International Symposium on Experimental Robotics, July 6-10, 2006, Rio de Janiero, Brazil

Vision-based Localization Using a Central Catadioptric Vision System

Mana Saedan[1], Chee Wang Lim[2], and Marcelo H. Ang, Jr[1]

[1] Department of Mechanical Engineering
National University of Singapore
9 Engineering Dr 1, Singapore 117576
{saedanm, mpeangh}@nus.edu.sg
[2] Singapore Institute of Manufacturing Technology
Agency for Science, Technology and Research
71 Nanyang Drive, Singapore 638075
cmlim@simtech.a-star.edu.sg

Summary. We present an appearance-based localization algorithm for an indoor environment that is inspired by human's localization and navigation capabilities. Our localization approach integrates the Monte-Carlo localization technique with an omnidirectional image matching algorithm. The approach yields robust localization outcome with reasonable accuracy even when operating in a large map with sparse reference images.

1 Introduction

The ability of human beings to use vision to navigate in daily life has inspired us to investigate vision to be used as a primary external sensor for robot localization. We can remember and relate places without explicitly knowing their exact locations. This motivates us to develop a vision-based localization algorithm, where the robot recognizes landmarks and uses them to navigate to the goal. Such approaches have been referred to as *appearance-based* methods [1]. Our system combines visual information with metrical information, resulting in a hybrid map consisting of a topology of landmarks (visual images) and their relative location and orientation (relative odometry).

Recently, a appearance-based localization has gained more popularity among mobile robotics researchers, perhaps due to the approach being still under explored and technological advances in computer and vision hardware. The main difference between appearance-based and other approaches is the method to represent the environment. The appearance-based approach relies on remembering features of the environment rather than explicitly modeling it. Many work in this field such as [2] and [3] take inspiration from the biological counterparts.

Early implementations of appearance-based localization [2] and [4] involved quite complicated image processing techniques. The ultimate aims of these methods are to find the image stored in the map that best matches a query image. They were either tested in a static environment or a specially structured one. More recently, many implementations attempted to deal with realistic environments. These algorithms incorporated probabilistic techniques with relatively simpler image processing algorithms. We take this approach to implement our robot localization. The Monte-Carlo localization (or particle filters) is specifically studied. The existing work by Gross et al [5], Menegatti et al [6] and Andreasson et al [7] demonstrate the success of incorporating the Monte-Carlo localization with omnidirectional vision sensors. Nevertheless, there are still challenges before appearance-based localization becomes pervasive.

An image contains rich information of the environment compared with other sensor data. However, high computational requirement to interpret the image prohibits it to be used directly in the localization algorithm. As a result, many appearance-based localization researchers focus their efforts to develop methods of extracting robust features from an image. For example, the Fourier coefficients are used in [6], and the SIFT features are adopted in [7]. In our previous work [8], we elaborate the technique to extract features directly from original (circular) omnidirectional images without projecting them to any other surfaces. This results in a marginal increase in computational speed and that the locations of features in the image are always spread over the entire image. Hence, they are naturally robust to occlusion.

The bottle neck of the appearance-based approach is the time spent in the image matching process in spite of the availability of low dimensional image features. This limits many approaches and makes them impractical in very large scale environments. Consequently, we try to ease the limitation by developing the localization technique to be able to work in a map that has relatively few reference images of an environment. Although our technique sacrifices the overall accuracy of the localization system for the applicability to a large scale map, we can still maintain robustness of the system.

In this paper, we present the main concepts of our localization technique. We first describe the image matching algorithm that is the crucial component in our localization implementation. Next, we elaborate the technique to incorporate the Monte-Carlo localization (MCL) with our image matching. We present experimental results, which demonstrate the efficacy of our localization. Finally, the conclusions are drawn to summarize our work.

2 Image Matching for the Appearance-based localization

In [8], we presented a method to extract information (image features) directly from an original omnidirectional image. The summary of image feature extraction is illustrated in Fig. 1. The method employs the wavelet-based

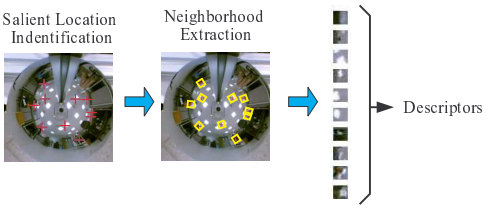

Fig. 1. Summary of image feature extraction procedure. The salient locations (cross) are determined from the input image. The descriptor vector is assigned to each neighborhood region (sub image.)

salient point identification to select some distinctive locations in the image. The neighborhood pixels around each salient location are then extracted in a rotational invariant manner to form a *descriptor-vector* describing the salient location. In our implementation, we fix the size of the omnidirectional image at 224×224 pixels. A total of 80 salient locations are extracted from the image. The *feature point* in an image consists of the *salient point location* (x, y) and its corresponding *descriptor-vector* (24 elements). The compactness of feature-dimensionality reduces computation time involved in the image matching.

Our localization implementation involves two levels of image matchings: global matching and local matching. The global image matching compares a query image with all images in the database, whereas the local image matching determines similarity between the query image and individual database image. Both global and local matching are independently used in different stages of our localization algorithm (described in Section 3).

2.1 Global Image Matching

The global image matching adopts the searching technique reported in [9]. The technique approximately searches the map-database for closest descriptors to the query vector given the maximum number of descriptor-candidates to be obtained. This maximum number of candidates is adjusted according to a compromise between computational speed and accuracy. In particular, we set the maximum number of candidates at 10% of total descriptors in the database.

The global matching requires a construction of the KD-Tree of all descriptors for all the images in the database (map). The global matching submits each descriptor of the query image to search for some closest descriptor-candidates from the tree. The Euclidean distance between the query descriptor and each candidate is computed. The closest candidates are chosen by a user-defined threshold in comparing Euclidean distances. A query descriptor should match only one descriptor from one reference image. If a query descriptor matches with more than one descriptor-candidates from the same reference image, the pair with smallest Euclidean distance is kept. After obtaining the

candidates from initial searching, the shortest Euclidean distance among a query descriptor and descriptor-candidates is identified. The pairs of (query and database) descriptors that have Euclidean distance smaller than 1.25 times the smallest Euclidean distance are selected as matching candidates. This searching process is then repeated with new query descriptor until all query descriptors are matched.

After all query descriptors are matched, the number of matching pairs are counted for each reference image. The reference images are sorted according to their numbers of matching pairs. The reference images that have numbers of matching pairs higher than 80% of the highest number of matching pairs are kept. This results in a reduced set of reference images where each image consists of a set of matching pairs. Each reference image is further examined to remove the matching pairs that are outliers. A matched pair consists of a pair of feature points in the reference and query image, but may be viewed from a different angle by the robot. The angle can be obtained by

$$\theta_{qd} = \tan^{-1}\left(\frac{y_d x_q - x_d y_q}{x_d x_q + y_d y_q}\right) \tag{1}$$

where (x_d, y_d) and (x_q, y_q) are the coordinates of the feature point (salient location) of the reference and query images, respectively. If the match is perfect, the angles computed for all matching pairs should be equal. However, in practice, the angles will vary. We therefore discretize the angles into bins of 45 deg and determine the number of matched pairs in each bin. Pairs in the bin with the highest number are retained and the pairs in the rest of the bins are considered as outliers and discarded. Each reference image will then have a score which is the number of remaining matched pairs. All the scores of the references images are compared to find the maximum. The maximum is used as the *global score*. The results from the global matching are the global score and the set of reference images that resemble the query image.

2.2 Local Image Matching

The local image matching analyzes similarity between a query image and a single reference image. Each descriptor of the query image is compared to all (80) descriptors of the reference image. The pair that has smallest Euclidean distance is recorded and its corresponding angle computed according to Eq. 1. This results in an angle for each feature point in the query image. The outliers are removed in a similar way as described in the previous section. The angle bin that has the maximum number of feature points is retained and is the *local score*, and the angles of the pairs in this bin are averaged. This average is an estimate of the relative orientation between the query and reference images.

3 Appearance-based Localization with the Monte-Carlo Localization

The environment is modeled using a hybrid map between topological and metrical maps. Since robot's odometry has accumulated errors, the absolute position of each map-node is obviously inaccurate. Therefore, our hybrid map contains only the relative metrical information obtained from the odometry reading. The map is represented by the graph where nodes are encoded with the image feature information, and edges are encoded with the relative position to/from adjacent nodes. The map can be learned quickly using only information from the odometry and the omnidirectional image. The state model of a robot in the hybrid map is modeled:

$$\boldsymbol{\xi}(t) = [n, x_r, x_r, \theta_r]^T \tag{2}$$

where n is the reference node, and (x_r, x_r, θ_r) is the position of a robot with respect to the reference node. The objective of our localization process is to identify the appropriate reference node that the robot is nearest to.

We aim to develop the appearance-based localization that requires minimal computational cost and memory usage to store a map by creating the map that has fewer reference images. Our localization system does not rely solely on the image matching, since the image matching results are inaccurate in many realistic situations. Consequently, the probabilistic approach is incorporated to deal with uncertainty and ambiguities of image matching results. The Monte-Carlo localization (MCL) is specifically studied. The Monte-Carlo localization represents the belief state of a robot by a set of discrete samples. Each sample consists of a state of the system of interest (Eq. 2) with an associated weight indicating its importance.

We observe characteristics of the similarity score from the local matching. The prominent characteristic is that more than 80% of the query images taken within 20 cm range from the corresponding reference node have similarity scores (local score) above 15, whereas the similarity scores of the query images taken farther from this area vary over a wide range. As a result, we develop the technique to update the importance weight of a sample with real observation (via local matching) only when the individual sample is within the 20 cm range from its reference node.

When a robot is in the sensitive area (within 20 cm range) of any map node, the similarity score between the query image and the reference image of the particular map node is likely to be much higher than the score of other map nodes. Therefore, samples in the that (nearest) reference node are updated with higher importance weight than the others. Moreover, rotational angle from local matching is added in the importance weight updating to decrease matching ambiguities, e.g., other reference nodes also having high similarity scores. The false reference node often has a random rotational angle with the query image even though their similarity score is high. Consequently, the

weight updating of a sample with real observation is evaluated by the function given in Eq. 3. The weight of a sample outside the range are calculated from the last actual updated weight with a linear discounting factor proportional to the distance from the reference node.

$$w_i = exp\left[-a(s(q,r) - S) - b|\phi(q,r) - \theta_r|\right] \tag{3}$$

where a, b are the positive arbitrary constants, S is our empirically derived minimum similarity score, and θ_r is from the state of the particle (orientation). $s(q,r)$ and $\phi(q,r)$ are the local similarity score and the rotational angle, respectively, between the query image and the reference image of the map node. In addition, the similarity score $s(q,r)$ is clipped at S if it is larger than S.

Another difference from existing techniques is that we utilize the global matching to speed up the convergence rate of the particle filter. In particular, particles are not placed uniformly over all map nodes in the case of unknown starting position. Furthermore, the global matching is used for re-initializing a portion of particles after majority of particles converges to one map node. This criteria is valid when one reference node has number of particles significantly higher than other nodes. We refer to our re-initialization algorithm as the *Disbelief* algorithm.

In the disbelief algorithm, some particles from the prominent node (the node with maximum number of particles) are randomly removed when the *global* score reaches a certain threshold. The removed particles are then placed at node-outputs from the global matching. Please note that the global score is computed every time a query image is taken.

3.1 Disbelief Algorithm

We investigate the accuracy the global matching to test its applicability in our disbelief algorithm. We test by putting a robot within 20 cm radius from any map node. We then execute our global matching algorithm to see if the set of nodes/reference images contains the correct map node. We run the algorithm for map database of size 5 to 50 map nodes, with many runs for each map size (number of nodes). The number of runs is proportional to the map size. The global matching produces correct results more than 84% of the time. Fig. 2 shows the average global score of the query images versus number of images in a map database. The sensitive area is set to be the area within 20 cm radius from the map node.

As shown in Fig. 2, the averaged global similarity score of the query images inside the sensitive area of the map node is significantly higher than the global score of the images outside the sensitive area. Hence, we empirically derive the minimum global score (disbelief threshold) to indicate that a robot is in one of the map nodes (hopefully in one of the output map nodes from the global matching) as:

$$T_d = 5 + 10exp(-0.05N) \tag{4}$$

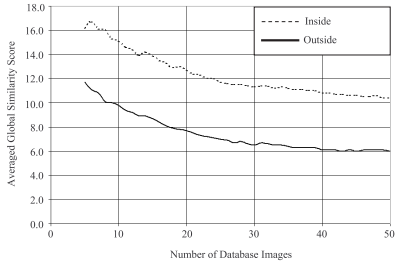

Fig. 2. The average global score of the query images inside and outside the sensitive area of the map nodes (20 cm radius from the corresponding map node).

where N is the total number of reference images in the database.

Consequently, the disbelief algorithm is set to re-initialize the particle filter when the global score of the query image is higher than the disbelief threshold defined in Eq. 4. Moreover, the disbelief algorithm is being activated when there is only one node that has number of particles significantly (e.g., more than 20%) higher than the other nodes. The disbelief algorithm randomly removes 40% of the particles from the prominent node, and then uniformly places the removed particles over each output nodes from the global matching.

4 Experimental Results

The robot in our experiments was equipped with the central catadioptric vision sensor as depicted in Fig. 3 (Left). The on-board computer was a notebook PC (Pentium M processor 1.73 GHz). The computer performed all required tasks such as image processing, robot control, and localization. The map database was constructed from the reference images that were captured at approximately 100 cm interval while the robot was manually controlled along the route shown in Fig. 3 (Right); a total of 72 reference images were taken. Moreover, The number of particles in all experiments was fixed at 720 particles. During localization phase, the query image was captured when either the robot had moved by 10 cm or turned by 30 degrees.

The first experiment was conducted to evaluate the overall performance of our localization. Two scenarios were tested by manually controlling a robot to travel along the route that the map was built. The robot was moving from the node number 1 to 72 in the first test, whereas the robot was moving in the reverse order during the second test. At the start of localization process, the robot did not have the true position information. The particles initialization was done by using the global matching to select starting nodes. The video clips of the user interface program of this experiment can be found at the url: *http://guppy.mpe.nus.edu.sg/∼manna/localization*. The big green dot in

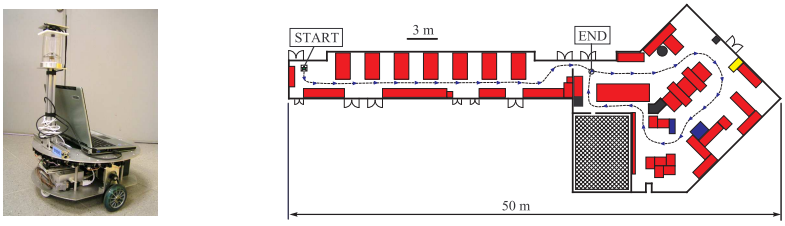

Fig. 3. (Left) the robot for the experiments was equipped with the omnidirectional camera. (Right) the floor plan of the laboratory, and the traveling route of the robot during map learning phase.

the user interface indicates the estimated reference node, and the blue dot indicates true location of the robot.

Table 1 presents the performance after completing both tests. The false localization time was counted if the estimated reference node is farther than 200 cm from the true position of the robot. The loss time was counted when the estimated reference node has less than 30% of the total particles. The results in Table 1 show that our localization algorithm is able to localize the robot within the position error range of 200 cm. However, both scenarios show some false localization (the estimated robot location is farther than 200 cm from the true location). This is because the method to select reference node is based on identifying the node with maximum number of particles. Therefore, there are some occasions that the correct reference node has lower number of particles than other nodes.

Table 1. Performance of our localization algorithm

	Localization Outcome (100%)		
Trial	Correct	False	Loss
1	79.9	16.2	3.9
2	66.9	28.3	4.8

The kidnapping situation was tested in the second experiment. After the particles has converged to correct location, the robot was manually lifted and then placed somewhere in the map without informing the robot. Four kidnapping attempts were conducted to verify whether the robot was able to recover from localization errors. The average time steps spent before the robot discovered the correct reference node was 125 steps. However, there were some kidnapping trials took much longer time to discover the robot's true position. The snapshots of the distribution of particles during the first kidnapping trial is depicted in Fig. 4. In the figure, the square with cross mark indicates the estimated reference node, and the star shows the true location of the robot.

The small dots are the particles where more particles near each other are seen as blobs. Note that some particles appear in infeasible areas such as being in tables or being outside the lab. This is because our map does not contain any information regarding these infeasible areas.

5 Conclusion and Research Perspective

We present our appearance-based localization technique that utilizes omni-directional image and robot's odometry information. The technique requires less reference nodes to localize a robot with reasonable accuracy. Hence, our algorithm is able to extend to larger working environment easily. The future development is to enhance the robot's capability to incrementally build the map from scratch, and to use that map to localize itself.

References

1. P. Zingaretti, E. Frontoni (2006) Appearance-based Robotics. In: IEEE Robotics and Automation magazine 13:59–68
2. A. Steinhage, G. Schöner (1999) Self-calibration based on invariant view recognition dynamic approach to navigation. In: Robotics and Autonomous Systems 20:133–156
3. M. O. Franz, B. Schölkpf, H. A. Mallot, H. H. Bülthoff (1998) Where did I take that snapshot? Scene-based homing by image matching. In: Biological Cybernetics 79:191–202
4. Y. Mutsumoto, K. Ikeda, M. Inaba, H. Inoue (2000) Exploration and navigation in corridor environment based on omni-view sequence. In: IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS 2000) 1505–1510
5. H.-M. Gross, A. Koenig, C. Schroeter, H.-J. Boehme (2003) Omnivision-based probabilistic self-localization for a mobile shopping assitant continued. In: IEEE/RSJ Int. Conf. on Intelligence Robots and Systems (IROS 2003) 1505–1511
6. E. Menegatti, M. Zoccarato, E. Pagello, H. Ishiguro (2004) Image-based Monte-Carlo localisation with omnidirectional images. In: Robotics and Autonomous Systems 48:17–30
7. H. Andreasson, A. Treptow, T. Duckett (2005) Localization of mobile robots using panoramic vision, local features and particle filter. In: IEEE Int. Conf. on Robotics and Automation (ICRA 2005)
8. M. Saedan, C. W. Lim, M. H. Ang, Jr. (2006) Omnidirectional image matching for vision-based robot localization. In: To be appear in the proceeding of the IEEE Int. Conf. on Mechatronic (ICM 2006)
9. J. S. Beis, D. G. Lowe (1997) Shape indexing using approximate nearest-neighbour search in high-dimensional spaces. In: Conf. on Computer Vision and Pattern Recognition 1000-1006

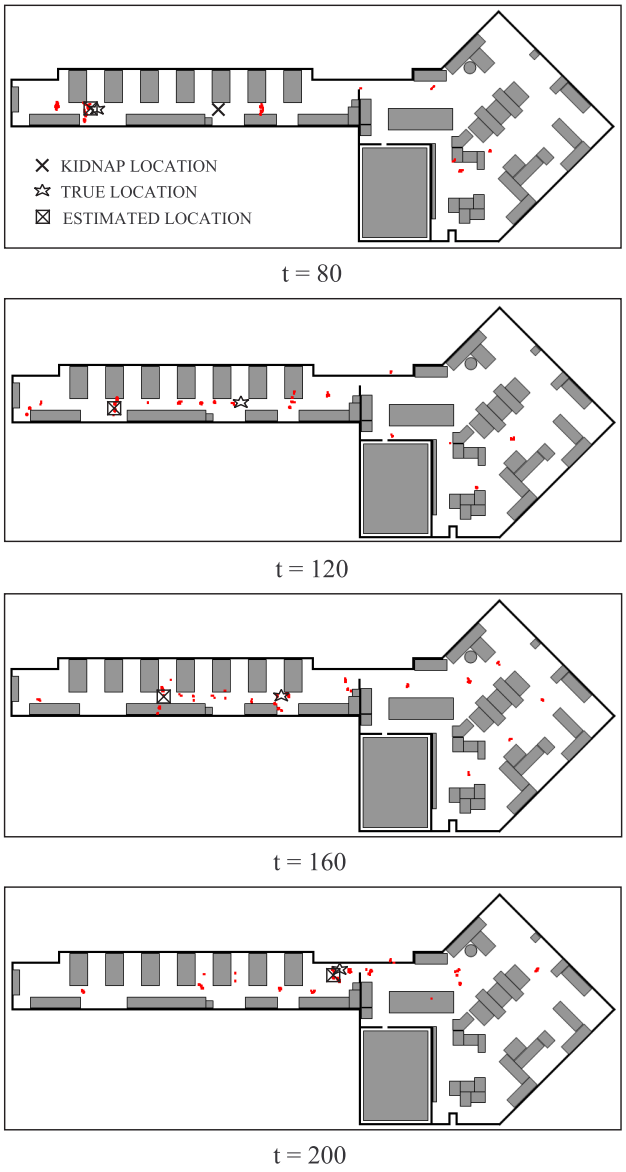

Fig. 4. Snapshots of particle distribution during the first kidnapping trials. The robot was kidnapped immediately after time step 80^{th}. The star indicates the true position of a robot, and the square with cross mark indicates the estimated location.