Online and Incremental Appearance-based SLAM in Highly Dynamic Environments

Aram Kawewong, Noppharit Tongprasit, Sirinart Tungruamsub, Osamu Hasegawa

4259-R2-52 Nagatsuta, Midori-ku, Yokohama, 226-8503, JAPAN
Department of Computational Intelligence and Systems Science
Tokyo Institute of Technology
{kawewong.a.aa, tongprasis.n.aa, tangruamsub.s.aa, hasegawa.o.aa}@m.titech.ac.jp

Abstract

This paper presents a novel method for online and incremental appearance-based localization and mapping in a highly dynamic environment. Using position-invariant robust features (PIRFs), the method can achieve a high rate of recall with 100% precision. It can handle both strong perceptual aliasing and dynamic changes of places efficiently. Its performance also extends beyond conventional images; it is applicable to omnidirectional images for which the major portions of scenes are similar for most places. The proposed PIRF-based Navigation method named PIRF-Nav is evaluated by testing it on two standard datasets as is in FAB-MAP and on an additional omnidirectional image dataset that we collected. This extra dataset is collected on two days with different specific events, i.e., an open-campus event, to present challenges related to illumination variance and strong dynamic changes, and to test assessment of dynamic scene changes. Results show that PIRF-Nav outperforms FAB-MAP; PIRF-Nav at precision-1 yields a recall rate about two times (approximately 80%) higher than that of FAB-MAP. Its computation time is sufficiently short for real-time applications. The method is fully incremental, and requires no offline process for dictionary creation. Additional testing using combined datasets proves that PIRF-Nav can function over a long term and can solve the kidnapped robot problem.

1. Introduction

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 (Filliat and Meyer 2003; Meyer and Filliat 2003). 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 (Cummins and Newman 2008a) even with the current Bag-of-Words (BoW) based methods (Newman et al. 2006; Wang et al. 2006; Filliat 2007; Newman et al. 2006; Ho and Newman 2007; Schindler et al. 2007) 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 (Kawewong et al. 2009). PIRF extraction is simple and fast. It is generated by averaging the existing scale-invariant feature transformation (SIFTs) (Lowe 2004), 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 (Levin and Szeliski 2004; Silpa-Anan and Hartley 2004; Zivkovic et al. 2005; Newman et al. 2006). 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 (Cummins and Newman 2008a), has localization under a strong perceptual aliasing condition. The second dataset, City Centre (Cummins and Newman 2008a), 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 (Cummins and Newman 2008a). 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.

2. 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 (Ulrich and Nourbakhsh 2003; Lamon et al. 2001; Torralba et al. 2003; Krose et al. 2001). Most such approaches using global features require much effort in the supervised training phase to learn the generative place models. Later, Bowling et al. (2005) 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 (Lowe 2004) or speeded up robust features (SURFs) (Bay et al. 2006) as the main visual feature for the localization and mapping system. Wolf et al. (2005) used an image retrieval system based on invariant features as the basis of a Monte Carlo localization scheme. Kosecka et al. (2005) 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 (Sivic and Zisserman 2003; Nister and Stewenius 2006) 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. (2006) use geometric information in a post-verification step to confirm putative matches. Schindler et al. (2007) 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 (2006) 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. (2007) 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 (2008a) 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. (2008) described the incremental creation of a visual vocabulary. The method is still based on the popular BoW scheme, as explained in an earlier study (Cummins and Newman 2008a). 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 (Angeli et al. 2008), 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 (2006) 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. (2001) 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 Fig. 11). 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 (Angeli et al. 2008; Cummins and Newman 2008a) and hence is applicable to real robots. We utilized two datasets used earlier (Cummins and Newman 2008a)—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.

### 3. Position-Invariant Robust Feature

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. Fig. 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 \( \psi \), where \( \psi = (d_{t-2}(d_t) + d_{t-1}(d_t) + d_t) / 3 \), in which \( d_x(d_y) \) denotes the descriptor in image \( I_x \) and matches descriptor \( d_y \) under the distant threshold \( \theta \). 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 \( I_t \) and \( d_1 \) and \( d_2 \) as the first and second nearest descriptors of \( d \) found in \( I_t \), \( d_1 \) would be considered as
the matched descriptor of \(d\), if and only if \(\frac{\cos^{-1}(d_1 \cdot d_2)}{\cos^{-1}(d_2)} < \theta\). 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.

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 \(\theta\), 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 \(\theta\) 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 \(n_{pirf,max}\). If \(n_{pirf} < n_{pirf,min}\), then \(w_{size}\) is decreased by 1 (i.e., reset to 2 if the default size is 3), and the PIRFs are re-extracted. In contrast, if \(n_{pirf} > n_{pirf,max}\), 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., \(n_{pirf} = 0\). 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

![Fig. 1. Extracted PIRF with threshold \(\theta = 0.5\). (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.](image-url)
of PIRFs. Despite their simplicity, PIRFs are robust, especially against highly dynamic changes in scenes (Kawewong et al. 2009), while being sufficiently tolerant to perceptual aliasing.

To extract PIRF, an important parameter $w_{\text{size}}$ 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 $w_{\text{size}}$. For example, consider the case in which the camera runs at 200 fps vs 0.5 fps. At 200 fps, $w_{\text{size}}$ would be too small and several redundant PIRFs are obtained. For this case, $w_{\text{size}}$ 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 $w_{\text{size}}$ might be reduced to its smallest value $w_{\text{size}} = 2$. Note that we have started trying PIRF with the 1000 km dataset of Cummins and Newman (2009), 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 $w_{\text{size}}$ 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. Incremental Appearance-based SLAM

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.

4.1 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 $w_{\text{size}}$ 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 $(I_{t-1} - I_t)$. 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.

4.2 Localization and Mapping

At time $t$, our map of the environment is a collection of $n$ discrete and disjoint locations $L = \{L_1, \ldots, L_n\}$. Each of these locations $L_i$, which was created from the past image $I_i$, has an associated appearance model $M_i$. This model $M_i$ is a set of PIRFs. Modified from naive Bayes, four main steps are used to calculate the similarity score for loop-closure detection. Fig. 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 $M_0$. In Steps 2–3, the score is recalculated considering the score of neighboring models. The normalized score $\beta$ is used to decide if the detected 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.

**Step 1: Simple Feature Matching**

First, the current observed model $M_t$ is compared to each of the mapped models $\{M_0, \ldots, M_{nt}\}$ using standard feature matching with distant threshold $\theta$. 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 $M_i$ for $1 \leq i \leq nt$. 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. (2008). 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, $M_t$ 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 (Sivic and Zisserman 2003):

\[
tf - idf = \frac{n_{wi}}{n_i} \log \frac{N}{n_w}
\]

In this equation, \( n_{wi} \) is the number of occurrences of the visual word \( w \) in \( M_i \), \( n_i \) is the number of visual words in \( M_i \), \( n_w \) 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 (Sivic and Zisserman 2003), 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. (2008) solves this by taking an advantage of the inverted index associated with the dictionary. At each current image \( I_t \), 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 \( I_t \) is found, the statistical score in equation (1) is added to the score of each image \( I_i \). That is, at each observed current image \( I_t \), the similarity score between \( I_t \) and previous image \( I_i \) is the summation of tf-idf of all visual words that appear in both \( I_t \) and \( I_i \):

\[
s_t = \sum_{w=1}^{n_{wi}} \frac{n_{wi}}{n_i} \log \left( \frac{N}{n_w} \right)
\]

where \( n_{wi} \) 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 straight-away. 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 (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 \( 0 \leq \frac{n_{wi}}{n_i} \leq 1 \), we set \( \frac{n_{wi}}{n_i} \) to 1 for all cases. The term \( \log \frac{N}{n_w} \) can be obtained by assigning \( N = n_i \) 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

\[
s_t = \sum_{k=1}^{m} \log \left( \frac{n_{wi}}{n_{wi,k}} \right)
\]

where \( n_{wi,j} \) is the number of models \( M_i \), \( 0 \leq j \leq n_{wi}, \ j \neq i \), containing PIRFs that match the \( k \)th PIRF of the input model \( M_t \). \( m \) is the number of all matched PIRFs between input model \( M_t \) and query model \( M_i \). 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.
Step 2: Considering Neighbors

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 $M_{\text{argmax} i(s_i)}$. 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

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

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 C). However, considering the graph in the Appendix, this parameter does not significantly affect the performance of PIRF-Nav. In addition, $\omega$ 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. (2008), in the sense that it gives the probability of transition from state $k$ to state $i$. By this scoring, a high score $\beta_i$ of the query model $M_i$ is obtained if and only if its neighbors $M_{i-\omega}, M_{i-\omega+1}, ..., M_i, ..., M_{i+\omega}$ also earn a high score.

Step 3: Normalizing the Score

Now that the similarity score $\beta_i$ 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 $\beta_i$ of model $M_i$, even though all of its neighbors obtain a good score, could be lower than another score $\beta_j$ of the single model $M_j$ if the matched features between $M_i$ and $M_j$ are numerous. For that reason, the score must be normalized.

Based on the score $\beta_i$ 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 Fig. 3 for example. $\beta_i$ obtains the highest score. We calculate the deviation and means over the set of scores $\beta_{i-\omega}, ..., \beta_0, ..., \beta_{i+\omega}$.

The normalization used here is employed from the method done in the incremental BoW method (Angeli et al., 2008). Considering the set of scores $\beta_{i-\omega}, ..., \beta_0, ..., \beta_{i+\omega}$, we calculate the standard deviation ($\sigma$) and means ($\mu$), where $\omega$ indicates the number of neighbors taken into consideration. The normalized score $C_i$ of a score $\beta_i$ is obtained as follows:

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

where

$$T = \sigma + \mu$$
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 Fig. 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

\[
    j = \text{argmax}_i \ C_i
\]  

(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.

**Step 4: Loop Closure Acceptance/Rejection and Re-localization**

The obtained location \( L_i \) 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 \( \beta_j \) is greater than \( \sigma + \mu \) by at least \( \tau_2 \). 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 Fig. 9). 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. 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

$$C_i' = \sum_{k=-\omega}^{\omega+\omega-1} (C_k \cdot p_T(i, k))$$

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

$$j = \text{argmax}_i C_i'$$
5. 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.

5.1 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 (2008a) 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 (2008a), 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 (2008a), 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.

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 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 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. Fig. 5 shows the walking route for data collection. The short dotted line indicates routes in which no images were collected.

5.2 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 (2008a) and the incremental BoW method of
Angeli et al. (2008). 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 (Angeli et al. 2008) 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 1,237 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.

5.3 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:

- \( \theta \): 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 \( \theta = 0.5 \) offers the highest performance for all datasets. Actually, PIRF-Nav works best with small \( \theta \) because a PIRF must be extracted from slow-moving features. We have also tried \( \theta = 0.5, 0.6, \) and \( 0.7 \), but the results are not much different, i.e., \( \theta = 0.6 \) offers about 3% lower recall rate than \( \theta = 0.5 \).

- \( n_{\text{pirf, min}} \) and \( n_{\text{pirf, max}} \): 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.

- \( \tau_1 \): 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 \( \tau_1 = 3 \) 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.

- \( \tau_2 \): 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 B)

- \( \omega \): This is the parameter indicating how many neighbors are examined for calculating the updated score in Step 2. If \( \omega \) is large, the scoring would become slightly more stable, but the accuracy of the localization would drop. For example, if \( \omega = 20 \), 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, \( \omega \) has been set to 3. Other values \( \omega = 4, 5, 6 \) 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; $\sigma$ and $\mu$ 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 $\mu$ and $\sigma$ 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. 7 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.

5.4 Results

The results of the experiments conducted for City Centre, New College, and Suzukakedai Campus are shown in Fig. 6, and as precision-recall curves in Fig. 7. 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 6 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.

Precision-recall curves are depicted in Fig. 7. The curves were generated by varying the threshold value $\tau_2$ at which a loop-closure was accepted. Ground truths were obtained from Cummins and Newman (2008a) 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 (2008a) and Angeli et al. (2008), 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, 78% for City Centre, and 78% for Suzukakedai Campus).

Fig. 6 (a) 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 (2008a). (b) Appearance-based matching results for City Centre dataset overlaid on an aerial photograph. The aerial photograph and the ground truth data are also obtained from Cummins and Newman (2008a). (c) 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. All results are 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. For all datasets, there are no incorrect matches that meet this threshold. These results are best viewed as videos (Extension 2, Extension 1, and Extension 3 for (a), (b) and (c) respectively).
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 Fig. 8. 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 (Cummins and Newman 2009) 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 (Cummins and Newman 2008b). 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 Fig. 10). 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 Fig. 10, apparently does not correspond with the computation time portrayed in Fig. 8; 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. (2008), where the dictionary size is about 40,000, which results in 40,000 integers to represent one image.

The graphs in Fig. 9 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 Fig. 9, 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. (2008); 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 Fig. 7(d). 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 Fig. 8(c), Fig. 9(d), and Fig. 10(d), 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 (processing time is about 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 (Cummins and Newman 2009) 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 (Angeli et al. 2008; Cummins and Newman 2008a), 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 Fig. 11 and Fig. 12. Fig. 11 (bottom) 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. Fig. 11 (top) 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. Fig. 12 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.

6. 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 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. (2008). 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. 7). 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.

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.

7. Conclusions

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.

Acknowledgment

This study was supported by an Industrial Technology Research Grant Program received in 2004 from the New Energy and Industrial Technology Development Organization (NEDO) of Japan. Also, the authors gratefully acknowledge Oxford Robotics Research Group for their provided database and source codes.

References


### Appendix A: Appendix of Multimedia Extensions

<table>
<thead>
<tr>
<th>Extension</th>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Video</td>
<td>Results for the City Centre Dataset</td>
</tr>
<tr>
<td>2</td>
<td>Video</td>
<td>Results for the New College Dataset</td>
</tr>
<tr>
<td>3</td>
<td>Video</td>
<td>Results for the Suzukakedai Dataset</td>
</tr>
</tbody>
</table>

### Appendix B: Relationship between threshold and accuracy

The appendix presents the graphs showing the relationship between threshold values and precision-recall rate for all dataset at every image scale.
Appendix C: Relationship between Sigma and Precision

The appendix presents the graphs showing the relationship between threshold values ($\tau_2$), sigma value used for Gaussian distribution in (4) and (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