Scene Association for Mobile Robot Navigation

Edward Johns and Guang-Zhong Yang
Institute of Biomedical Engineering, Imperial College London
{ej09, g.z.yang}@imperial.ac.uk

***Abstract*—Accurate, efficient and robust location recognition is a fundamental task for any mobile robot. This paper presents a new approach using visual features to efficiently represent a series of locations along a path in an indoor environment. In the training stage, local features which are detected across multiple images from a single tour are combined to represent a real-world landmark, modelled by the expected variance of its descriptor. Those landmarks which represent the scene in the most efficient and discriminative manner are then retained, and this selection is optimized with respect to the scale of the environment. In the recognition stage, features detected in an image are matched to the landmarks in memory, based upon a novel similarity measure drawing from feature co-occurrence statistics.**

I. INTRODUCTION

Accurate location recognition forms an integral part of any mobile robot system, whether it is for interaction with the local environment or as a component in a navigation strategy. Visual methods have proved to be popular due to the quantity of information captured from a single image, the low cost of image sensors and the close relation to the human sensory system. The challenge is generally addressed by one of two approaches [1]. First, model-based techniques [2] compute a 3D reconstruction of the environment, by matching visual features in an image to those in the model. Second, appearance-based methods [3] compare the robot's current view with a database of images in memory, each forming a node in a topological map.

Within the field of appearance-based location recognition, there have again been two main approaches adopted. First, an image can be considered in a holistic manner by extracting global properties, such as colour histograms [4] or image gradients [5]. Matches are then made either by direct matching to an image database [6], or in a more qualitative sense by employing supervised learning [7]. Second, an image can be considered as an arrangement of local features, each describing the visual content in a local region [8]. Many examples of such local features exist, which are immune to variations such as illumination, scale and affine viewpoint [9][10]. Images are then matched either by quantizing features and comparing feature histograms [11], or by finding matches between each individual feature [12].

One of the benefits of using local features for location recognition is their relative robustness to occlusion compared to more global approaches. This is particularly apparent for indoor environments where small changes in viewpoint within a room can reveal or occlude large portions of a scene. Additionally, it is an advantage to be able to relate features in an image to specific objects in the environment, to facilitate a more semantic understanding of the environment and greater contextual awareness.

Of the many local features that have been developed, SIFT features [13] are a popular choice for image matching due to their tolerance of illumination, scale, rotation and small affine viewpoint variations, whilst remaining highly discriminative. Many approaches to location recognition using local features involve finding matches between every feature in a test image, and all features from all images in memory [14]. The standard method for determining a match between two features, the nearest-neighbour approach, applies a threshold to the ratio of the closest two matches [15], to ensure a high level of confidence in the match. Whilst this technique is powerful for features which are highly discriminative, such as in object recognition, or for location recognition in local environments, it is not well suited to large-scale location recognition. As the number of features stored in memory increases, the fraction of matches that exceed the threshold may decrease, as each feature becomes less distinct within the larger database. Furthermore, a feature match will be true simply if it is the closest match by some threshold, even if is still somewhat dissimilar to the feature it is being matched to.

In order to eliminate these false positive matches, a number of verification steps have been developed, including utilising the geometric and spatial arrangements of features [16][17], reducing the feature set to a subset representing each image in a more discriminative manner [18], and estimating the epipolar geometry between images [19]. Whilst the use of epipolar geometry can help to eliminate false matches, its performance again decreases as the size of the environment increases, because the algorithm requires at least seven true positive matches, and both the number of matches, and the proportion of true positive matches, decreases with scale.

As an alternative to the treatment of every extracted feature as an independent feature, this paper presents an approach which links together the same feature detected across multiple images. The linked features are then associated with landmarks in the real-world scene, rather than merely features from a single image. In the localisation stage, scene association then links features in the current view back to these real-world landmarks. The method is more suited to practical location recognition than standard techniques matching individual features, for the following four reasons. First, an understanding can be generated of how a feature's appearance varies across multiple

viewpoints; as such, the threshold of [15] is not required because it is now known where a feature should lie in descriptor space, for it to be considered a true match. Second, eliminating features which are only found in single images, and combining features detected across several images into one unified feature, reduces both the matching processing time and memory requirements of the system. Third, retaining only the most frequently-occurring features increases the likelihood of generating good feature matches when revisiting the scene. Finally, features that are recognized from multiple viewpoints often form part of the background of a scene, hence introducing greater robustness to dynamic objects in the foreground.

This paper presents three key contributions. First, an understanding of the expected variance of real-world landmarks is demonstrated. Second, a novel graph matching approach is presented, based upon the co-occurrence statistics of detected features. Third, an optimization algorithm is introduced, showing that the selection of features can be optimized with respect to the scale of the environment.

For the remainder of the paper, local features detected in a single image are denoted *features*, whilst features which are consistently detected across multiple viewpoints and represent a real-world point, are denoted *landmarks*. The features employed in this work use the SIFT descriptor previously discussed, which consists of a spatial arrangement of histograms surrounding the feature centre, each representing orientations of image gradients. Both features and landmarks use the SIFT descriptor.

II. Generating Landmarks

A. Detecting Landmarks

The first stage in computing the landmarks is the detection of features that are found in multiple adjacent images in an image sequence. This image sequence is generated by a robot during a tour of the environment, capturing images at discrete intervals, with each image representing a distinct location along the tour. Features are initially extracted from all images, $\mathbf{I} = \{I_1, \ldots, I_N\}$, representing all previously visited N locations. For each image I_i, features are tracked sequentially across adjacent images in the sequence, $I_{i+1}, \ldots, I_N$, using a feature descriptor distance threshold, t. If the descriptor distance between features is less than t, then a potential match is found. For experiments in this paper, $t = 0.6$ is employed for normalized descriptors. Matches are then verified by use of epipolar geometry [19]. A feature track terminates once there is no match between the currently tracked feature and all features in the next image in the sequence. Each feature track then represents a landmark in real-world space, which has been detected in a series of adjacent images. This is demonstrated in Fig. 1.

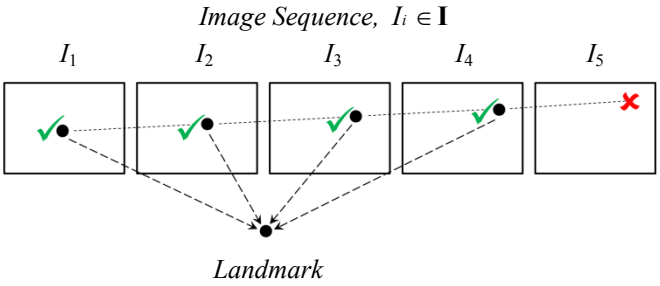

Fig. 1. Detecting consistent features across multiple sequential images. In this example, a feature is found to occur across images $I_1 \ldots I_4$, and each instance of this feature then contributes towards a landmark.

As an example of the overall concept of the paper's contribution, Fig. 2 shows the evolution of features into landmarks, and their subsequent optimization to a reduced set (discussed in section IV).

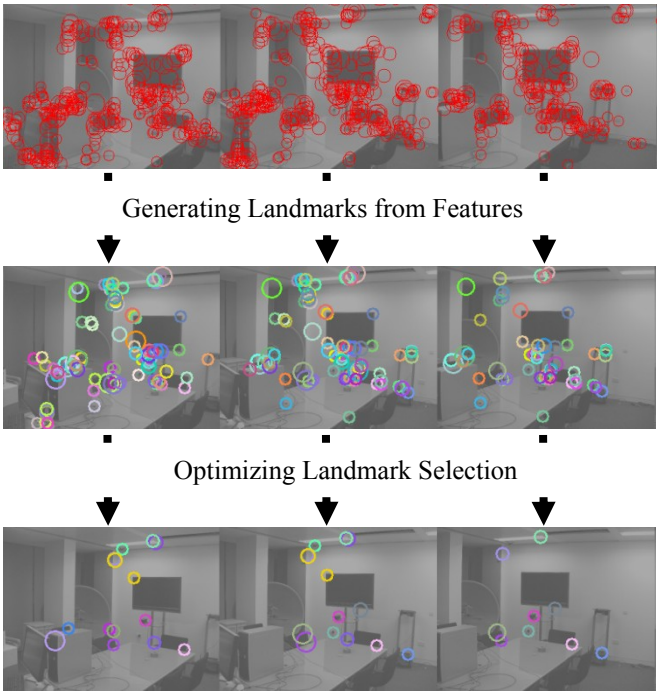

Fig. 2. Demonstrating the generation and optimization of landmarks. Images in the top row consist of a large number of SIFT features. Those which are detected across multiple consecutive images are combined to form landmarks, as shown in the images of the middle row. The bottom row shows the selection of landmarks which has been optimized to represent the scene in the most efficient manner. Each colour represents a single landmark.

B. Computing Landmark Descriptors

Once features have been tracked across a sequence of images, a landmark is generated by fusing these features. The nature of the SIFT descriptor allows for the same feature, viewed across small affine viewpoint changes, to have similar descriptors. However, the extent of this similarity is ambiguous when only one feature is used to learn the descriptor. The use of multiple features to learn a

range of descriptors for each landmark, allows a greater understanding of how the descriptor is expected to vary from differing viewpoints.

One approach to learning this variance would be to specify upper and lower limits for each dimension of the descriptor (representing each orientation bin of each histogram). However, with some landmarks built from only two or three features, there is often insufficient information to reliably predict the variability of each individual dimension. Therefore, instead of generating an expectation of the values for each bin in the SIFT descriptor, an expectation of each overall histogram is considered. This is appropriate because the variance within each histogram of the SIFT descriptor differs due to each histogram's unique locations relative to the landmark centre. Some locations will retain their structure across different viewpoints (consider a vertical line on a wall as a camera moves horizontally) whilst others will vary by greater extents (such as locations near the edge of the descriptor, which are more sensitive to external textures crossing over the feature boundary). As an example, consider a landmark detected at the corner of a computer monitor, in the middle of a room. The histograms representing the body of the monitor will exhibit only a small variance across different views of the monitor, whereas the histograms representing the background surrounding the monitor may exhibit a large variance, due to the inclusion of background clutter into the histogram. The system can therefore learn which histograms are expected to vary, and which are expected to remain constant as the camera's location changes.

Each histogram for the landmark is then described by a mean vector, μ, computed by averaging the histogram across all constituent features. For a SIFT descriptor with m histograms, each containing n bins, the mean representing the entire landmark is then an m-by-n matrix. Each histogram is also described by a single variance of σ^2, representing the variance in Euclidean distance to the histogram mean, across all constituent features. A value of d_{max} is then computed for each histogram, representing the greatest distance to the histogram mean, across all constituent features. For this paper, the descriptor parameters recommended in [15] are used, such that m = 16 (for a 4x4 window) and n = 8.

In order to fully utilize the power of the understanding that has been gained of the landmark's variance, a wider descriptor, computed at twice the scale of the feature and represented by this histogram variance, is incorporated into the landmark. This is the *context descriptor*, whereas the standard SIFT descriptor is the *local descriptor*, which does not include histogram variance as this is minimal at the local scale. Instead, the local descriptor simply has a mean and variance of the overall descriptor. The context descriptor adds a powerful awareness of the contextual variance surrounding the landmark, and by combining the local and context descriptors, a more discriminative overall descriptor is achieved. This compensates for the reduction in discriminative power induced by allowing landmark descriptors to operate within a certain variance.

III. Matching to Landmarks

A. Computing Descriptor Similarities

Once landmarks representing the robot's environment have been generated and stored in memory, matches can then be made to features detected in a new image captured by the robot. The first stage of matching a feature from a captured image, to a landmark in memory, is to compare the respective descriptors. A descriptor similarity measurement between a feature and a landmark is computed by considering the expected variance of the landmark's descriptor. The expected descriptor distribution for features constituting a landmark is modelled by a Gaussian function, with a mean of μ and variance of σ^2 for each histogram (context descriptor) or overall descriptor (local descriptor). The descriptor similarity S_D is computed as follows:

$$S_D = \begin{cases} \kappa \exp(-\dfrac{(x-\mu)^2}{2\sigma^2}) & \text{if } |\mu - x| > d_{\max} \\ 1 & \text{otherwise} \end{cases} \qquad (1)$$

$$\text{where} \qquad \kappa = \exp(\frac{{d_{\max}}^2}{2\sigma^2})$$

Here, $|\mu - x|$ is the Euclidean distance between the mean descriptor value of the landmark, and the descriptor value of the feature. The value of κ is set such that a feature descriptor which lies at the point exactly $d_{\max}$ away from the landmark mean descriptor is given a similarity of 1, and a distance less than $d_{\max}$ is given a similarity of 1. For all other distances, the similarity falls away as the distance from the mean increases, at a rate dictated by the Gaussian model. This is demonstrated in Fig. 3.

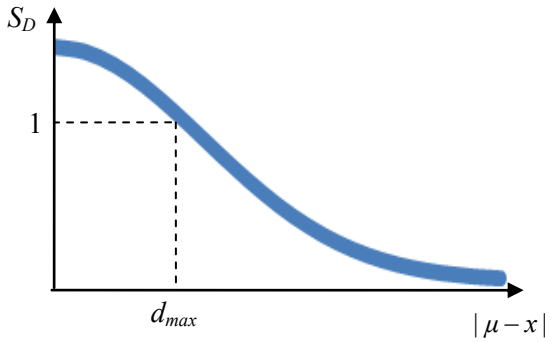

Fig. 3. Modelling the expected descriptor variation of features that constitute a landmark. For descriptor distances less than d_{max}, the similarity is 1. Otherwise, the similarity is based upon the expected descriptor variance.

The similarity for the entire context descriptor is computed by averaging the similarities for each histogram. The similarity for the local descriptor is computed by considering the overall descriptor mean and variance. The

overall descriptor similarity between a feature and a landmark is then computed by multiplying the similarities of the local and context descriptors. In this way, a feature and a landmark are only similar if they share both a similar local and similar contextual appearance, adding a high level of discriminative power to each landmark.

In order to speed up the rest of the matching process, and to eliminate improbable feature matches of low similarity, a minimum value of S_{min} was introduced. Potential feature matches with descriptor similarities below this value are discarded. For experiments in this paper, $S_{min} = 0.3$.

B. Computing Neighbourhood Similarities

Whilst the combination of local and context descriptors adds a high level of discrimination to features, false positive matches may still occur in large environments. As such, it is necessary to prune out the false positives by considering a more spatially-aware understanding of each landmark. Existing techniques for addressing this often take the spatially k-closest features to represent a feature's neighbourhood [20], or those which lie within a certain spatial distance from the feature [16]. One problem with these methods is that neighbouring features may become occluded across small viewpoint changes. Another problem is that features near the edge of an image will automatically have a large number of neighbouring features missing. To address these, a novel method is proposed, which computes the maximum neighbourhood similarity with respect to distance from the central feature. In this way, should occlusions occur near the central feature, then their impact on the overall neighbourhood match will be reduced as the distance from the central feature increases.

In order to find the neighbourhood similarity between a feature $f_i \in F$ in the current image, and a landmark $g_j \in G$ in memory, all other landmarks, g'_n, which co-occur in the same image as g_j, are re-ordered with respect to their distance from the central landmark g_j. This is done by considering the average location over each landmark's constituent features. Then, moving outwards from g_j across the image, the overall neighbourhood similarity is calculated by including all landmarks $g_1 \ldots g_k$ which lie inside the current neighbourhood size, k, and finding the best match between each neighbourhood landmark, g'_n, and each neighbourhood feature, $f'_m \in F$. The maximum value of this similarity, with respect to k, is then stored as the overall neighbourhood similarity between f_i and g_j. This is demonstrated in Fig. 4, where the maximum will occur at $k = 3$. At $k = 1$, the similarity is zero because the triangle is only detected in the landmarks, and not in the features; at $k = 3$, the similarity has increased because the diamond and square are detected in both the landmarks and features; at $k = 4$, the similarity drops again, as the hexagon is found only in the landmarks.

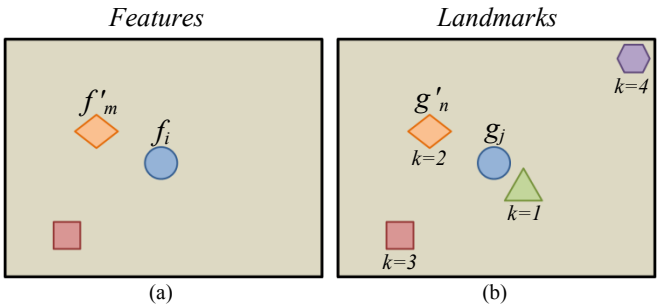

Fig. 4. Computing neighbourhood similarities between feature f_i and landmark g_j. (a) represents the features found in an image captured by the robot. (b) represents the landmarks generated during a training tour. The maximum neighbourhood similarity between feature f_i and landmark g_j occurs at $k = 3$.

The co-occurrence statistics of landmarks are also taken into account, in order to weight the contributions of each neighbouring landmark to the neighbourhood similarity, by how often these landmarks actually occur in the same image. The conditional probability, $p(g'_n \mid g_j)$, of a neighbouring landmark g'_n occurring in the same image as the central landmark, g_j, is calculated by dividing the number of images containing both landmarks, by the total number of images containing g_j. The overall neighbourhood similarity S_N, between feature f_i and landmark g_j, is then calculated by dividing the maximum neighbourhood similarity by the hypothetical maximum neighbourhood similarity if all neighbouring features and landmarks were a perfect match.

$$S_N(f_i, g_j) = \max_k \frac{\sum_{n=1}^{n=k} p(g'_n \mid g_j) \max_m S_{DA}(f'_m, g'_n)}{\sum_{n=0}^{n=k} p(g'_n \mid g_j)} \qquad (2)$$

$$\text{where} \quad S_{DA}(f'_m, g'_n) = S_D(f'_m, g'_n) S_A(f'_m, g'_n)$$

Here, $S_{DA}(f'_m, g'_n)$ represents the similarity between a neighbouring feature f'_m and landmark g'_n, in terms of both their descriptor similarity, $S_D(f'_m, g'_n)$, and their arrangement similarity, $S_A(f'_m, g'_n)$, relative to their respective central feature f_i, and landmark g_j. During the initial generation of a landmark, each co-occurring landmark is recorded, and the mean and variance for both spatial distances and angles between the two landmarks are recorded. Then, the spatial distance similarity and angle similarity between a neighbourhood feature and a neighbourhood landmark, are calculated in a similar manner to equation (1), with differences in spatial distance and angle replacing the differences in descriptor values. The angular and spatial distance similarities are multiplied to give the arrangement similarity, S_A, which is then itself multiplied by the descriptor similarity between the neighbourhood feature and neighbourhood landmark, to generate the overall neighbour similarity, S_{DA}. With the maximum neighbourhood similarity computed, the overall similarity between a landmark and a feature is then equal to the product of the descriptor similarity and the neighbourhood similarity, $S(f_i, g_j) = S_D(f_i, g_j) S_N(f_i, g_j)$, $0 < S(f_i, g_j) < 1$.

C. Computing Location Likelihoods

Once similarities have been calculated between every feature in a captured image, and every landmark in memory, the likelihood for each location can be computed, by considering the overall similarity to the images captured during the robot's tour. For a robot to be located at location l, it is expected that all landmarks present at this location during the tour, will also be present at any other time the robot is at l. Hence, the overall likelihood L of each location l is computed as follows:

$$L(l) = \frac{\sum_{g \in G_l} \max_{f \in F} S(f,g)}{|G_l|} \tag{3}$$

Here, $f \in F$ are the features in the image currently viewed by the robot, $g \in G_l$ are the landmarks which occur in the tour image at location L, and $|G_l|$ represents the total number of landmarks in the set G_l. The most likely location of the robot is then of course determined by the tour image with the highest likelihood.

IV. Optimizing Landmark Selection

The set of generated landmarks represents scenes from a number of viewpoints, but it is not yet optimized to represent the scenes in the most efficient manner. This is because:

i) Many of the landmarks will have large variances, or may be close to a large number of features not classified by this landmark, and hence will introduce a large number of false positive matches. A good landmark will therefore have small similarities to all those features not classified by this landmark.

ii) Many landmarks will be similar to each other and will not be highly discriminative when classifying a feature. A good landmark will therefore be highly dissimilar to all other landmarks.

A quality function, $Q(g_j)$, is now introduced, which determines the suitability of landmark g_j for inclusion in the final set. The quality function is an indication of the expected ratio of true positive feature matches to false positives feature matches, based on the matching of every feature, $f^t{}_i \in F^t$, in the training set. The descriptor similarity between each feature $f^t{}_i$ and landmark $g_j \in G$ is determined, and weighted by the probability that landmark g_j is the classification, rather than any other landmark. This probability is determined by the similarity between $f^t{}_i$ and g_j relative to all other landmarks. This then satisfies both requirements in i) and ii).

$$Q(g_j) = \frac{\sum_{f^t{}_i \in \tilde{G}_j} p(g_j \mid f^t{}_i) S_D(f^t{}_i, g_j)}{\sum_{f^t{}_i \notin \tilde{G}_j} p(g_j \mid f^t{}_i) S_D(f^t{}_i, g_j)} \tag{4}$$

$$\text{where} \qquad p(g_j \mid f^t{}_i) = \frac{S(f^t{}_i, g_j)}{\sum_{g_k \in G_j} S(f^t{}_i, g_k)}$$

Here, $f^t{}_i \in \tilde{G}_j$ represents all the constituent features of landmark g_j, when g_j was generated from the original tour.

The top η landmarks with the highest quality are then retained for each tour image. However, with each landmark spanning several images, it is highly unlikely that exactly η landmarks are assigned to each image. As such, in order to distribute landmarks evenly across the set of images, an iterative algorithm is employed, as described below.

1. Find the image, I_{min}, with the least number of landmarks assigned to it, η_{min}. If $\eta_{min} \geq \eta$, end the algorithm.
2. From the set of potential landmarks that occur in I_{min}, compute the $\eta - \eta_{min}$ landmarks which have the highest quality value.
3. Assign these landmarks to I_{min}, and update the number of landmarks that occur in each image accordingly. Jump back to 1.

Due to possible errors in the feature tracking algorithm of section IIA, some generated landmarks would have very large descriptor and spatial variances, and their inclusion in the final set only reduced the overall matching performance. As such, a threshold q_{min} was introduced, such that a landmark was only added to an image if its quality was above q_{min} – even if this means that the image would have less than η overall landmarks. The results in this paper use a value of $q_{min} = 0.01$. It should also be noted that whilst the optimization stage is computationally heavy, it is computed off-line once the robot has completed its training tour.

V. Experimental Results

In order to test the method described, a robot was manually driven through a training tour of an indoor environment, capturing images at roughly 1 metre apart. Landmarks were then computed from this training set. A test set containing the same number of images was then captured, following a similar tour to the training tour, but introducing slight deviations. The training set and test set were captured at different times throughout the day, to verify the illumination-invariance of the SIFT descriptor.

Experiments were then carried out to investigate how the value of η, the minimum number of landmarks per image, would affect the system's performance, by varying η in the optimization stage. Fig. 5 shows the results of this experiment for a training image sequence of 150 images,

computed on a 2.2 GHz Intel Core i5 processor. The match rate is defined as the percentage of correctly identified locations, and a confidence measure is also introduced. This is the average confidence that the most likely location is, in fact, the correct location, and is defined as (L_1 / L_1 + L_2), where L_1 and L_2 are the first and second highest location likelihoods, respectively.

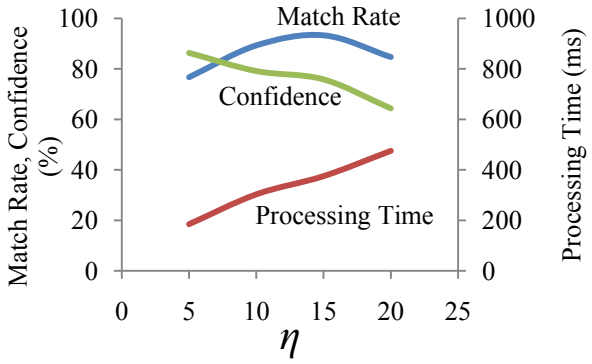

Fig. 5. Effect of η on system performance for a training set of 150 images.

There is a clear peak in the match rate, with an optimum number of landmarks per image. This can be explained by considering the optimization stage. Landmarks are added to their respective images in order of their quality, and hence with a smaller number of landmarks per image, the overall quality of the landmarks is large. As the number of landmarks increases, those with less discriminative properties are introduced, reducing the overall confidence of each matched image. However, with too few landmarks, the information stored representing each image is insufficient to generate an appropriate number of feature matches, and any false positive matches will have a much larger impact on the results. Therefore, a compromise is necessary between landmark discriminative power, and overall information quantity per image.

Furthermore, experiments showed that the optimum value of η is not distinct for each environment size. As the scale of the environment increases, it then becomes necessary to attach greater information to each image in memory, in order to combat the increase in false positive feature matches. It was found that for a certain environment size consisting of a certain number of images in the training tour, there is an optimum value of η, denoted η_{opt}. Thus, the system can be optimized with respect to the size of the environment, by adjusting the single parameter η. As can be seen in Fig. 5, the processing time of the matching algorithm also increases rapidly with η, and hence it is also somewhat fortunate that indefinitely increasing η to improve the accuracy is not necessary. Fig. 6 shows the value of η_{opt} for a range of environment sizes, together with the match rate at this optimum value of η.

The processing time of the location recognition includes around 180 ms for extracting the SIFT features. With smaller environments, the matching stage can occupy less than 10% of the overall processing time, relative to feature extraction. Even in large environments of 200 unique locations, employing η_{opt} allows matching to occur at 2 frames per second at a match rate of 88%, which is promising for real-time location recognition as part of a wider navigation system.

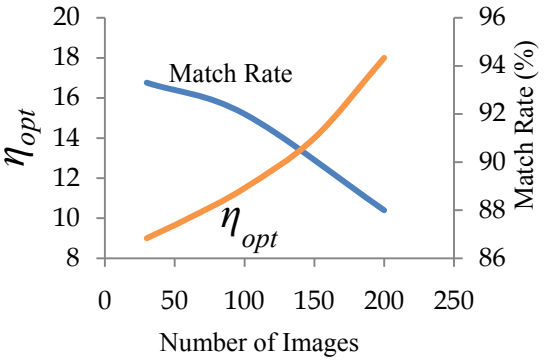

Fig. 6. Demonstrating the optimum number of landmarks per image and match rate at varying environment scales.

REFERENCES

[1] G. N. DeSouza and A. C. Kak, "Vision for Mobile Robot Navigation: A Survey", in *Trans. PAMI*, 24(2), 2002

[2] A. Davison, I. Reid, N. Molton and O. Stasse, "MonoSLAM: Real-Time Single Camera SLAM", in *Trans. PAMI*, 2007

[3] R. Sim and G. Dudek, "Comparing Image-based Localization Methods", in *Proc. Intl. Joint Conf. on Artificial Intelligence*, 2003

[4] M. Swain and D. Ballard, "Color Indexing", in *IJCV*, 7(1), 1991

[5] A. Jain and A. Vailaya, "Image Retrieval Using Color and Shape", in *Pattern Recognition*, 29(8), 1996

[6] Y. Rubner, C. Tomasi and L. J. Guibas, "The Earth Mover's Distance as a Metric for Image Retrieval", in *IJCV*, 40(2), 2000

[7] A. Pronobis and B. Caputo, "Confidence-based Cue Integration for Visual Place Recognition", in *Proc. IROS* 2007

[8] K. Mikolajczyk, T. Tuyltelaars, C. Schmid, A. Zisserman, J. Matas, J. Schaffalitzky, T. Kadir and L. Van Gool, "A Comparison of Affine Region Detectors", in *IJCV*, 65(1/2), 2005

[9] K. Mikolajcyk and C. Schmid, "An Affine Invariance Interest Point Detector" in *Proc. ICCV*, 2002

[10] J. Matas, O. Chum, M. Urban and T. Pajdla, "Robust Wide Baseline Stereo from Maximally Stable Extremal Regions", in *Proc. BMVC,* 2002

[11] D. Filliat, "A Visual Bag of Words model for Interactive Qualitative Localization and Mapping", in *Proc. ICRA*, 2007

[12] J. Kosecka and X. Yang, "Global Localization and Relative Pose Estimation Based on Scale-Invariant Features", in *Proc. ICPR*, 2004

[13] D. Lowe, "Object Recognition from Local Scale-invariant Features", in *Proc. ICCV*, 1999

[14] S. Se, D. Lowe and J. Little, "Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks", in *Intl. Journal of Robotics Research*, 2002

[15] D. Lowe, "Distinctive Image Features from Scale-invariant Keypoints", in *IJCV* 60(2), 2004

[16] A. Ascani and E. Frontoni, "Feature Group Matching for Appearance-based Localization", in *Proc. IROS*, 2008

[17] J. Kim, O. Choi and In So Kweon, "Efficient Feature Tracking for Scene Recognition using Angular and Scale Constraints", in *Proc. IROS*, 2008

[18] F. Li and J. Kosecka, "Probabilistic Location Recognition using Reduced Feature Set", in *Proc. ICRA* 2006

[19] O. Chum, T. Werner and J. Matas, "Epipolar Geometry Estimation via RANSAC Benefits from the Orientated Epipolar Constraint" in *Proc. ICPR* 2004

[20] F. Schaffalitzky and A. Zisserman, "Automated Scene Matching in Movies", in *Proc. Challenge of Image and Video Retrieval*, 2002