Adaptation and Learning for Image Based Navigation

Supreeth Achar and C.V. Jawahar
Center for Visual Information Technology
International Institute of Information Technology
Hyderabad, 500032, India
{supreeth@research. , jawahar@}iiit.ac.in

Abstract

Image based methods are a new approach for solving problems in mobile robotics. Instead of building a metric (3D) model of the environment, these methods work directly in the sensor (image) space. The environment is represented as a topological graph in which each node contains an image taken at some pose in the workspace, and edges connect poses between which a simple path exists. This type of representation is highly scalable and is also well suited to handle the data association problems that effect metric model based methods. In this paper, we present an efficient, adaptive method for qualitative localization using content based image retrieval techniques. In addition, we demonstrate an algorithm which can convert this topological graph into a metric model of the environment by incorporating information about loop closures.

1. Introduction

With recent advancements in computer vision, many robotic vision systems have been shown to be practical in real world. Traditional robotic vision systems employ stereo or structure from motion (a complete 3D reconstruction) for navigation in a 3D world [3]. However, model-free or image based methods [16] have recently emerged as interesting alternatives which enable a robot to operate without an explicit metric reconstruction of the environment.

An autonomous robot typically requires some sort of representation or 'map' of the environment it is working in to enable navigation. This map can be provided *apriori* or can be built by the robot as it explores the environment. Sensor readings taken by the robot can be correlated with this map to determine where the robot currently is, a process referred to as *localization*. The process by which a robot determines a feasible path to a goal using the map and knowledge of the current pose is called *path planning*. As the robot moves towards the goal it will tend to slowly deviate from the intended path due to factors such as wheel slippage and innacuracies in the robot motion model. This can be overcome by periodically relocalizing the robot as it moves using an *apriori* estimate of the robot's pose (this is called local localization). Alternatively, a robot equipped with a camera can use visual servoing techniques to follow a path, without the problem of accumulating pose errors. A thorough description of the problems in robot navigation can be found in [19]

A large amount of research in robot navigation is devoted to the problem of automatically building a metric map of the environment while concurrently using the partially constructed map to localize the robot. This problem is called SLAM (Simultaneous Localization and Mapping). Visual SLAM systems using both monocular and stereo imaging to reconstruct a 3D model of the environment have been studied in depth [2]. In [12] and [18] systems are presented that use monocular vision and structure from motion techniques to provide a realtime estimate of the trajectory being followed by a robot.

On the other hand, image based methods for robot navigation store very little or no metric information in the environment representation. The 'map' takes the form of a topological graph in which each node contains sensor readings (in this case images) taken at some position in the workspace [14]. Nodes are linked with an edge if there is a simple, collision free path between the poses corresponding to the two nodes.

In the context of image based navigation, the localization process is formulated as an image retrieval problem. The graph contains a large collection of images taken from all over the environment. The image acquired at the robot's current position is used as the query. Localization is performed by finding images stored in the graph which are similar to the robot's current view. The more similar a database image is to the robot's current view and the greater the overlap between the two images, the more likely it is that the robot is close to the corresponding node in the graph. Con-

978-0-7695-3476-3/08 $25.00 © 2008 IEEE
DOI 10.1109/ICVGIP.2008.72

tent based image retrieval can now be performed accurately and efficiently even over very large image collections containing millions of images [13]. Hence image retrieval techniques can be employed for fast and effective robot localization.

The focus of this paper is on how learning can improve the performance of an image based robot navigation system. We show that qualitative localization of a robot can be performed effectively using an adaptive vocabulary based approach to image retrieval. The visual vocabulary used by the system is not fixed, it adapts dynamically to better describe the type of visual features that occur in the environment. This makes it possible for the robot to work better in new environments which are dissimilar in appearance to those it has worked in previously. In addition, we present a method that allows the robot to gradually learn the metric structure of the environment over time from the topological graph that it builds.

The remainder of the paper is organized as follows. Section 2 describes the use of CBIR for qualitative localization and the adaptive vocabulary approach in particular. Section 3 discusses how the topological graph is used by the robot to plan and execute paths. Section 4 introduces our method for learning metric structure from the topological graph. Section 5 contains results from experiments performed.

2. Qualitative Localization as Image Retrieval

The workspace representation used takes the form of a topological graph $G = (V, E)$. The number of nodes in the graph is denoted by n and each node $v \in V$ contains an image captured at some pose in the workspace. Edges join pairs of nodes between which the straight line path between the corresponding poses is known to be collision free. Edges can also optionally store a rough estimate of the displacement between the two nodes determined from epipolar geometry and odometric measurements. This information is not strictly necessary but it helps during the path planning and execution processes described later.

We formulate the localization problem as an image retrieval task. When a robot navigates in an environment, it builds an experience in the form of a collection of images which are stored as nodes in a graph. The question *Where am I now?* is answered by retrieving the images from the collection most similar to the current view. This is directly applicable for a robot which navigates in an environment which it is familiar with. The same framework and representation can also be used while exploring unseen areas [15].

In a metric scheme, the localization process is expected to return an estimate of the current 3D pose of the robot with respect to some global coordinate frame given a map and the current sensory inputs. In an appearance based approach with a topological graph, there is no notion of metric space or a global coordinate frame. Hence localization to an absolute position is not possible. Instead, the localization process finds a node in the graph which is close to the current pose of the robot through the direct comparison of images. The localization merely says that the robot is close to a particular location in the graph. This type of localization is referred to as qualitative localization.

Thus the goal of qualitative localization is to find the node $v \in V$ in G that was taken at a pose in the workspace close to the current pose. Two images taken at similar poses in a workspace are likely to have a significant degree fo overlap in content between them and hence it is natural to formulate qualitative localization as an image retrieval problem. Once the robot is localized, the control signal required to reach the goal destination can be calculated (Section 3). The comparison between CBIR and qualitative localization is summarized in Table 1.

Table 1. CBIR for Localization

Database	Image collection of workspace
Query	Robot's current view
Desired Result	Image(s) in database closest to current pose
Similarity	Occurrence of similar patches
Scoring	Degree of overlap between images

The problem of image retrieval for qualitative robot localization is in some ways different from that faced in traditional content based image retrieval (CBIR) tasks. In a typical CBIR application, the input is an image containing some object and the goal is to find all images in the database containing the same object or an object of the same class. In qualitative localization, the goal is not only to find an image from the database image containing the same objects that are present in the current view, it is more important to find an image taken from a camera pose that is as close as possible to the current view. A qualitative localization system does not have to deal with the problem of retrieving images containing objects similar but not identical to those in the query image. An ideal visual qualitative localization system would be robust to changes in direction, source(s) and intensity of illumination etc. They can cause large changes in scene appearance in both natural and man made environments. The localization process should also ideally be able to handle dynamic environments in which some elements (people and chairs for example) do not remain stationary. The localization method should also be able to adapt to new environments containing features unlike those seen previously.

Approaches to visual qualitative localization typically extract some type of features from images and use a similarity measure to match stored images against the present view. The approaches can be broadly divided into two major categories, those that use global features for describing an image and those that use local features. Like in CBIR, one of the most popular global image descriptors in qualitative robot localization is the colour histogram [23]. Colour histograms are simple and often effective but they tend to give very coarse localization results. It has also been proposed to use the entire image itself as a global descriptor, but because of the computational expense involved some form of dimensionality reduction is generally considered to be desirable. In [6], the use of kernel principal components of an image as global features was proposed. In [9], Fourier domain analysis of images captured from an omnidirectional camera was used to generate a Fourier signature for an image, which was used for localization.

The use of local features for qualitative localization provides a significant degree of robustness to occlusions and changes in viewpoint. Extraction of local features from an image is typically computationally more expensive than global features, but the successful use of local features in image retrieval demands the investigation of their applicability in qualitative robot localization. One approach would be to directly match local features (like SIFT descriptors) between the current view and the images stored in the graph as is done in [22]. This method is provides good results. But because the current view needs to be matched against each database image it does not scale well as the number of images is increased. The Bag-of-Words (BoW) approach [17] of modelling images as collections of 'visual words' built from local feature descriptors has made it possible to perform efficient and accurate image retrieval using local features over very large image databases. In [5], the current view is matched to images in the database on the basis of visual words by using a simple voting mechanism in which the number of words each database image has in common with the current view is counted. The matches from the highest scoring images in the first step are geometrically validated by fitting them using a homography. If an image in the database has a sufficiently large number of geometrically validated matches with the current view, the robot is localized to that image. In [1] a Bayesian approach to Bag-of-Words localization is presented. A generative model for the probability of a set of visual words occurring in an image is learnt from a training dataset. This is used to estimate the probability of two given images coming from nearby poses. Similar looking, highly distinctive views are given high probabilities of being from the same pose, while views that appear frequently in the workspace are given lower probability scores.

The Bag-of-Words based robot localization schemes described above use fixed vocabularies. These vocabularies are built during a separate training process over a set of images that are considered to be representative of what the robot is expected to see while it navigates through an environment. It is assumed that a sufficiently large vocabulary will allow the CBIR system to function effectively over any image collection. A better alternative would be a dynamic set of visual words for describing an image that adapts to best represent the images of the robot's environment. This would improve the robot's ability to operate in new environments which are visually dissimilar to those it had seen previously. Adaptive Vocabulary Forests [21] provide a method for doing this. A forest is grown incrementally as new images are added to the collection. Using a set of vocabulary trees helps to overcome problems of quantization near cell boundaries that occur when using a single tree. As new images are added to the collection, nodes are added. Nodes that have not been accessed over a long period of time are considered obsolete and gradually pruned out of the trees.

We extract scale and affine invariant interest points [10] from each image. A typical 640×480 image generates around 200 to 300 such interest points. For each interest point we determine the 128 dimensional SIFT [8] feature descriptor. Each tree has a set of inverted files associated with it, one file for each visual word. The files contain the indices of all images in the collection containing that particular visual word. This inverted file structure makes it possible to quickly process queries. When a query image is given to the system, visual words are extracted from it. We score database images according to the number of words they contain that appear in the query. Each tree has its own set of visual words and generates a score for the database images. These scores are totalled and the database image with the highest score is returned as the closest match. Figure 1 shows some example queries and the results returned by the localization system. The result images clearly match well with their respective queries.

3. Navigation using a Topological Map

The topological graph can be used for more than just localization. It enables the robot to plan and execute paths from its current position to a destination in the environment. This makes image based approaches a complete solution to the problem of robot navigation. The robot is given an image of the destination pose. The first step in path planning is to localize the current view and the destination image. Once the two nodes in the graph closest to the current position and destination are determined, a path through the graph that links them needs to be found. We call this path a visual path and use Dijkstra's algorithm to determine the optimal visual path. The edge weight w_{ij} between the nodes i and j used for this path planning process is given by

$$w_{ij} = \alpha|\theta_{ij}| + \beta||T_{ij}|| \quad (1)$$

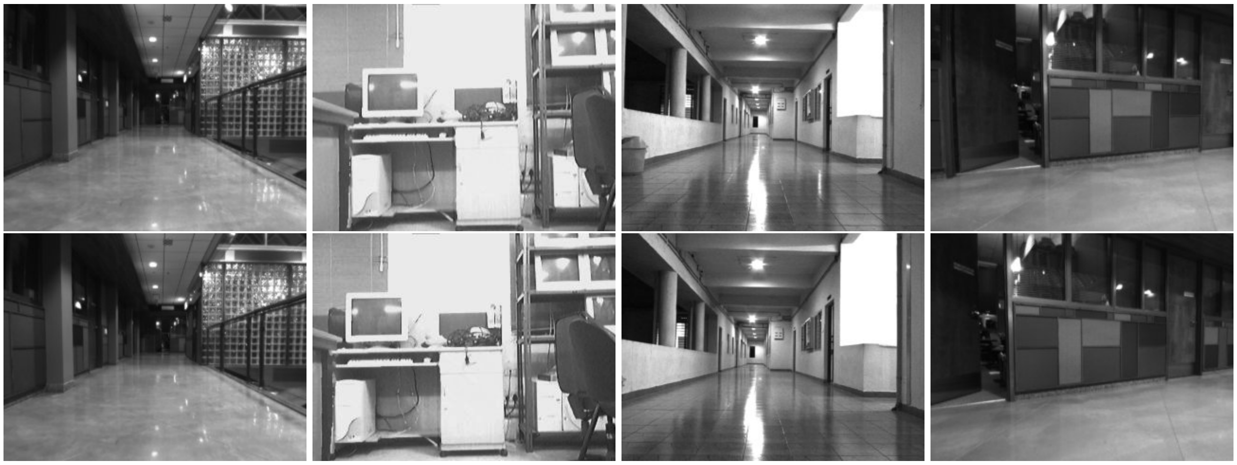

Figure 1. Four examples of qualitative localization in different environments. The top row shows four different query images and the bottom row shows the corresponding matches returned

where θ_{ij} is the rotation angle between the positions i and j, T_{ij} is the relative displacement vector between positions i and j. The parameters α and β are constant scale factors chosen in inverse proportion to the rotational and translational velocity of the robot respectively. Hence the path planner returns fastest known path to the destination in the form of a set of intermediate waypoint nodes the robot must move to in order to reach the destination. Intermediate waypoints are needed because the destination viewpoint may have little or no overlap with the current view.

Once a path through the graph leading to the destination pose has been determined, the robot needs to execute it. The simplest strategy for executing a given visual path through the topological graph is to associate a motion command with each edge in the graph and have the robot perform that action at each step along the path. This scheme can fail if the robot deviates from the path due to odometric errors. The fact that an image taken at each intermediate waypoint is available along with the considerable overlap between consecutive waypoint images suggests a visual servoing solution to the path execution problem. In [16] visual servoing is used to help a robot execute paths in an outdoor environment. In [14], it is argued that exact convergence to intermediate waypoints along a path is not necessary. They propose a qualitative servoing control law that leads the robot towards the next waypoint while not actually enforcing convergence. In [5], the essential matrix between the current view and an intermediate waypoint is estimated and decomposed into camera rotation and translation components. The robot moves in small increments along the direction of camera translation and then rotates to align itself with the waypoint repeating this process until the next waypoint comes into view.

We use a look-and-move strategy to navigate the robot from one waypoint to another waypoint. Features are matched between the image of the current waypoint and another neighboring image in the topological graph which is separated by a baseline. The essential matrix between the two images is estimated using RANSAC [4]. Once estimated the essential matrix can optionally be stored in the graph so that it does not have to be recalculated in the future. The matched features are then triangulated using the estimated essential matrix. The resulting rough 3D model is scaled using the odometric information that has been stored along the edge in the graph. The features in this rough local 3D model are then matched to those in the current view of the robot. The pose of the robot with respect to the current waypoint is then estimated using the pose from 3 points algorithm [4]. The robot then moves directly to the waypoint. Due to errors in reconstruction, pose estimation and odometry, the robot may not converge exactly at the waypoint. However, perfect convergence to intermediate waypoints is not required as these nodes only act as consecutive checkpoints in the sensor space to reach the goal. Moreover, the navigation errors do not accumulate from waypoint to waypoint as they are corrected at every step. The small 'local reconstructions' do not become a part of the robot's representation of the environment, they are only intermediates used to generate a suitable control signal for navigation.

4. Learning a Metric Model of the Environment

In traditional visual SLAM and structure from motion techniques, there are small errors and ocassional gross errors in the camera motion estimates between frames. Even without gross inaccuracies, the errors tend to build up and eventually render the camera pose estimates of frames that appear late in the sequence completely inaccurate. As a result, when the robot travels along a loop and returns to the starting location, the pose estimates at the start and end of the loop are likely to be different, which causes inconsistencies in the built map. Bundle adjustment techniques [20] can help reduce this problem, but can not eliminate it completely and tend to be computationally expensive. To avoid this problem , it is necessary for a mapping robot to have some way of associating the views at the beginning and end of a loop with each other and thus detecting loop closure. Topological graphs are ideally suited to solve the loop closing problem. The graph stores the robot view at each pose during the mapping process. When a newly acquired image matches closely with a node already present in the graph, a link can be created between them thus closing the loop.

This section presents a method through which a topological graph can be converted into a metric model (but not a complete 3D model) by associating a pose estimate with each node in the graph. We assume that the robot moves on a flat planar surface and can rotate only around an axis perpendicular to this plane. This is in general a valid assumption for a wheeled robot working in an indoor environment. The formulation presented could easily be extended to dealing with a full 3D motion model where the robot has 6 degrees of freedom.

Let the database contain n images of the environment. We set up the coordinate frame such that the robot moves in the (x, y) plane and rotates about the z axis. The unknown real pose of the i^{th} image in the database is (x_i, y_i, θ_i). The actual relative pose between the i^{th} and j^{th} image is $(x_{ij}, y_{ij}, \theta_{ij})$, which in polar coordinates is equivalent to $(r_{ij}, \phi_{ij}, \theta_{ij})$ where $r = \sqrt{x_{ij}^2 + y_{ij}^2}$ and $\phi = arctan(\frac{y_j - y_i}{x_j - x_i}) - \theta_i$. The estimated relative pose is $(\hat{r_{ij}}, \hat{\phi_{ij}}, \hat{\theta_{ij}})$. The inverse of the variance or confidence in the relative pose estimate is $< k_{ij}^r, k_{ij}^\phi, k_{ij}^\theta >$.

To obtain relative pose estimates between the i^{th} and j^{th} images, the essential matrix E between these two views is computed using the 5 point algorithm [11]. The essential matrix is then decomposed to give the rotation matrix R and translation vector t between the views as

$$E = [T]_\times R$$

Where $[A]_\times$ is the 3×3 skew symmetric matrix for which the cross product $A \times B = [A]_\times B$. The translation vector t defines only the direction of translation and not the magnitude. It is scaled by the odometric estimate of the distance travelled between the poses where the two frames were captured. Since this pose estimation is done only for pose pairs that are close to each other, the odometric estimates are sufficiently accurate. The translation component in the (x, y) plane and rotation about the z axis are extracted from R and t to give the interframe motion estimate $(\hat{r_{ij}}, \hat{\phi_{ij}}, \hat{\theta_{ij}})$.

To determine the uncertainty of relative pose estimate, a measure of the accuracy of each essential matrix estimate is needed. We assume that the only source of error in the essential matrix calculation are the errors in the image coordinates at which the feature points are detected. Small, zero mean Gaussian perturbations are added to the five sampled image points used to generate the essential matrix and decomposing this 'noisy' essential matrix into R and t. This process is repeated and the variance of the estimated displacement parameters is calculated over all the samples.

The objective of the optimization process is to find the best possible set of camera pose estimates for the n images. The first image added to the graph is assumed to be the origin. Therefore only the poses of the remaining $n-1$ images need to be computed. The problem is modelled as a system of springs. Each spring's energy depends on how much its length is changed along r and how much it is twisted along ϕ and θ. The energy of a single spring in the system is $E = k^r \Delta r^2 + k^\phi \Delta \phi^2 + k^\theta \Delta \theta^2$. If image i and image j have a relative pose estimate between them, they are connected by a spring with natural length $(\hat{r_{ij}}, \hat{\phi_{ij}}, \hat{\theta_{ij}})$ and spring factors k_{ij}^r, k_{ij}^ϕ and k_{ij}^θ along the r, ϕ and θ directions. The optimization is thus performed over $3(n-1)$ pose variables to minimize an energy function $U = f(r_i, \phi_i, \theta i)$. More precisely,

$$U = \frac{1}{2} \sum_{i,j} k_{ij}^r \Delta r_{ij}^2 + k_{ij}^\phi \Delta \phi_{ij}^2 + k_{ij}^\theta \Delta \theta_{ij}^2$$

$$U = \frac{1}{2} \sum_{i,j} k_{ij}^r (r_{ij} - \hat{r_{ij}})^2 + k_{ij}^\phi (\phi_{ij} - \hat{\phi_{ij}})^2 + k_{ij}^\theta (\theta_{ij} - \hat{\theta_{ij}})^2$$

The energy function is formulated in terms of interpose displacements that are related to the absolute pose variables by the following relations.

$$r_{ij} = \sqrt{(x_j - x_i)^2 + (y_j - y_i)^2}$$

$$\phi_{ij} = \arctan\left(\frac{y_j - y_i}{x_j - x_i}\right) - \theta_i$$

$$\theta_{ij} = \theta_j - \theta_i$$

By substituting for the absolute pose variables in the energy function and differentiating we get

$$\frac{\partial U}{\partial x_i} = \sum_j \frac{k_{ij}^r (r_{ij} - \hat{r_{ij}})(x_i - x_j)}{r_{ij}} + \frac{k_{ij}^\phi (\phi_{ij} - \hat{\phi_{ij}})(y_j - y_i)}{r_{ij}^2}$$

$$\frac{\partial U}{\partial y_i} = \sum_j \frac{k_{ij}^\phi (r_{ij} - \hat{r_{ij}})(y_i - y_j)}{r_{ij}} - \frac{k_{ij}^\phi (\phi_{ij} - \hat{\phi_{ij}})(x_j - x_i)}{r_{ij}^2}$$

$$\frac{\partial U}{\partial \theta_i} = -\sum_j k_{ij}^{\phi}(\phi_{ij} - \hat{\phi_{ij}}) + k_{ij}^{\theta}(\theta_{ij} - \hat{\theta_{ij}})$$

Finally, U is minimized numerically using gradient based techniques. For the optimization, the initial pose for the i^{th} node is initialized to $(\tilde{x_{ij}}, \tilde{y_{ij}}, \tilde{\theta_{ij}})$. This initialization is performed using the relative pose estimates. For each path P_j through the graph from the origin to node i, a pose estimate is obtained by adding the relative displacements between node pairs along that path. The weighted mean of the pose estimates from each path is used to determine the initialization. Paths are weighted in inverse proportion to their length from the origin. At first glance it might appear that the algorithm will run in $O(kn^2)$ time where k is the number of iterations the optimization method performs as each iteration requires the computation of $3n$ partial derivatives and each derivative seems be the sum of n terms. However each node in the graph is connected only to the nodes lying in close proximity to it and so most of the k_{ij} terms will be zero. Hence the actual complexity of the algorithm is only $O(kn)$.

The method described above can be compared to bundle adjustment, as it is an optimization process that aims to refine a model of an environment. However, bundle adjustment optimizes the pose of world points and the cameras simultaneously, while our method optimizes only over camera poses. Information regarding world points is passed in the form of parameters to the optimization process.

Qualitative localization based on image retrieval is useful in detecting loop closure, a difficult problem in SLAM and Structure from Motion. The topological graph representation is easy to update when a loop is detected. This suggests the possibility of using topological mapping as an aid or pre-processing step to metric modelling of the environment. Once the workspace has been explored and the connectivity and general structure of the environment have been determined and encoded into the topological graph, it can be used to build a metric model.

5. Experiments and Results

Our experimental setup consists of an indigenously designed and built differential drive robotic platform. The robot is equipped with encoder feedback for providing odometric information, ultrasonic range finders and a monocular camera mounted on a pan-tilt head. The camera used is a Flea2 color camera (from *PointGrey*) fitted with a $5mm$ lens which gives a field of view (FOV) of approximately 50°. Computations are performed by an onboard laptop. In addition to experiments on our robot, we also used the $floor3$ dataset available on the Robotics Data Set Repository [7].

Localization The effectiveness of image retrieval for qualitative localization was tested on the $floor3$ dataset. The dataset consists of a sequence of 512 frames captured by a robot as it moves through one circuit along a closed path in a corridor. Even numbered frames from the sequence were used to create a topological graph of the environment and were added into the adaptive vocabulary forest. All the odd numbered frames were used as query images for testing the accuracy of the global localization. Query images that matched to the closest corresponding frame in the graph were marked as good matches, query images that best matched the second closest frame in the graph were considered OK matches, queries that returned results 3 or 4 frames from the best match were considered poor matches and any other retrieval result was considered a mismatch. Figure 3 shows the index of the image retrieved for each query, the points plotted fall very close to a straight line of unit slope which is the ideally expected behaviour. Table 2 gives the number of matches falling into each category.

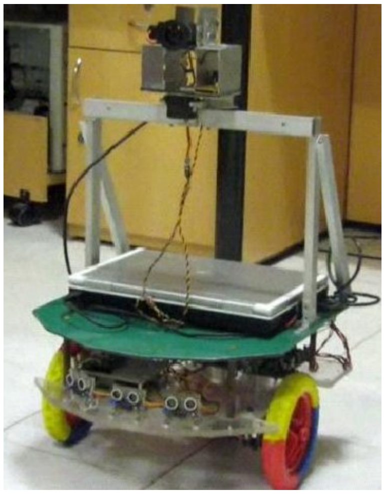

Figure 2. Robotics platform used in experiments

Table 2. Localization Accuracy

Match Type	Number of Matches	Percentage
Good	233	91.01%
OK	16	6.25%
Poor	7	2.74%
Mismatch	0	0%
Total	256	100.0%

Path Planning and Execution The path planning and execution using visual servoing were tested on our robotics platform in a laboratory environment. The robot was given a topological graph of the environment and an image taken from a destination pose. Starting from random locations in

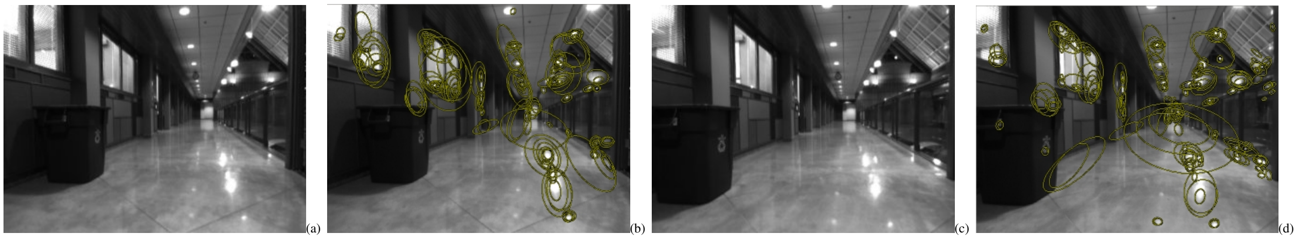

Figure 4. a. A sample query image, b. Keypoints extracted from query, c. The resulting match in the graph, d. Keypoints from the matching image

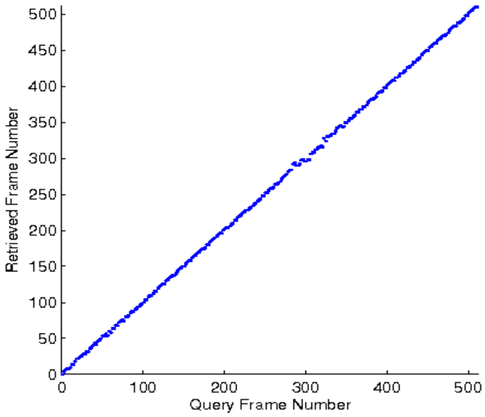

Figure 3. Queries and Results

the lab, the planning algorithm selected paths to the goal destination, images at waypoints along these paths were used for servoing to the destination. Fig. 5 shows the initial and goal position views for one instance of the path planning algorithm along with the intermediate waypoints the robot used to navigate towards the goal.

Metric Model Learning The algorithm for learning a metric model of the environment described in Section 4 was tested on the $floor3$ dataset. The robot started at the origin and moved in a clockwise direction returning to its starting position after completing one circuit around the corridors. Ground truth position information is available for each frame captured by the robot, the blue dashed line in Figure 6 shows the actual path followed by the robot. The magenta coloured line shows the estimate of the trajectory as determined without the metric model learning. Despite the fact that the robot returns to where it started from, the initial and final estimated positions do not coincide due to errors that built up during the camera motion estimation. Since the last few frames in the sequence match closely with the first few frames, the qualitative localization system is able to detect a loop closure near the end of the circuit. When the camera pose estimates are corrected using the metric model learning algorithm, the resulting trajectory estimate is shown in red. The trajectory estimate is closed ensuring that the map remains consistent. Also, as shown in Table 3, the pose estimates along the trajectory are more accurate after the model learning algorithm.

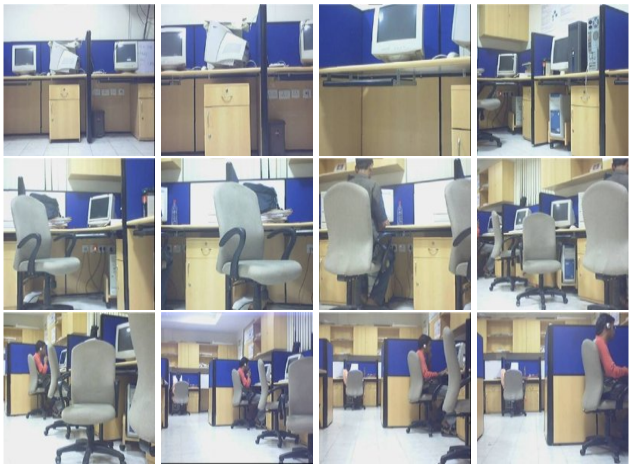

Figure 5. Path Execution: The sequence of waypoints used while travelling from start pose (top left) to the goal pose (bottom right).

Table 3. Pose Estimation Accuracy

	Mean translation error (in mm)	Mean rotation error (in °)
Before	670.9	3.83
After	384.3	1.94

6. Conclusion

We have demonstrated how appearance/image based methods provide a viable alternative to their model based counterparts in solving robot navigation problems. The use

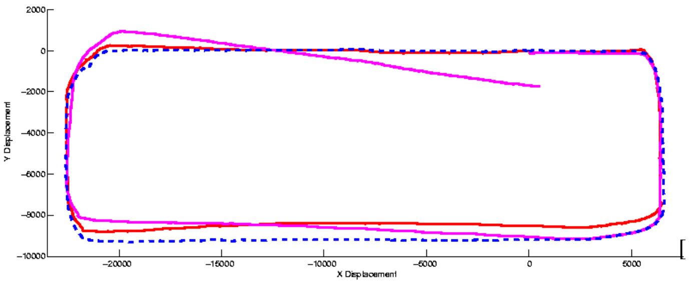

Figure 6. Estimate of the robot trajectory before and after metric model learning

of adaptive vocabulary forests enables the robot to learn a representation that works well with the robot's environment and which can adapt to changes that take place. The use of content based image retrieval for global localization is highly effective and can be scaled to large environments, an essential quality in real world robotics applications. It also allows for effective loop closure detection which ensures that maps built remain consistent. This loop closing also allows the robot to gradually learn the metric structure of the environment from the topological model.

References

[1] M. Cummins and P. Newman. FAB-MAP: probabilistic localization and mapping in the space of appearance. *The International Journal of Robotics Research*, 27(6):647–665, 2008.

[2] A. J. Davison, I. Reid, N. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6):1–16, June 2007.

[3] G. N. DeSouza and A. C. Kak. Vision for mobile robot navigation: A survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(2):237–267, Febraury 2002.

[4] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24:381–395, June 1981.

[5] F. Fraundorfer, C. Engels, and D. Nister. Topological mapping, localization and navigation using image collections. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, pages 3872–3877, 2007.

[6] T. Hashem and Z. Andreas. Global visual localization of mobile robots using kernel principal component analysis. In *IEEE/RSJ Intl. Conf. on Intelligent Robots and Systems*, 2003.

[7] A. Howard and N. Roy. The robotics data set repository (radish), 2003.

[8] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, November 2004.

[9] E. M. T. Maedab and H. Ishiguro. Image-based memory for robot navigation using properties of omnidirectional images. *Robotics and Autonomous Systems*, 47(4):251–267, 2004.

[10] K. Mikolajczyk and C. Schmid. Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, 60(1):63–86, 2004.

[11] D. Nister. An efficient solution to the five-point relative pose problem. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(6):756–770, 2004.

[12] D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *Proc. IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, volume 1, pages 652–659, 2004.

[13] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *IEEE Conf. on Computer Vision and Pattern Recognition*, volume 2, pages 2161–2168, June 2006.

[14] A. Remazeilles and F. Chaumette. Image-based robot navigation from an image memory. *Robotics and Autonomous Systems*, 55(4):345–356, 2007.

[15] D. Santosh, S. Achar, and C. V. Jawahar. Autonomous image-based exploration for mobile robot navigation. In *IEEE International Conference on Robotics and Automation*, 2008.

[16] S. Segvic, A. Remazeilles, A. Diosi, and F. Chaumette. Large scale vision-based navigation without an accurate global reconstruction. In *IEEE Conf. on Computer Vision and Pattern Recognition*, June 2007.

[17] J. Sivic and A. Zisserman. Video google: A text retrieval approach to object matching in videos. In *International Conference on Computer Vision*, 2003.

[18] J. Tardif, Y. Pavalidis, and K. Daniilidis. Monocular visual odometry in urban environments using an omnidirectional camera. In *International Conference on Intelligent Robots and Systems*, 2008.

[19] S. Thrun, W. Burgard, and D. Fox. *Probabilistic Robotics*. MIT Press, Cambridge, 2005.

[20] W. Triggs, P. F. McLauchlan, R. I. Hartley, and A. Fitzgibbon. *Vision Algorithms: Theory and Practice*, chapter Bundle Adjustment for Structure from Motion. Springer-Verlag, 2000.

[21] T. Yeh, J. Lee, and T. Darell. Adaptive vocabluary forests for dynamic indexing and category learning. In *International Conference on Computer Vision*, 2007.

[22] W. Zhang and J. Kosecka. Image based localization in urban environments. In *International Symposium on 3D Data Processing, Visualization and Transmission, 3DPVT*, 2006.

[23] C. Zhou, Y. Wei, and T. Tan. Mobile robot self-localization based on global visual appearance features. In *IEEE International Conference on Robotics and Automation*, 2003.