2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

Autonomous Image-based Exploration for Mobile Robot Navigation

D Santosh*, Supreeth Achar, C V Jawahar

Abstract— **Image-based navigation paradigms have recently emerged as an interesting alternative to conventional model-based methods in mobile robotics. In this paper, we augment the existing image-based navigation approaches by presenting a novel image-based exploration algorithm. The algorithm facilitates a mobile robot equipped only with a monocular pan-tilt camera to autonomously explore a typical indoor environment. The algorithm infers frontier information directly from the images and displaces the robot towards regions that are informative for navigation. The frontiers are detected using a geometric context-based segmentation scheme that exploits the natural scene structure in indoor environments. In the due process, a topological graph of the workspace is built in terms of images which can be subsequently utilised for the tasks of localisation, path planning and navigation. Experimental results on a mobile robot in an unmodified laboratory and corridor environments demonstrate the validity of the approach.**

I. Vision-based Robot Navigation

Vision-based robot navigation [1] has long been a fundamental goal in both robotics and computer vision research. While the problem is largely solved for robots equipped with active range-finding devices, for a variety of reasons the task still remains challenging for robots equipped only with vision sensors. Cameras have evolved as attractive sensors as they help in the design of economically viable systems with simpler sensor limitations.

Several techniques have been proposed and extensively studied in literature to address this problem. They can be broadly classified into model-based and appearance-based approaches [1]. Model-based approaches correspond to the conventional algorithms employing a metric model of the robot's workspace [2]. Features are tracked in the images and a $3D$ reconstruction of them is computed in an off-line process. Localisation is performed by matching features in the model with those observed in the current image and the pose is computed from 3D-2D correspondences. The accuracy of this approach is highly dependent on the features used for tracking, robustness of the feature descriptor and the method for image matching and view reconstruction. In contrast, appearance-based (or view-based) algorithms [3], [4], [5], [6], [7] avoid the need for a metric model by working directly in the sensor space. The environment is generally represented as a topological graph in which each node represents a position in the workspace and stores the sensor readings (*i.e.*, images from the camera) observed at that pose. Pairs of nodes corresponding to positions with a direct path between them are connected with an edge. In this context, localisation reduces to an image-retrieval problem that involves finding within the database the image description that is most similar to its current view. A path to follow is described by a set of images extracted from the database. Control of the robot is either performed by hardcoding the action required to move from one node to another in the graph [4] or by employing a more robust approach in the form of visual servoing [5]. This paradigm is relatively new and is attracting active interest, as the modelling of objects is substituted by the memorisation of views, which is far easier than 3D modelling [3], [6], [7].

Though the appearance-based approaches developed until now have helped us gain a state of maturity in this field, there are certain aspects that need to be further addressed. The proposed algorithms assume that an image database or a topological graph of the workspace is already available to the robot [3], [4], [5]. This information is acquired manually during a training phase where a human-operator guides the robot through the workspace [4]. This is a tedious process as it involves human intervention every time a robot moves to a new workspace, which is particularly difficult for very large environments. This drawback severely limits the applicability of such approaches. Further, the robotic system is restricted to the limited amount of information that is acquired by it during the training stage. As map building is done off-line, it limits the robot workspace only to the explored regions that are visualised during the training stage [3], [5]. It would be rather preferable to dynamically extend the robot workspace into unseen regions in its surroundings. Although a few methods have been developed to automatically organise images of a workspace into a graph (representing the spatial relationship between them) [3], [5], [7], they still demand user intervention to obtain the images and to ensure sufficient sampling of the entire workspace. It must be emphasised that task of exploration is an important aspect of any mobile robot navigation algorithm and forms the basis for the design of several important algorithms (localisation, servoing etc). Hence an autonomous efficient exploration algorithm is very much critical for the overall success of the navigation paradigm.

Rather than limiting image-based paradigms to a simple teach-and-replay scheme, they can be extended to autonomously learn and navigate unknown environments. In this paper, we present a method for systematically exploring an unknown bounded indoor workspace for the first time. The method is analogous to the popular frontier based exploration strategy [8] but involving a robot equipped only with a

D Santosh, S Achar, C V Jawahar are with the Center for Visual Information Technology, International Institute of Information Technology, Hyderabad 500032, India {santosh@research,supreeth@students}.iiit.ac.in, jawahar@iiit.ac.in

*Currently with the Robotics Institute, Carnegie Mellon University, USA.

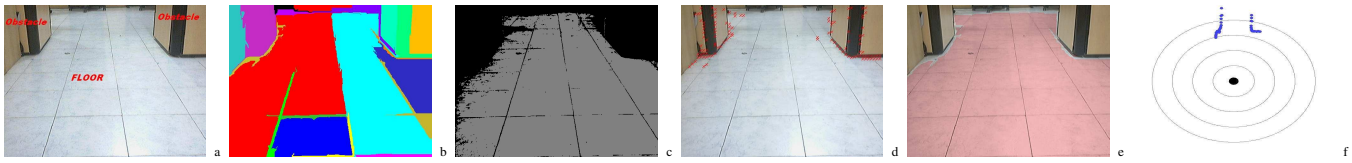

Fig. 1. Inferring horizons using geometric context-based segmentation (a) Original Image (b) Segmented (super-pixeled) Image (c) HSV Image (d) Edge-filtered Image (e) Output Image showing floor region (the boundary indicates the horizon) (f) Polar Plot (blue points indicate obstacle boundaries relative to the robot (black circle))

monocular camera. Specifically, we describe a method to extract frontier boundaries directly from a single image by exploiting geometric context information (Sect. II-A). Using the detected frontiers, the robot is navigated to the unexplored regions (Sect. II-B). The mapping is performed in terms of a topological graph (Sect. II-C). Based on this exploration strategy, we subsequently describe algorithms for localisation, planning and control (Sect. II-D) of a mobile robot in the explored environment. Thus in principle it enables the achievement of a holistic image-based navigation framework.

II. Proposed Image-based Exploration Approach

Most image-based navigation techniques assume that a sequence of images is acquired during a human-guided training step that allows the robot to find paths for moving from its initial position to a goal pose. To overcome this limitation, for the first time, the problem of exploring an unfamiliar environment using a single limited-field of view camera is considered.

The purpose of exploration is to systematically discover and memorise unknown regions of an environment so that a robot can navigate reliably throughout the environment. This topic has received considerable attention in the literature in the context of range-based sensors, the most popular being the frontier-based exploration strategy [8]. In general, all the approaches have, in common, the concept of information gain *i.e.,* moving to the destinations in the world that are most informative for mapping and for increasing the confidence about its location. This problem has attracted recent interest in the context of vision sensors, specifically in the domain of visual SLAM [9]. These algorithms often involve the use of metric maps and can be broadly classified into one of two categories, online algorithms that incrementally update a $3D$ map of the environment [6] and off-line algorithms that process a large set of readings in a batch manner [10]. Further, these methods are dependent on human control or active range sensing for planning and obstacle avoidance. Our proposed strategy is different from the above approaches in terms that exploration is performed autonomously and the mapping is done directly using images, which makes it particularly suited for image-based navigation paradigms.

Our basic approach is similar to the popular frontier-based strategy where the central idea is to gain new information about the world by moving the robot to the boundary between open space and uncharted territory (*i.e.,* the frontiers). The focus here is to estimate the obstacle-free regions from the images and drive the robot towards these navigable regions for increasing its knowledge of the workspace. In this context, the frontiers are more appropriately referred as *horizons*. More precisely, the robots takes an image from its current position and detects all possible horizons from it. The detected horizons are maintained in an open list. One of the frontiers is selected and the robot is moved towards it. It then acquires images from its new position and adds them to a topological map. By moving to successive frontiers, the robot can constantly increase its knowledge of the world and extend its map into new territories until the entire environment has been explored.

A. Inferring Horizons

In this section, we describe the method to infer the horizons directly from an image. To achieve this step, the natural scene structure in regular man-made indoor environments is exploited. In such environments, the floor in the entire workspace is usually a level plane and obstacles when present start at or near the floor level (assuming no overhanging obstacles). Also the appearance of the floor is reasonably different from that of the surrounding walls and the obstacles present in the scene. The basic idea is to segment and identify the ground plane region in the image from the obstacle-occupied areas.

Methods for ground plane extraction typically exploit the appearance or geometry of the ground region. In [11], appearance-based models of outdoor scenes were used to identify the ground plane, vertical structures and the sky region from a single view and to build a rough 3D reconstruction of a scene. In this paper, ground plane extraction from a single image is performed using colour and texture cues. The input image is first over-segmented into a number of super-pixels *i.e.,* contiguous regions with fairly uniform colour and texture [12] (See Fig. 1(b)). Each super-pixel is then labelled as belonging to the ground or non-ground region using the colour and texture cues. The colour-based segmenter assigns a score to every super-pixel representing how likely it is to belong to the floor region based on the colour of its constituent pixels. This is done using an adaptive Hue-Saturation-Value (HSV) histogram approach that models the distribution of ground region pixel colours [13] (See Fig. 1(c)). The membership score (X) of each segment S_i is calculated as

$$X(S_i) = \frac{1}{|S_i|} \sum_{p_j \epsilon S_i} U(H(p_j) - T), \qquad (1)$$

where U is a heavy-side function, $H = f(h, s, v)$ is the histogram probability of the bin corresponding to the hue, saturation, value tuple and T is a threshold determined

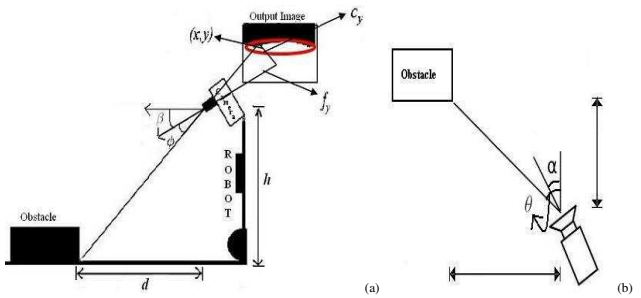

Fig. 2. Determining distance d and orientation θ to obstacles: (a) Profile View (b) Top View of the robot

based on the entropy of the histogram. Super-pixels with a score above a particular threshold are labelled as part of the floor. When placed in a new environment, the HSV histogram is initialised by assuming that a small region directly in front of the robot is open space and that hence a trapezoidal region near the bottom of the initial view can be marked as a part of the ground plane. The texture-based segmenter finds possible boundaries between the ground and non-ground regions by identifying edges across which large changes in image texture take place (using an Sobel operator) as shown in Fig. 1(d). Texture information is measured at regular intervals in small patches on both sides along each edge. If there is a large difference in texture between two patches, and one of the patches is in a super-pixel marked as definitely floor and the second does not, then the second super-pixel is labelled as non-floor. *Texture* here primarily refers to the mean RGB value and the variance of intensity. A new HSV histogram is then re-calculated over all super-pixels in the floor region and is used to assign labels to unlabelled super-pixels using (1). Thus by combining candidate boundary hypothesis with ground plane membership score of each super-pixel, the exact extent of the floor plane can be estimated (See Fig. 1(e)). The boundary pixels of the floor region in the image constitute the horizons.

B. Horizon Boundary Computation

Given the pixel information of the horizons in the image, the polar plot of distance to them can be computed using simple trigonometric relations. Fig. 2(a) shows the profile view of the robot while Fig. 2(b) shows its top view. The height h of the camera center from the ground plane and its tilt angle β from the horizontal are fixed and assumed to be known. Also the camera is assumed to be pre-calibrated *i.e.,* internal parameters K of the camera (specifically the image center c and the focal length f) are already available.

$$K = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

Using the concept of similar triangles, angle ϕ can be derived as $\phi = \arctan(\frac{y-c_y}{f_y})$, where (c_x, c_y) is the camera center and (f_x, f_y) is the focal length. The distance d to the obstacle can then be determined as $d = hcot(\beta + \phi)$. The orientation (or the bearing angle) can be computed as $\theta = \alpha + \arctan(\frac{x-c_x}{f_x})$, where α is the camera pan angle. The polar plot is represented in the form of radial distances d to the visible obstacles indexed by θ *i.e.,* $d(\theta)$. Fig. 1(f) displays the polar plot obtained for the considered image view. It must be emphasised that the computation of the polar plot is simplified by utilising the fact that the ground level is planar and the height of the robot is fixed.

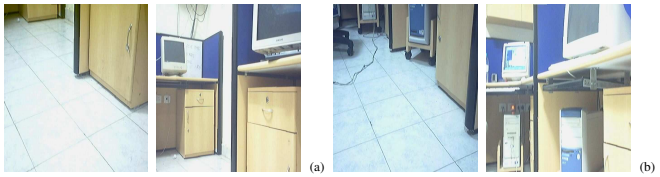

Fig. 3. Two types of images are acquired by the robot. In each sub-figure, the first image is captured with the camera tilted downwards to observe the floor region (for use in the exploration process); while the second image, taken with zero tilt, is utilised in building the topological graph for the purpose of localisation and navigation.

To infer the obstacle free regions, the resultant polar plot is scanned through (radially) for detecting continuous free interval regions, spanning at least c° and at a minimum distance D away from the robot. The value of D is set depending on the minimum distance the camera can view while tilted downwards (here $25cm$) and c is chosen based on the minimum width for the robot to pass through (here 30°). For the detected frontiers, their heading direction is chosen to be the median angle ($\theta^{'}$) of the interval region and the distance of the frontier from the current position is set to $d(\theta^{'})$. If a continuous frontier spans more than 60°, it is split into two sub-frontiers and explored separately. In case of multiple frontiers, the algorithm prefers the ones closer to its heading direction. This biases the robot in favor of moving directly forward rather than following zig-zagged paths. However, it would be preferable to consider a more formal notion of *information gain* for selecting the frontiers.

It should be noted that the computed polar plot is used only for the purpose of finding suitable target locations to explore and is not incorporated into the robot's representation of the workspace. Hence a highly accurate range plot is not necessary.

C. Modelling as Topological Graphs

The visual memory of the robot is modelled as a topological graph. Each node in the graph represents a position in the robot workspace and stores the images acquired by the robot at the pose. Note that the stored images are different from the images used in the horizon computation step and are acquired at a zero-tilt angle (See Fig. 3). The images are acquired at equally spaced pan angles covering a full 360° panorama. These images are added along with their SIFT feature points to the topological graph. Edges in the graph connect navigable straight line paths between two nodes *i.e.,* two nodes are linked by an edge if the robot moved directly from one to the other during the exploration process. They are bidirectional as the robot could move back and forth amongst the nodes. Further, the edges store local geometric information between the adjacent nodes inferred from the epipolar geometry. The epipolar geometry between

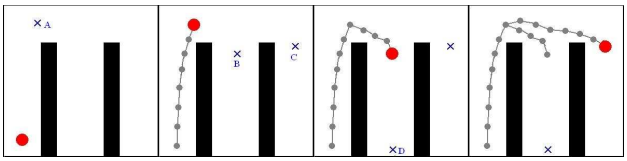

Fig. 4. An example showing four iterations of exploration. The robot is shown in red, unexplored frontiers are marked with blue crosses, nodes and edges in the graph are marked with grey dots and lines respectively. On the last iteration, the robot backtracks to a previous node before travelling to the frontier using the planning and servoing algorithms

two camera views, referred as the essential matrix E, is calculated using the five-point relative pose algorithm [14]. The algorithm can successfully compute E matrix even in cases when the two views contain a dominant plane (which occur frequently in man-made environments), thus tackling planar as well as non-planar scenes. The decomposition of E yields the rotation matrix R and translation vector t (upto scale) [15]. To disambiguate the scale, some prior knowledge about the real world motion is required. The odometric information is utilised for this purpose.

It must be emphasised that once the robot selects a particular frontier to visit, it stops at regular intervals along its way to capture more images. This ensures that the sampling of the workspace is sufficiently dense for the localisation and navigation algorithms to work robustly. In case the path from the current position to the destination frontier is not a simple direct path, it employs the path planning and servoing algorithms to navigate to the desired pose (Sect. II-D). In Fig. 4(b), we observe that the robot detects two frontiers at positions B and C relative to its current pose. It first explores frontier B (and in the process detects another new frontier at D). Next for exploring frontier C, it backtracks to its previous position, using the planning and servoing algorithm, and then proceeds towards C. The overall exploration algorithm is summarised in Algo. 1.

$frontiers \leftarrow \emptyset$;
$initializeGraph(\ )$;
$currentPose \leftarrow initialize(\ )$;
repeat
  $\mathcal{I} \leftarrow captureImages1(currentPose)$;
  $newFrontiers \leftarrow getFrontiers(\mathcal{I})$;
  $frontiers \leftarrow frontiers \cup newFrontiers$;
  $destination \leftarrow nextUnexploredFrontier(\ )$;
  while $(currentPose \neq destination)$ **do**
    $servoRobot(destination)$;
    $\mathcal{I}^{'} = captureImages2(currentPose)$;
    $updateGraph(\mathcal{I}^{'})$;
  end while
  $frontiers \leftarrow frontiers - \{destination\}$;
until $frontiers = \emptyset$;
$storeGraph()$;

Algorithm 1: Image-based Exploration Algorithm

Thus using the proposed image-based exploration algorithm, an autonomous robot can systematically explore an unknown environment and build a topological representation of it. The graph resulting out of this process can be subsequently utilised for localising the robot in this environment and navigating it to perform goal-oriented tasks. In the following, the algorithms required for localisation, planning and servoing using the output of the above exploration approach are described.

Qualitative Localisation In image-based navigation systems, localisation is performed by finding the node in the graph whose image best matches that of the current view [3], [4], [7]. Features typically used for matching include colour histograms, Fourier signatures and local feature descriptors. It must be emphasised that localisation is *qualitative* in nature as the absolute robot pose with respect to a reference frame is not determined; rather the retrieval process only informs that the robotic system is in the vicinity of one of the images from the database [6]. In this work, the image matching is performed by comparing the locally invariant SIFT features extracted from the images [16]. It must be recalled the SIFT features of each image are already indexed in the graph structure. The robot is localised to the image with the largest number of SIFT matches to the current view.

Path Planning Planning of a path between the current robot pose and a desired destination first requires the localisation of the current and the desired image views. Once the two nodes in the graph closest to the current position and the destination are determined, a path through the graph that links them is found. This path will take the form of a number of intermediate image way-points that the robot must move to in order to reach the destination. To find the shortest path (in cases where more than one path is available), Dijkstra's algorithm is used. The edge weight w_{AB} between any two adjacent nodes A and B used for this path planning process is given by $w_{AB} = \alpha|\theta_{AB}| + \beta||T_{AB}||$, where θ_{AB} is the rotation angle between the positions and T_{AB} is the relative displacement vector. α, β are constant scale factors chosen such that w_{AB} also becomes a measure of the time required for the robot to move from node A to B. Hence the path planner returns the fastest available path in the graph to the destination pose.

D. Visual Servo Control

Once a path from the current position to a destination has been determined by the path planner, a servoing algorithm is required to move the robot towards its destination via the intermediate way-points. One could either employ a feed-forward or a feedback based strategy. In case of the former, the meta information stored in the edges of the graph (*i.e.,* the relative displacement values) could be used to drive the robot directly to the desired pose via the way-points. However, to achieve an asymptotic regulation of the robot pose, visual servo control strategies can be utilised as follows.

A look-and-move strategy is used to navigate the robot from one way-point to the next. Wide baseline feature matching [16] is performed to match the the image of the current way-point and another neighboring image in the topological

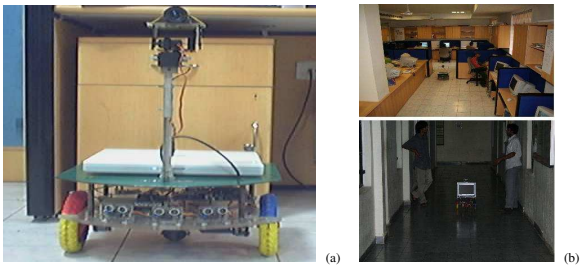

Fig. 5. Experimental testbed: (a) A differential drive robot with a pan-tilt camera and an on-board laptop. (b) The upper right image shows a birds eye view of the lab environment while the bottom displays the corridor environment

graph that is separated by a baseline. The matched features are triangulated using the pre-computed essential matrix (see Sect. II-C) to obtain a coarse $3D$ reconstruction (upto scale). Ambiguity in scale is resolved using the odometric information (stored along the edges in the topological graph). The features in the reconstructed model are matched to those in the current view of the robot. The pose of the robot (rotation R and translation t) with respect to the current way-point is then estimated using the pose from three-points algorithm [17]. The estimated rotation and translation are applied to the robot to displace it to the next way-point. Due to errors in the reconstruction, pose estimation and odometric information, the robot may not converge exactly at the way-point. However, perfect convergence to intermediate way-points is not desired as these nodes only act as consecutive checkpoints in the sensor space to reach the goal. Moreover, the navigation errors do not accumulate from one way-point to next as they are corrected at every step.

III. Experimental Results and Analysis

Our experimental setup is comprised of an indigenously designed and built differential drive robotic platform (with kinematics similar to a unicycle) as displayed in Fig. 5(a). The robot is equipped with encoder feedback, ultrasonic range finders (only for collision detection) and an internally-calibrated camera on a pan-tilt head. The camera used is a Flea2 colour camera (from *PointGrey*) fitted with a $5mm$ lens which gives a field of view (FOV) of approximately $50°$. The proposed algorithms were implemented in Linux on a 1.6 GHz Dell Inspiron 640m laptop mounted on-board. They were tested in indoor workspaces, including laboratory and corridor environments (See Fig. 5(b)). Some of the results are demonstrated in the accompanying video.

Small-scale Exploration First, the exploration algorithm was tested in two of our laboratories which measured approximately $11m \times 6m$. As the field of view of the camera was around $50°$, eight images were taken by panning the camera at each pose to acquire a complete $360°$ FOV (An omni-directional camera could be employed to avoid this step). During the process of frontier detection, the camera was tilted downwards by $30°$ to ensure the floor region close to the robot was within its field of view (See Fig. 3). The exploration algorithm stopped the robot every $50cm$ to take images of the workspace. Fig. 6 illustrates

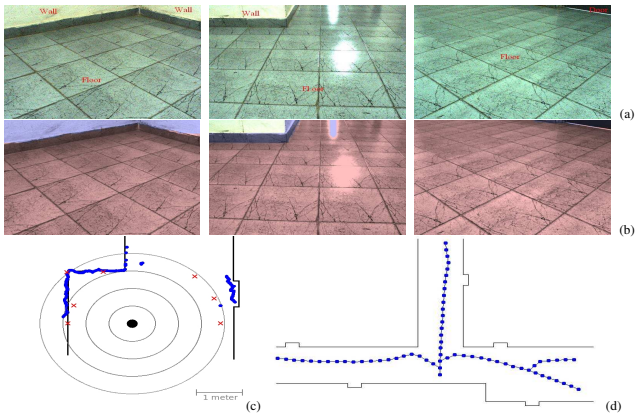

Fig. 7. Exploration in a 'T' shaped corridor environment. Fig.a shows the frontal input images (left, center, right orientations) taken near one of the corners in the corridor. Fig.b shows the corresponding output images with segmented floor region. Fig.c shows the resultant polar plot computed (using all the three images) overlaid on manually measured ground truth. Fig.d displays the poses taken by the robot during the actual exploration run (manually overlaid on a ground truth map).

the process of the frontier-detection method at few example poses in the workspace. The final graph contained about 396 image nodes.

Medium-scale Exploration The exploration algorithm was also tested around the intersection of two corridors. The area marked for exploration measured approximately $15m \times 12m$. Fig. 7(a-c) show the frontier computation process at a particular pose in the corridor. Fig. 7(d) displays the final path traversed by the robot while exploring this workspace. The maximum allowed depth of the graph was limited to 20 steps (approx $10m$) from the starting position for this experiment. The map overlaid is the manually-measured ground truth and the positions shown are the actual positions at which the robot was, when it added the node to the graph. It can be observed that there is some backtracking performed by the robot for returning to an unexplored frontier position near the junction of the corridor. This was done using the path planning and servoing algorithm. The final graph contained 520 image nodes.

Localisation The effectiveness of the exploration algorithm was evaluated by using the resulting topological graph for localisation and navigation experiments. Localisation was performed by matching SIFT features between the view from the current robot pose and the images stored in the topological graph [16]. Fig. 8 shows some query images and the best matches found in the graph. It can be observed that the retrieved images are very close to the query image.

Planning & Servoing Experiments were performed to evaluate the sufficiency of the built topological graph for the purpose of navigation. The robot was placed randomly in the explored workspace and was provided with different destination images to reach. Using the paths selected by the planning algorithm, images along these paths were retrieved

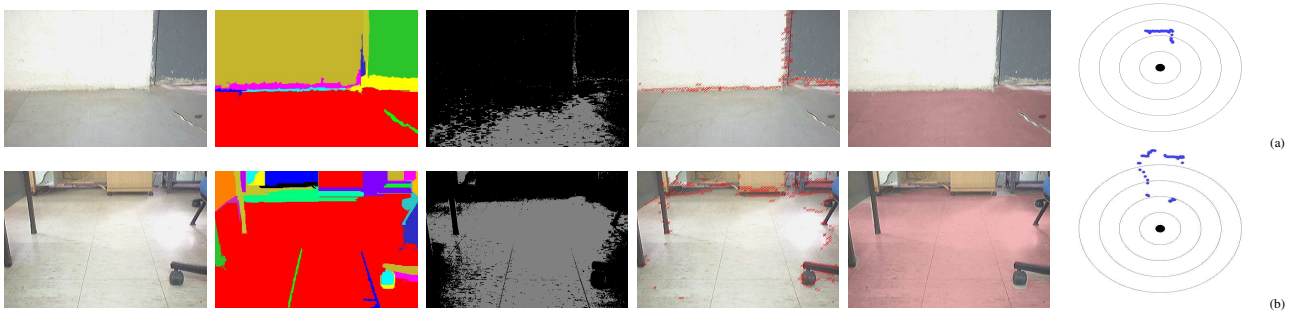

Fig. 6. Horizon Inferencing algorithm on two sample images. The input (first) image is first segmented into super-pixels (second image). Super-pixels belonging to the floor region are identified using a HSV histogram process (third image). This result is combined with the texture-based result computed on the input image (fourth image). The final result (fifth image) displays the identified floor region using the result of both colour and texture cues. The final image displays the computed polar plot.

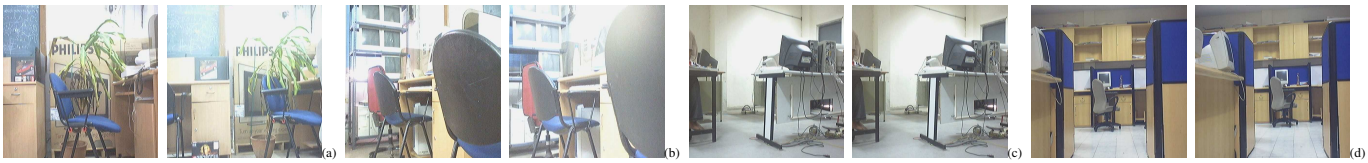

Fig. 8. Localisation result: For each sub-figure, left column shows the query image while the right column displays the retrieved image

and used in the servoing algorithm. Fig. 9 shows an instance of the servoing algorithm in one of the lab environments. It can be observed that the robot is effectively guided to its goal using the retrieved image path.

Fig. 9. Navigation Result: The sequence of images (seen from the robots view) while servoing from start pose (top left) to the goal pose (bottom right).

IV. Conclusion

This paper described a novel image-based exploration algorithm for the task of autonomous vision-based robot navigation. The proposed algorithm detected the frontier boundaries from the images captured by a monocular camera and utilised them to explore the unknown regions of the environment. A topological graph with images acting as nodes was used for modelling the explored workspace. The approach facilitated the robot to autonomously expand its workspace and memorise newly discovered information. We believe that this approach will be essential for mobile robots to progress in the direction of increased applicability.

References

[1] G. N. DeSouza and A. C. Kak, "Vision for mobile robot navigation: A survey," *IEEE PAMI*, vol. 24, no. 2, pp. 237–267, 2002.
[2] E. Royer, M. Lhuillier, M. Dhome, and T. Chateau, "Towards an alternative GPS sensor in dense urban environment from visual memory," *British Machine Vision Conference*, 2004.
[3] G. Blanc, Y. Mezouar, and P. Martinet, "Indoor navigation of a wheeled mobile robot along visual routes," *IEEE ICRA*, 2005.
[4] Y. Matsumoto, K. Sakai, M. Inaba, and H. Inoue, "View-based approach to robot navigation," *IEEE/RSJ IROS*, 2000.
[5] A. Remazeilles and F. Chaumette, "Image-based robot navigation from an image memory," *Robotics and Autonomous Systems*, vol. 55, no. 4, pp. 345–356, April 2007.
[6] R. Sim and G. Dudek, "Comparing image-based localization methods," *Intl Joint Conf on Artificial Intelligence*, pp. 1560–1562, 2003.
[7] F. Fraundorfer, C. Engels, and D. Nister, "Topological mapping, localization and navigation using image collections," *IROS*, 2007.
[8] B. Yamauchi, "A frontier-based approach for autonomous exploration," *IEEE Intl. Sym. on Computational Intelligence in Robotics and Automation*, pp. 146–151, 1997.
[9] N. Karlsson, E. di Bernardo, J. Ostrowski, L. Goncalves, P. Pirjanian, and M. Munich, "The vSLAM algorithm for robust localization and mapping," *ICRA*, pp. 24–29, 2005.
[10] A. W. Fitzgibbon and A. Zisserman, "Automatic camera recovery for closed or open image sequences," *ECCV*, pp. 311–326, 1998.
[11] D. Hoiem, A. Efros, and M. Hebert, "Recovering surface layout from an image," *IJCV*, vol. 75, no. 1, pp. 151–172, 2007.
[12] P. Felzenszwalb and D. Huttenlocher, "Efficient graph-based image segmentation," *IJCV*, vol. 59, pp. 167–181, 2004.
[13] I. Ulrich and I. Nourbakhsh, "Appearance-based obstacle detection with monocular color vision," *AAAI*, pp. 866–871, 2000.
[14] D. Nister, "An efficient solution to the five-point relative pose problem," *IEEE PAMI*, vol. 26, no. 6, pp. 756–777, 2004.
[15] R. Hartley and A. Zisserman, *Multiple view geometry in computer vision*. Cambridge University Press, 2003.
[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *IJCV*, vol. 60, pp. 91–110, 2004.
[17] M. A. Fischler and R. C. Bolles "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. of the ACM* vol. 24, pp. 381–395, 1981