efficiently to identify places. As described herein, we propose a Position-Invariant Robust Feature (PIRF) [62] as an image local feature that solves these three problems.

The PIRF is developed upon existing local descriptors such as Scale-Invariant Feature Transformation (SIFT) [41] and Speeded Up Robust Feature (SURF) [42]. Local features extracted from many individual images are filtered to derive the descriptors which appear repeatedly in almost every scene (taken at the same place). These descriptors are averaged to generate a single representative descriptor called PIRF. Filtering is done autonomously using simple feature matching as in an earlier study [41]. Considering figure 3.1, given several SIFTs extracted from many interesting points of an image (left), and assuming that some SIFTs appear repeatedly in a few more sequential images, PIRFs are then generated by interpolating those corresponding SIFTs. Figure 3.1 (right) shows PIRFs extracted from standard SIFTs. In this case, almost all PIRFs are only of distant objects, whose appearances are very stable. This solves the *first* problem. One place can be

represented using several PIRFs because one place might include many distant objects; each requires many PIRFs for representation. For later reference in this paper, we call these representative PIRFs (for one individual place) a PIRF-dictionary. For example, three places require three PIRF-dictionaries for representation. Because of the descriptive power of the existing descriptor, detected objects can be described precisely using a set of PIRFs (representatives of slow-moving SIFTs or SURFs). This solves the *third* problem described earlier. However, it is also clear that collecting many PIRFs will eventually pose the problem of many duplicated PIRFs, which confuse recognition in the long run. Therefore, we additionally propose a technique for eliminating these redundant PIRFs. The technique is incremental; at any time, it can rapidly search for redundant PIRFs and delete them from memory. This solves the *second* problem above.

We also describe a simple approach to use PIRFs for scene recognition. The recognition system is portrayed in figure 3.2. Assume that an environment contains five separate places. Each place has its own PIRF-dictionary $\mathcal{D}^i,\ i \leq 5$ for representation. First, a set of SIFTs is extracted from a testing image. Feature matching is performed to match each single SIFT to a set of PIRFs in each dictionary. A place, whose dictionary contains the highest number of PIRFs that can be matched to the extracted SIFTs, is justified as the winner. In figure 3.2, for instance, both the first and the second image belong to place 1 because the number of matches between SIFTs and PIRFs in $\mathcal{D}^1$ is the highest.

To demonstrate the advantages of PIRF, we test it on 1000+ outdoor omnidirectional images collected from two campuses. Training and testing data are collected, respectively, on holidays and weekdays to address the difficulties of highly dynamic changes over time. The results show that PIRF obtains a markedly higher rate of recognition than other features. Additionally, we test PIRF on indoor images collected by a real robot. The images are taken from the standard Cognitive Systems (COsy) (for Cognitive Assistants Localization Database (COLD)) [50], which is available online at http://cogvis.nada.kth.se/COLD/. Results show that, although PIRF cannot outperform other features in indoor scenes, its performance is good and not so different from the others.

We also describe a simple robot navigation method that uses PIRF (designated as PIRF-Nav) as a basic feature, to confirm PIRF's importance in

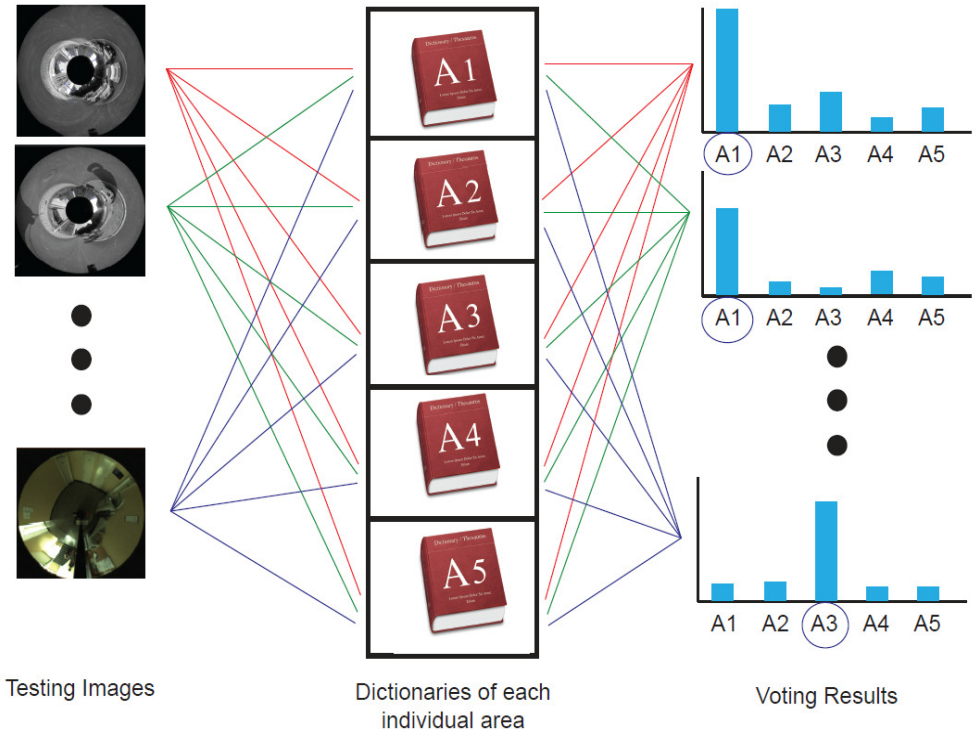


Figure 3.2 Illustration of PIRF voting for scene recognition. Each dictionary votes for the matched PIRF included in an input image. The voting result justifies the location of the scene.

relation to robotics. The PIRF-Nav is tested on one campus. The image dataset is identical to that used previously for scene recognition. Results show that PIRF-Nav outperforms Incremental Spectral Clustering [36], [63] in both time and accuracy.

3.2 Notable Works Related to PIRF

Various approaches used in the past have addressed Scene and Place recognition. Many effective features and various modes of use have been proposed. Histograms of image properties, *e.g.* color [60] or image derivatives have been used widely in place recognition. However, after SIFT [41] was popularized among the vision community, it came to dominate feature choice in place recognition systems [36], [64], [65]. The SIFT features are invariant to scale and are robust to rotation changes. The 128 dimensional SIFT

descriptors have high discriminative power, but are simultaneously robust to local variations [66]. For place recognition, SIFT outperforms edge points [46], pixel intensities [67], and steerable pyramids [68].

As the most appealing descriptor for practical uses, SIFT has been used widely in appearance-based navigation [37], [69] because matching digital image contents among different views of scene requires a good distinctive invariant feature. Although SIFT satisfies such a requirement, its computation time and memory cost are high. As a remedy, Ledwich and Williams [70] reduced SIFT features by taking advantage of the structure of the indoor environment where average view-depths of most images are short. This makes vertical planes such as walls dominate an image's composition. The rotational invariance of SIFT has also been removed by assuming that the viewpoint for indoor images will be stable to rotation around the view axis, resulting in a non-rotated orientation of the keypoint descriptors located on vertical surfaces. The method is specifically useful for indoor environments.

Meanwhile, Oliva and Torralba [39] suggested that recognition of scenes can be achieved using "global configurations", without detailed object information. Consequently, statistical analysis of SIFT distribution became popular. Torralba *et al.* [61] use global image features to generate Gaussian Mixture Models for place recognition, using fixed variance. The method gives limited tolerance for appearance variation and is not invariant to translation or scale changes. Lazebnik *et al.* [46] use the *k*-means algorithm to cluster SIFT features, and cluster centers were used as the codebook to solve 15-class scene recognition. Cummins and Newman [35] integrate bag-of-visual-words (BoW) into the recursive probabilistic Bayesian framework and achieve performance beyond the localization. That method can determine that a new image has come from a previously unseen place. Later, Angeli *et al.* [65] proposed incremental BoW. Starting from an empty dictionary, the system can gradually collect new words while localizing places. Recently, Wu and Rehg [40] proposed the spatial Principle Component Analysis on Census Transform (sPACT) as a feature for scene recognition and categorization. Its performance has proven to be better than that of the BoW method [46]. The authors reported the highest accuracy over the KTH-IDOL dataset [45].

In robotics, scene recognition and localization are also important topics. Several previously proposed methods should also be acknowledged here. Robot localization can be done based on vision alone [50], [71], [72], [73], or based on

combinations of vision and other sensors (*e.g.*, laser-scanner, odometer) [60], [74], [75], [76], [35]. The methods can be performed either offline or online for use indoors or outdoors. Clemente *et al*. [74] generated metric maps based only on a hand-held camera. The key ingredient of the method is the inverse-depth representation [77], which can estimate the depth of local features even in a single observation. The camera motion is an important concern for representing the main map. Royer *et al.* [72] reported their success in building the 3D reconstruction of map from sequential scenes by calculating the robot motion. Distinct from [72] and [77], the method of Luo [78] emphasizes single-image recognition: neither robot motions nor its position is incorporated in the method. The incremental support vector machine (SVM) is used to train the localization system for indoor use. The authors apply two techniques to extend SVM to an incremental version: *Fixed Partition* and *Error-Driven*. Nonetheless, the incremental step used in their work contains some images partitioned by the user. The method must go offline until the step has been completed. In addition, the adopted error-driven technique requires a human to stand by the robot's side and tell it whether the recognition is corrected or not. Valgren and Lilienthal [36], [63] use incremental spectral clustering (ISC) to cluster images and thereby create a topological map. Their method was reported as a fully incremental mapping method in highly dynamic outdoor environments (across seasons). The number of nodes in the map (data segmentation) is determined autonomously. The incremental BoW [65] is also a fully incremental mapping method. An empty dictionary can be updated incrementally. This method is considered as state-of-the-art vision-based mapping. However, the method addresses only the loop-closure detection problem; all obtained images up to the current time are included in the process of probabilistic decision-making. They did not report the recognition rate of a single image. In other words, their method is especially applicable for robotics. Furthermore, they did not address the problem of highly dynamic changes in scenes; all images in the study seem to be collected on one day.

Here we confirm again that PIRF is new. From a computer vision perspective, PIRF is more discriminative than global features, but less noisy than using standard local features. Unlike other object-based approaches, PIRF can autonomously detect the landmarks which are robust against highly dynamic changes of scenes. From a robotic perspective, navigation based on PIRF (PIRF-Nav) outperforms ISC in terms of time and accuracy. PIRF-Nav is

incremental like [65]; it can recognize single images and close the loops, while its performance is beyond localization like the method of [35]; it can determine that a new observation come from a previously unseen places. Its robustness against dynamic change also leaves some room for additional improvement and applications (*i.e.*, using PIRF as the local feature for generating BoW).

3.3 Definition of PIRF

A Position-Invariant Robust Feature (PIRF) is a single local feature that is robust to any position along the path within the same place. The idea comes from observing that outdoor scenes generally include faraway objects. These objects are useful to identify the place because their appearance is stable, irrespective of position changes. Precisely, PIRF is a single local descriptor computed as an average of existing local descriptors, such as SIFT [41] or SURF [42], which has wide baseline visibility. Actually, a PIRF must be extracted from sequential images because it must retrieve all associated features from these images and compress them into a single PIRF. Many single PIRFs are collected to form an individual PIRF-dictionary of a place (one place contains many sequential images). An individual dictionary is a signature of an individual place. In other words, a video sequence is segmented into N places (partitions). Each place *i* contains n_i sequential images. It seems very difficult to find a fine number of features that appear in all images. Therefore, the place is divided into many sub-places before extracting PIRF. Figure 3.3 portrays extraction of PIRFs from a single place. By repeating this extraction process for every place, N dictionaries can be obtained, where each contains PIRFs used for representing an individual place. The algorithm has three main stages: *Sequential Image Matching, PIRF Extraction, and Place Recognition*.

3.3.1 Sequential Image Matching

Given N as the current number of all visited places in an environment, n_i is the number of sequential images $\mathbb{I}_{\mathrm{i}} = \{I_1, \dots, I_{n_i}\}$ of the i^{th} place, where $i \leq \mathrm{N}$. Matching is performed sequentially for every pair of images; namely $(I_1 - I_2), \dots, (I_{n_i-1} - I_{n_i})$. We use the same matching criteria as that used in an

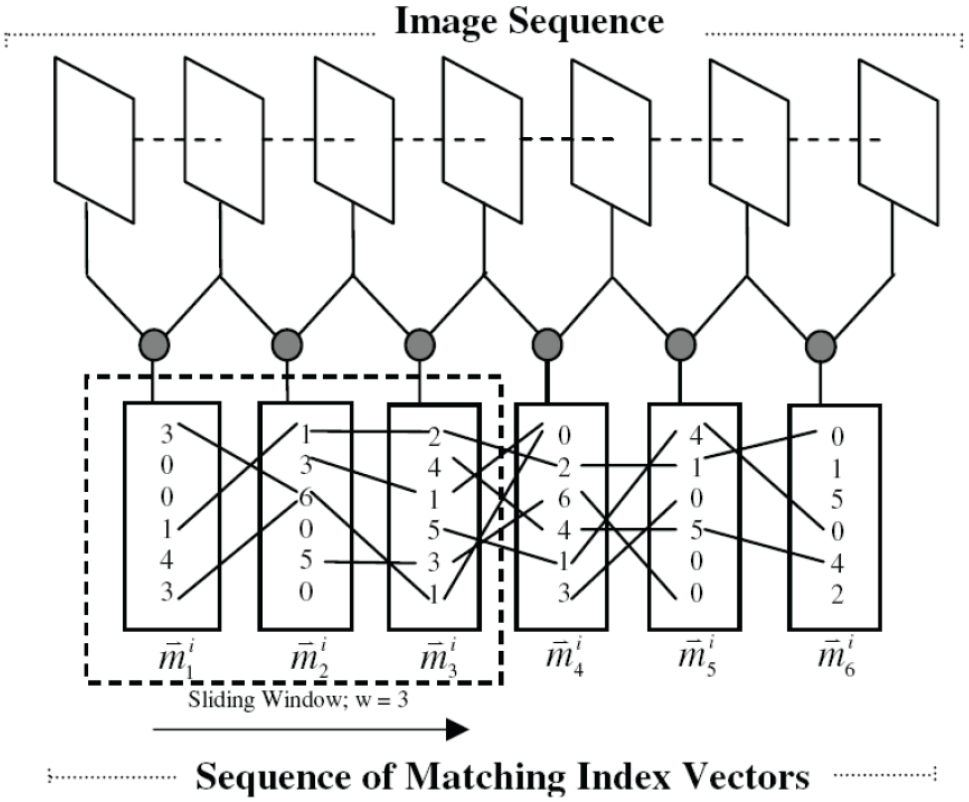


Figure 3.3 Sample PIRF extraction of the i^{th} place. Given the number of sequential images n_i = 7 and the size of sliding window w = 3. Number of extracted SIFT from each image is 6. Every image pair is compared using feature matching, resulting in six matching index vectors. A vector element is the index of the corresponding feature in the next image. For example, for the first sub-place ($\vec{m}_1^i, \vec{m}_2^i, m_3^i$) of I_1, I_2, I_3, I_4, there are only three features appearing in all images: (1,3,6,1), (4,1,1,2), (6,3,6,1). (1,3,6,1) is interpreted, respectively, as the 1st, 3rd, 6th, and 1st feature of image I_1, I_2, I_3, I_4. These four features are interpolated to obtain a single representative PIRF. Therefore, there would be 3, 4, 4, 3 PIRFs for the 1st, 2nd, 3rd, and 4th sub-place respectively, 14 PIRFs in all for the whole i^{th} place.

earlier study [41]. The threshold value is set to 0.6. After every pair of images has been matched, the matching result is retained as the matching index vector, $\vec{m}_q^i = \left(m_{1,q}^i, \dots, m_{k_q,q}^i\right)$, where $q < n_i$, $k_q$is the number of local features of I_q. For example, $\vec{m}_1^1 = (10,0)$ is interpreted as the first matching between I_1 and I_2 of the i^{th} place results in, out of two features, only one matched feature. The first feature of I_1 matches the tenth feature of I_2, whereas the second features of I_1 are not found in the image I_2. As described herein, we select the

SIFT of [41] as our descriptor.

3.3.2 PIRF Extraction

Considering the $(n_i)^{\text{th}}$ image (the last image of the i^{th} place), after $n_i - 1$ matching index vectors $\vec{m}$ are derived, then the PIRF is extracted. However, an object with a stable appearance irrespective of the changed position is difficult to find because the path might be long or curved. Therefore, we instead extract those features which are positionally invariant in relation to the sub-place. Considering the sequence of vector $\vec{m}_q^i$ as the sequential input data, sliding windows feature extraction is performed to collect PIRFs from many sub-places instead of the whole place. For example, if $w = 3$, then the first sub-place contains $\vec{m}_1^i, \vec{m}_2^i, \vec{m}_3^i$ corresponding to I_1, I_2, I_3, I_4, and the second sub-place contains $\vec{m}_2^i, \vec{m}_3^i, \vec{m}_4^i$ corresponding to I_2, I_3, I_4, I_5. The window size is *w*; the window is shifted by one, which means that, given D_j^i as the PIRF-dictionary containing a set of descriptors corresponding to the j^{th} window (sub-place), there would be $n_i - w + 1$ dictionary for representing the place when the extraction is completed.

The depiction of Algorithm 1 shows how extraction is performed. Given n_i images of the i^{th} place, and a set of matching index vector $\vec{m}$ derived from sequential matching in the previous sections, a sliding window of size w is created to extract the PIRF of sub-places. For each sub-place j, all matching index vectors $\vec{m}$ are processed to find those local features which appear repeatedly in all images of the current window (line 3). All corresponding $w - 1$ features would be retrieved and put into the temporary matrix M if such features were found (line 4). These features are interpolated using averaging to obtain a single representative feature $\vec{\psi}$. **This feature $\vec{\psi}_{x,j}^i$ is the x^{th} single PIRF of the j^{th} sub-place of the i^{th} place.** Each extracted PIRF is gradually collected into the PIRF-dictionary of the j^{th} sub-place D_j^i (line 6). This extraction process is repeated until the window is slid to the last image of the place. Each sub-place has its own dictionary $\mathrm{D}_j^i, j \leq n_i - w + 1$. These dictionaries are finally concatenated mutually to form the dictionary of the i^{th} place $\mathcal{D}^i$ (line 9). Given d_j as the number of PIRFs in the dictionary of the j^{th} sub-place, $\mathcal{D}^i$ as the PIRF dictionary of the i^{th} place, and $n_{\mathcal{D}}^i$ as the total number of PIRFs in $\mathcal{D}^i$, the PIRF dictionaries for representing all visited places $\mathbb{D}$ are derived as presented below:

Algorithm 1: PIRF Extraction of the i^{th} place

Require: $\vec{m}$ is the matching index vector, as shown in Fig. 3.3
Require: w is the sliding window size
1: for j = 1 **to** $n_i - w$
2: **for** i_2 = 1 **to** w
3: **if** $isFoundInAllImage(m^i_{i_2,j}, w)$ **then**
4: $\mathrm{M} \leftarrow retrieveAllCorrespondedFeature(i, j, i_2, w)$
5: $\vec{\psi} \leftarrow interpolate(\mathrm{M})$
6: $\mathrm{D}^i_j \leftarrow addNewEntry(\mathrm{D}^i_j, \vec{\psi})$
7: **end if**
8: **end**
9: $\mathcal{D}^i \leftarrow addNewEntry(\mathcal{D}^i, \mathrm{D}^i_j)$
10: end

$$\mathrm{D}^i_j = \begin{bmatrix} \vec{\psi}^i_{1,j} \\ \vdots \\ \vec{\psi}^i_{d_j,j} \end{bmatrix}, \ \mathcal{D}^i = \begin{bmatrix} \mathrm{D}^i_1 \\ \vdots \\ \mathrm{D}^i_{n_i-w} \end{bmatrix}, \quad \mathbb{D} = \begin{bmatrix} \mathcal{D}^1 \\ \vdots \\ \mathcal{D}^{\mathrm{N}} \end{bmatrix} \tag{3.1}$$

$$n^i_{\mathcal{D}} = \sum_{j=1}^{n_i} d_j \tag{3.2}$$

Therein, $\mathbb{D}$ is useful to represent all visited areas in the environment. Extraction is incremental because the new area can be simply added to the library. Additionally, it is worth noting that extracted PIRFs must match images only $\left(\sum_{i=1}^{\mathrm{N}} n_i\right) - 1$ times, whereas the spectral clustering (SC) requires $\left(\sum_{i=1}^{\mathrm{N}} n_i\right) \times \left(\left(\sum_{i=1}^{\mathrm{N}} n_i\right) - 1\right)/2$ times to form the affinity matrix. Although incremental spectral clustering (ISC) [36] performs fewer image comparisons than SC, the comparisons are still much more numerous than those using our method.

3.3.3 Place Recognition

Now that all N individual places, $\mathbb{P} = \{\mathrm{p}_1, \ldots, \mathrm{p}_{\mathrm{N}}\}$, are well represented using a set of corresponding PIRF-dictionaries $\mathbb{D} = \{\mathcal{D}^1, \ldots, \mathcal{D}^{\mathrm{N}}\}$, we can

describe how these PIRF dictionaries are used to recognize places. Majority voting is selected as the recognition framework.

Majority voting (MV) is a popular combination scheme because of its simplicity and its performance on real data. Its performance has been demonstrated experimentally in many studies such as handwriting recognition [79] and personal authentication. We select MV because of its main concept related to the independence of recognizers. Based on theoretical analyses, MV is apparently effective if the recognizers are independent. Considering our problem, we assume that each place is independent. By applying MV to our problem, each place vote for the matched descriptors is found in the testing image. Additionally, MV is suitable for the task of incremental map-building in robotics, as described in [64], because a similarity threshold for image comparison is not needed. The image is assigned to the place with the maximum number of votes.

Consider the problem in which a single omnidirectional image I is to be assigned to one of N possible existing places $(\mathrm{p}_1, \ldots, \mathrm{p}_\mathrm{N})$. First image I is extracted and a set of descriptors, $\mathbb{Z} = (\vec{z}_1, \ldots, \vec{z}_n)$, is derived, where $\vec{z}$ is a single image descriptor and n is the number of descriptors. Of N places, each checks if the descriptor $\vec{z}_k, 1 \le k \le n$ is similar to any PIRF in its dictionary $\mathcal{D}^i, 1 \le i \le \mathrm{N}$. The vote is counted and the score is increased by one for every matching: we initialize $\mathrm{S}_i \to 0$ for every i,

$$\mathrm{S}_i = \mathrm{S}_i + 1 \qquad if \qquad \min_{1 \le j \le n_{\mathcal{D}}^i} \left| \vec{z}_k - \vec{\psi}_j^i \right| < \tau \quad , \tag{3.3}$$

where τ is the similarity threshold for feature matching (we found earlier that $\tau = 0.6$ yields the best performance). The vote from places can be done in parallel, thereby enabling rapid recognition. After voting has been completed, the system recognizes the image I as

$$\text{assign} \qquad I \to \mathrm{p}_{\mathrm{argmax}_i(\mathrm{S}_i)}$$

with confidence

$$c_{\mathrm{argmax}_i(\mathrm{S}_i)} = \frac{\mathrm{S}_i}{\sum_{j=1;\, j \ne i}^{\mathrm{N}} (\mathrm{S}_j)}. \tag{3.4}$$

We have now described image classification to an existing class. In the next

section, we consider incremental topological mapping by which the input images might come to belong to either an old or a *new* place.

3.3.4 Reducing the number of PIRFS

In the view of long-term recognition, a main concern for using PIRFs is the amount of PIRF in the current system. One PIRF-dictionary is used as a signature of one place. Therefore, the number of dictionaries depends on the number of visited places. For long-term recognition, the number of places is infinite, which means that the number of PIRF-dictionaries would also be infinite. Two techniques are used to solve this problem: reducing (i) the number of PIRFs or (ii) the number of dictionaries. The first technique is to slow the growth rate of PIRF. However, even though the number of PIRFs grows very slowly, it will finally reach the memory limits. The second technique is used here to delete or forget an unnecessary dictionary. The techniques are simple but efficient.

3.3.4.1 Reducing PIRFs

Most PIRFs are of distant objects whose appearances are robust against the change of viewpoints along the path. These objects, on the other hand, are also likely to be detected as distant objects in other places as well. For example, Tokyo Tower is visible in many places throughout Tokyo. Seeing the tower does not help in identifying the place. Therefore, the PIRFs which capture the Tokyo Tower are useless and should be deleted. These PIRFs might be treated as "redundant PIRFs". To eliminate these PIRFs, training images can be re-used: some recognized images were retained for the following retest. By this retest, the system knows which PIRFs match to the right object, and which PIRFs do not. Particularly, given $\mathbb{D}_t = \{\mathcal{D}_1, \dots, \mathcal{D}_{n_t}\}$ as the set of PIRF-dictionary up to time *t*, and I as an input testing image, the dictionary with the highest matched PIRFs, denoted by $\mathcal{D}_{win}$, wins recognition with confidence (vote quality) *c*. This recognition result is used to update the scores of all PIRFs in all dictionaries. For every matching between descriptor x of image I and the corresponding PIRF $\vec{\psi}_i^{win}$ of dictionary $\mathcal{D}_{win}$ where $1 \le i \le n_{\mathcal{D}}^{win}$, increase the score α_i^w of the PIRF by 1. In contrast, for every matching between descriptor x of image I and the corresponding PIRF $\vec{\psi}_i^{\mathbb{D}_t - \{\mathcal{D}_{win}\}}$, decrease the score of the PIRF by 1. A *high-score* PIRF can be

inferred as a useful PIRF; it often matches features of a distinctive object in the place, whereas a *low-score* PIRF can be interpreted as a PIRF which either captures confusing objects (an object that is visible from many places) or which captures highly sensitive objects (an object which is visible to only a few camera positions in such places). After a re-test, almost all PIRFs would already be assigned scores. Sorting the PIRFs by their scores, the number of PIRFs can be reduced by rate R (*i.e.*, R=0.75, R=0.50). The reduction drastically shrinks the PIRFs without a marked drop in accuracy. A PIRF with $\alpha = 0$ is simply treated as a redundant PIRF.

Generally, this reduction technique is performed *offline* because it must run a batch retest on previous images to update scores of PIRFs. However, "when to update the scores" is flexible; the robot can wait until it is free to take time thinking of the past and to update the dictionaries. This reduction could be postponed if the robot was busy with some task. This process can also be done *online* by taking advantage of the assumption of physical robots. Actually, scores can be updated every time the system recognizes a new image. What the system must know is whether the recognition result is correct or not. This is solvable by assuming that the robot actually obtains more than two images before making a decision. Therefore, once the system recognizes input image I as place p_w (with corresponding dictionary $\mathcal{D}_w$), it continues recognizing the first and second next images to confirm further that the images really belong to $p_{w;}$ then it updates the score of PIRFs. Details about this online reduction method are described in the next section of robotic applications.

Figure 3.4 portrays PIRF scores obtained by running a retest on all 382 training images. Most PIRFs were used at least once. The score separates the good and the bad PIRFs. We later show that, even after reducing the size of PIRF by 50%, the recognition rate is still high, which underscores the effectiveness of the reduction.

3.3.4.2 Forgetting Places

Certainly, long-term recognition will eventually confront the problem of memory overload because the number of places is infinite. For that reason, a robot must "forget" some places that are considered to be of no use, or at least temporarily remove such places from the searching space to speed up the localization time. In fact, PIRF-based recognition uses the PIRF-dictionary as

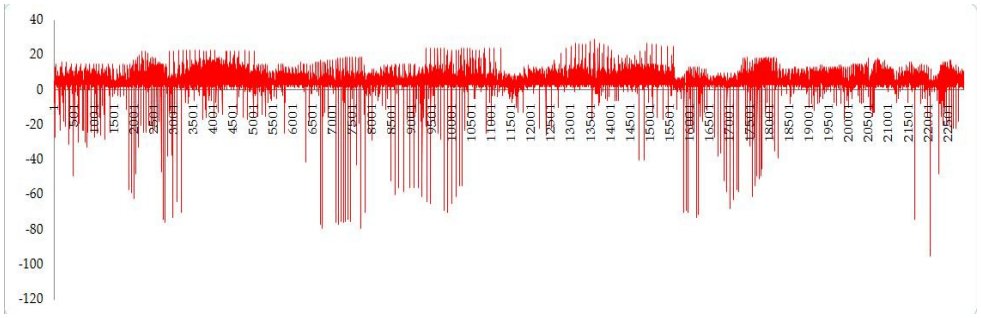


Figure 3.4 Updated score of 22901 PIRFs corresponding to 15 places of Suzukakedai Campus. The re-test was done over the same set of 382 images. The *x*-axis is the frequency score of the PIRFs, the axis is the frequency score of the PIRFs, and the *y*-axis are the indices of all 22901 PIRFs. We found that the dictionary with a high average score of PIRFs (i.e., PIRFs of index 11414–15629 of the 6th place $\mathcal{D}^6$) is extracted from the isolated place where most of the faraway objects (i.e. high building) are not shared by other places.

the signature of the whole individual place. Once the robot is sure that a place will never be visited again, the corresponding dictionary can be simply deleted, or moved to other memory spaces. Although the environments used for this study are large in scale, they are not so large as to require the place-forgetting procedure. The technique for a larger environment, *i.e.* 100+ places is described here.

3.4 Application to Robotics Navigation

In robotic topological mapping, determining the number of topological nodes (how to partition image data into classes) is an important concern. Some previous works [61], [45], [40] ignored this problem by partitioning image samples manually into classes. Later, Valgren and Lilienthal [36] proposed incremental SC (ISC) so that the algorithm becomes fully incremental. Nevertheless, partitioning based only on appearance might yield too many nodes. For example, the work described in an earlier study [63] has 160 nodes (classes).

In this section, we describe a simple but effective method to use PIRF in topological mapping and localization: the obtained performance is substantially better than that of ISC. Our topological mapping is expected (i) to

be fully incremental, and (ii) to output a reasonable number of nodes (matched well to the environment). Using the concept of Spatial Semantic Hierarchy (SSH) described elsewhere in the literature [81], we simply add the junction detection module to the control layers of robot. This module takes an omnidirectional image as input, unwraps it into a panoramic image, and then classifies it as either a *junction* or *non-junction* image. This module would signify the upper recognition system to set the partition boundary if it judged the image as that of a junction. This guarantees that the number of nodes or places in the map matches well to the environment; it depends only on the number of detected junctions. For example, 580 training images of Suzukakedai Campus (see Figure 3.5(top)) are segmented into 23 places with 19 junctions. After the partitioned images are obtained, we can perform the recognition as in the previous section.

For this study, we implement a junction detection system resembling that described in an earlier study [80]: color histograms are used instead of Gaussian Mixture Models. The only difference is that our detection is of omnidirectional images. The assumptions resemble those of a prior study [80] where the lowest area of the image is the road; the upper part is the background. The road and off-road area pixels are sampled to create models of 30×30 histograms of red (R) and green (G) (because this setup yields the best result according to [80]) for representing the road and the background.

As depicted in Figure 3.5(top), junction detection from 580 omnidirectional images is possible (red circle). Particularly, the system samples the road and background pixels of each image and then performs binary classification (junction/non-junction). For all 580 images, most junctions were detected correctly with some small error contained (the system detects the junction a few images before or after the correct one). Precisely, four junctions were missed out of all 25 junctions. The image is first converted to a panoramic image and is segmented by the system. The segmentation would then be cropped at the part believed to contain all road vanishing points (we manually set the cropping size to 0.2–0.5 h, where h is the image height). Then, this cropped image is filtered using a Gaussian filter with sigma=1. The resulting image is averaged by each column to derive the vector which would be smoothed using a pseudo-Gaussian. The number of valid paths is determined by the number of peaks above the mean value. Because of limited space, we do not describe about the junction method in detail.

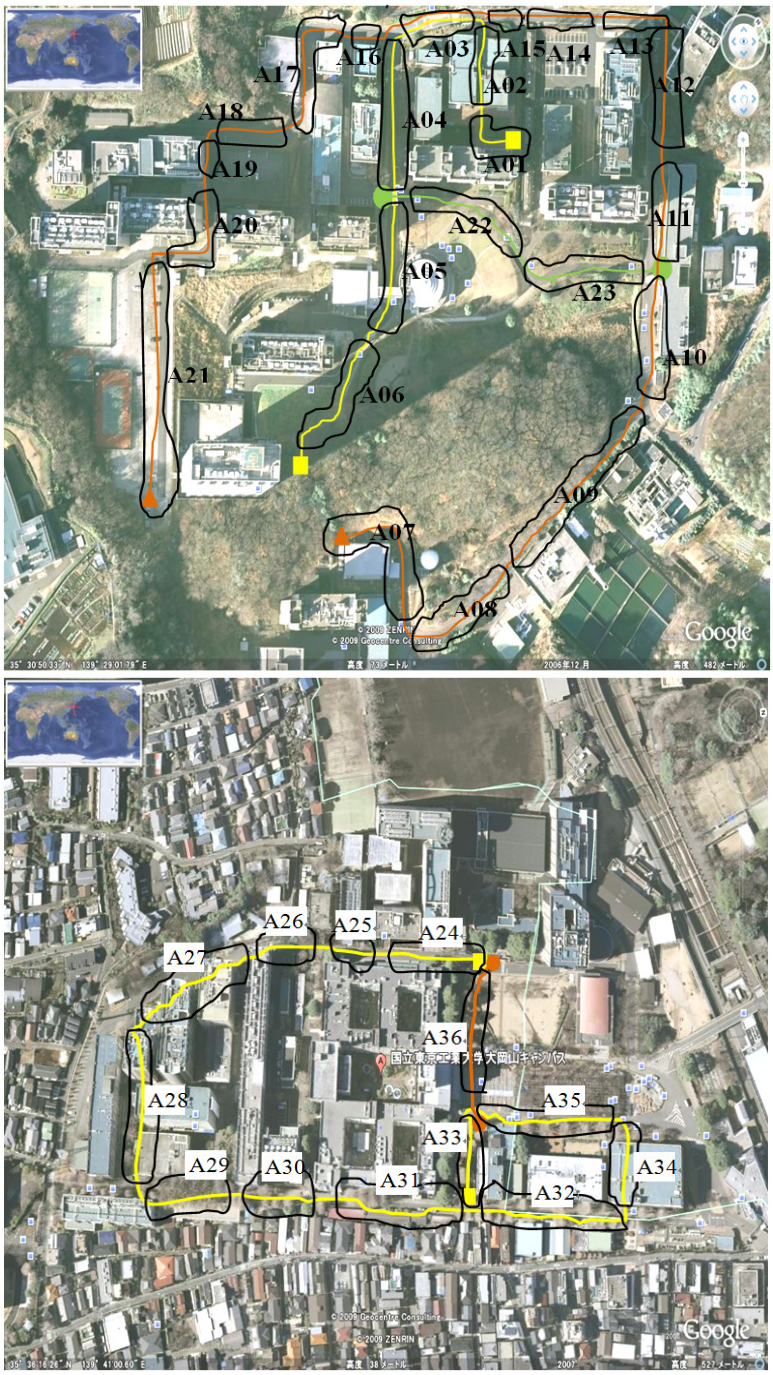


Figure 3.5 Map of the outdoor experiment sites. Twenty-three places manually segmented by hand from Suzukakedai campus (Top). An additional 13 places of Okayama campus were added (bottom). For Suzukakedai, train data were collected on holidays under clear weather during daytime, whereas test data were collected on weekdays under various conditions (i.e. cloudy, sunny, night). For Okayama, train and test data are collected randomly on weekdays under various conditions over 3 months.

Now that the junctions have been detected and all images have been partitioned into places, localization is performed. Robot localization might differ slightly from scene recognition; the recognition results of some images in the current place are obtainable before making a decision. Therefore, one step of the robot might contain n_t images, where t is a time step. For each step, the robot recognizes all n_t images and then summarizes the votes of the nearest place p^* with reliability score r^*. At this point, all n_t images would have been recognized. The PIRFs would have been extracted from these images and would have been collected to the new dictionary $\mathcal{D}^{new}$. If the score r^* is greater than threshold θ, then the current place recognized as p^* and $\mathcal{D}^{new}$ is neglected. Otherwise, the place is a new previously unseen place. A new dictionary $\mathcal{D}^{new}$ would be augmented to a set of dictionaries, $\mathbb{D}_{t+1} = \mathbb{D}_t \cup \{\mathcal{D}^{new}\} = \{\mathcal{D}^1, \dots, \mathcal{D}^N, \mathcal{D}^{new}\}$.

To examine the process in more detail, both the average value of confidence and the maximum rate of recognition for the nearest place are considered to calculate the reliability score r. Given $\mathbb{I}_t = \{I_1, \dots, I_{n_t}\}$ as the sequentially observed image of the current place ($n_t \geq 2$), with p_i and c_i respectively signifying the assigned place and confidence. The binary valued function as

$$\Delta_{ki} = \begin{cases} 1 & \text{if } p_i = k \\ 0 & \text{Otherwise} \end{cases} \tag{3.5}$$

and p^* as the nearest place class to which most input images have been assigned, where the following hold

$$p^* = p_j \text{ if}$$

$$\sum_{i=1}^{n_t} \Delta_{ij} = \max_{1 \leq k \leq N} \sum_{i=1}^{n_t} \Delta_{ki} \ . \tag{3.6}$$

The set of images $\mathbb{I}$ is recognized as place $p^* = p_j$ if

$$\mathrm{r}^* = r_j = \frac{1}{n_t}\left[\omega_1 \cdot \left(\sum_{i=1}^{m_t} \Delta_{ij}\right) + \omega_2 \cdot \left(\sum_{i=1}^{m_t} \Delta_{ij} c_i\right)\right] < \theta, \tag{3.7}$$

where θ is the threshold set by the user. For this study, we use θ=0.6, ω_1=0.6, and $\omega_2 = 0.4$ (the importance lays on the number of correct votes. Results show that recognizing new places as existing classes usually elicits low scores, whereas recognizing old places to the corresponding class gains a much higher score. However, PIRF-Nav requires at least two places (nodes) in the map as an initialization before starting the incremental process. While

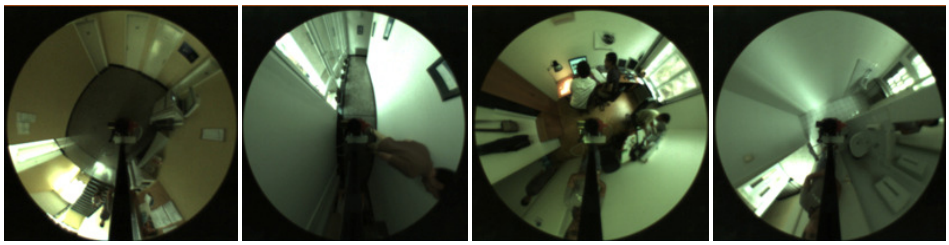
(a)

(b)

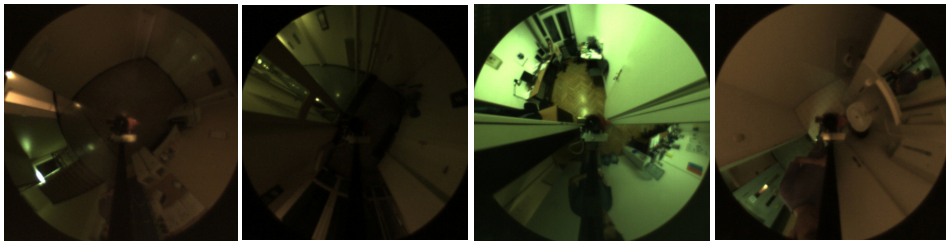
Figure 3.6 **(a)** Sample of images from place A21 (top) and A01 (bottom) of Suzukakedai. The training image was collected on a holiday (top), whereas the testing image was obtained on weekdays (bottom). **(b)** Some training images are taken in daytime (top), while the testing image was obtained in the evening (bottom). Both images are unwrapped merely for illustration.

executing, the system obtains new input images and makes a decision for each input image.

As described in previous section, the PIRF reduction can be done in an online manner. Instead of running a batch retest with a great number of re-used images, we incrementally update the score for every recognition step using the real testing images.



(a)



(b)

Figure 3.7 **(a)** Example of images of the Ljubljana lab taken by iRobot ATRV-Mini from the COLD database of [50]. From two available sub-datasets, we select the standard path database, which is taken from the *Printer Area*, *Corridor, A shared office*, and *a bathroom* (shown respectively from left to right). In the study described in this paper, we use two different set of sequences taken under cloudy, sunny, and nighttime conditions, constituting *ca.* 6000/6000 sequential images of 640 × 480 pixels for use in training/testing. **(b)** Samples images taken in nighttime from the same four places.

3.5 Experiments and Results

Four main experiments are done to prove the advantages of PIRF. Experiments 1, 2, and 3 examine scene recognition in both indoor and outdoor scenes. These experiments show that PIRF offers a markedly better rate of accuracy than other features for the task of highly dynamic outdoor scenes, while retaining good result for indoor. Experiment 4 is to show that the PIRF-based navigation system outperforms ISC in terms of time and accuracy.

Three image databases were used for this study: *Suzukakedai Campus*, *O-okayama Campus* and *COLD*. Cognitive Systems for the Cognitive Assistants Localization Database (COLD) dataset [50] was captured in a four-room office environment, including *a printer area, corridor, two-person office,* and *a bathroom*. Images were taken by a robot. The purpose of this dataset is to recognize which room the robot is in based on a single image. First regarding

the outdoor images datasets (Suzukakedai Campus and O-okayama Campus), we collected them by setting a tripod with height *ca.* 1.7 m. mounted with a camera (60D, DSLR; Nikon Corp.) with an omnidirectional lens. We walk along the road on campus while capturing omnidirectional images every few meters. The camera positions along the road are various (*i.e.*, the position must be moved to the footpath when the car is passing). The images are taken without concern about pedestrians or cars running past by. Some images contain a big blurred object, which actually is a running car. All images' original resolutions were 3872×2592, but they were scaled down to 640×428 for use in all experiments. For Suzukakedai Campus, most training data were collected on *holidays* under clear weather, whereas the testing data are collected on *weekdays* under various weather conditions, resulting in 580 images for training, and 489 images for testing. All images were collected according to all three routes portrayed in Figure 3.5(top). For O-okayama Campus, we collected more images from places A24-A36 in respect to the path portrayed in Figure 3.5 (bottom). For this campus, people crowded in the images taken on both holidays and weekdays, so all data were taken on weekdays at various times and weather conditions. Data were collected during 3 months, resulting in 450 images in all for training, and 493 images for testing. Figure 3.6 and Figure 3.7 portray differences between training images and testing images. All experiments were written and run using software (Matlab 7.6.0.; The MathWorks, Inc.).

3.5.1 Experiment 1: Recognizing Outdoor Scenes

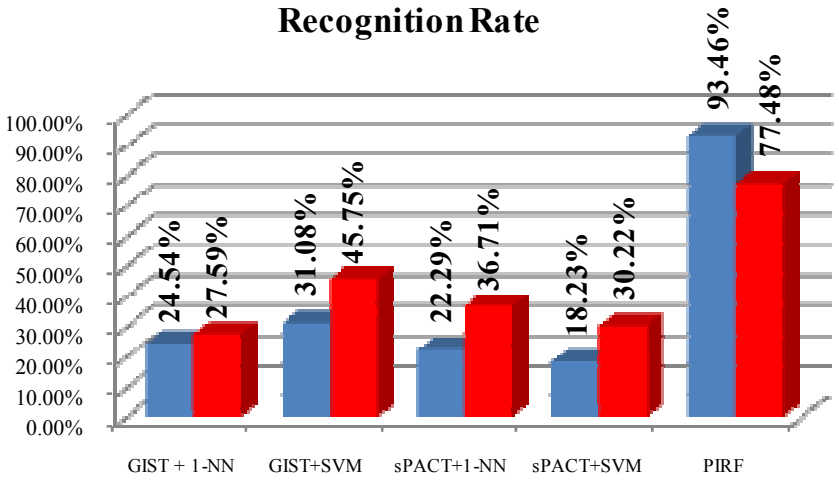
This experiment is further divided into two sub-experiments conducted respectively at Suzukakedai campus and Okayama campus. At Suzukakedai, 580 omnidirectional images were input to the system for training. The testing is done over 489 testing images. The training images are labeled by hand according to the junction detection result obtained in experiment 4, resulting in 23 separate places (A01–A23). We do this because, in the discussion related to experiment 4, we can directly use the accuracy of this experiment as the accuracy of PIRF for comparison to the ISC method in experiment 4. The difficulty of these datasets is the great difference between training and testing data. Changes occurring between images taken on holidays and weekdays are considerable, as shown in the sample image of Figure 3.6 (a).

For O-okayama, the dataset is a bit different from Suzukakedai. O-okayama images were not collected on different holidays and weekdays because this main campus is always crowded on both holidays and weekdays. Data were collected during 3 months to attempt recognition despite changes occurring over a long period of time.

Two baselines were used for comparison. The first one (i) is the 80-D Gist vectors used in the work of Torralba *et al.* [61]. With six orientations of steerable pyramid and four scales applied to the monochrome image, 580 Gist vectors were derived from 580 training images. However, we do not use the HMM as in [61] because the transition matrix of labeled sequence data is not available. Therefore, we try to use First Nearest Neighbor (1-NN) and Support Vector Machines (SVM) as the classification framework for Gist. We also tried 3-NN and 5-NN, but the results are mostly equivalent to those of 1-NN. For the second baseline (ii), the spatial Principal component Analysis on Census Transform (sPACT) proposed by [40] is our choice because of its recently highest result for indoor scene recognition over the IDOL database of [45]. The training images are first converted by the census transformed (CT) image; then the CT histograms are created. Principle Component Analysis (PCA) is then performed on the CT histograms to extract the most important components among the distribution of CT histograms. In this study, we also apply the level 2 spatial pyramid, as done in [45]. As hinted in [45], choosing the right classifier for a specific application is important. Consequently, the classifiers used with sPACT are both NN and SVM, in the same way as that done for the first baseline.

Results are presented in figure 3.8 (a). Actually, PIRF-based recognition yields about a two times higher rate than the others both for Suzukakedai and O-okayama. It is noteworthy that the recognition rate is considerably lower, at only 77.48%. We suspect that this occurs because Okayama campus is a main campus crowded with many people. Most parts of campus are not wide open compared to the Suzukakedai campus. The campus contains crowded buildings so that the problem of perceptual aliasing occurs. The Okayama campus contains many artificial structures that make it similar to indoor areas with highly dynamical changes. Suzukakedai contains more natural distant objects than Okayama, thus having a higher rate.

We also report the confusion matrix of Suzukakedai in figure 3.8 (b): errors are, in general, not distributed uniformly. Taking a look into the place A21



(a)

Place	A01	A02	A03	A04	A05	A06	A07	A08	A09	A10	A11	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21	A22	A23
A01	11																						
A02		17																					
A03			23																				
A04				15																			
A05					14																		
A06						39																	
A07							22														2		
A08						1		22															
A09									17												6		
A10										8											1		
A11											24	1											
A12												15									2		
A13						1							14					1			2		
A14														21				1			1		
A15					1										14								
A16			2													10							
A17																	29						
A18			1		1													17			1		
A19																			8		2		
A20			1																	16	1		
A21																					36		
A22																						24	
A23																							31

(b)

Figure 3.8 Recognition Results **(a)** The overall performance of PIRF is shown in comparison with other methods in term of accuracy (Red->O-okayama, Blue->Suzukakedai). **(b)** The confusion matrix of recognition results of 489 images of Suzukakedai (Row-> corrected classes, Column-> predicted classes).

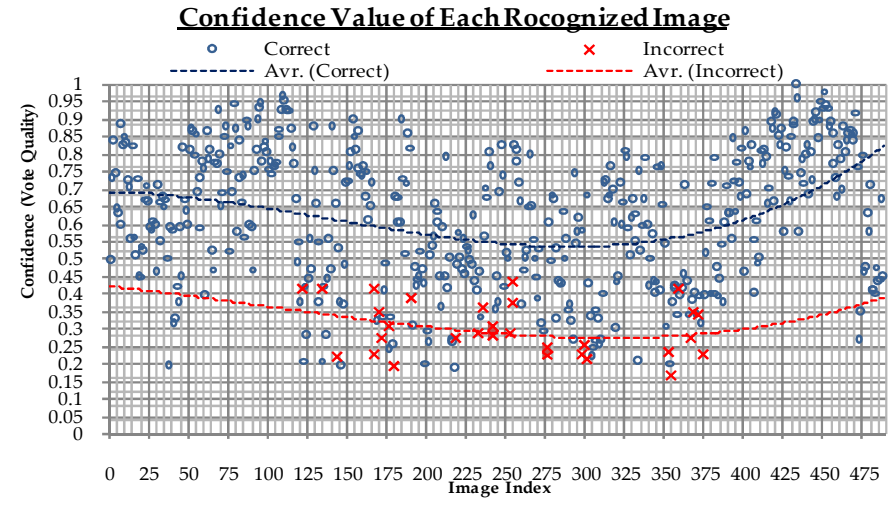
(which confuses many places), for instance, we found that the place is a large open-wide area (sample of image from A21 is portrayed in figure 3.6 (a) where many distant objects are shared with other places). Although A21 shared some distant objects with many places, place recognition is still efficient because votes on other objects are useful for determination.

We believe that some reasons make the PIRF-based recognizer outperform

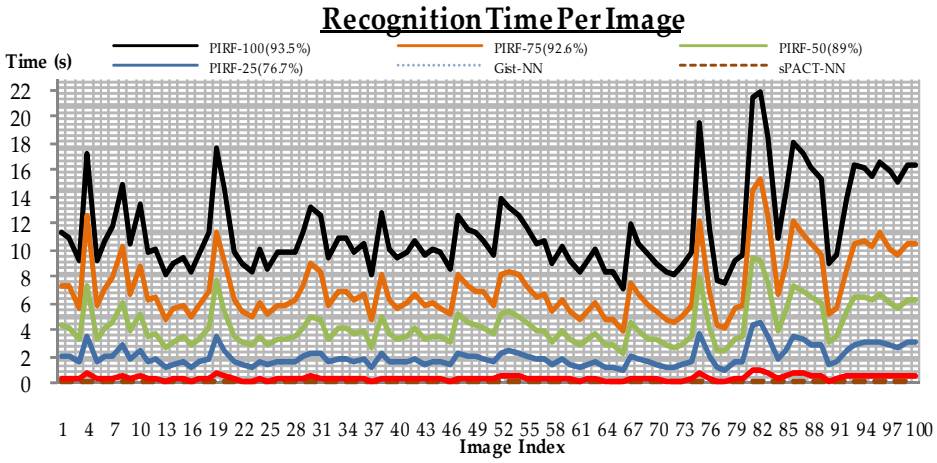
others in this experiment. Distant objects, which are robust to positional changes, usually appear smaller than nearby objects. Consequently, global features that capture a whole scene, such as Gist, include many unstable objects, *e.g.*, cars, doors, a gate, people. The sPACT provides a lower rate of recognition than our PIRF because of its basic nature of feature extraction; sPACT converts the whole image into the Census Transformed. Although its performance is recently considered the highest for the IDOL database, its accuracy is lower when tested on our highly dynamic outdoor scenes. Although sPACT is a local descriptor with greater descriptive power than Gist features, sPACT still includes many dynamic objects of the scenes. Being sensitive to changes in camera positions, nearby objects can strongly affect the recognition system. Note that PIRF requires sequentiality only for feature extraction. It does not take advantage of sequentiality in recognition process.

Distinctive from the others, PIRF assigns emphasis to distant objects while neglecting most nearby objects. The underlying concept of PIRF-recognition differs from those methods that perform segmentation (*i.e.*, normalized cut) before feature extraction. In fact, a PIRF can be extracted without going offline for image segmentation. Unlike the BoW approach, PIRF does not quantize the descriptors; it can therefore preserve the distinctiveness of original local descriptors. Consequently, PIRF can mostly overcome the problem of highly dynamic changes because almost all unstable closed objects are ignored (see Fig. 3.13 for sample-matched PIRF in testing images).

The confidence values in figure 3.9 (a) underscore the quality of the recognition provided by PIRF. Almost correct recognitions display high confidence values, which can be interpreted as the quality of vote in the same sense of [64]. This fact proves that PIRF is sufficiently discriminative for place identification; the right dictionary only matches to the right place. Consider figure 3.6 (a), for example, with two different scenes taken on a holiday and weekday. While training on holiday images (top), PIRF captures the distant building because its appearance is robust to position changes. Therefore, although the testing image might be changed dynamically and might share some distant objects with other places, the number of votes on distant buildings would be adequately higher than other votes. However, it should be noted that, during image index 262–301 (see figure 3.9 (a)), the confidence values are quite low because place A09 is mostly covered by trees along both sides of the path. These trees block the view of distant buildings. Therefore,



(a)



(b)

Figure 3.9 **(a)** The confidence value of each image recognized by our system. The average lines for correct and wrong recognition are estimated using 3rd order polynomials. This confidence value is similar the quality of vote in the work of [64]. The calculation method is the same. **(b)** The recognition times per image are shown. PIRF-*n* denotes PIRFs reduced to *n*%. The PIRF-50(Parallel) is the PIRF which has been reduced by 50% and each dictionary vote for the matched PIRF in parallel. A comparison is made with Gist and sPACT.

distant objects are insufficient for justification. Nevertheless, the recognition rate remains good because some nearer objects, which are apparently stable for sub-places, have been used instead.

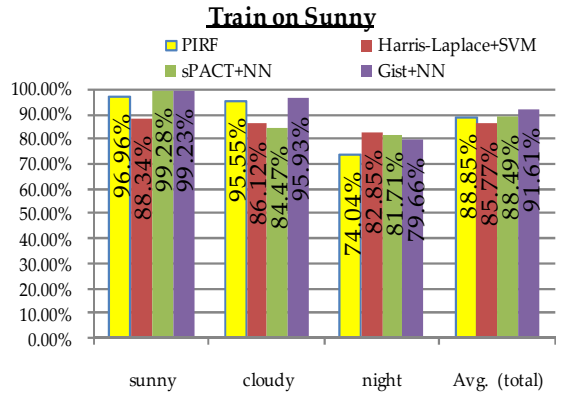
With respect to the learning time used for model training and feature extraction, PIRF is faster than sPACT or Gist. For every image in the

Suzukakedai Campus, the average times for creating the CT histogram, Gist and PIRF are, respectively, 29.2931 s, 4.8219 s, and 3.2312 s. Based on this result, PIRF and Gist are suitable for learning in real-time applications. A comparison of two images can be done quickly during the robot's exploration. In terms of recognition time (per image), PIRF is clearly slower than Gist or sPACT because each encodes an image into only one feature vector. In fact, PIRF trades off the recognition time for better accuracy. However, the recognition time of PIRF-based method can be reduced further using the reduction technique described in section 3.3.4 to reduce the number of PIRFs and slow the PIRF growth rate. Figure 3.9 (b) presents the accuracy values obtained for different number of PIRFs. Even with 50% reduction of PIRF, the accuracy remains higher than the other baselines. It is particularly interesting that parallel votes (each dictionary votes simultaneously) can reduce the time to less than a second per image (Red Line in figure 3.9 (b)). Although that reduced time is still longer than that of Gist, it might be acceptable for robotic navigation, for which the image capture rate must correspond to the robot's motions. Experiment 4 will show that PIRF-Nav executes more quickly than ISC, although its recognition time is longer than that of either sPACT or Gist.

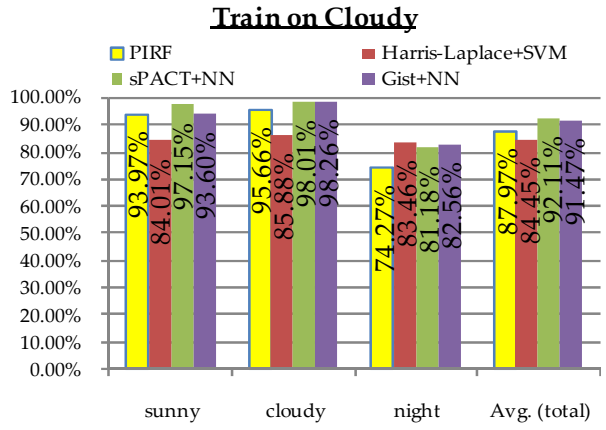
3.5.2 Experiment 2: Recognizing Indoor Scenes

In the second experiment, the PIRF are tested on COsy Localization Database (COLD) of [50]. The COLD database includes data collected by three robots. In this work, we select the data collected from *Ljubljana* laboratory (we designate the experiment using this database as "Ljubljana" hereinafter). We select two sequences of each weather condition: cloudy 1 & 2, sunny 1 & 2, and night 1 & 2, in all six sequences (*ca.* 2000 images for each sequence). Training is done on one sequence and the testing is done on the rest. We repeat the experiments six times (all combinations) and obtain averaged results as portrayed in figure 3.10. Although we clearly claimed that PIRF is especially suitable for outdoor scenes, it is also very common for any long-term system to recognize both outdoor and indoor scenes. To prove that PIRF can also be used indoors efficiently, we conduct this experiment. Furthermore, this experiment proves that PIRF is applicable to images collected by a real robot at various times and in weather conditions (figure 3.7(a-b) portrays the sample images of COLD taken in sunny and at night. The baselines used in this

(a)



(b)



(c)

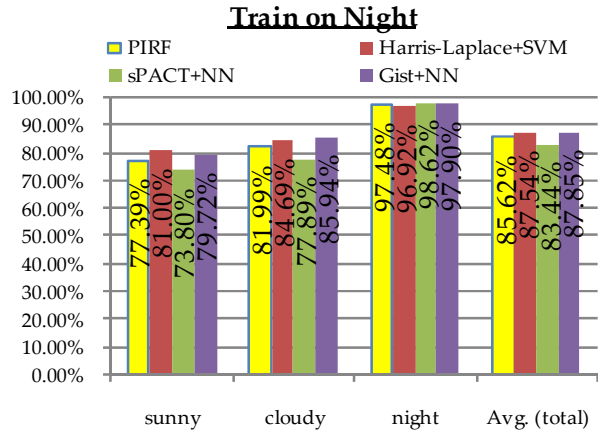


Figure 3.10 Averaged recognition results with COLD-Ljublajana standard sequences. Training is done on one sequence and testing on the other sequences. (a) Train on sunny. (b) Train on cloudy. (c) Train on night. The comparisons are done to sPACT of [40] using NN as classifier, to Gist of [61] using NN as classifier, and to results of Harris-Laplace with SVM [50].

experiment are the same as those used in experiment 1: sPACT and Gist, and

the referred result of [50].

Although Gist and sPACT yield the highest accuracy for this indoor database, PIRF also works well. Especially during daytime (trained or tested with either sunny or cloudy), PIRF offers a high rate of about 93%–94%. In daytime, the scene is clear and a sufficient number of extracted SIFTs are used for PIRF generation. With more PIRFs, the vote quality is high: it can recognize either sunny or cloudy scenes correctly. When testing at night, although the number of PIRFs is sufficient for recognition, the number of SIFT extracted from a testing image is insufficient for representing an image. It is also true that darkness can reduce the number of SIFTs. Consequently, the quality of votes for the nearest place is low. It is also noteworthy that the unbalanced number of sample images does not affect the performance of PIRF-based systems. In this experiment, a corridor was traversed many times by the robot, gathering *ca.* 1500 sequential images. The PIRF-dictionary of corridor is also the biggest. However, the confusion matrix in figure 3.11 shows that the different dictionary size does not engender biased recognition.

3.5.3 Combined Sites

In the experiment, we combine all experiment sites together: Suzukakedai, Okayama, and Ljubljana sunny 1 & cloudy 1 sequences. Testing images from outdoor scenes are the same as those used in experiment 1, although the testing images for Ljubljana are those of cloudy 1 (trained by sunny1). This experiment shows that PIRF-dictionaries are sufficiently distinctive even for recognizing a larger environment. The efficiency is unaffected by the imbalance of training samples in places with different sizes (*i.e.*, the corridor is longer than the bathroom). The results are shown as a confusion matrix in figure 3.11.

Torralba *et al.* [21] navigated the robot indoors and outdoors. We do the same by combining the PIRF-dictionaries of Suzukakedai, Okayama, and indoor Ljubljana. The testing images are the same set used previously. The results are presented in figure 3.10. Results show that the accuracies of PIRF are approximately equal––2009/2112 (95.12%), 423/489 (86.50%), and 375/493 (76.06%)–– respectively, for Ljubljana, Suzukakedai and O-okayama. The PIRF-dictionary is not much affected by the imbalanced image samples in each place.

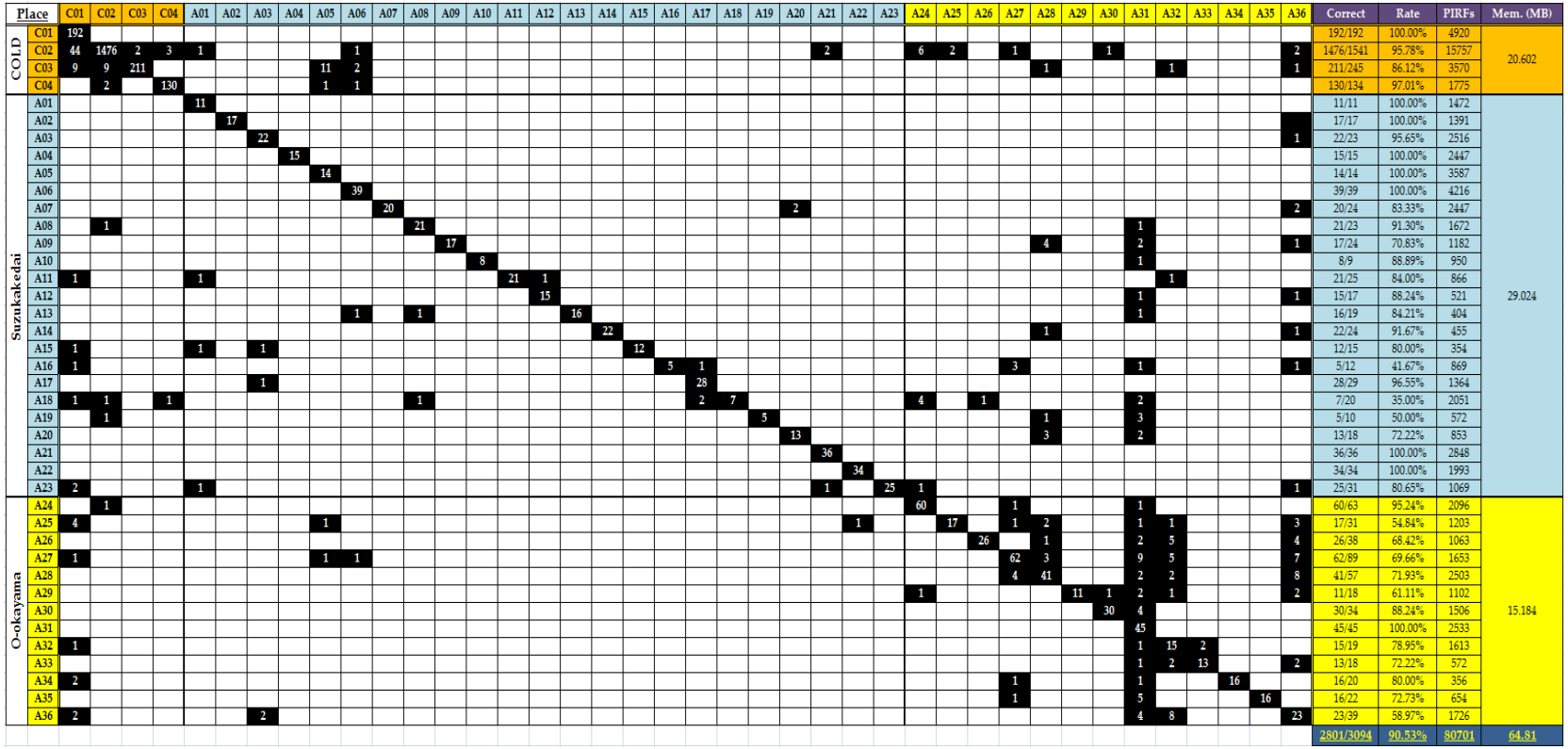


Figure 3.11 Results of combined sites. The matrix shows confusion among 40 dictionaries of 40 places (Row-> corrected classes, Column-> predicted classes). The number of PIRFs for each place and the memory required are shown.

Based on the obtained results, we conclude that two main factors affect the PIRF's efficiency. **(i)** The characteristics of the place. Places with numerous objects blocking the distant view of the camera inject bad PIRFs into the dictionary. With only a few distant objects, the PIRF must capture some nearby objects instead. In addition, in cloudy weather, sometimes a distant view in an image is too bright (this is the problem of photography in which the illumination condition the space between the distant view and the camera position is too different). In this case, only a very few SIFTs would be extracted from distant objects. **(ii)** The size of the place. A few image samples can cause failure of PIRF extraction in the sense that only a few wide-baseline features were found. For example, A16 and A19 obtained a low rate of accuracy because they are much smaller than other places, whereas A18 obtains low accuracy because its high slope blocks most of the distant views. It must be clarified that the imbalance of sample images of places does not affect the recognition as long as the PIRFs in the dictionary are sufficiently distinctive; this depends directly on the characteristics of the place. For example, 1104 PIRFs are sufficient for representing A01. The numerous PIRFs of C02 (15757 PIRFs) cannot confuse A01, although only 2533 PIRFs of A31 confuses many places. Examining place A31 (which confuses many places) for instance, we found that the places are also open-wide areas where many distant objects are shared with other places. Several buildings are visible in this place. It is important to note that PIRF suits a wide-open area: wide-open areas (*i.e.* A31, A21) themselves obtain a very high rate of recognition (100%), but they can also confuse other areas. This might be resolved simply by re-examining the confusion matrix and deleting some PIRFs that are confusing. Nevertheless, overall, the results show that the PIRF-dictionary is sufficiently distinctive to offer a better recognition rate than other features.

3.5.4 Experiment 4: Incremental Topological Mapping

In this experiment, we show that PIRF is useful to solve appearance-based topological mapping in an incremental manner like that of ISC, but with less computation time. The baseline used in this experiment is the incremental spectral clustering (ISC) of [36], [63]. We let both PIRF-based navigation (PIRF-Nav) and ISC create the map incrementally. The loops have been closed with neither false negatives nor false positives, although the junction detection

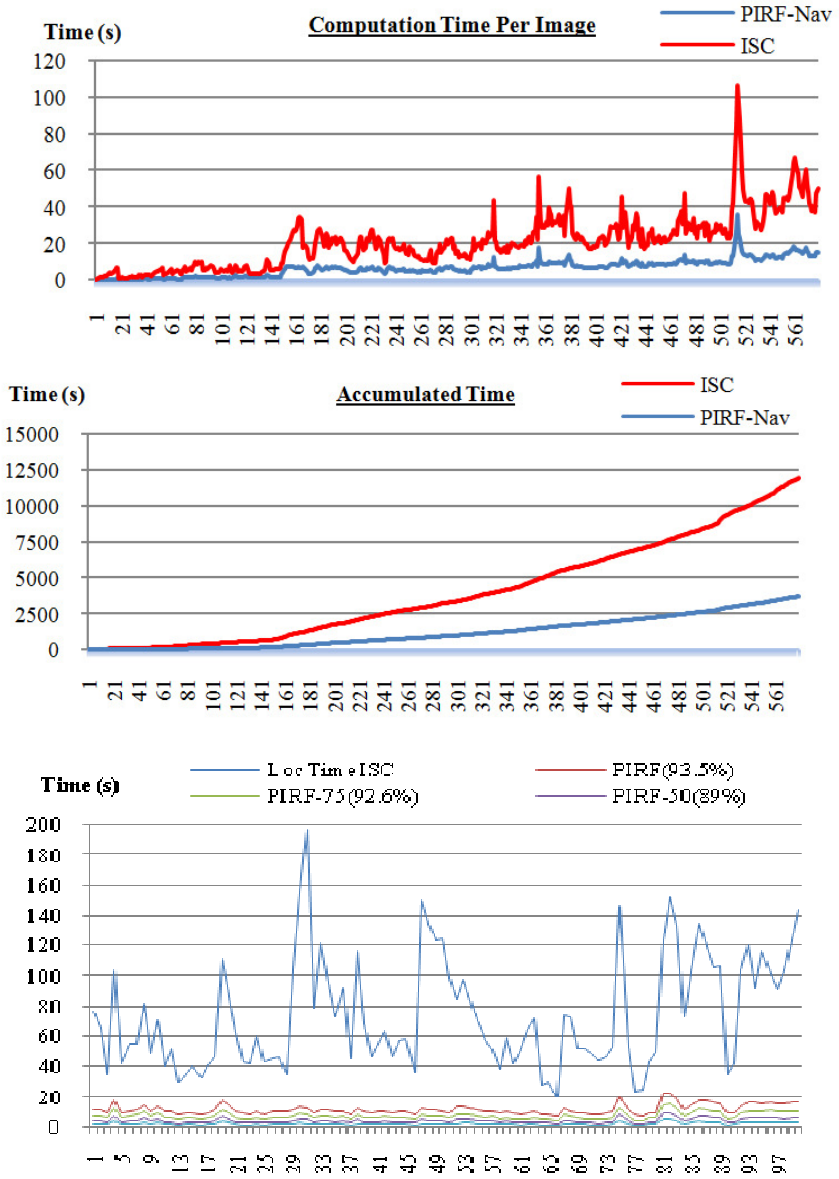


Figure 3.12 Comparison of the computation time (top) and accumulated computation time excluding feature extraction for ISC and PIRF-Nav (middle). (Bottom) Computation time for each recognition and the corresponding accuracy. Times are shown for the first 100 testing images. PIRF-75, -50, and -25 have 25%, 50% and 75% respective size reduction. (Programmed by Matlab)

missed 4 junctions from among all 25 junctions (16.0% false negative). A comparison of time between ISC and PIRF-Nav is presented in figure 3.12.

In terms of mapping time, PIRF-Nav builds the map in *ca.* 30% of the usual

time and tends to do so faster in larger environments (figure 3.12 (top & middle -row)). The ISC uses much more time because of its necessary affinity matrix generation. Precisely, ISC requires 167910 comparisons (46888.28 s), ISC requires 40757 comparisons (11996.80 s), and PIRF requires 579 comparisons (3723.23s). In terms of recognition time, PIRF-Nav also recognizes a single-image rapidly (figure 3.12 (bottom)). The ISC clusters the map into 159 nodes. Therefore, the minimum number of comparisons necessary for each recognition is 159. We further reduce the number of PIRFs to reduce the localization time. We set the reduction rate to 25%, 50%, and 75% to reduce the PIRFs, and classify the testing images again. The result presented in figure 3.12 (bottom) respectively shows that the 75% and 50% reduced PIRFs (PIRF-75 and PIRF-50) still yield the same accuracy despite reducing the localization time by *ca.* 25% and 75%.

Regarding accuracy, we implement the standard spectral clustering (SC) as our baseline instead of ISC because SC yields a better rate of classification if k is appropriated. Therefore, we ran SC for many different k and found that k = 43 offers the highest accuracy. To classify the images by SC without its associated position data, we simply label 489 testing images with respect to figure 3.5 (a). For example, we consider that SC clusters training images no. 1–10 as cluster 1, and images no. 11–25 as cluster 2. Consequently, the testing images taken for area A01 according to figure 3.5 (a) are expected to match the images in either cluster 1 or 2. As such, classification of the image from A01 by SC would be considered correct if the nearest cluster is 1 or 2. As a result, SC offers 40.29%, while PIRFs offers 93.46%, two times higher than SC.

Both SC and ISC represent places with a set of reference images. In a highly dynamic environment, training images taken on holidays appear very different from testing images taken on weekdays. This can cause failure of the recognition. For example, A03 depicts the main road at the entrance of the campus; A21 shows the parking areas. These two places look very different between peak and off-peak times. A comparison between SC and PIRF might imply that several raw SIFTs cannot recognize the highly dynamic scenes. By SC, a set of reference images are retained for matching in the recognition process. The matching is done by local feature matching. If a major portion of an image is changed, then major SIFTs of training images might be unable to match those SIFTs in the testing image.

Some may wonder why ISC has been selected as our baseline for this

Figure 3.13 The sample images described by PIRF. The images are taken from A28, A21, and C03 respectively. For outdoor scenes, the PIRF tries to describe the distant object such as building. For indoor, it captures the most stable objects such as walls.

experiment. Because PIRF is intentionally proposed to deal with the dynamical changes in the scenes, we select the incremental mapping method which has been reported to succeed in handling the dynamical changes across seasons of [63]. The state-of-the-art approaches like [35] did not especially address this problem.

3.6 Chapter Summary

In this chapter, the results show that PIRF recognizes large sets of both indoor and outdoor images without the help of supervised learning tools such as SVM or HMM. Precisely capturing the points of interest from distant objects, the number of local descriptors can be markedly reduced while preserving their discriminative power.

Regarding accuracy, PIRF clearly outperforms other features in outdoor scenes; it does well even in an indoor environment where distant objects are not ubiquitous. Instead of natural distant objects, PIRF captures nearby objects with a stable appearance. Figure 3.13 presents samples of testing images with PIRFs matched to the correct dictionaries for both indoors and

outdoors. In figure 3.13 (Bottom) PIRF captures the decorations on the wall instead of distant objects.

One concern related to long-term scene recognition is that the number of samples for each place is unbalanced because some places take less time to walk through, but others can take much more time (*i.e.*, the "corridor" dictionary from Ljubljana contains more than 10,000+ PIRFs, whereas the dictionary of "bathroom" contains only *ca.* 2000 PIRFs). Regarding this concern, thanks to the discriminative power of SIFT, all 40 combined dictionaries of PIRFs are sufficiently distinctive to support recognition.

Another remarkable advantage of PIRF is the reduction rate of memory. Because PIRFs are sufficient to represent the place, the reference images are no longer needed. Most previous approaches work with a database of reference images [37], [60] for which the size depends heavily on the area size. As depicted in Fig. 3.11, the memory necessary to store all 3091 images (for combined sites) is *ca.* 991 MB, whereas the memory required for storing PIRFs is 57.65 MB (71669 PIRFs, no reduction). Therefore, using PIRF reduces the necessary memory size by *ca.* 94%.

The junction detection module, which is used to partition data into classes instead of clustering, is not the main emphasis of this study. This problem might be regarded as a problem of robot perception. A robot who cannot detect the junction would be like a man who forgets to notice an intersection. Such a junction can be treated simply as a normal straight path. Moreover, without a junction detection system, PIRF can be used simply with preliminary partitioned data like those described in earlier studies [45], [78], [40].

A profound effect of using PIRF is the utilization of stable distant object information. However, PIRF has some limitations and limited future research directions for improvement. One disadvantage of the current PIRF implementation is that (i) it strongly relies on the efficiency of the local descriptors. SIFT is a highly discriminative local descriptors. Therefore, the extracted PIRF can capture distinctive features from objects precisely. On the other hand, this disadvantage makes PIRF flexible for use with other local descriptors, *i.e.* speeded up robust feature (SURF). Second, (ii) PIRF requires input images as the sequences. Although we have claimed that PIRF can solve the kidnapped robot problem (appearance-based localization), it requires that the length of image sequence be sufficient for PIRF extraction (*i.e.* three images). In other words, PIRF is currently limited to the recognition problem;

its descriptive power is too great to be used in the problem of categorization or understanding. Third, in this paper, we use PIRF in a simple manner to recognize scenes. (iii) Collecting numerous PIRFs from many places might finally produce a problem of duplicated features. In addition to our PIRF reduction, vector quantization might be another good choice. Because PIRF is a distinctive feature in a highly dynamic environment, bag-of-PIRF might be a good solution for highly dynamic environments. This idea does not contradict to what we suggested earlier that PIRF is efficient because it omits the clustering process. Clustering those features of selected distant objects should preserve more salient information than clustering all features included in the scene. Finally, although the time in PIRF extraction is faster than that of either Gist or sPACT, its recognition time is slower (although it is faster than ISC). In this study, one place required about 500–1000 PIRFs for representation. Room exists for improvement here; one might be able to compress these PIRFs further to speed up the recognition time.

The PIRF features, despite their simple implementation, can achieve promising localization performance, especially in terms of computation time. Comparing the current ISC method [63], PIRF is useful to build the topological map incrementally in considerably less time. Although the current PIRF still requires more than a single image for PIRF extraction, the current result appears promising for future improvement. The localization time (single-image classification) is also shortened considerably because the dictionaries are sufficient for place recognition instead of databases of reference images. Importantly, we do not claim that PIRF-Nav is the most suitable navigation approach for PIRF. We merely describe a simple navigation approach to demonstrate that PIRF is useful for the robotic development community. The standard local descriptors used by the BoW approach [35], [65] do not perform well in highly dynamic scenes. We believe that one might create more efficient robot navigation by considering PIRF, *i.e.* BoW created from three feature spaces, one of which is PIRF.

CHAPTER 4

PIRF-BASED ONLINE AND INCREMENTAL LOCALIZATION AND MAPPING

This chapter introduces a new method for online and incremental localization and mapping base on only appearance data. Appearance-based localization and mapping have recently become hot topics of discussion in robotics because of the difficulty of detecting loop closure in metric SLAM [82], [83]. Advanced computing technologies and the low prices of cameras have helped to supplement traditional metric SLAM methods with appearance-based visual information. Consequently, commonly used sensors such as laser scanners, radars, and sonar tend to be associated with, or replaced by, a mono or stereo camera. With the popularity of this topic, numerous approaches have been reported for fast and accurate localization and mapping. However, there are still some important problems to overcome. Perceptual aliasing and dynamic changes in the systems are important concerns for localization and mapping in robotics. At 37%, the recall rate of localization with 100% precision for localization in City Centre is considerably low [35] even with the current Bag-of-Words (BoW) based methods [84], [85], [64], [86], [87] that render real-time and online performance.

The position-invariant robust feature based navigation system (PIRF-Nav) was designed to perform localization and mapping in exactly the same manner as the current state-of-the-art method FAB-MAP, but with a substantially higher recall rate with 100% precision. This high performance is achieved by making use of the design of our previously proposed PIRF [62]. PIRF extraction is simple and fast. It is generated by averaging the existing

scale-invariant feature transformation (SIFTs) [41], which appear to be "slow moving" relative to the change in camera positions. These slow-moving features are identified using simple feature matching: a local feature that appears repeatedly in many sequential images would be regarded as a slow-moving local feature. Extracting PIRFs in this manner also enables landmark filtering. Unlike BoW-based approaches that encode an entire image into a set of visual words, PIRFs selectively capture only objects that are likely to exist in a place permanently. This ability renders PIRFs robust against dynamic changes in scenes. Representing each image with a set of PIRF features instead of SIFT features enables a compact representation that is robust against noise, while preserving the distinctive power of unique local features. Loop-closure is detected using PIRF-Nav by taking into account similar PIRF-based scores among observations, and then pairwise similarity among all observations are computed to yield a square similarity matrix, similar to that obtained in some previous studies [88], [89], [90], [84]. The similarity scoring of PIRF-Nav is modeled on the concept of *term frequency–inverted document frequency (tf-idf)* and the discrete Bayes' filtering scheme. Only a few PIRFs are necessary for image representation (about 50–100 PIRFs per image; about 50,000–100,000 features for 1000 images). Therefore, PIRF-Nav is sufficiently fast for real time. We demonstrate the PIRF-Nav performance by testing it on three image datasets. The first dataset, New College [35], has localization under a strong perceptual aliasing condition. The second dataset, City Centre [35], has localization under dynamic changes of the scene in the center of the city. The last dataset that consists of 1079 omnidirectional images, collected by us on different days, presents the task of localization under "highly" dynamic changes. By the term "highly dynamic," we mean the unusual dynamic changes that occur occasionally because of some special event (e.g., open-campus event). The dataset also includes illumination variance by collecting data in different weather conditions (i.e., sunny, cloudy, etc.). These conditions were not addressed in the earlier studies [35]. Finally, all three datasets are combined to illustrate long-run performance of PIRF-Nav. We achieve 3000+ image localizations in an online incremental manner without offline dictionary generation.

4.1 Related Works

Appearance-based localization and mapping are popular, and there are several approaches to them. Initially, many methods represented appearance using global features, where a single descriptor is computed for an entire image [60], [91], [61], [92]. Most such approaches using global features require much effort in the supervised training phase to learn the generative place models. Later, Bowling et al. [93] described an unsupervised approach that uses a sophisticated dimensionality reduction technique, but the method yields localization results in a subjective form that are inconvenient to interpret and still calls for high computational cost. Furthermore, the use of global features is not robust to some effects such as varied lighting, perspective change, and dynamic objects that alter portions of a scene between visits.

Work in computer vision has engendered development of local features such as scale, rotation, and some lighting changes that are robust to transformations. Many recent appearance-based localization methods therefore use SIFTs [41] or speeded up robust features (SURFs) [42] as the main visual feature for the localization and mapping system. Wolf et al. [94] used an image retrieval system based on invariant features as the basis of a Monte Carlo localization scheme. Kosecka et al. [95] represented a model of an environment by a set of locations and spatial relations among the locations. Each location is represented as a set of views and their associated local SIFTs. Their method has a salient disadvantage in computation time. Localization based on matching numerous features consumes too much time for use in real-time systems. Consequently, the idea of a visual vocabulary from computer vision community [96], [97] built upon local invariant features was applied widely to address this problem. The visual vocabulary model treats an image as a *bag-of-words* (BoW) much like a text document, where a "word" corresponds to a region in the space of invariant descriptors. One of the main advantages of using BoW is that it enables rapid visual search through the application of methods developed for text retrieval. Wang et al. [85] use geometric information in a post-verification step to confirm putative matches. Schindler et al. [87] described how to improve the visual vocabulary generation to yield more discriminative visual words, and discussed the application of a technique for city-scale localization with a database of 10

million images.

Nevertheless, most studies till date address only the problem of "localization." It is the problem of recognizing an input image to an already mapped location, where the map is known *a priori*. This guarantees that the image has come from somewhere within the map. This limitation renders these methods unsuitable for real implementation in SLAM. For using an appearance-based approach to detect loop closure in SLAM, the system must cope with the possibility that the current view comes from a previously unvisited place, and therefore has no match within the map. Chen and Wang [98] solved this in a topological framework, but their solution to the perceptual aliasing problem is unsatisfactory (two different places appear to be similar). Goedeme et al. [37] described an approach to this issue using Dempster–Shafer theory with sequences of observations to confirm or reject putative loop closures. Because localization and mapping were done separately, this approach was unsuitable for online unsupervised mobile robot applications. More recently, Cummins and Newman [35] proposed a probabilistic framework for online visual SLAM. The method, FAB-MAP, considers the correlation among visual words using Chow–Liu trees. This approach, based on the BoW scheme, is also robust against perceptual aliasing: a generative model of appearance is learned offline by approximating the occurrence probabilities of the words included in the offline-built dictionary. The main asset of this model is its capability of evaluating the distinctiveness of each word, thereby accounting for perceptual aliasing at the word level; its principal disadvantage lies in the offline process needed for model learning and dictionary computation. These disadvantages become more serious when there is a remarkable change in the environment (e.g., when a human wants to use the robot in different countries). A good dictionary of one environment is not useful in a different environment. Our results would later show that FAB-MAP performance decreases when the environments differ considerably.

Most recently, Angeli et al. [65] described the incremental creation of a visual vocabulary. The method is still based on the popular BoW scheme, as explained in an earlier study [35]. The system can start with an empty dictionary. The model requires around 40,000 visual words for 500+ images, which is much more than FAB-MAP. The authors trade off the increased number of words with the ability to generate a dictionary online. The

performance in terms of accuracy of the recall rate of loop-closure detection, as stated by the authors at the end of their earlier paper [65], is less than or approximately similar to that of FAB-MAP. In light of these reasons, we selected FAB-MAP as our baseline for evaluating PIRF-Nav in terms of accuracy.

In order to design a method for selecting the most salient visual feature, Li and Kosecka [99] reduced a set of SIFTs by first estimating the posterior probability of the feature. Then they calculated the information entropy of each feature. This information is used to select the most useful feature. However, this method requires a set of training images. The selection process must be performed offline. Moreover, this method is limited to localization, where an input image is guaranteed to be from some previously visited location. Nevertheless, a PIRF can be extracted in an incremental manner. Its performance is beyond localization and is therefore applicable to the incremental online loop-closure detection. Se et al. [100] also described a similar underlying concept. Their system, much like our basic idea, shows that the matched SIFTs between different frames offer a good salient feature for tracking. However, there are some significant differences between their method and ours. First, the PIRF-Nav is proposed for a large-scale outdoor scene where the matching can be done between images taken at about ~10 m. different observer's positions. PIRF-Nav also takes into account the problem of moving objects; it can localize the place even though the major part of scene has been changed (see figure 4.16). Furthermore, using PIRF-Nav, the odometry data are not required for obtaining the coordinate of each feature.

The proposed PIRF-Nav can achieve a high rate of recall (approximately twice that of FAB-MAP) with 100% precision under identical conditions as FAB-MAP, while requiring no offline process such as dictionary generation. Moreover, PIRF-Nav can run in real time similar to other systems [65], [35] and hence is applicable to real robots. We utilized two datasets used earlier [35]––City Centre and New College––to sufficiently evaluate the PIRF-Nav performance under strong perceptual aliasing and under dynamic scene changes. We also test our PIRF-Nav using the omnidirectional image dataset, which was collected on different days under highly dynamic changes, where different events were held. The result of combining all three datasets proves that PIRF-Nav can be used efficiently with omnidirectional images, can run in the long term, and can accommodate the kidnapped robot problem. The

experiment is also expanded to testing on the New Colleges dataset. We also present a more detailed evaluation and description of the proposed PIRF-Nav.

4.2 Using PIRF

A position-invariant robust feature, designated as a PIRF, is a local descriptor generated from SIFTs (or SURFs), which appears repeatedly in an image sequence. To extract a PIRF, we must set the sliding window to extract a PIRF for w_{size} sequential images. In other words, the number of SIFTs in a single image can be reduced markedly by considering only the descriptors that can be matched to those of neighboring images. Figure 4.1 portrays a sample of the resulting extracted PIRF from three sequential images. Standard feature matching is performed for the current image and its neighbor. The number of image matchings depends on the window size, which is set by the user. For example, if the window size w_{size} is 3, then the current image I_t must be matched to I_{t-2} and I_{t-1}. Strictly speaking, the image I_t would be matched only to I_{t-1}. The resulting matched descriptor would then be used to retrieve its matches recursively from image I_{t-2} because the matching result between I_{t-2} and I_{t-1} would have already been obtained. The corresponding SIFTs, which appear repeatedly in all three images, are retrieved and averaged to obtain a single representative descriptor. This descriptor is a PIRF ψ, where $\psi = \left(d_{t-2}\left(d_{t-1}\left(d_i\right)\right) + d_{t-1}\left(d_i\right) + d_i\right)/3$, in which $d_x(d_y)$ denotes the descriptor in image I_x and matches descriptor d_y under the distant threshold θ. To match the descriptors, we use the same standard feature matching as that was used by Lowe (2004). That is, given d as a descriptor in image I_1 and d_1 and d_2 as the first and second nearest descriptors of d found in I_2, d_1 would be considered as the matched descriptor of d, if and only if $\frac{cos^{-1}(d \cdot d_1)}{cos^{-1}(d \cdot d_2)} < \theta$. The process is repeated for every descriptor d_i in the current image I_t. It is worth clarifying that the matching is performed only *once* with a neighboring image I_{t-1}. Precisely at image I_t, given that the matching results between $I_{t-2} - I_{t-1}$ were obtained at time t-1, for every matched descriptor between $I_{t-2} - I_{t-1}$, we can retrieve all corresponding descriptors recursively from I_{t-2}. In doing so, the system must perform only one matching for an input image I_t, which makes our system applicable to real-time localization and mapping because the cost for one matching is acceptable for one step of the robot.

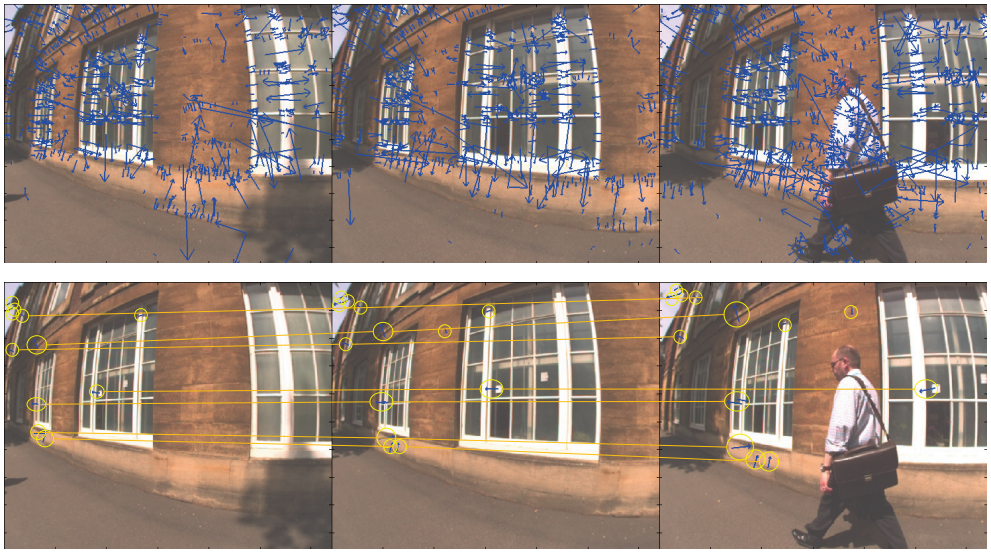


Figure 4.1 Extracted PIRF with threshold θ = 0.5. (Top-row) Three sequential images described using SIFTs. (Bottom-row) PIRFs extracted from SIFTs. Only "slow-moving" descriptors are selected to generate the mean feature vectors, PIRFs.

The extracted PIRFs are used instead of SIFT to represent an image, and this eliminates numerous SIFTs that appear to be sensitive to changes of the observer's position. Typically, we found that only about 50–100 PIRFs are derived from an image with about 3000 SIFTs. It is very apparent that the number of PIRFs, for some constant threshold θ, depends directly on the sliding window size w_{size}. The number of PIRFs would be equal to the number of matched SIFTs between two images if the windows were set to their smallest size: $w_{size} = 2$. On the other hand, if the window size is too large, then PIRFs might not be found because local descriptors that exist in many images are unlikely to exist. Therefore, it can be said that the size of the sliding window w_{size} and the threshold value θ_1 are the two main factors that affect the system's computational cost. To make it suitable for real-world applications, at least one of these two parameters should adapt dynamically. This flexibility of parameters is not easy to achieve because information of the input scenes is not available *a priori*. Among these two parameters, we found that adaptively changing the window size w_{size} is more tractable. First, the default window size is set to $w_{size} = 3$. After PIRF extraction, the resulting number of obtained PIRFs n_{pirf} can be checked to determine whether it is less than n_{pirf_min} or greater than n_{prif_max}. If $n_{pirf} < n_{pirf\_min}$, then w is decreased by 1 (i.e., reset to 2 if the default size is 3), and the PIRFs are re-extracted. In contrast, if $n_{pirf} > n_{prif\_max}$, then the

w_{size} is increased by 1 and the PIRF is re-extracted. For the latter case, the re-extraction can be repeated until n_{pirf} is in an acceptable range. Sometimes, a case arises in which n_{pirf}, even at the smallest size w_{size} = 2, yields an insufficient number of PIRFs, i.e., $n_{pirf,\ w=2} < n_{pirf\_min}$. The system handles this eventuality by simply neglecting the input image and regarding it as a new previously unexplored location; this can prevent the system from yielding a false positive. In other words, this image is treated as a "bad image" that would be discarded and would not be included in the appearance-based representation for localization. Insufficient PIRFs mean insufficient information for localization. However, this case rarely occurs. In all experiments of this study, only 3–5 images per dataset become the new location because of the insufficient number of PIRFs. Despite their simplicity, PIRFs are robust, especially against highly dynamic changes in scenes [62], while being sufficiently tolerant to perceptual aliasing.

To extract PIRF, an important parameter w_{size} related to the fps of the camera and the egomotion of the camera through the scene needs to be considered. However, our PIRF extraction handles this using the adaptive value of w_{size}. For example, consider the case in which the camera runs at 200 fps vs 0.5 fps. At 200 fps, w_{size} would be too small and several redundant PIRFs are obtained. For this case, w_{size} will automatically be incremented until the resulting PIRFs meet the acceptable range. For the case of 0.5 fps, images would be fairly far from each other and w_{size} might be reduced to its smallest value w_{size} = 2. Note that we have started trying PIRF with the 1000 km dataset of Cummins and Newman [101], where the images are captured around at an interval of 10 m. The resultant number of PIRFs is still sufficient for calculation, although the value of w_{size} would usually be reduced to 2. If the fps is very low (i.e., each picture taken every 50 m), our system would not be applicable because the number of PIRFs is insufficient. Nonetheless, our PIRF-Nav is mainly proposed for mobile robots. Unlike other vehicles, general robots do not have a speed high enough for the fps rate to be outside of an acceptable range.

4.3 PIRF-Nav Model

The environment is modeled as a set of discrete locations, each location

being described using a set of PIRFs. An incoming observation is converted into a PIRF-based representation and then for each location, we simply calculate the similarity between the PIRFs of the observation and the PIRFs of that location. The system then decides whether the observation came from a place not in the map by considering the quality of the similarity score. The proposed system uses the PIRF as the only visual feature, and takes advantage of it to obtain high performance. This system, called a PIRF-based navigation system or *PIRF-Nav*, is outlined in detail in the following subsections.

4.3.1 Modeling Appearance

In our system, scenes are simply represented as a collection of PIRFs, much like the traditional representation using raw SIFTs. An important difference is that the required number of PIRFs for representing a location is much lesser than the number of SIFTs required for similar purposes. Precisely, at time step *t*, the system must retain w_{size} images in the buffer for PIRF extraction. For example, if the window size for PIRF extraction is set to $w = 3$, then the system must retain three images { I_{t-2}, I_{t-1}, I_t} in memory and two matching results of ($I_{t-2} - I_{t-1}$) and ($I_t - I_t$). Analyzing these matching results takes less time and is tractable, as mentioned earlier in this study. It provides us the corresponding descriptors, which are slow-moving. All PIRFs of image I_t can be generated quickly. This set of PIRFs is simply used to model the image appearance.

4.3.2 Localization and Mapping

At time t, our map of the environment is a collection of nt discrete and disjoint locations L = $\{L_1, \ldots, L_{n_t}\}$. Each of these locations Li, which was created from the past image Ii, has an associated appearance model Mi. This model Mi is a set of PIRFs. Modified from naive Bayes, four main steps are used to calculate the similarity score for loop-closure detection. Figure 4.2 depicts the overall processing of the PIRF-Nav. First, an obtained image is described by the PIRFs. Simple feature matching is performed in Step 1 to obtain the basic similarity score s if the number of PIRFs is sufficient. The system proceeds to the next step if the maximum score exceeds the threshold and is not the null hypothesis M0. In Steps 2–3, the score is recalculated considering the score of neighboring models. The normalized score β is used to decide if the detected

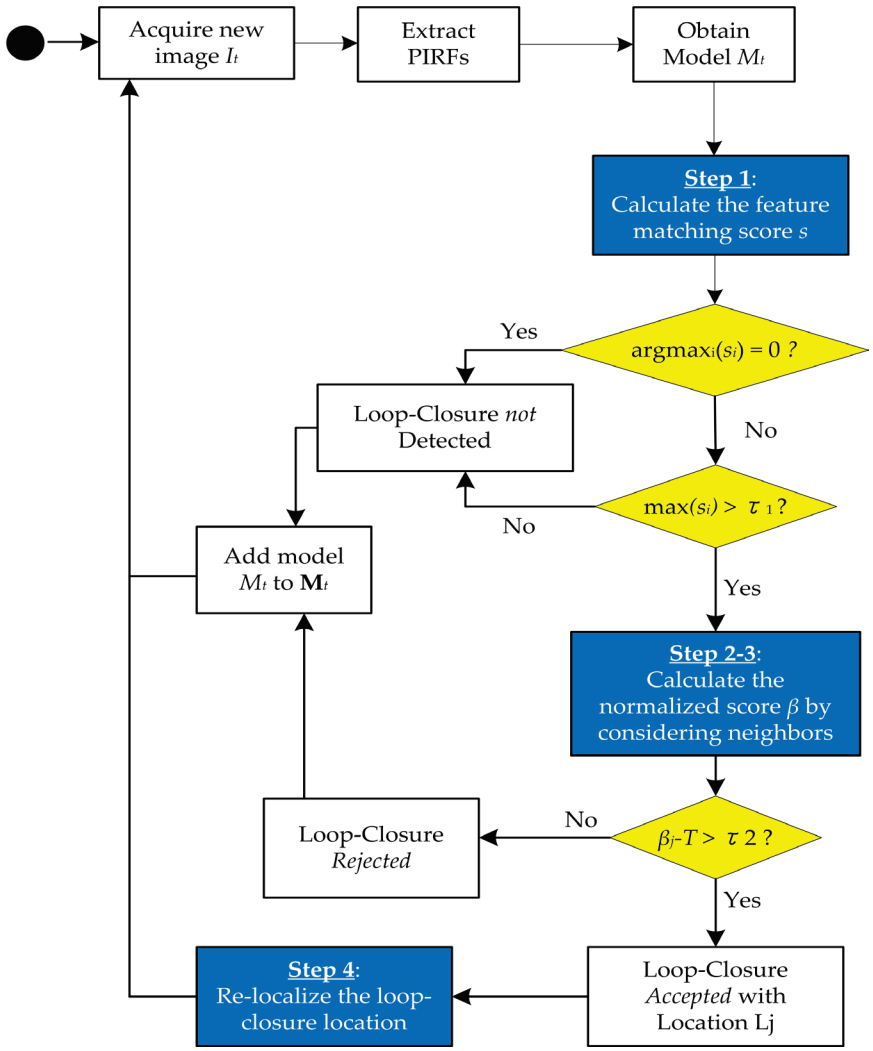


Figure 4.2 Overall processing diagram of PIRF-Nav (see text for details).

loop-closure is accepted or rejected. If accepted, re-localization (Step 4) is performed by doing a second summation over the similarity scores obtained from step 3 to re-assign the most probable location for loop-closure. The details for each step are described as follows.

4.3.2.1 Step 1: Simple Feature Matching

First, the current observed model M_t is compared to each of the mapped models $\mathbf{M}_t = \{M_0 \ldots, M_{n_t}\}$ using standard feature matching with distant threshold theta θ. Each matching outputs the similarity score between the

input model and the query model. M_0 is the model of the location L_0, which is a virtual location for the event "no loop closure occurred at time t." In fact, this event is evaluated as the event "a loop closure is found with the model M_0." M_0 is a virtual model of the virtual image I_0 built/updated at each step by randomly sampling PIRFs from all models M_i: $1 \le i \le n_t$. For example, we can sample two PIRFs from 100 past models, yielding 200 PIRFs for modeling M_0 for the current time. This technique has been proven to be effective in Angeli et al. [65]. By sampling the PIRF, models of locations that have never been visited resemble M_0 the most. In other words, M_0 will contain about 3–4 times more PIRFs than other models. In this study, we sample five PIRFs from every new input model. New PIRFs from current model M_t are accumulated into M_0 without deletion unless the number of PIRFs reach the maximum size (i.e., 3000 for this study). For instance, if we set the maximum number of PIRFs in M_0 to 3000, sample PIRFs from first 600 models would be accumulated to M_0. After this point, for every new model, M_0 randomly selects five PIRFs to be deleted and stores the five new PIRFs to the model. However, the virtual model sometimes can be re-created particularly to prevent biased sampling. For this study, we re-create virtual models at every 300 input images. That is, M_0 is reset to null. Three thousand PIRFs are randomly sampled from a set of currently obtained models. For example, if there are 1000 models obtained, the system will sample three PIRFs from each model to create the M_0.

After the first step, if M_0 turns out to be the winner, the system terminates the process and determines that M_t belongs to a new previously unseen location. Otherwise, the system performs feature matching to obtain a similarity score s between M_t and the existing models. This similarity is assessed using the score. Instead of counting matched features, we calculate the score by considering the *term frequency-inverted document frequency (tf-idf)* weighting [96]:

$$\text{tf} - \text{idf} = \frac{n_{wi}}{n_i} \log \frac{N}{n_w} \tag{4.1}$$

In this equation, n_{wi} is the number of occurrences of the visual word w in Mi, ni is the total number of visual words in Mi, nw is the number of models containing word w, and N is the total number of all existing models. The weighting is a product of the word frequency term $\frac{n_{wi}}{n_i}$ and the inverse document frequency term $\log \frac{N}{n_w}$. This scoring increases emphasis to words

that are seen frequently in a small number of images, and penalizes common words (i.e., words that appear everywhere) according to the most recent statistics. In the classification/retrieval [96], models are ranked by their normalized scalar product (cosine of angle) between query model and all existing mapped models. However, this model-to-model (image-to-image) comparison is too exhaustive for real-time robotic navigation. Angeli et al. [65] solves this by taking an advantage of the inverted index associated with the dictionary. At each current image It, each time a word is found, the list of past images in which it has previously appeared are retrieved. When a word that has appeared in image Ii is found, the statistical score in equation (4.1) is added to the score of each image Ii. That is, at each observed current image It, the similarity score between It and previous image Ii is the summation of tf-idf of all visual words that appear in both I_t and I_i:

$$s_i = \sum_{w=1}^{\alpha_i} \frac{n_{wi}}{n_i} \log\left(\frac{N}{n_w}\right) \tag{4.2}$$

where α_i is the number of visual words that appear in I_t and I_i.

We consider this scoring as accurate and thus would like to apply this scoring to our method. Unfortunately, we cannot apply this scoring for use with PIRF straightaway. In PIRF, no common vocabulary is used to represent the image. The number of PIRFs detected in an input image is quite small: it is about 50–100 PIRFs. Consider the word frequency term $\frac{n_{wi}}{n_i}$ in equation (4.2). It gives us the weight value of the frequency of the word in the image. However, if n_i is too small, this term would become very sensitive to noise. Therefore, in order to apply this scoring to our method, the word frequency term would be ignored. Because $0 \le \frac{n_{wi}}{n_i} \le 1$, we set $\frac{n_{wi}}{n_i}$ to 1 for all cases. The term $\log\frac{N}{n_w}$ can be obtained by assigning $N = n_t$ and n_w as the number of models containing PIRF w. In other words, with a small number of PIRFs, scoring based on only the inverted document frequency term is sufficient for our method by avoiding the noise that might occur in the word frequency term. By ignoring the word frequency term $\frac{n_{wi}}{n_i}$, the scoring function can be written as

$$s_i = \sum_{k=1}^{m_i} \log\left(\frac{n_t}{n_{w_k}}\right) , \tag{4.3}$$

where n_{w_k} is the number of models M_j, $0 \le j \le n_t$, $j \ne i$, containing PIRFs that match the k^{th} PIRF of the input model M_t, m_i is the number of all

matched PIRFs between input model M_t and query model M_i (m_i is equivalent to α_i in (4.2)). This model closely resembles scoring based on the number of matched PIRFs between M_t and M_i. The difference is that we assign more weight to the PIRF that is likely to be distinctive. Matching between the input and query model will earn a high score when most of the matched PIRFs appear only in the query model.

4.3.2.2 Step 2: Considering Neighboring

Once we obtain the similarity score s_i between the input model M_t and query model M_i, we proceed to the next step of scoring. Generally, this score can be used simply to determine that the potential loop-closure is found for the model $M_{\mathrm{argmax}_i(s_i)}$. However, accepting or rejecting loop-closure detection based on the score from a single image is sensitive to noise. To obtain high precision, the system must not assert loop-closure detection with high confidence based only on a single similar image. This avoidance of detection can be achieved by additionally considering the similarity score of neighboring image models. Therefore, we create another scoring function as

$$\beta_i = \sum_{k=i-\omega}^{i+\omega} \left(s_k \cdot p_T(i,k) \right) , \qquad (4.4)$$

where the term $p_T(i,k)$ is the transition probability generated from a Gaussian sigma = 2 on the distance in time between *i* and *k* (note that we did not use any motion models in this study). The Gaussian parameter used here is determined in order to obtain the best result (See appendix B). However, considering the graph in the Appendix, this parameter does not significantly affect the performance of PIRF-Nav. In addition, ω stands for the number of neighbors examined. The use of the term p_T is much like the time evolution model of the probability density function in eq. (5) of Angeli et al. [65], in the sense that it gives the probability of transition from state *k* to state *i*. By this scoring, a high score β_i of the query model M_i is obtained if and only if its neighbors $M_{i-\omega} \ldots, M_{i-1}, M_{i+1}, \ldots, M_{i+\omega}$ also earn a high score.

4.3.2.3 Step 3: Normalizing the Score

Now that the similarity score β_i of the model M_i can be obtained properly, it is necessary to normalize. Practically speaking, the resulting number of

matched features between images is difficult to predict. Matching two images taken from the same place might output only a few matched features, although some matching between different images might yield many matched features. That is, a score β_i of model M_i, even though all of its neighbors obtain a good score, could be lower than another score β_j of the single model M_j if the matched features between M_t and M_j are numerous. For that reason, the score must be normalized.

Based on the score β_i of model M_i, we calculate the standard deviation and mean over all these scores to determine the loop-closure acceptance/rejection. To determine the loop-closure acceptance or rejection, we need to consider the deviation among scores. If the input image is to be localized to some previously visited places, the similarity score of this place to the nearest model and neighbors should be high, while the score of the other model should be low. Otherwise, the loop-closure would be rejected. However, the standard deviation and mean directly depend on the number of data. Consider figure 4.3 for example. β_i obtains the highest score. We calculate the deviation and means over the set of scores $\beta_{i-l_n}, \dots, \beta_i, \dots, \beta_{i+l_n}$.

The normalization used here is employed from the method done in the incremental BoW method [65]. Considering the set of scores $\beta_{i-l_n}, \dots, \beta_i, \dots, \beta_{i+l_n}$, we calculate the standard deviation (σ) and means (μ), where l_n indicates the number of neighbors taken into consideration. The normalized score C_i of a score β_i is obtained as follows:-

$$C_i = \begin{cases} \frac{\beta_i - \sigma}{\mu}, & \text{if } \beta_i \geq T \\ 1, & \text{Otherwise} \end{cases} \tag{4.5}$$

where

$$T = \sigma + \mu \tag{4.6}$$

By this normalization, the model with the sufficiently high score would be rewarded, while the low score of the other model would be penalized. The illustration is shown in figure 4.3. There are l_n = 7 neighbors taken into consideration, resulting in a set of 15 scores. The summation of deviation and mean T is calculated. The normalized score C is shown at the bottom graph.

The obtained similarity score for all possible models determines the most potential loop-closure location L_j, where

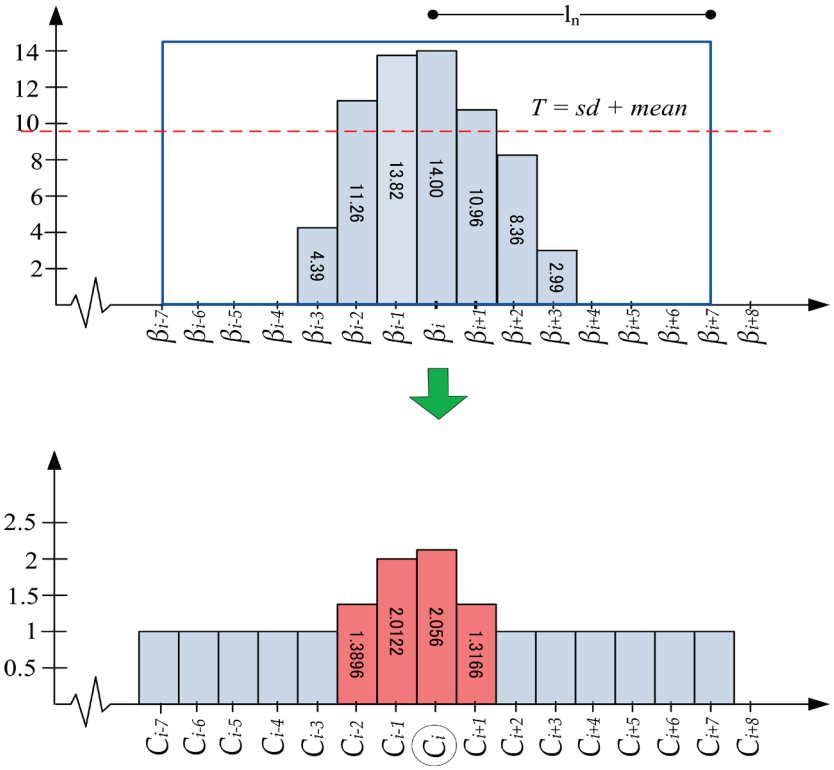


Figure 4.3 Sample normalizing the similarity score of the query model M_i and input model M_t.

$$j = \text{argmax}_i \, C_i \quad . \tag{4.7}$$

The parameter l_n is used to limit the number of scores for calculating the T value, because the T value can be very different in the long run as the number of mapped models increase. It should be noted that the difference of the l_n value does not greatly affect the overall performance; it is used only to make sure that the calculated T value for every time step is in the same scale. Considering the maximum score obtained in Step 1, it is also based on the same scale. Because the number of PIRFs for each image is limited to a maximum of 150 as described in Section 3, the maximum number of matched PIRFs would always be less than 150.

4.3.2.4 Loop Closure Acceptance/Rejection and Re-Localization

The obtained location L_j would be *accepted* as loop-closure if $\beta_j - T > \tau_2$. We use the T value to determine the loop-closure acceptance/rejection

because it indicates both the deviation and mean value of the scores. The difference between the similarity score and T can reflect the quality of localization. That is, the detected loop-closing location L_j would be accepted if and only if β_j is greater than $\sigma + \mu$ by at least τ_2. After the localization, unlike other methods, the current model M_t would be discarded. The number of mapped model n_t would not be increased (see figure 4.12). The system would store the model M_t as the new model in the map and continue acquiring new images only if the loop-closure is *rejected*. This enables us to discard redundant models of previously visited locations. In this study, we simply discard the input model M_t if the loop-closure is accepted. However, it might be helpful if the model is used to update the existing model, so that it can be more tolerant to dynamic changes.

For the case of loop-closure acceptance at L_j, the system must perform a few more tasks. Ideally, the neighboring model scores of location L_j should decrease symmetrically from a model score C_j. However, scenes in dynamic environment always contain moving objects that frequently cause the occlusion. Some landmarks might be occluded. Therefore, we compensate for this by performing the second summation over the neighboring score model to achieve a more accurate localization. The sample of this problem is shown by Fig. 4.4. The upper graph shows the score obtained from step 2. The value of the maximum score C_j , j = 87, is sufficiently high to satisfy the acceptance/rejection condition. However, the graph is not symmetrical. The second summation can refine the graphs to be more symmetrical, and the location can be re-located. It should also be noted that if the graph obtained from step 2 is already symmetrical, the second summation will just emphasize the score of the winning location.

Particularly, given a set of beta scores $C_{\omega+1}, \ldots, C_{n_t-(\omega+1)}$, the scores are re-calculated using the equation (4), resulting in

$$C_i^{'} = \sum_{k=i-\omega}^{i+\omega} \left(C_k \cdot p_T(i,k) \right) \tag{4.8}$$

where $\omega + 1 \leq i \leq n_t - (\omega + 1)$. The loop-closure location L_j is finally re-located by these scores:-

$$j = \text{argmax}_i \, C_i^{'} \tag{4.9}$$

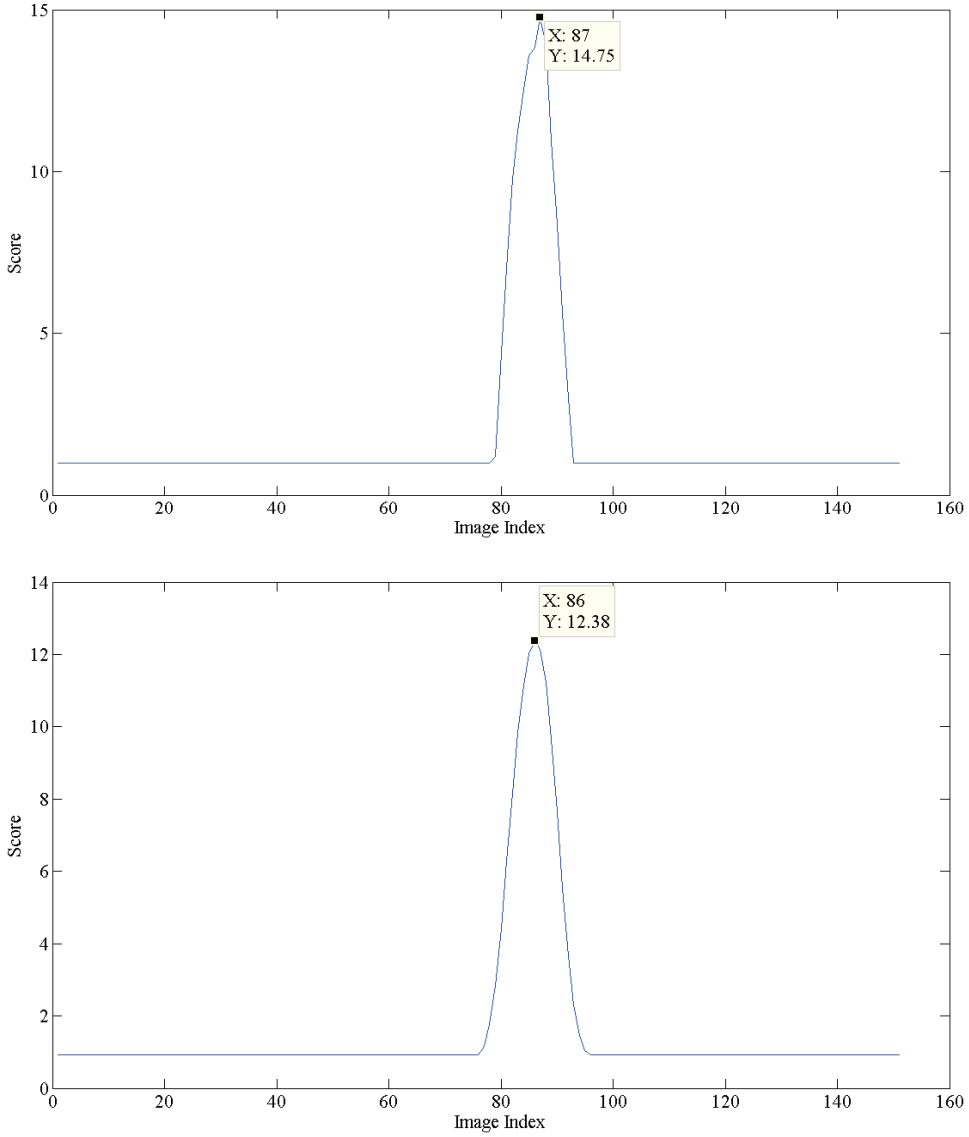


Figure 4.4 Sample case where normalized beta score is not symmetrical (upper). By performing the second summation, the graph becomes more symmetrical (bottom).

4.4 Results and Experiments

We tested the proposed PIRF-Nav using three outdoor image datasets. Each one of them is used to examine various difficulties, including dynamic changes and perceptual aliasing. The results obtained from each dataset prove that PIRF-Nav is efficient. Finally, we combine all datasets to evaluate the performance of PIRF-Nav in the long run. The computation time is acceptable

for real-time applications with an impressive rate of recall with 100% precision.

4.4.1 Datasets

As described, three outdoor image datasets are used to evaluate the performance of PIRF-Nav. The first dataset (New College), provided by Cummins and Newman [35] was chosen to test the system's robustness to perceptual aliasing. It features several large areas of strong visual repetition, including a medieval cloister with identical repeating archways and a garden area with a long stretch of uniform stone wall and bushes.

The second dataset (City Centre), which was also provided by Cummins and Newman [35], was collected intentionally to test the matching capability in the presence of scene changes. It was compiled using the data collected along public roads near City Centre, featuring many *dynamic* objects such as traffic and pedestrians. Moreover, it was collected on a windy day with bright sunshine, which renders abundant foliage and shadow features unstable.

For these two datasets provided by Cummins and Newman [35], a real mobile robot collected the images. The robot moved through its environment, collecting images to the left and right of its trajectory, approximately every 1.5 m. There were a total of 2,474 images collected from City Centre and 2,146 images from New College. Each has 640 × 480 resolution. For this study, left and right images are combined before being processed by our system, that is, one location is associated with one left image and one right image, and 1,237 images from City Centre and 1,073 images from New College with a resolution of 1280 × 480 were obtained. Figure 4.5 shows image sample from all datasets featuring different conditions in scenes.

Actually, PIRF-Nav is proposed for highly dynamic environments, as stated in the title. Therefore, we collected an additional dataset to improve these PIRF advantages. The third dataset—Suzukakedai Campus (our campus)—was chosen to test the matching ability further, even in *highly dynamic* environments: the first and the second visits are different events. For this dataset, images are grabbed using a single monocular handheld camera (HDC-TM300; Panasonic Inc.) with an omnidirectional lens and with a frame rate of 0.5 frame/s. Images were collected during two days, on which different specific events were held. The first sequence (Seq. 1) was collected on an

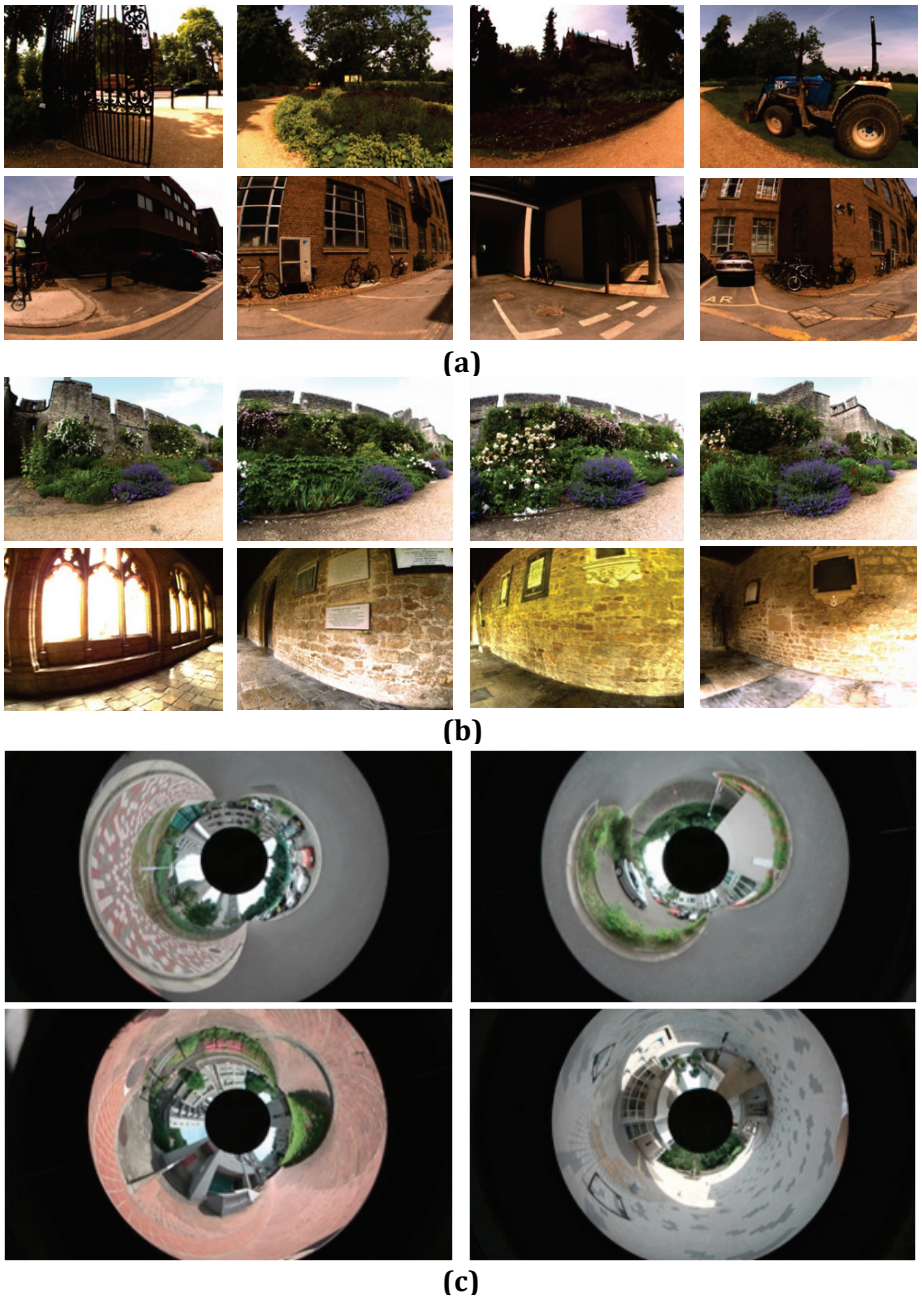


Figure 4.5 Samples of images from all datasets. **(a)** Sample images from City Centre Dataset with condition of dynamic changes. **(b)** Sample images from New College. The data set challenges on the perceptual aliasing problem. **(c)** Sample images from Suzukakedai. The dataset further challenge on highly dynamic conditions.

afternoon during an open-campus event under clear weather. Many people attended the event. Tents and booths were set up for this event. The second sequence (Seq. 2) was collected on a cloudy evening of a normal day with

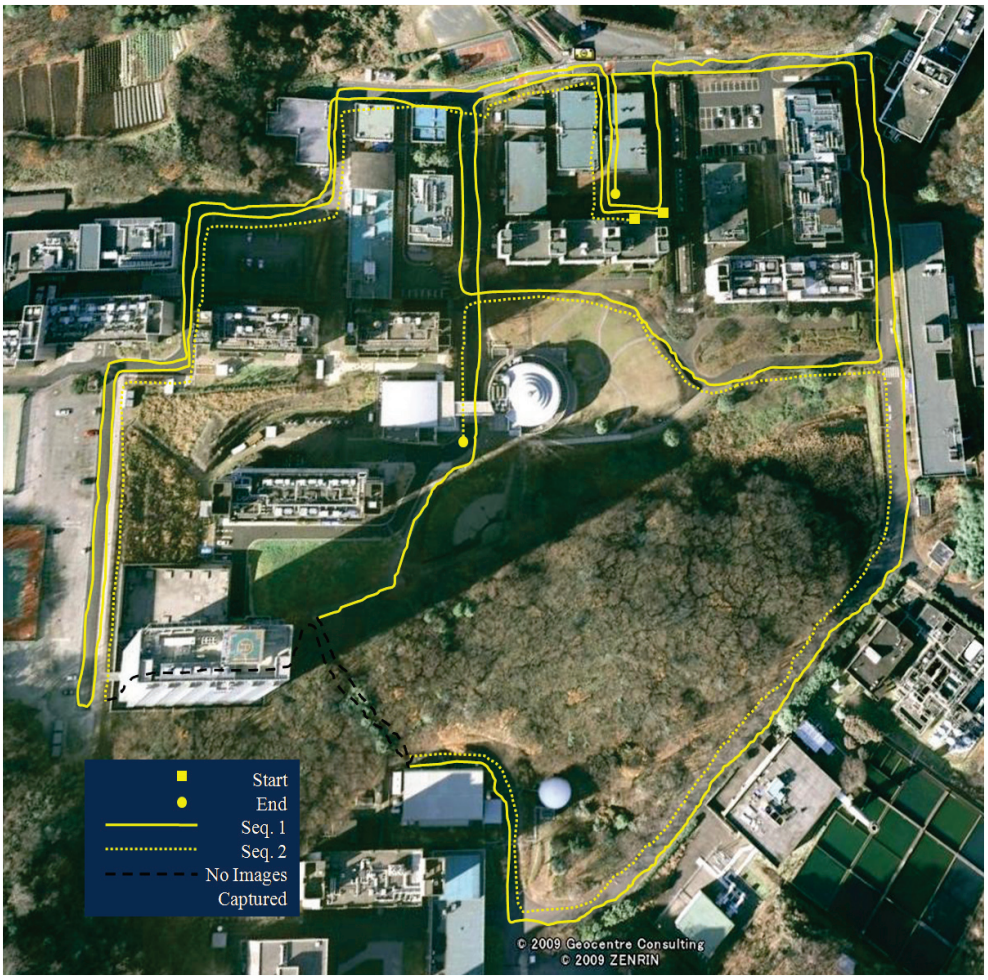


Figure 4.6 Route of walks for data collection on two different days for the Suzukakedai Campus dataset.

fewer people. All tents and booths had been removed. Major portions of many scenes changed because of this event. We consider this change as highly dynamic because marked changes occur for some special events. Collecting this dataset yielded a totals of 1,079 omnidirectional images (689 + 390) with 1920×1080 resolution. Figure 4.6 shows the walking route for data collection. The short dotted line indicates routes in which no images were collected.

4.4.2 Baseline

Among recent appearance-based methods, we consider FAB-MAP as the most suitable baseline for comparison. Based on the popular BoW scheme,

FAB-MAP uses the complex probabilistic framework to handle the problem of perceptual aliasing properly. It considers the problem at the word level (i.e., the input information level). In fact, two recent well-known BoW-based methods for this task exist: FAB-MAP of Cummins and Newman [35] and the incremental BoW method of Angeli et al. [65]. However, a true comparison in terms of accuracy between these two methods has not been done because the latter was mainly proposed as a complementary method; a dictionary of the latter method can be generated and updated in an online manner, whereas FAB-MAP requires a preliminarily well-formed dictionary. An earlier study [65] also described this requirement. In terms of accuracy, it is more appropriate to compare the result of PIRF-Nav with the FAB-MAP.

Although FAB-MAP on the Suzukakedai Campus dataset can be implemented straightaway, the implementation at City Centre and New College requires a slight modification for a fair comparison with PIRF-Nav. In our study, one location comprises two images (left-hand-side and right-hand-side images), but the original FAB-MAP represents one location with only a single image. That is, FAB-MAP performs loop-closure detection on 2,474 images, whereas PIRF-Nav does it on 1237 images. To cope with this unfair condition, we combine the results of left-image and right-image of FAB-MAP into one. Particularly, for each location, FAB-MAP has two answers, one for the left and another for the right image. If either one of these answers is correct, the localization would be considered as correct. Consequently, with 100% precision, the recall rate on the 2,474 images of City Centre increased from 37% to recall rate of 43% on the 1,237 images, and the 48% recall rate on the 2,146 images of New College increased to 61% recall rate on the 1,073 images. These combined results facilitate a fair comparison.

For the Suzukakedai campus, where the FAB-MAP can be implemented without any changes, no proper dictionary for FAB-MAP exists. The dictionary used for this dataset is identical to that used in the City Centre and New College. However, our PIRF-Nav requires no preliminary dictionary generation process. In fact, PIRF-Nav can start on any dataset without any modifications even though the datasets are collected from different countries with different building structures. Failure of FAB-MAP on the Suzukakedai dataset confirms that “good” vocabularies are indispensable for FAB-MAP.

4.4.3 Initialization and Testing Conditions

As described previously, there are a few parameters that play an important role in PIRF-Nav. We explain the importance of all parameters and their appropriate values used for this study as follows:

- θ : This is the distance threshold for PIRF extraction and feature matching. It controls the number of extracted PIRFs for appearance representation. If this parameter is too large, then more PIRFs would be obtained. Some might be noisy and useless, e.g., PIRFs that will never be matched during localization. In this work, we found that θ = 0.5 offers the highest performance for all datasets. Actually, PIRF-Nav works best with small θ because a PIRF must be extracted from slow-moving features. We have also tried θ = 0.5, 0.6, and 0.7, but the results are not much different, i.e., θ = 0.6 offers about 3% lower recall rate than θ = 0.5.
- n_{pirf_min} and $n_{pirf\_max:}$ These are the minimum and maximum number of PIRFs for representing a single image. These numbers are set manually to 10 and 100, respectively. The values control the number of extracted PIRFs per image to be around 10–150. We have tried to set these parameters at 100 and 200, which yields about 100–250 PIRFs per image. Although the performance is not much different from settings at 10 and 150, the computation increases greatly. Therefore, we select these values as the most appropriate value for PIRF extraction of images for all datasets.
- τ_1 : This is the threshold value for loop-closure detection/non-detection. This parameter can filter out images with very few matched PIRFs between the input model and the query model because a mere few PIRFs are insufficient for generating a reliable similarity score. In other words, this parameter indicates which image is considered "insufficient information." In this work, we set τ_1=3 as the threshold value for all experiments because it offers the highest recall rate for all datasets. This parameter can be set to other values such as 4 or 5. The overall accuracy will not differ much by changing this value. Using this parameter can reduce the PIRF-Nav computation time. For images with insufficient information (too few PIRFs), the system can simply retain the input model as the model for the new location and start fetching new input images.
- τ_2: This is the threshold for determining the loop-closure acceptance or

rejection. This parameter indicates the difference between the beta score of the most probable loop-closing location and the term $\sigma + \mu$. Actually, this parameter is difficult to set in order to obtain 100% precision. As in other works (Angeli et al., 2008; Cummins & Newman 2008), we select the threshold by considering the result to obtain 100% precision for all datasets. The relation between the recall rate, the precision rate, and this threshold are shown by the graphs in the Appendix. A little drop in threshold can reasonably increase the recall rate while retaining high precision. Considering all datasets in this study, $\tau_2 = 3.1$ yields the highest accuracy with 100% precision. (See Appendix A)

- ω : This is the parameter indicating how many neighbors are examined for calculating the updated score in Step 2. If ω is large, the scoring would become slightly more stable, but the accuracy of the localization would drop. For example, if ω = 20, then there would be 20 models that cannot be used in localization (first 10 models and last 10 models). However, this parameter has little effect on the overall accuracy of PIRF-Nav. For this study, ω has been set to 3. Other values ω = 4, 5, 6 have also been tried, but the accuracy is mostly equivalent while the computation time is increased. However, this parameter is expected to depend directly on the camera velocity. This value is expected to be larger because many neighbors will have high scores if the robot walks slowly while the capturing rate is high.
- l_n : The parameter indicates the number of model scores to be included for calculating the standard deviation and means. Actually, this parameter is not very significant for databases less than 4000 images; σ and μ will not be very different even though they can be calculated for all over scores. However, these two values can gradually change as the number of score models is increasing. Therefore, we need to set some fixed number of score models for calculating μ and σ to make the score of every time step be in the same scale and comparable.

With these parameters, for each dataset, each collected image is processed using the proposed PIRF-Nav and is used either to initialize a new place, or, if loop-closure is detected and accepted, to localize the correct loop-closure location. No additional dataset exists for offline dictionary generation.

Although there seems to be several parameters that users need to manually

set, their values are not sensitive to overall performance. The results in Fig. 4.10 show that the recall rate considerably increases by sacrificing just a small amount of precision. With the parameters used in this paper, although it cannot be guaranteed to always yield 100% precision in any environment, its resulting recall rate should be sufficiently good for loop-closure detection. In any case, a system that can always offer 100% precision in any environment is difficult to obtain.

Three main experiments and one additional experiment were conducted in this study. The three main experiments were the testing on each dataset—New College, City Centre, and Suzukakedai Campus. An additional experiment was performed by combining all three experiments (1,237+1,073+1,079 = 3,389 images). This last experiment tested whether PIRF-Nav can run incrementally over the long term. It is useful in many places and can accommodate the kidnapped robot problem (i.e., taking the robot from England to Japan). Despite the increased number of mapped places, the precision rate for loop-closure detection can remain at exactly 100% with a high recall rate.

4.4.4 Results

The results of the experiments conducted for City Centre, New College, and Suzukakedai Campus are shown in figure 4.7 – 4.9, and as precision-recall curves in figure 4.10. Although the FAB-MAP is run on the full-scale image, PIRF-Nav has also been tested with three different levels of scale reduction: 0.25, 0.5, and 0.75. Because PIRF-Nav is based on feature matching, it requires one image matching for every input image. Additionally, all programs in this study were written in MATLAB and executed with a 3.2 GHz CPU (Intel Corp.), therefore the computation was slightly slower than that of FAB-MAP. We resolve this by showing that, even with a reduced image, PIRF-Nav offers a much higher recall rate with 100% precision than FAB-MAP. Although it is reasonable to say that PIRF-Nav's computation time can be shortened further, it is more appropriate to present what we have obtained instead of what we expect to obtain.

Figure 4.6 – 4.9 presents navigation results overlaid on the aerial photographs. These results were generated using images with 50% scale reduction. The system correctly identifies a large portion of possible loop-closures with good scores. No false positives met the score threshold.

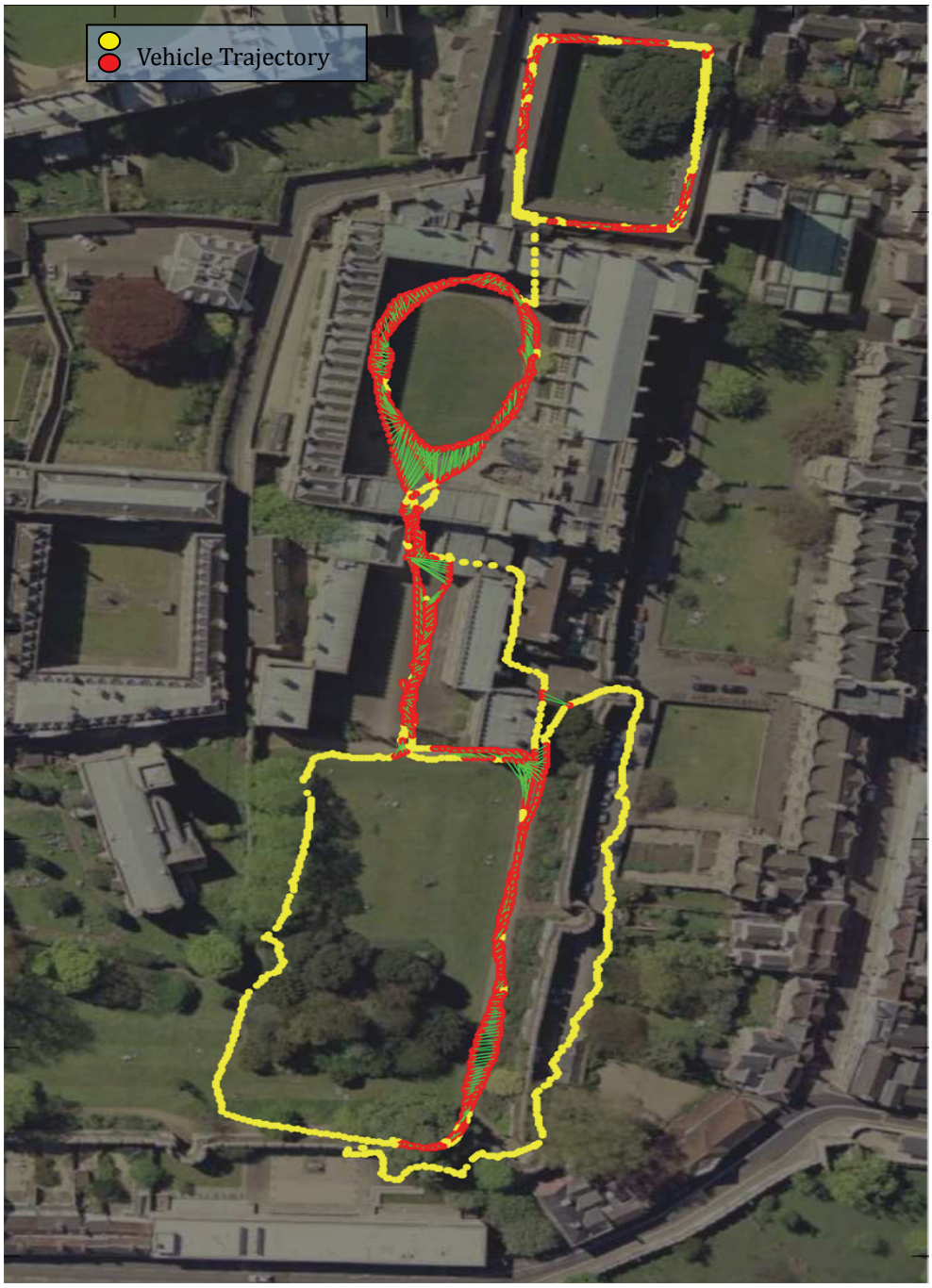


Figure 4.7 Appearance-based matching results for New College dataset overlaid on an aerial photograph. The aerial photograph and the ground truth data are obtained from Cummins and Newman [35]. A result is for image scale = 0.5. Two images that were assigned a similarity score $\beta - T > 3$ (based on appearance alone) are marked in red (darker color) and joined with a line. There are no incorrect matches that meet this threshold.

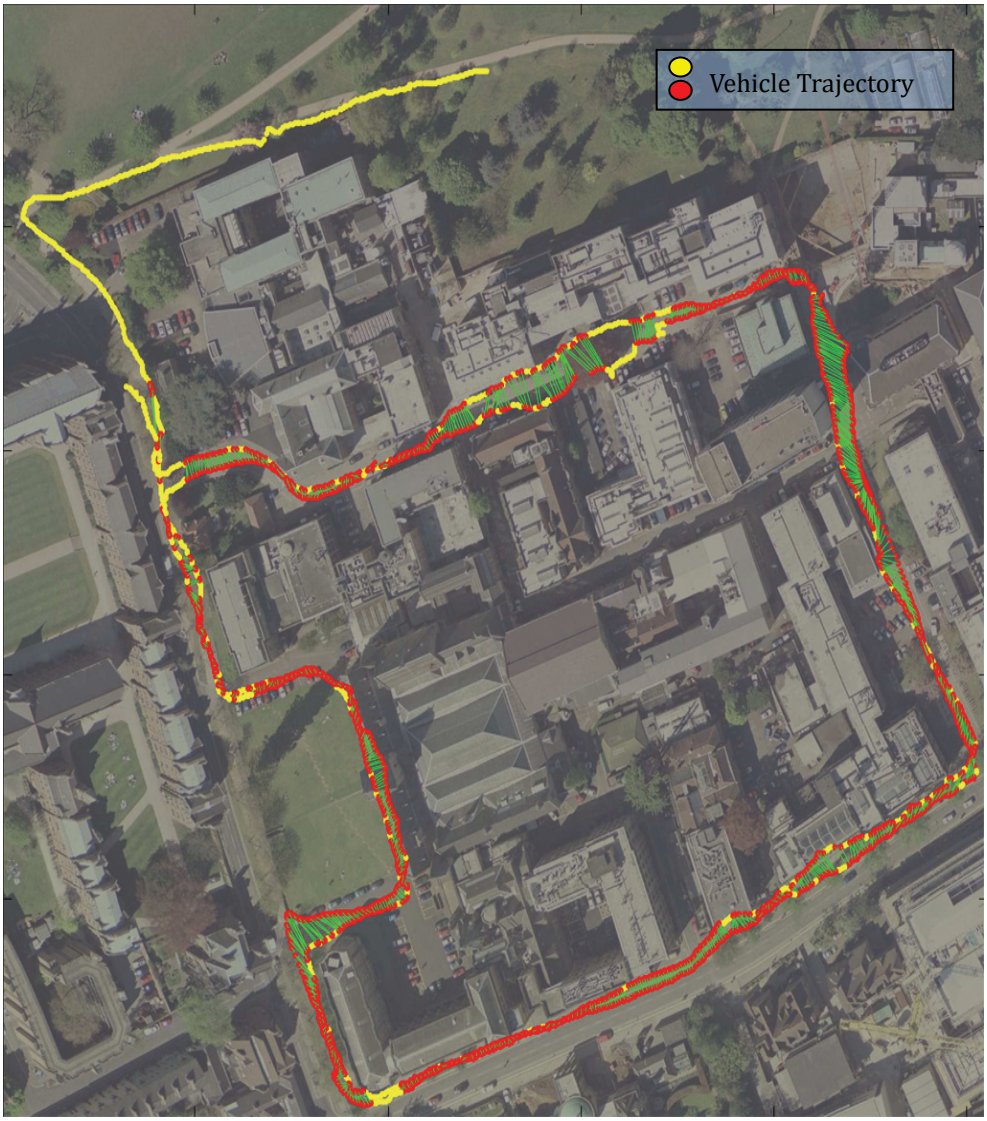


Figure 4.8 Appearance-based matching results for City Centre dataset overlaid on an aerial photograph. The aerial photograph and the ground truth data are obtained from Cummins and Newman (2008a). A result is for image scale = 0.5. Two images that were assigned a similarity score $\beta - T > 3$ (based on appearance alone) are marked in red (darker color) and joined with a green line. There are no incorrect matches that meet this threshold. By this result, approximately 85% of loop-closing places have been correctly detected as loop-closure without any false positives which results in 100% precision.

Figure 4.9 Appearance-based matching results for Suzukakedai dataset overlaid on an aerial photograph. Ground truth is generated from the GPS data. Any pair of matched images that was separated by less than 10 m based on GPS was accepted as a correct correspondence. A result is for image scale = 0.5. Two images that were assigned a similarity score $\beta - T > 3$ (based on appearance alone) are marked in red (darker color) and joined with a green line. There are no incorrect matches that meet this threshold.

Precision-recall curves are depicted in figure 4.10. The curves were generated by varying the threshold value τ_2 at which a loop-closure was accepted. Ground truths were obtained from Cummins and Newman [35] for City Centre and New College dataset, and labeled by GPS for the Suzukakedai dataset. The threshold value for 100% precision varied depending on the image size. For parameters, $\tau_2 = 3$ has been set for all image scales. These thresholds offer 100% precision for all datasets for each image scale. The effects of threshold values on precision and recall for each image scale are

shown as graphs in the Appendix. The graphs show that changing of threshold values does not affect the recall rate too much.

Considering 100% precision, PIRF-Nav achieves a high recall rate using an image scale more than 0.5. It is particularly interesting that a 100% image scale does not always offer the highest recall rate (about 70–80% for the City Centre and New College datasets), although the image scales of 0.5 and 0.75 offer better rates of recall. The highest rate is obtainable using an image at scale = 0.75; about 85% for City Centre and New College datasets. Unfortunately, the image size at scale >0.75 consumes too much time for real-time applications by the current PIRF-Nav. Therefore, PIRF-Nav requires a reduced size of an image at scale <= 0.5 to run in real time. In other words, compared to the approach of Cummins and Newman [35] and Angeli et al. [65], PIRF-Nav, in its current form, still has slightly lower performance in terms of computation time. However, PIRF-Nav compensates for this shortcoming with the ability to run on various image scale sizes. Even at a 50% image scale, PIRF-Nav offers about two times the rate of recall of the FAB-MAP in real time (78% for New College, and 84% for City Centre).

It is noteworthy that the recall rate of the Suzukakedai campus dataset is apparently markedly lower than those of City Centre and New College because it contains sequences obtained under markedly different conditions (specific events). By considering loop-closure only on the Seq. 1 (first day), for which all images were collected on the same day, the system achieves a 68% recall rate with 100% precision for the image scale of 0.5. This rate is not very different from those at City Centre and New College. In fact, FAB-MAP did not perform well with this dataset. We believe that the main reason is the dictionary. The structures of most buildings in England and Japan differ greatly. Vocabulary captured from streets in England is not useful to describe the feature in scenes in Japan. Nevertheless, PIRF-Nav can run in various environments without the need for offline dictionary generation. The Suzukakedai Campus result also proves that PIRF-Nav is compatible with omnidirectional images.

Computation times of PIRF-Nav for each image scale of all datasets are shown in the graphs of figure 4.11. For the computation time, we are interested only in the image scale in which the PIRF-Nav can process in real time: scale = 0.5 and 0.25. According to the graphs, the computation time is acceptable for real-time application. The average time is about 2–3 s, including SIFT extraction. Unlike other approaches, the computation of PIRF-Nav

depends on the number of extracted PIRFs of the current image, which creates high peaks in the graph. Results show that time is acceptable for real-time applications. For City Centre and New College, images are captured approximately every 2 s. Consequently, each combined left and right image is obtained about every 4 s. The average time of about 2–3 s for one image is acceptable. The time can be reduced further to yield high speed with an image scale of 0.25 with averaged time per image within 1 s, including SIFT extraction. This high-speed mode of PIRF-Nav yields a lower rate of recall, but the recall rate is approximately equal to that of FAB-MAP. The computation time of PIRF-Nav is apparently even faster than the original FAB-MAP. However, the latest FAB-MAP 2.0 [101] has been improved to be applicable to cars instead of robots. It processes one image within 0.1 s, which is much faster than our system. Nevertheless, the PIRF-Nav is currently being proposed for robots by emphasizing the recall rate improvement. Although no clear results for testing FAB-MAP 2.0 exists on City Centre and New College, its accuracy should be equivalent because the authors specifically examine speeds instead of accuracy, as was done in FAB-MAP 1.5 [102]. In any case, it would also be interesting to consider extending PIRF-Nav for use in automobiles, similarly to FAB-MAP 2.0. We leave this as a topic for future studies.

The computation time of PIRF-Nav might also be considered as the number of feature matchings required for each input image (see figure 4.13). For all datasets, at an image scale of 0.5, PIRF-Nav's maximum number of required feature matchings is about 4000–5000K, which implies that the maximum number of feature matchings for every input image is approximately equal to that required for image-to-image matching (i.e., match 2000–4000 SIFTs to 2000–4000 SIFTs of 4000k – 16000k matching). In other words, for every input, PIRF-Nav requires a processing time approximately equal to that required for image matching twice (one for PIRF extraction and another for loop-closure detection).

It is noteworthy that the number of feature matchings, as shown in figure 4.13, apparently does not correspond with the computation time portrayed in figure 4.11; the numbers of feature matchings for image scales 0.5 and 0.75 are not very different. However, their computation times differ greatly: the PIRF extraction for image scale 0.5 is much greater than scale 0.25, although feature matchings for loop-closure detection are equivalent. Furthermore, it is noteworthy that the number of feature matchings becomes zero for some

images. This process of zeroing the matchings is done by Step 1 of the algorithm; images with either insufficient amount of PIRFs or a low similarity score are rejected.

For memory, the average number of PIRFs for each model is approximately 100 (50–150). The memory required for one image representation is therefore $100 \times 128 = 12{,}800$ floats. This necessary memory size is slightly less than that required by the former incremental method of Angeli et al. [65], where the dictionary size is about 40,000, which results in 40,000 integers to represent one image.

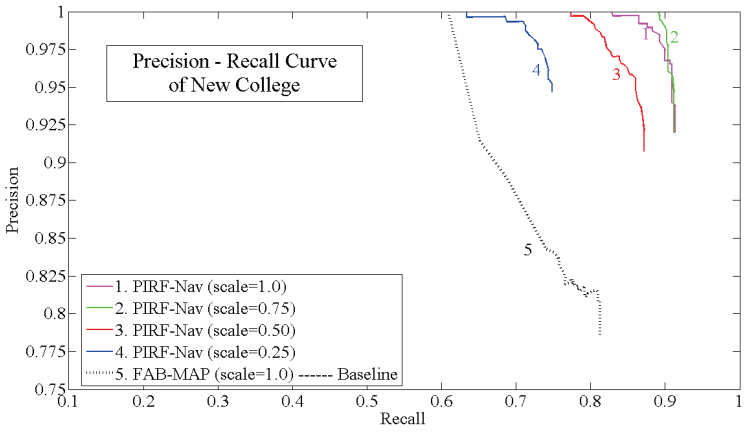
The graphs in figure 4.12 show the number of extracted PIRFs accumulated for representing the map (all visited locations) at each time step. In the current PIRF-Nav, the system simply discards the input model, which is detected as the loop-closure location, so that the number of accumulated PIRFs becomes approximately stable for loop-closure detected locations. The dotted line in figure 4.12, shown by hand, roughly indicates images being detected as loop-closure. For all datasets, the maximum number of PIRFs that must be retained is less than 60,000 PIRFs. This number of PIRFs is acceptable compared to the method of Angeli et al. [65]; 40,000 visual words for 500+ images are not much different from 60,000 PIRFs for 1000+ images. By further consideration of the graph, we can also say that the number of extracted PIRFs can be limited to the range of 50–150 for any datasets at every image scales, which guarantees that PIRF-Nav will not face the problem of retaining highly overloaded features.

In addition to testing of three datasets (City Centre, New College, and Suzukakedai), we conduct the last experiment by combining all datasets. This experiment proves two more advantages of PIRF-Nav: performance in the long run and kidnapped robot problem. The precision-recall curve of this experiment is portrayed in figure 4.14 (a). For this experiment, we specifically examine only image scales of 0.25 and 0.5 because larger scales are unacceptable for real-time applications. The computation time, the number of accumulated PIRFs, and the number of feature matchings are presented in figure 4.14 (b), figure 4.15 (a), and figure 4.15 (b), respectively. In terms of accuracy, PIRF-Nav can still obtain a good recall rate (65%) with 100% precision. In addition, by sacrificing only a small amount of precision (i.e., with 97% precision), the recall rate increases considerably to 72%. In terms of time, even with 3000+ images, the computation time is still acceptable for real-time

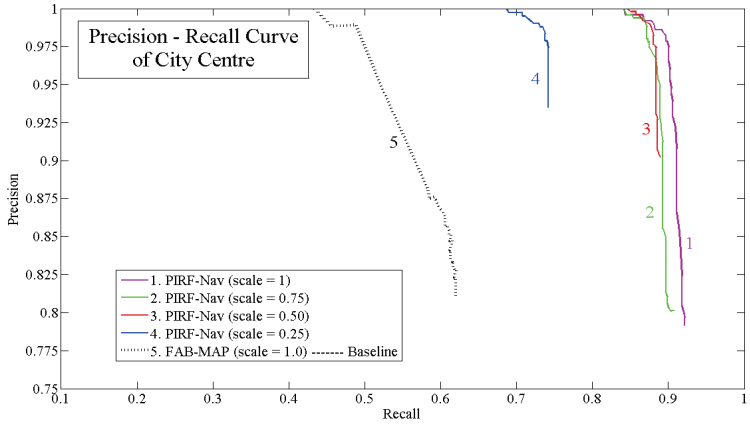
applications. The averaged processing time per image increases to around 4–5 s. However, because of the fact that the combined scene of each location from New College and City Centre is obtained every 4 s, this processing time is not too far from real-time performance. This processing time is also not too different from that reported by FAB-MAP 1.0 (comp. time is *ea.* 3–6 s per image for City Centre).

The increased number of PIRFs does not affect the system accuracy much (the recall rate for each dataset is still approximately the same) because the PIRFs are sufficiently distinctive for use as good signatures for each place in spite of the presence of dynamic changes and perceptual aliasing. The required feature matching number is still 4000K–16000K. The PIRFs retained in the system for the last image are approximately 120K. This number is apparently too large for only 3000+ images, as FAB-MAP 2.0 [101] requires 100K words for 100K images. However, this amount of PIRFs might be acceptable for mobile robots in the same sense as FAB-MAP 1.0 [65], [35], as offline dictionary generation is not needed. In future studies, we plan to extend the PIRF-Nav to be sufficiently fast for application to car navigation systems such as FAB-MAP 2.0, as discussed in the next section.

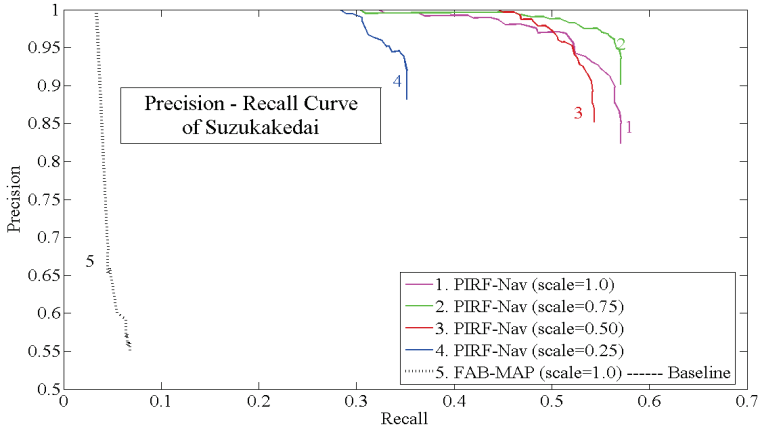
Some examples of typical image matching results are presented in figure. 4.16, figure 4.17 and figure 4.18. Figure 4.16 highlights robustness to drastic viewpoint changes in New College. The PIRF-Nav can achieve this by considering the neighboring scores. For this sample, two peaks in the score exist, which might imply that the robot used to visit this place previously at least two times. The graphs show in each step how the score has been updated and used to calculate the pdf for re-localization. Through re-localization, an image which earns a lower score in step 1 might obtain the highest pdf in the step 4. Figure 4.16 shows matching performance in the presence of scene changes in City Centre. For this matching, PIRF-Nav mostly ignores the truck because it was believed to be an unreliable object. We emphasize that these results are not outliers; they represent typical system performance. Figure 4.18 shows more matching in the presence of highly dynamic changes in the Suzukakedai campus. Empty parking lots in the query image become full in the input image, which does not affect PIRF-Nav because it mostly ignores such nearby unstable objects.



(a) New College



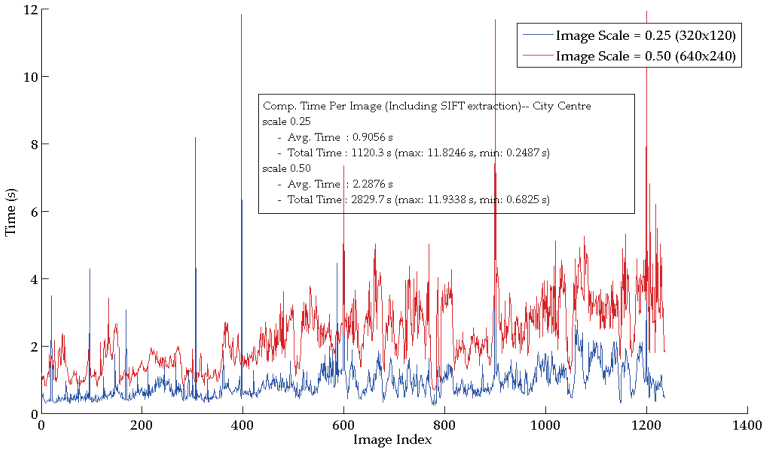
(b) City Centre



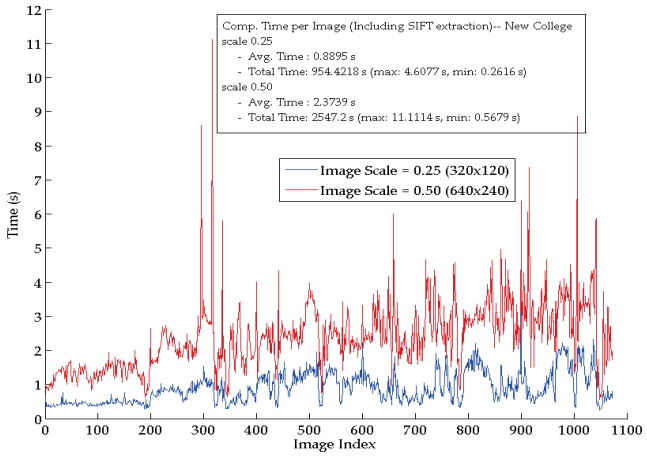
(c) Suzukakedai Campus

Figure 4.10 Precision-recall curves for all experiments at all image scales: **(a)** New College, **(b)** City Centre, **(c)** Suzukakedai Campus,

(a)



(b)



(c)

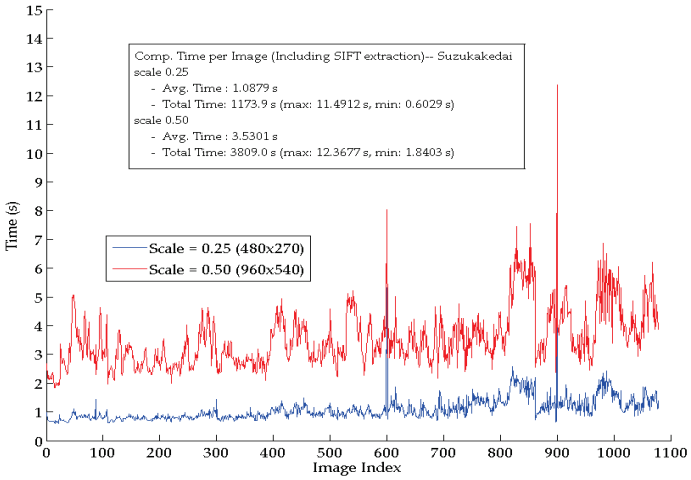
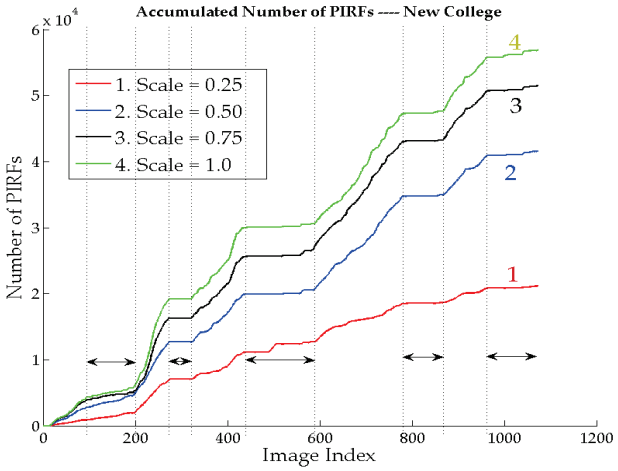
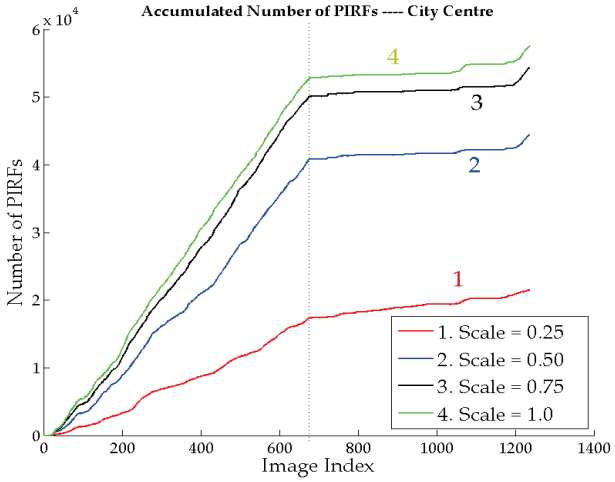


Figure 4.11 Processing times for images for all datasets at image scales 0.25 and 0.5 including SIFT extraction. All programs were written in MATLAB 7.6.0.

(a)



(b)



(c)

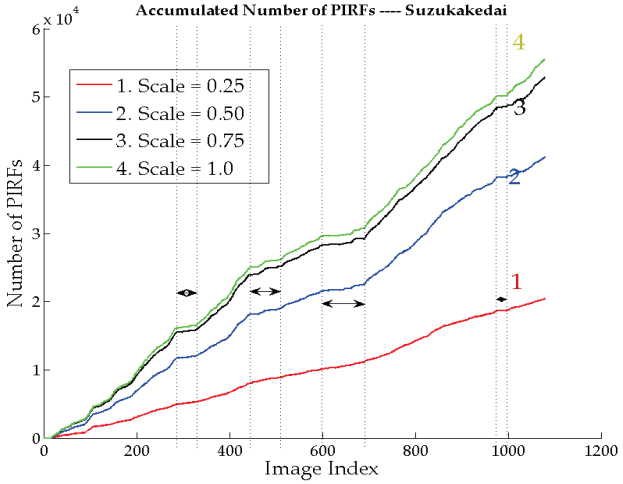
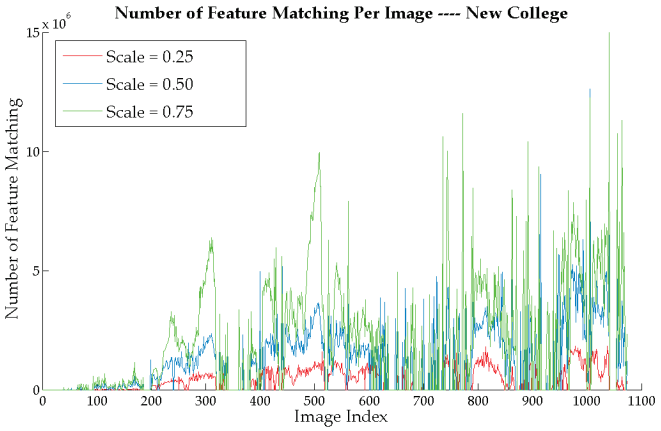
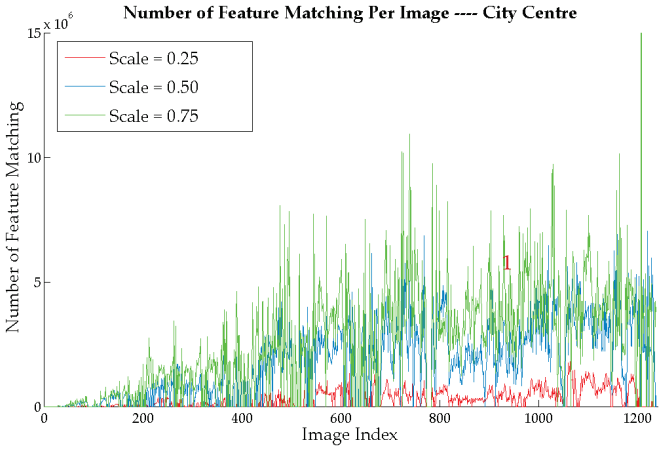


Figure 4.12 Accumulated number of PIRFs: **(a)** New College, **(b)** City Centre, **(c)** Suzukakedai Campus. **(a)–(c)** graphs are plotted for every image scale of 0.25, 0.5, 0.75 and 1.0, with $\tau_2 = 3$. The dotted line, plotted by hand, roughly shows loop-closure images. The number of PIRFs during loop-closure detection becomes more stable.

(a)



(b)



(c)

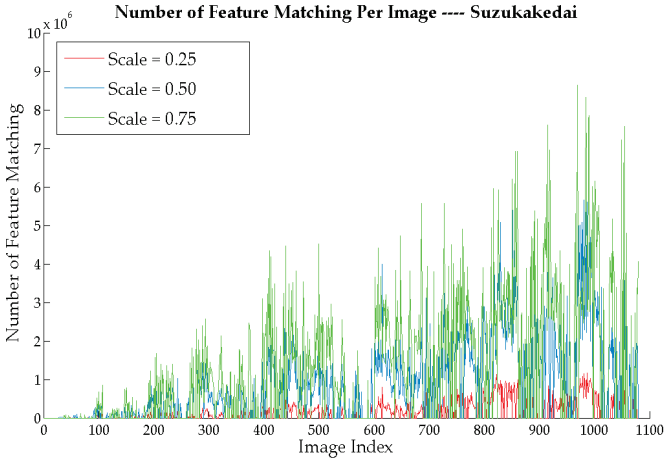
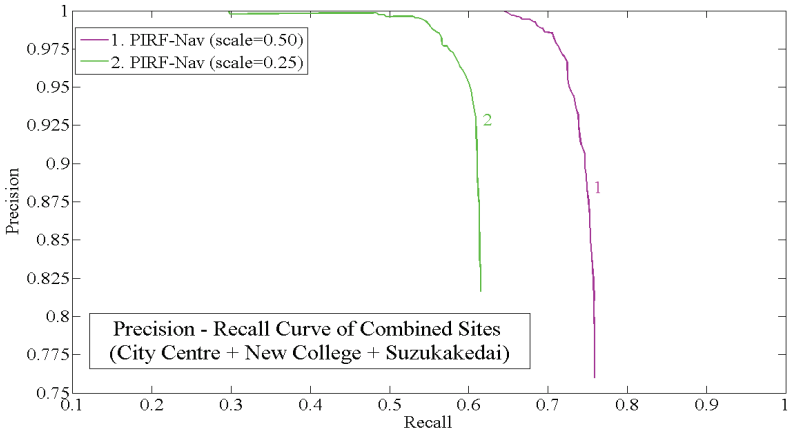


Figure 4.13 Number of feature matchings necessary for loop-closure detection of every dataset at $\tau_2 = 3$: **(a)** New College **(b)** City Centre **(c)** Suzukakedai Campus. **(a)–(c)** graphs are plotted for image scales of 0.25, 0.5, and 0.75. For each dataset (a)–(c), the averaged number of matching at image scale < 0.5 is approximately equal to that required for matching two single images. The number of matchings can be reduced further by using a scale of 0.25, although the performance is the same as that of FAB-MAP.

(a)



(b)

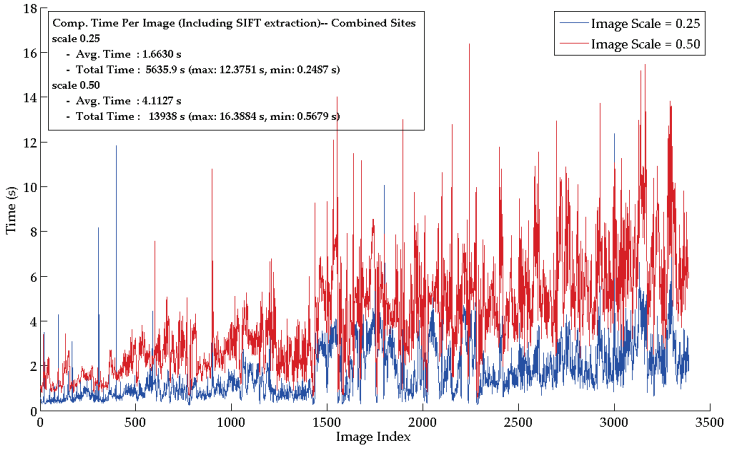
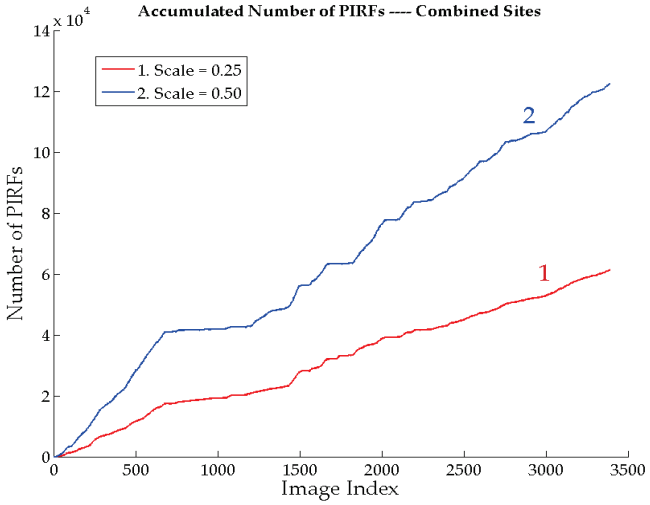


Figure 4.14 (a) Precision-recall curve for combined dataset. (b) Computation time per image for combined dataset including SIFT extraction. All programs were written in MATLAB.

4.5 Discussion

Actually, PIRF-Nav is a completely incremental and online appearance-based method allowing loop-closure detection in real time. The results obtained from experiments on three outdoor datasets––City Centre, New College, and Suzukakedai––confirm that PIRF-Nav outperforms the current BoW method in terms of accuracy. The implementation of multi-steps loop-closure detection/non-detection and loop-closure acceptance/rejection enable the system to close the loop without the need to calculate the full

(a)



(b)

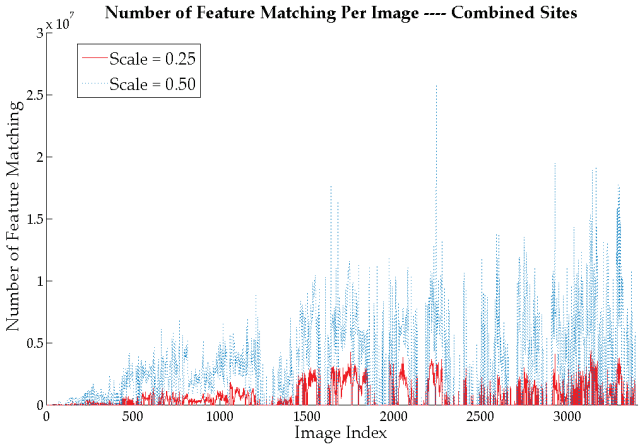


Figure 4.15 **(a)** Accumulated number of PIRFs for combined dataset. **(b)** Number of feature matchings necessary for loop-closure detection for combined dataset. All graphs are plotted only for image scale 0.25 and 0.5 because they offer real-time performance.

posteriori. Using PIRF as the main visual feature is also a success. As distinct from a BoW scheme, PIRF selects the most distinctive local feature instead of clustering all of them into the visual word, which enables the PIRF-Nav to cope efficiently with dynamic changes of scene. Furthermore, the use of PIRF can handle the problem of strong perceptual aliasing efficiently by retaining the most unique feature of the places and use them as the signature of the place. The scoring method based on tf-idf enables us to calculate the similarity score for loop-closure detection properly. The test of Suzukakedai Campus shows that PIRF-Nav is also applicable for use with omnidirectional images. In contrast, the BoW-based method encodes an entire scene into a set of

unordered visual words. In some cases of dynamic changes, BoW also includes many noisy words in the appearance model. This inclusion of words would be problematic if the scene changed markedly. We believe that this might be one reason why FAB-MAP obtains lower accuracy of the City Centre than at the New College.

Another advantage of PIRF-Nav is the ability of incremental loop-closure detection as in the method of Angeli et al. [65]. The combined dataset experiment illustrates clearly that PIRF-Nav can run in many environments without confusion, although FAB-MAP still lays its disadvantage on a preliminarily well-generated dictionary.

Despite its success, our PIRF-Nav leaves great room for improvement. First, its computation time of PIRF-Nav can be speeded up further, which might be done in one of two ways. The first way is implementation of the whole system with more appropriate language instead of MATLAB, which is expected to speed up the time slightly. The second means is to structure the PIRFs. As described in this paper, PIRFs for each model are simply stores as the sequential data in the memory. Every mode of image retrieval requires $O(n^2)$. If the PIRF has been stored in some efficient way, i.e., tree-like BoW, its retrieval time could be hastened markedly. In addition, application of some other method of rapid image matching could decrease the time. For each image, PIRF-Nav constantly requires time that is equivalent to that required for double image matching. Reducing the time for image retrieval and image matching for PIRF-extraction can speed up the time greatly. We plan to implement this in the future to make PIRF-Nav applicable to car navigation similarly to FAB-MAP 2.0.

For a slight but simple improvement, PIRF-Nav can be combined with some post-processing techniques such as epipolar geometry (Nister 2004) to increase its performance, as done by Angeli et al. (2008) and Cummins and Newman (2009). Using multiple-view geometry algorithm, the threshold can be decreased. Some wrong matches that gain a probability more than threshold would be rejected by this algorithm. The PIRF-Nav can drastically increases the recall rate by sacrificing a small amount of precision (as depicted in Fig. 4.12). Therefore, a lower threshold can improve the PIRF-Nav result considerably. In addition, accessing to the motion models is expected to provide the system a good location priori for use in re-localization based on Bayes' filtering.

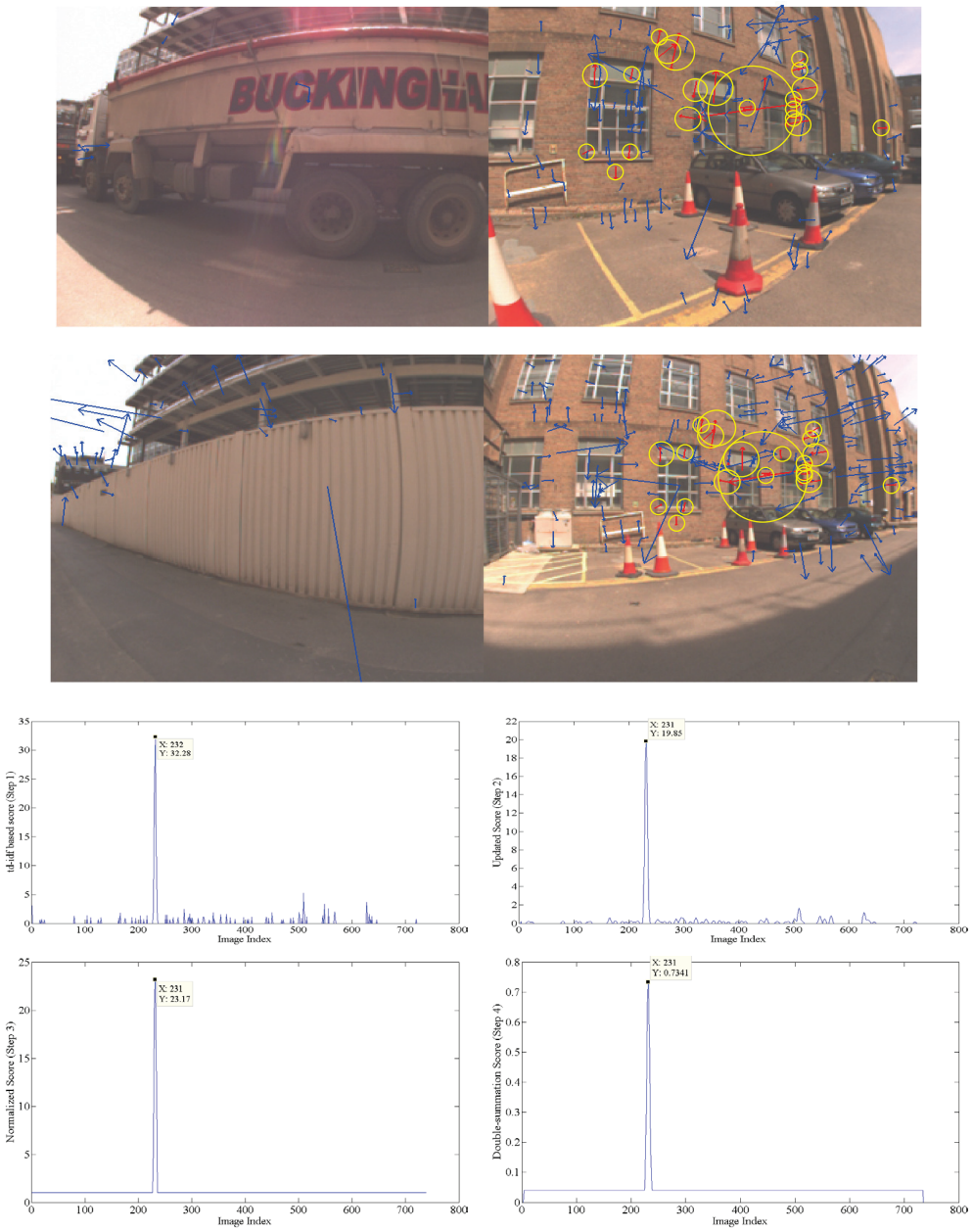


Figure 4.16 Some examples of images detected correctly as loop closure locations despite scene changes. Blue arrows show PIRFs extracted from the input image and query image. The red arrows are matched PIRFs between two images. The images taken from City Centre failed localization by FAB-MAP. Four graphs show the resulting calculated score from each step. PIRFs did not capture the truck because it realized that the truck is a nearby object that is unlikely to be present for a long time.

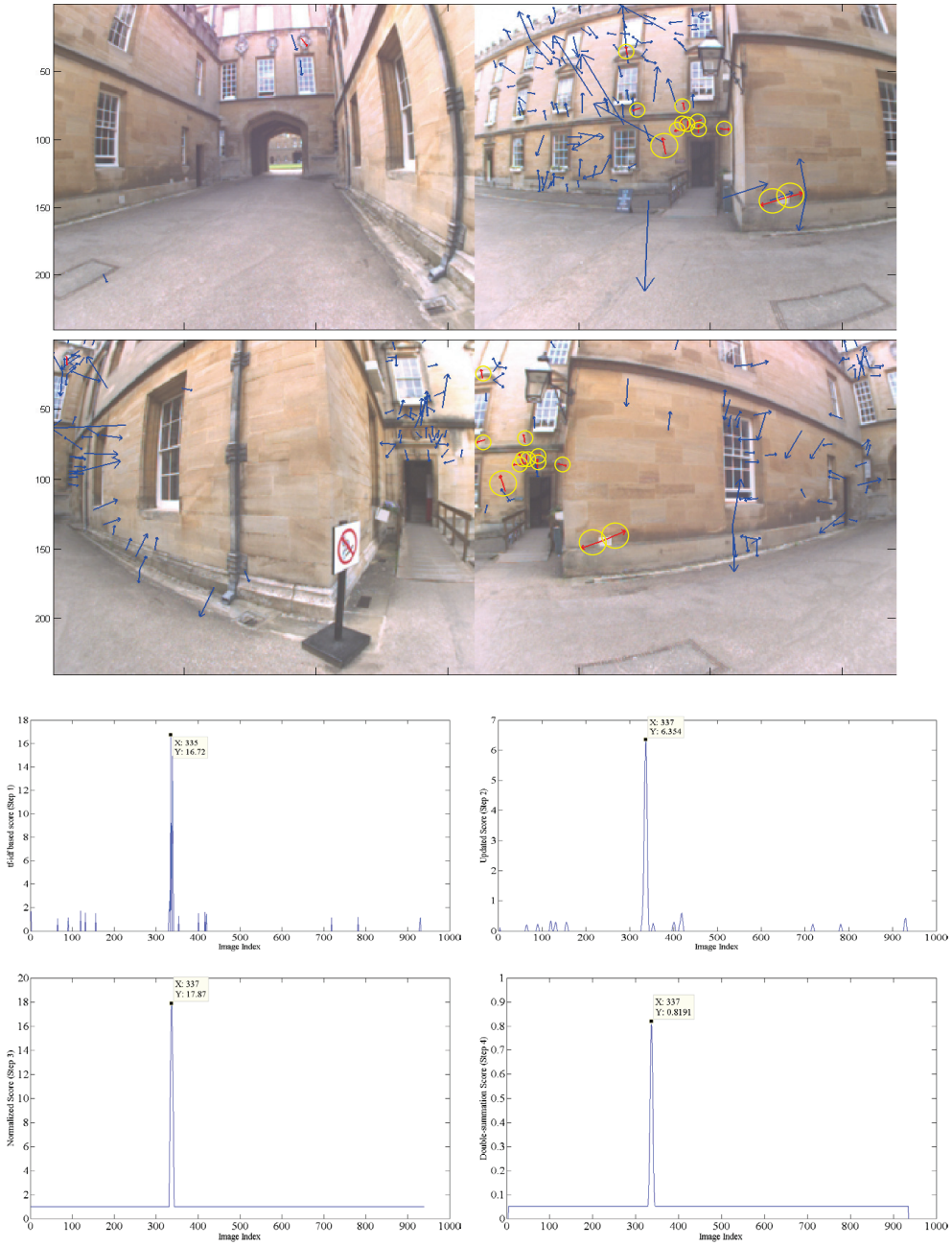


Figure 4.17 Some examples of images detected correctly as loop closure locations despite significant observer's position changes. Blue arrows show PIRFs extracted from the input image and query image. The red arrows are matched PIRFs between two images. The images taken from City Centre failed localization by FAB-MAP. Four graphs show the resulting calculated score from each step. Loop can be closed in spite of a much different viewpoint. PIRFs mostly captures only landmarks that seem to appear in any scenes taken from various viewpoints.

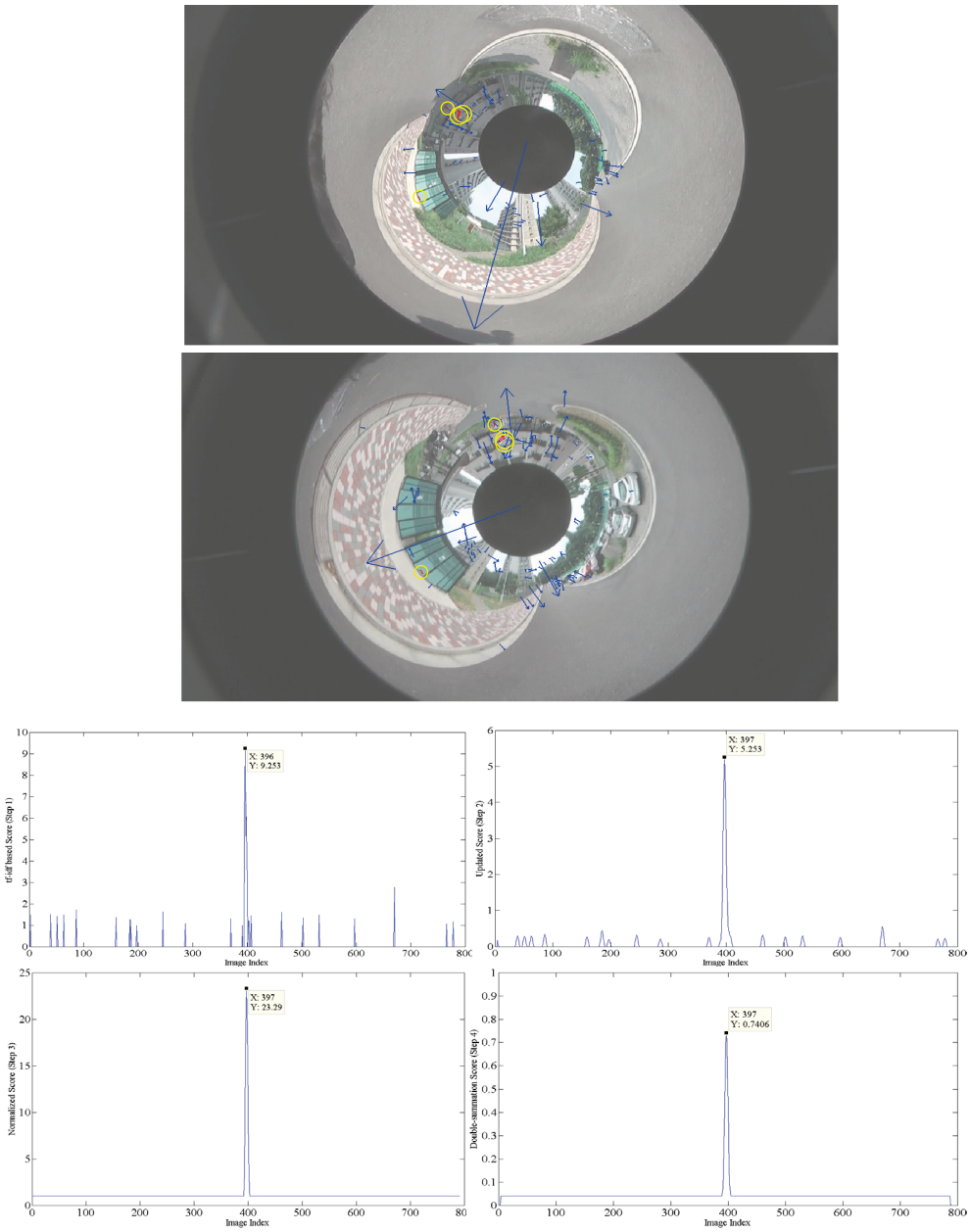


Figure 4.18 Sample images from Suzukakedai that were detected correctly as the same place. The input image is collected on weekdays with full parking lots (right); although the query image (left) was collected on holidays with empty parking lots (special events were hold on holidays with crowded people but in other different places). The four graphs display the scores calculated from each step. Major portion of images which belong to road surface have been ignored by PIRF-Nav because it is a common nearby objects that changes overtime. Discarding these unreliable image's component can prevent the occurrence of perceptual aliasing.

Actually, PIRF can also be investigated more extensively. Instead of simple averaging the SIFTs to obtain PIRF, we can more carefully consider the orientation of each bin in the SIFT, which might yield a more compacted PIRF with similar or better performance.

4.6 Chapter Summary

As described herein, we have presented an online incremental appearance-based localization and mapping based using Position Invariant Robust Feature named PIRF-Nav. The proposed system outperforms the state-of-the-art BoW based method, FAB-MAP, in terms of accuracy; it offers about a two times higher recall rate at 100% precision. The system requires no offline processing. Testing using a combined dataset proves that PIRF-Nav can offer such a capability. All performance has been evaluated by testing using two standard datasets provided by Cummins and Newman (2008a) and our own collected dataset. In fact, PIRF-Nav is based on scoring the feature matching by combining the tf-idf into the simple feature matching to earn a reliable score and then convert such a score into a probability that is useful with discrete Bayes filtering. The method can run incrementally in real time, even for 3000+ images.

CHAPTER 5

SUMMARY OF THE THESIS

In this thesis, we address the problem of online and incremental localization and mapping in dynamic outdoor for robotics. We first describe about the recent work in robotics community and point out that the problem of visual SLAM is very important. In order to solve this, we need to propose two main contributions: PIRF and PIRF-Nav.

For PIRF, its main advantage is its robustness against dynamic changes. Because PIRF captures only objects which is likely to be permanent in place, recognition base on these reliable landmarks thus obtain high rate of accuracy. Also, because the number of PIRF for representing each single image is very small, clustering process like K-Means is not necessary. This enables PIRF to be use for online and incremental SLAM.

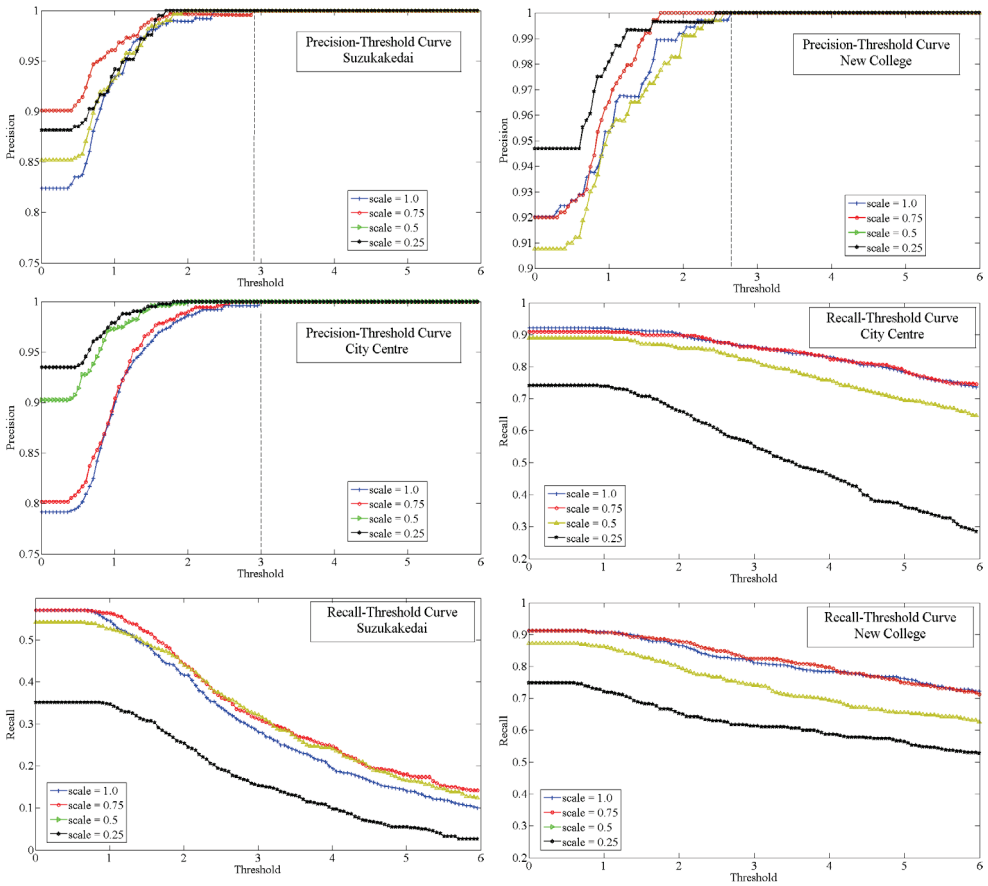
PIRF-Nav is the system proposed to combine the PIRF with the current localization and mapping in robotics. The results apparently show that PIRF is a very good feature choice for the task and also the proposed combination method works well in many environments. With the obtained recall rate of three datasets, we can claim that PIRF-Nav is the most efficient appearance-based localization and mapping in dynamic environments.

PIRF-Nav can still be developed further in aspect of time computation. In this thesis, the PIRF-Nav seems to be applicable to only mobile robots but not the vehicle because the systems requires about 2-3 s for processing one image. Also, even though the size of memory required to store PIRF grows very slowly, it finally reach the limitation of memory. One may solve this by trying to re-constructing the model by doing some clustering process to obtain the more compact size of models.

APPENDIX

Appendix A

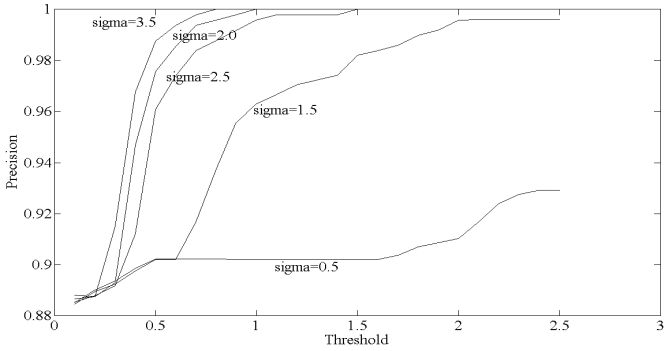
The appendix presents the graphs showing the relationship between threshold values and precision-recall rate for all dataset at every image scale.



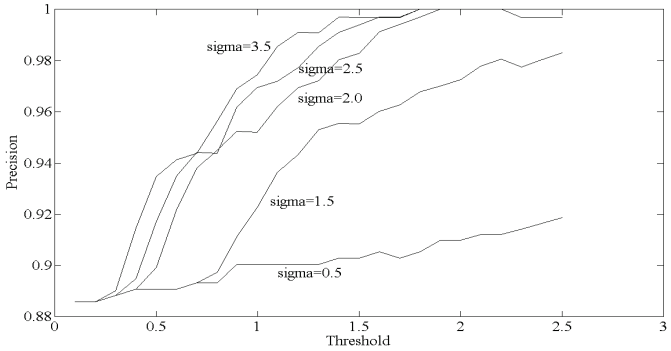
Appendix B

The appendix presents the graphs showing the relationship between threshold values (τ_2), sigma value used for Gaussian distribution in (4.4) and (4.8), and precision rate for each dataset. From the obtained graphs, we can simply use high sigma value in order to obtain precision-1. However, high sigma reduces the recall rate. In this study, sigma value around 2 - 3.5 seems to yield a reasonable good result both for precision and recall. Also, it is noteworthy that sigma value does not much affect the precision; all precisions are greater than 80% at any thresholds for any datasets.

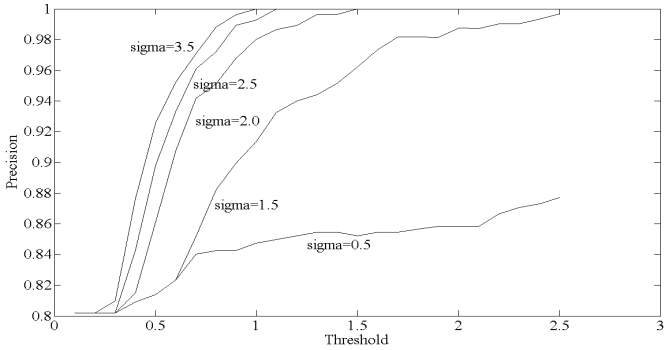
City Centre



New College



Suzukakedai



Publications

Peer-reviewed Journal Articles

[1] Aram Kawewong, Noppharit Tongprasit, Sirinart Tangruamsub and Osamu Hasegawa, “Online and Incremental Appearance-based SLAM in Highly Dynamic Environments, ” *International Journal of Robotics Research (IJRR)*, June, 2010.

[2] Aram Kawewong, Yutaro Honda, Manabu Tsuboyama, and Osamu Hasegawa, "Reasoning on the Self-Organizing Incremental Associative Memory for Online Robot Path Planning, " *IEICE Transaction on Information and Systems*, vol. E93-D, no. 3, pp. 569-582, March, 2010.

[3] 本田雄太郎，Aram Kawewong，坪山学，長谷川修："半教師ありニューラルネットワークによる場所細胞の獲得とロボットの自律移動制御"，信学論 D，2009，採録決定。

[4] Aram Kawewong and Osamu Hasegawa, "Classifying 3D Real-World Texture Images by Combining Maximum Response 8, 4th Order of Auto Correlation and Colortons, " *Journal of Advanced Computational Intelligence and Intelligent Informatics (JACIII)*, vol. 11, no. 5, pp. 511-521, 2007.

Peer-reviewed Conference Articles

[5] Noppharit Tongprasit, Aram Kawewong and Osamu Hasegawa, "A Fast Online Incremental Loop-Closure Detection for Appearance-based SLAM in Dynamic Crowded Environment, " *in Proc. Meeting on Image Recognition and Understanding (MIRU),* 2010.

[6] Noppharit Tongprasit, Aram Kawewong and Osamu Hasegawa, "Data Partitioning Technique for Online and Incremental Visual SLAM, " *in Proceedings of International Conference on Neural Information Processing (ICONIP)*, pp. 769-777, 2009.

[7] Aram Kawewong, Sirinart Tangruamsub and Osamu Hasegawa, "Wide-baseline Visible Features for Highly Dynamic Scene Recognition," *in Proceedings of International Conference on Computer Analysis of Images and Patterns (CAIP)*, pp. 723-731, September, 2009.

[8] Sirinart Tangruamsub, Manabu Tsuboyama, Aram Kawewong and Osamu Hasegawa, "Mobile Robot Vision-Based Navigation Using Self-Organizing and Incremental Neural Networks," *in Proceedings of International Joint Conference on Neural Networks (IJCNN)*, pp. 1103-1110, June, 2009.

[9] Aram Kawewong, Yutaro Honda, Manabu Tsuboyama and Osamu Hasegawa, "Common-Patterns Based Mapping for Robot Navigation, " *in Proceedings of IEEE International Conference on Robotics and Biomimetics (ROBIO)*, pp. 608-614, February 2008.

[10] Aram Kawewong, Yutaro Honda, Manabu Tsuboyama and Osamu Hasegawa, “A Common-Neural-Pattern Based Reasoning for Mobile Robot Cognitive Mapping, “ *in Proceedings of International Conference on Neural Information Processing (ICONIP)*, vol. 1, pp. 32-39, 2008.

[11] Atiwong Suchato, Proadpran Punyabukkana, Prin Mana-aporn, Thanadit Pumprao, and Aram Kawewong, “EngCC: A Thai Language Speech-automated Contact Center, ” *in Proceedings of The Sixth Symposium on Natural Language Processing (SNLP)*, pp. 247-252, 2005.

[12] Aram Kawewong and Osamu Hasegawa, “Combining Rotationally Variant and Invariant Features Based on Between-Class Error for 3D Texture Classification, ” *in Proceedings of IEEE International Conference on Computer Vision (ICCV) Workshop*, pp. 107-112, October 2005.

[13] Aram Kawewong and Osamu Hasegawa, "3D Texture Classification by Using Pre-testing Stage and Reliability Table, " *in Proceedings of IEEE International Conference on Image Processing (ICIP)*, vol. 2, pp. 1330-1333, September, 2005.

Bibliography

[1] R. S. Sutton and A. G. Barto, Reinforcement Learning: An Introduction (Hard Cover), MIT Press, Cambridge, 1998.

[2] A. Arleo, F. Smeraldi and W. Gerstner, "Cognitive Navigation Based on Nonuniform Gabor Space Sampling, Unsupervised Growing Networks, and Reinforcement Learning," *IEEE Trans. Neural Networks*, vol. 15, no. 3, May 2004.

[3] T. Strosslin, D. Sheynikhovich, R. Chavarriaga, W. Gerstner, "Robust Self-Localisation and Navigation Based on Hippocampal Place Cells," *Neural Networks*, vol. 18, no. 9, Nov. 2005.

[4] A. Arleo, F. Smeraldi, S. Hug, and W. Gerstner "Place Cells and Spatial Navigation based on Vision, Path Integration, and Reinforcement Learning," *Proc. Neural Information Processing Systems*, 2001.

[5] G. Z Grudic, V. Kumar and L. Ungar, "Using Policy Gradient Reinforcement Learning on Autonomous Robot Controllers," *Proc. Int'l Conf. Intelligent Robots and Systems*, 2003

[6] H. Igarashi, "Path Planning of a Mobile Robot by Optimization and Reinforcement Learning," *Artif. Life Robotics*, 2002.

[7] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics (Hard Cover)*. Cambridge: MIT Press, 2005.

[8] R. Smith and P. Cheesman, “On the Representation of Spatial Uncertainty, ” *Int’l Jour. Robotics Research*, vol. 5, no. 4, pp. 56-68, 1987.

[9] H. F. Durrant-Whyte, “Uncertain Geometry in Robotics, ” *IEEE Trans. Robotics and Automation*, vol. 4, no. 1, pp. 23-31, 1988.

[10] N. Ayache and O. Faugeras, “Building, Registrating, and Fusing Noisy Visual Maps, ” *Int’l Jour. Robotics Research*, vol. 7, no. 6, pp. 45-65, 1988.

[11] J. Crowley, “World Modeling and Position Estimation for a Mobile Robot Using Ultra-Sonic Ranging, ” in *Proc. IEEE Int’l Conf. Robotics and Automation*, pp. 674-681, 1989.

[12] R. Chalita and J. P. Laumond, “Position Referencing and Consistent World Modeling for Mobile Robots, ” in *Proc. IEEE Int’l Conf. Robotics and Automation,* pp. 138-143, 1985.

[13] R. Smith, M. Self and P. Cheeseman, “Estimating Uncertain Spatial Relationships in Robotics, ” in *Autonomous Robot Vehicles*, Eds. New York: Springer-Verlag, pp. 167-193, 1990.

[14] J. J. Leonard and H. F. Durrant-Whyte ,”Simultaneous Map Building and Localisation for an Autonomous Mobile Robot, ” in *Proc. IEEE Int’l Workshop on Intelligent Robots and Systems (IROS)*, pp. 1442-1447, 1991.

[15] J. J. Leonard and H. F. Durrant-Whyte, *Directed Sonar Navigation*, Norwell, MA: Kluwer, 1992.

[16] W. D. Renken, “Concurrent Localization and Map Building for Mobile Robots Using Ultrasonic Sensors, ” in *Proc. IEEE Int’l Workshop Intelligent Robots and Systems (IROS)*, 1993.

[17] H. Durrant-Whyte, D. Rye and E. Nebot, “Localisation for Automatic Guided Verhicles, ” in *Int’l Symposium on Robotics Research (ISRR)*, pp. 613-625, 1996.

[18] M. Csorba and H. F. Durrent-Whyte, "A New Approach to Simultanoeus Localisation and Map Building, " in *Proc. SPIE*, 1996.

[19] J. J. Leonard and H. J. S. Feder, "A Computational Efficient Method for Large-Scale Concurrent Mapping and Localisation, " in *Proc. Int'l Symposium on Robotics Research (ISRR),* pp. 169-176, 2000.

[20] J. A. Castellanos, J. D. Tardos and G. Schmidt, "Building a Global Map of the Environment of a Mobile Robot: The Importance of Correlations, " in *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA)*, pp. 1053-1059, 1997.

[21] G. Guivant, E. M. Nebot and S. Baiker, "Localization and Map Building Using Laser Range Sensors in Outdoor Applications, " *Jour. Robotics Research,* vol. 17, no. 10, pp. 565-583, 2000.

[22] S. B. Williams, P. Newman, G. Dissayanake, and H. F. Durrant-Whyte, "Autonomous Underwater Simultaneous Localisation and Map Building, " in *Proc. IEEE Int'l Conf. Robotics and Automation*, pp. 1793-1798, 2000.

[23] K. S. Chong and L. Kleeman, "Feature-based Mapping in Real, Large Scale Environments Using an Ultrasonic Array, " *Int'l Journal of Robotics Research*, vol. 18, no. 1, pp. 3-19, 1999.

[24] M. Deans and M. Hebert, "Experimental Comparison of Techniques for Localization and Mapping Using a Bearing-only Sensor, " in *Proc. Int'l Symposium Experimental Robotics (ISER)*, pp. 395-404, 2000.

[25] B. Kuipers and Y. T. Byan, "A Robot Exploration and Mapping Strategy based on a Semantic Hierarchy of Spatial Representations, " *Jour. Robotics and Autonomous Systems*, vol. 8, pp. 47-63, 1991.

[26] H. Choset and K. Nagatani, "Topological Simultaneous Localization and Mapping (SLAM): Toward Exact Localization without Explicit Localization, " *IEEE Trans. Robotics and Automation*, vol. 17, no. 2, pp. 125-137, 2001.

[27] C. O. Dunlaing and C. K. Yap, “A ‘Retraction’ Method for Planning the Motion of a Disc, ” *Algorithmica,* vol. 6, pp. 104-111, 1985.

[28] H. Choset and J. Burdick, “Sensor based Motion Planning: The Hierarchical Generalized Voronoi Graph, ” *Int’l Jour. Robotics Research*, vol. 19, no. 2, pp. 96-125, 2000.

[29] C. M. Smith and J. J. Leonard, “A Multiple-hypothesis Approach to Concurrent Mapping and Localization for Autonomous Underwater Vehicles, ” in *Proc. Int’l Conf. Field and Service Robotics*, 1997.

[30] J. Modayil, P. Beeson and B. Kuipers, “Using the Topological Skeleton for Scalable Global Metrical Map-Building,” *Proc. Int'l Conf. Intelligent Robots and Systems*, 2004

[31] B. Kuipers et al., “Local Metrical and Global Topological Maps in the Hybrid Spatial Semantic Hierarchy,” *Proc. Int'l Conf. Robotics and Automation*, 2004.

[32] J. Buhmann, W. Burgard, A. B. Cremers, D. Fox, T. Hofmann, F. Schneider, J. Strikos and S. Thrun, “The Mobile Rhino, ” in *Artificial Intelligence Magazine*, vol. 16, no. 1, 1995.

[33] H. J. Chang et al., “P-SLAM: Simultaneous Localization and Mapping With Environmental-Structure Prediction,” *IEEE. Trans. Robotics*, vol. 23, no. 2, Apr. 2007.

[34] J. Blanco et al., “Toward a Unified Bayesian Approach to Hybrid Metric-Topological SLAM,” *IEEE. Trans. Robotics*, vol. 24, no. 2, Apr. 2008.

[35] M. Cummins, and P. Newman, "FAB-MAP: Probabilistic Localization and Mapping in the Space of Appearance," *Int'l Journal of Robotics Research*, vol.27, pp. 647-665, 2008.

[36] C. Valgren, and A. Lilienthal, "Incremental Spectral Clustering and Seasons: Appearance-Based Localization in Outdoor Environments" *Proc. IEEE Int'l Conf. Robotics and Automation*, 2008.

[37] T. Goedeme, et al., "Omnidirectional Vision Based Topological Navigation," *Int'l Jour. of Computer Vision*, vol. 74, no.3, pp. 219-236, 2007.

[38] N. Dalal and B. Triggs, "Histogram of Oriented Gradients for Human Detection, " in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition (CVPR)*, 2005.

[39] A. Oliva and A. Torralba, "Modeling the Shape of Scene: A Holistic Representation of the Spatial Envelope," *Int'l Jour. Computer Vision*, vol. 42, no. 3, pp. 145-175, 2001.

[40] J. Wu and J. M. Rehg, "Where am I: Place instance and category recognition using spatial PACT," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition,* 2008.

[41] D. G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l Jour. Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[42] H. Bay, T. Tuytelaars and L. V. Gool, "SURF: Speeded Up Robust Features," *Proc. European Conf. Computer Vision*, 2006.

[43] A. Friedman, "Framing Pictures: The Role of Knowledge in Automatized Encoding and Memory for Gist, " *Jour. Experimental Psychology: General*, vol. 108, pp. 316-355, 1979.

[44] M. C. Potter, "Short-term Conceptual Memory for Pictures, " *Jour. Experimental Psychology: Human Learning and Memory*, vol. 2, pp. 509-522, 1976.

[45] A. Pronobis, B. Caputo, P. Jensfelt and H. I. Christensen, "A Discriminative Approach to Robust Visual Place Recognition," *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems*, 2006.

[46] S. Lazebnik, C. Schmid and J. Ponce, "Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2006.

[47] K. R. Castleman, *Digital Image Processing*, Prentice Hall, Englewood Cliffs, NJ, USA, 1996.

[48] D. G. Lowe, "Object Recognition from Local Scale-Invariant Features, " in *Proc. IEEE Int'l Conf. Computer Vision (ICCV)*, pp. 1150-1157, 1999.

[49] T. Lindeberg, "Scale-space Theory: A Basic Tool for Analysing Structures at Different Scales, " *Jour. Applied Statistics*, vol. 21, pp. 224-270, 1994.

[50] M. M. Ullah, A. Pronobis, B. Caputo, J. Luo, P. Jensfelt and H.I. Christensen, "Towards Robust Place Recognition for Robot Localization, " *Proc. IEEE Int'l Conf. Robotics and Automation,* pp. 530-537, 2008.

[51] H. Durrant-Whyte and T. Bailey, "Simultaneous Localization and Mapping: Part I," *IEEE Robotics & Automation Magazine*, vol. 13, no. 2, pp. 99-110, 2006.

[52] T. Bailey and H. Durrant-Whyte, "Simultaneous Localization and Mapping (SLAM): Part II," *IEEE Robotics & Automation Magazine*, vol. 13, no. 3, pp. 108-117, 2006.

[53] O. Martinez Mozos, C. Stachniss, and W. Burgard, "Supervised Learning of Places from Range Data using adaboost," *Proc. IEEE Int'l Conf. Robotics and Automation*, 2005.

[54] Y. Abe, M. Shikano, T. Fukuda and F. Arai, “Vision Based Navigation System for Autonomous Mobile Robot with Global Matching,” *Proc. IEEE Int’l Conf. Robotics and Automation*, pp. 1299-1304, 1999.

[55] S. Thrun, “Finding Landmarks for Mobile Robot Navigation,” *Proc. IEEE Int’l Conf. Robotics and Automation*, pp.958-963, 1998.

[56] S. Maeyama, A. Ohya, and S. Yuta, “Long Distance Outdoor Navigation of an Autonomous Mobile Robot by Playback of Perceived Route Map,” *Proc. Int’l Symp. Experimental Robotics,* pp. 185-194, 1997.

[57] J. Shi and J. Malik, “Normalized Cuts and Image Segmentation,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 22, no. 8, pp. 888-905, 2000.

[58] X. Ren and J. Malik, “Learning a Classification Models for Segmentation,” *Proc. IEEE Int’l Conf. Computer Vision*, 2003.

[59] L. Renniger and J. Malik, “When Is Scene Identification Just Texture Recognition?” *Vision Research*, vol. 44, pp. 2301-2311, 2004.

[60] I. Ulrich and I. Nourbakhsh, “Appearance-Based Place Recognition for Topological Localization,” *Proc. IEEE Int’l Conf. Robotics and Automation*, pp. 1023-1029, 2000.

[61] A. Torralba, K. P. Murphy, W. T. Freeman and M. A. Rubin, “Context-Based Vision System for Place and Object Recognition,” *Proc. IEEE Int’l Conf. Computer Vision*, pp. 1023-1029, 2003.

[62] A. Kawewong, S. Tangruamsub and O. Hasegawa, “Wide-baseline Visible Features for Highly Dynamic Scene Recognition, ” in *Proc. Int’l Conf. Computer Analysis of Images and Patterns (CAIP)*, 2009.

[63] C. Valgren, T. Duckett and A. Lilienthal, "Incremental Spectral Clustering and Its Application to Topological Mapping," *Proc. IEEE Int'l Conf. Robotics and Automation*, 2007.

[64] D. Filliat, "A visual bag of words method for interactive qualitative localization and mapping," *Proc. IEEE Int'l Conf. Robotics and Automation,* pp. 3921-3926, 2007.

[65] A. Angeli, D. Filliat, S. Doncieux and J. Meyer, "Fast and Incremental Method for Loop-Closure Detection Using Bags of Visual Words," *IEEE Trans. Robotics*, vol. 24, no. 5, pp. 1027-1037, 2008.

[66] K. Mikolajczyk and C. Schmid, "A Performance Evaluation of Local Descriptors," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 27, no. 10, pp. 1615-1630, 2005.

[67] L. Fei-Fei and P. Perona, "A Bayesian Hierarchical Model for Learning Natural Scene Categories," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, 2005.

[68] J. J. Kivinen, E. B. Sudderth and M. I. Jordan, "Learning Multiscale Representation of Natural Scenes Using Dirichlet Processes," *Proc. IEEE Int'l Conf. Computer Vision,* 2007.

[69] J. Kosecka and F. Li, "Vision Based Topological Markov Localization," *Proc. IEEE Int'l Conf. Robotics and Automation,* 2004.

[70] L. Ledwich and S. Williams, "Reduced SIFT Features For Image Retrieval and Indoor Localisation," *Proc. Aust. Conf. Robotics and Automation*, 2004.

[71] A. C. Murillo, C. Sagues, J. J. Guerero, T. Goedeme, T. Tuytelaars and L. V. Gool, "From Omnidirectional Images to Hierarchical Localization," *Robotics and Autonomous Systems*, vol. 55, no. 5, pp. 372-382, 2007.

[72] E. Royer, M. Lhuillier, M. Dhome and J. M. Lavest, “Monocular Vision for Mobile Robot Localization and Autonomous Navigation,” *Int’l Jour. Computer Vision*, vol. 74, no. 3, pp. 237-260, 2007.

[73] A. Tapus and R. Siegwart, “Incremental Robot Mapping with Fingerprints of Places,” *Proc. IEEE Int’l Conf. Intelligent Robots and Systems,* 2005.

[74] L. A. Clemente, A. J. Davison, I. D. Reid, J. Neira and J. D. Tardos, “Mapping Large Loops with a Single Hand-Held Camera,” *Proc. Robotics: Sciences and Systems*, 2007.

[75] H. Andreasson, T. Deckett and A. J. Lilienthal, “A Minimalistic Approach to Appearance-Based Visual SLAM,” *IEEE. Trans. Robotics,* vol. 24, no. 5, pp. 991-1001, 2008.

[76] P. Newman, D. Cole and K. Ho, “Outdoor SLAM using Visual Appearance and Laser Ranging,” *Proc. IEEE Int’l Conf. Robotics and Automation*, pp. 1180-1187, 2006.

[77] J. Civera, A. J. Davison, and J. M. Martinez Montiel, “Inverse Depth Parameterization for Monocular SLAM,” *IEEE Trans. Robotics*, vol. 24, no. 5, pp. 932-945, 2008.

[78] J. Luo, A. Pronobis, B. Caputo and P. Jensfelt, “Incremental Learning for Place Recognition in Dynamic Environments,” *Proc. IEEE Int’l Conf. Intelligent Robots and Systems,* 2007.

[79] J. Kittler, et al., “On Combining Classifiers,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 20, no. 3, pp. 226-239, 1998.

[80] C. Tan, T. Hong, T. Chang and M. Shneier, “Color Model-Based Real-Time Learning for Road Following,” *Proc. Int’l IEEE Intelligent Transportation Systems Conf.*, 2006.

[81] B. Kuipers, “The Spatial Semantic Hierarchy,” *Artificial Intelligence*, vol. 119, no. 1-2, pp. 191-233, 2000.

[82] D. Filliat and J. A. Meyer, “Map-based Navigation in Mobile Robots—I. A Review of Localisation Strategies, “ *Cognitive Systems Research*, vol. 4, no. 4, pp. 243-282, 2003.

[83] J. A. Meyer and D. Filliat, “Map-based Navigation in Mobile Robots—III. A Review of Map-learning and Path-planning Strategies, “ *Cognitive Systems Research*, vol. 4, no. 4, pp. 283-317, 2003.

[84] P. Newman, D. Cole and K. Ho, “Outdoor SLAM Using Visual Appearance and Laser Ranging, “ in *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA)*, 2006.

[85] J. Wang, H. Zha and R. Cipolla, “Coarse-to-fine Vision-based Localization by Indexing Scale-Invariant Features, “ *IEEE Trans. System, Man and Cybernetics, vol.* 36, no. 2, pp. 413-422, 2006.

[86] K. Ho and P. Newman, “Detecting Loop Closure with Scene Sequences, ” *Int’l Jour. Computer Vision (IJCV)*, vol. 74, no. 3, pp. 261-286, 2007.

[87] G. Schindler, M. Brown and R. Szeliski, “City-scale Location Recognition, ” in *Proc. IEEE Int’l Conf. Computer Vision and Pattern Recognition (CVPR)*, 2007.

[88] A. Levin and R. Szeliski, “Visual Odometry and Map Correlation, ” in *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition (CVPR)*, 2004.

[89] C. Silpa-Anan and R. Hartley, “Localisation Using an Image-map, ” in *Proc. Aust. Conf. Robotics and Automation*, 2004.

[90] Z. Zivkovic, B. Bakker and B. Krose, “Hierarchical Map Building Using Visual Landmarks and Geometric Constraints, ” in *Proc. IEEE/RSJ Int'l Conf. Intelligent Robots and Systems (IROS)*, 2005.

[91] P. Lamon, I. Nourbakhsh, B. Jensen and R. Siegwart, “Deriving and Matching Image Fingerprint Sequences for Mobile Robot Localization, ” in *Proc. IEEE Int’l Conf. Robotics and Automation (ICRA)*, 2001.

[92] B. J. A. Krose, N. A. Vlassis, R. Bunschoten and Y. Motomura, “A Probabilistic Model for Appearance-based Robot Localization, ” *Image and Vision Computing*, vol. 19, no. 6, pp. 381-391, 2001.

[93] M. Bowling, D. Wilkinson, A. Ghodsi and A. Milstein, “Subjective Localization with Action Respecting Embedding, ” in *Proc. Int’l Sym. Robotics Research (ISRR)*, 2005.

[94] J. Wolf, W. Burgard and H. Burkhardt, “Robust Vision-based Localization by Combining an Image-retrieval System with Monte Carlo Localization, ” *IEEE Trans. Robotics*, vol. 21, no. 2, pp. 208-216, 2005.

[95] J. Kosecka, F. Li and X. Yang, “Global Localization and Relative Positioning based on Scale-invariant Keypoints, ” *Robotics and Autonomous Systems*, vol. 52, no. 1, pp. 27-38, 2005.

[96] J. Sivic and A. Zisserman, “Video Google: A Text Retrieval Approach to Object Matching in Videos, ” in *Proc. IEEE Int’l Conf. Computer Vision (ICCV)*, 2003.

[97] D. Nister and H. Stewenius, “Scalable Recognition with a Vocabulary Tree, ” in *Proc. IEEE Int’l Conf. Computer Vision and Pattern Recognition (CVPR)*, 2006.

[98] C. Chen and H. Wang, “Appearance-based Topological Bayesian Inference for Loop-closing Detection in a Cross Country Environment, ” *Int’l Jour. Robotics Research*, vol. 25, no. 10, pp. 953-983, 2006.

[99] F. Li and J. Kosecka, “Probabilistic Location Recognition Using Reduced Feature Set, ” in *Proc. Int’l Conf. Robotics and Automation (ICRA)*, 2006.

[100] S. Se, D. Lowe and J. Little, "Vision-based Mobile Robot Localization and Mapping Using Scale-Invariant Features, " in *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA)*, 2001.

[101] M. Cummins and P. Newman, "Highly Scalable Appearance-Only SLAM – FAB-MAP 2.0, " in *Proc. Robotics: Sciences and Systems (RSS)*, 2009.

[102] M. Cummins and P. Newman, "Accelerated Appearance-Only SLAM, " in *Proc. IEEE Int'l Conf. Robotics and Automation (ICRA)*, 2008.

[103] S. Thrun and A. Bücken, "Integrating Grid-based and Topological Maps for Mobile Robot Navigation, " in *Proc. The 13th National Conf. Artificial Intelligence (AAAI)*, 1996.