Appearance-based Indoor Navigation by IBVS using Line Segments

Suman Raj Bista, Paolo Robuffo Giordano, François Chaumette

Appearance-based Indoor Navigation by IBVS using Line Segments

Suman Raj Bista[1], Paolo Robuffo Giordano[2] and François Chaumette[1]

***Abstract*—This paper presents a method for image-based navigation from an image memory using line segments as landmarks. The entire navigation process is based on 2D image information without using any 3D information at all. The environment is represented by a set of reference images with overlapping landmarks, which are acquired during a prior learning phase. These reference images define the path to follow during the navigation. The switching of reference images is done exploiting the line segment matching between the current acquired image and nearby reference images. Three view matching result is used to compute the rotational velocity of a mobile robot during its navigation by visual servoing. Real-time navigation has been validated inside a corridor and inside a room with a Pioneer 3DX equipped with an on-board camera. The obtained results confirm the viability of our approach, and verify that accurate mapping and localization are not necessary for a useful indoor navigation as well as that line segments are better features in the structured indoor environment.**

***Index Terms*—Visual-Based Navigation, Visual Servoing.**

I. INTRODUCTION

THERE must exist a close relationship between the perceived environment and the controller of a robot for its autonomous navigation. Such a relationship is often defined w.r.t. the features extracted from the sensors (e.g. images from a camera) and associated with the real world landmarks. For this, we need some internal representation of the environment. The environment can be represented either in the 3D space or in the sensor space. The first approach relies on the knowledge of an accurate and consistent 3D model of the navigation space. The navigation is then performed by matching the global model with a local model deduced from sensor data. Such a model can be computed from different features like lines, planes, or points [1], or estimated from a learning step. Most of the simultaneous localization and mapping (SLAM) methods [2], [3], [4] fall in this category. The second approach, also known as appearance-based approach, does not require a 3D model of the environment, but it has instead the advantage of working directly in the sensor space. To simplify the process of appearance-based navigation, the navigation environment is generally represented topologically in a graph [5], [6], [7]. The nodes of the graph give characteristic features or zones of the environment (locations) obtained using the sensor data, and arcs give adjacency relations between locations. Such maps are built in a prior offline mapping phase. Navigation is then usually performed by computing a similarity score between the view acquired by the camera and the different images of the database, or by using the features extracted from previous images via tracking and generating associated control command. This similarity can be based on global descriptors, like considering the whole image [8], [9], color histograms [10], or image gradient [11]; or by using local descriptors, like photometric invariants [12] or local feature points like corners, Scale-Invariant Feature Transform (SIFT)/ Speeded Up Robust Features (SURF) points or Maximally Stable Extremal Regions (MSER) [5], [6], [7], [13].

The work in [7] has demonstrated indoor navigation of a mobile robot using a visual memory for both perspective and omni-directional cameras. The robot is controlled by visual servoing based upon the regulation of successive homographies. In [5], [6], the authors have demonstrated a hybrid model for topological navigation based on a visual memory in an outdoor environment. Local 3D reconstruction has been used for verifying the key-point matches and automatic key-frame selection using SIFT, Multi Scale Harris, and MSER features. However, the motion control was still based upon 2D features, in particular, the centroid of matched points. They also show that it is not necessary to converge towards each intermediate position (key frames) as long as it is possible to reach the final position. Hence, the use of qualitative servoing [14] eliminates the necessity of a database accurate enough to get satisfying trajectories regarding the initial and desired positions, contrary to [13] where the robot converges to the intermediary position using visual servoing by minimizing the error between the current and successive desired positions of visual landmarks.

From the above literature, one can then conclude that accurate mapping and localization are not mandatory for visual navigation. Robots are able to navigate using this approach in urban environments and in all places where local point based features are abundant. In this respect, the goal of this paper is to adopt this approach for indoor navigation by exploiting *line segments* as visual landmarks. Indeed, a typical navigation task in an indoor environment can be divided into two parts: a) Navigation through corridors and b) Navigation inside rooms. For the latter case, it is more likely to have abundant distinctive local features and global features whereas, for the former case, the perceived surface may not give enough features points for navigation. Moreover, similar texture or lack of texture may

Manuscript received: August, 31, 2015; Revised November, 25, 2015; Accepted January, 13, 2016.

This paper was recommended for publication by Editor Jingshan Li upon evaluation of the Associate Editor and Reviewers' comments.

This work has been supported by the Brittany Council and Oseo Romeo 2 project.

[1] S. R. Bista and F. Chaumette are with Inria at Irisa, Rennes, France suman-raj.bista@inria.fr, francois.chaumette@inria.fr.

[2] P. Robuffo Giordano is with the CNRS at Irisa, Rennes, France prg@irisa.fr.

Digital Object Identifier: Not Available Yet

result in false matching. However, in indoor environments, line segments are abundant. In addition to this, line segments are more robust to partial occlusions and more resilient to motion blur [4], [15]. Also, line segments in the image can be detected quickly and accurately by line algorithms like Line Segment Detector (LSD) [16] and Edge Drawing Lines (EDLines) [17]. Because of all these reasons, this work will explore the use of line segments as visual landmarks. This, however, requires a suitable modification/extension of all the steps that have been previously designed for point features.

A. Navigation based on line segments

Tracking/matching of multiple line segments is still an open problem in computer vision. This is due to inaccurate locations of line endpoints, fragmentation of lines, lack of strongly disambiguating geometric constraints for an image pair, and lack of distinctive appearance in low-texture scenes [18], [19], [20]. Despite these problems, [3], [4], [21] have demonstrated line segments-based navigation, but using a 3D model-based approach. In [3], a model-based SLAM using 3D lines as landmarks has been presented, where unscented Kalman Filters are used to initialize new line segments and generate a 3D wire-frame model of the scene that can be tracked with a robust model-based tracking algorithm. The authors of [21] have extended the monocular SLAM using points [2] to line segments, where Kalman filters are used to track the lines. Both methods rely on control points (a set of sample points placed along the line) for tracking, which, however, is not suitable when line segments are close to each other because of failure in tracking. Recently [4] has used Nearby Line Tracking to track lines and an Extended Kalman Filter (EKF) is used to predict and update the state of the camera and line landmarks.

In [22], [23], [24], a vision-based corridor navigation algorithm has been proposed that uses the vanishing point extracted from corridor guidelines for the Nao humanoid robot, a wheelchair and a mobile robot respectively. The first two works are map-less methods. In [24], the vanishing point is used for the heading control whereas an appearance-based process is used to monitor the robot position along the path. A set of reference images are acquired manually at relevant positions along the path which correspond either to areas in the workspace where some special action can be undertaken (e.g doors, elevators, corners, etc.) or viewpoints where very distinctive images can be acquired. During navigation, these reference images are compared with current images using the Sum of Squared Differences (SSD) metric. The solution in [15] not only uses two pairs of natural line and point, but also the odometer data (to determine the height of the landmarks) for the visual localization.

B. Main Contribution

Our main contribution is a complete method for indoor navigation (automatic construction of a navigation route, initial localization that enables the robot to start from any position within the map, successive localization and a control law for choosing the rotational velocity) that coarsely follows the learned path by just using the information provided by the 2D line segments detected in the image *without* need of accurate mapping, localization and robot odometry. To our knowledge, the closest works to ours that use image memory are [5], [6], [7], which, however, still use 3D information for navigation based on point features. The approach proposed in this paper is instead different from the available literature as our method *only* exploits 2D line segments detected in the image and does not depend upon specific types of lines (e.g. vertical lines or corridor lines). Indeed, we show that the information obtained from the 2D line segment matching between the current acquired image and nearby reference images is enough for automatic switching of key images and for robot control *without* 3D reconstruction.

The next section describes the complete framework for mapping and navigation. Section III presents experimental results with a real robotic system, which demonstrate the validity of the proposed navigation scheme, and advantage w.r.t. classical point features. Finally, concluding remarks are reported in Section IV.

II. NAVIGATION FRAMEWORK

A. Constraints

We consider a non-holonomic mobile robot of unicycle type equipped with a fixed perspective camera as the only sensing modality. The intrinsic parameters of the camera are constant and coarsely known. The presented framework is concerned only with a goal-directed behavior without considering obstacle avoidance, which will be considered in future works. Thus, in the navigation experiments we assume that other moving objects will adopt collision-free trajectories, while a human supervisor is responsible for handling the emergency stop button. The devised control scheme exhibits a qualitative path following behavior, since the learned path in general is not tracked precisely. It is therefore suitable to prefer the center of the free space during the acquisition of the learning sequence. During navigation, it is assumed that the robot is initially inside the mapped environment. The localization outside the mapped location is out of scope of this paper.

B. Line Segments Matching

For matching the line segments, there exists a considerable number of works on this topic [18], [19], [20], [25], [26]. In this work, we use the line matching method proposed by [20] to generate pairwise matches, which utilizes Line Band Descriptors to get candidate matches at first, and then exploits geometric constraints and topological filters to eliminate the false matches. To detect line segments, EDLines detector [17] has been used. These methods have been selected because of their high accuracy and computational speed.

For two views, the line segments do not provide strong geometric constraints as opposite to points (epipolar geometry). The trifocal tensor provides instead a strong geometric constraint for lines, but it requires line correspondences in three views. Let $T = [T_1, T_2, T_3]$ be the $3 \times 3 \times 3$ trifocal tensor, T_1,T_2 and T_3 be the individual 3×3 matrices of T, and $l_1 \leftrightarrow l_2 \leftrightarrow l_3$ be the line correspondences in three views: these

are 3×1 vectors representing line parameters. By letting $[l_1]_\times$ represent the skew-symmetric matrix associated to vector l_1, and l_2^T represent the transpose of vector l_2, the trifocal tensor T can be estimated by the following relationship [1]

$$(l_2^T[T_1, T_2, T_3]l_3)[l_1]_\times = 0. \quad (1)$$

The estimation of the trifocal tensor with Random Sample Consensus (RANSAC) [1] can be used to verify the line segment correspondences in three views. However, the cost function associated with the trifocal tensor is computationally more expensive than in the fundamental matrix case when used with RANSAC. In addition, at least 13 line correspondences are required to compute the trifocal tensor (instead of only 6 point correspondences) for three views. Nevertheless, the number of outliers in three views matching is quite low compared to two views only, which makes it possible for the RANSAC-based estimation to converge in few iterations. The process is described in [1].

In our method, two view matches are used in initial localization, switching of key images and generating three view correspondences. Three view matches are used in mapping, switching of key images and motion control. For three view matching, the current key image and the two most recently acquired images are used during the mapping, whereas, the two key images and the currently acquired image are used during the navigation. When obtaining the three view correspondences, only the matched lines between the first two images are used to match with the third image in order to reduce the cost of matching (see Fig. 1).

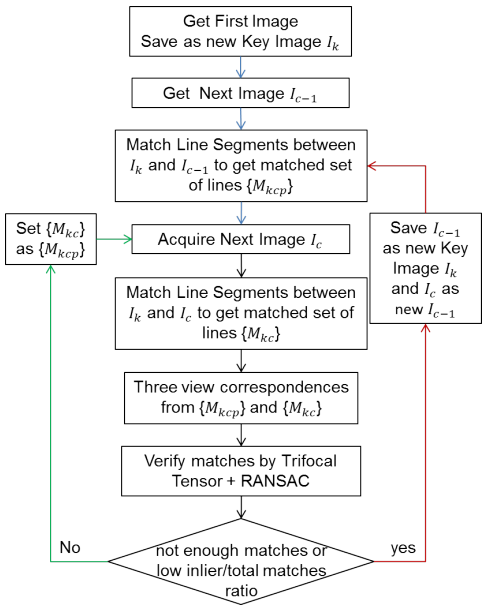

Fig. 1. Building the map from line segments.

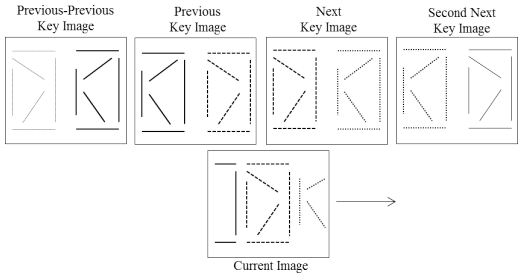

Fig. 2. The map consists of key images and line segments. Adjacent key images share some line segments with the current image. These corresponding line segments with the current acquired image are used for motion control with the aim of following the arc defined in the map.

C. Mapping from line segment

Mapping or learning a path starts with driving the robot on a reference path under manual control. The acquired images are used to automatically create a map based on matching of the line segments across the views and organizing them within an adjacency graph. A correct mapping is important for a successful navigation. It is not always necessary to perform the mapping in real time. Therefore, it is better to use verification of the matching using trifocal tensor and RANSAC to get a better set of key images for representing the environment.

The key image selection procedure is sketched in Fig. 1. The first acquired image is always stored in a database as a key image (first node in the topological map). Let I_{c-1} and I_c be the two most recently acquired images and I_k be the most recent key image. For the case just after the new key image is set, I_{c-1} and I_c are the two images acquired successively after I_k. The detected line segments of I_k are matched with I_{c-1} to get the first set of matched lines $\{M_{kcp}\}$. The lines in I_k present in $\{M_{kcp}\}$ are matched with the detected line segments in I_c to get the second set of matched lines $\{M_{kc}\}$. The common line segments in $\{M_{kcp}\}$ and $\{M_{kc}\}$ give three view correspondences. If there are not sufficient number of lines (for example less than 20) after three view matching, or a low ratio (for example less than 0.5) of inlier to total number of matches after trifocal tensor estimation with RANSAC, I_{c-1} is saved in the database as a recent key image I_k, and I_c becomes the new I_{c-1}. Otherwise, $\{M_{kc}\}$ becomes $\{M_{kcp}\}$ and the line segments of next acquired image I_c and I_k are matched to get a new set of $\{M_{kc}\}$. This idea is similar to tracking the line segments of the key image in successive frames. Then the process continues. The last acquired image is also stored in the database, which helps to determine when the robot has to stop at the end of the navigation. Three view matching is always done between the current key image I_k and the two most recently acquired images I_{c-1} and I_c. Hence, the output of the mapping process is a set of key images that represents the arc the robot has to follow during the navigation. The neighboring key images share some common line segments as shown in Fig. 2, which makes it possible to consider multiple key images in a neighborhood for defining the heading angle of the robot.

D. Navigation in the map using line segments

After the mapping phase, a set of key images that represents the nodes of the adjacency graph of the environment is available. For simplicity, a linear map is considered here. The navigation process can be divided into two tasks: a) initial localization in the map, and b) successive localization in the map and motion control. The topological location corresponds

to the actual arc of the graph, which determines the two key-images used for visual servoing. For smooth motion and switching of key images, one more key image next to current key images in the forward direction is also used (see Fig. 2).

1) Initial Localization: The navigation starts with the initial localization where the first image acquired (I_a) is compared with all the images in the database based upon line segment matching. Initial localization helps to determine the initial position of the robot in the map. This enables to start the robot from any position in the mapped location. The key image with the maximum number of matches is selected. Let it be I_k. Then the adjacent key image with second maximum matches is also selected. This image can be either I_{k+1} or I_{k-1}. If the robot is assumed to be moving along the same direction as the images arranged in the database, I_a is between I_{k-1} and I_k, or I_k and I_{k+1}. For simplicity, we denote the previous key image as I_P and the next key image as I_N. Hence, the position of I_a in the topological map is in between I_P and I_N as shown in Fig. 3.

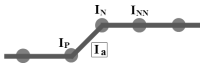

Fig. 3. Localization in the map represented by the topological graph.

2) Successive Localization: After initial localization in the map, further localizations can be done by just comparing with few adjacent images in the database. The previous key image I_P, the next key image I_N and the second next key image I_{NN} are compared with the current acquired image I_a. Let $n(...)$ be the number of lines matched between the images. Then switching of key images is done when at least one of the following criteria is fulfilled for two consecutive acquired images I_a and I_{a+1}:

$$n(I_a, I_N, I_{NN}) > n(I_P, I_a, I_N) \text{ or}$$

$$n(I_a, I_{NN}) > n(I_a, I_N) \;\&\&\; n(I_a, I_{NN}) > n(I_p, I_a).$$

The first criterion is based on the result of three view matching between the images inside the brackets, whereas the second criterion is based on two view matching of images. The second criterion is essentially useful when there are no three view matches or very few number of three view correspondences. Such a condition may sometimes occur with sharp turns in corridors having no texture at all. After switching the images, I_N becomes I_P, I_{NN} becomes I_N, and next key image from I_N becomes I_{NN}. Then the process repeats. When the end of the database is reached, I_{NN} will not be available and I_N will be the last image acquired during the mapping. So, the navigation needs to be stopped. Otherwise, the robot will be moving out of the mapped environment.

E. Motion Control

For navigation, the robot is not required to accurately reach each reference image of the path, or to accurately follow the learned path. In practice, the exact motion of the robot should be controlled by an obstacle avoidance module [5], which will be the future work. Therefore, Image-Based Visual Servoing (IBVS) [27] is the adequate strategy for such purpose. The rotational velocity is derived from the matched lines between I_a, I_N and I_{NN}, whereas the translational velocity is kept constant and reduced to smaller constant value when turning. Such turnings are automatically detected by looking at the commanded rotational velocity.

Let us define a vector of visual features as $\mathbf{s}$, the camera velocity expressed in camera frame as $\mathbf{u_c} = (v_{cx}, v_{cy}, v_{cz}, \omega_{cx}, \omega_{cy}, \omega_{cz})$ and the robot velocity as $\mathbf{u} = (v_r, \omega_r)$, where v is the linear velocity and ω is the rotational velocity around the given axes. The velocity of $\mathbf{s}$ can be related via an interaction matrix $\mathbf{J_s}$ [27] to $\mathbf{u_c}$ as

$$\dot{\mathbf{s}} = \mathbf{J_s u_c}. \tag{2}$$

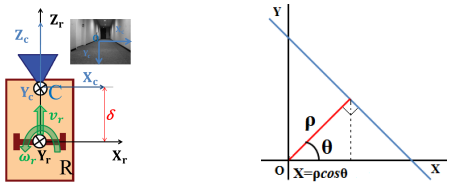

Fig. 4. Top view of robot (orange) equipped with a perspective camera (blue) with its optical axis perpendicular to axis of robot rotation, (Left) and Representation of line in polar form (Right).

For the considered unicycle-like robot (Fig. 4 (left)), $\mathbf{u_c}$ can be expressed in terms of (v_r, ω_r) as

$$\mathbf{u_c} = (-\delta\omega_r, \quad 0, \quad v_r, \quad 0, \quad -\omega_r, \quad 0), \tag{3}$$

where δ is the distance between the camera center and the robot center of rotation. From (2) and (3), we obtain

$$\dot{\mathbf{s}} = \mathbf{J}_v v_r + \mathbf{J}_\omega \omega_r, \tag{4}$$

where $\mathbf{J}_v$ and $\mathbf{J}_\omega$ are the Jacobian associated with v_r and ω_r respectively. In order to drive $\mathbf{s}$ to desired value $\mathbf{s}^*$, we control ω_r as [27]

$$\omega_r = -\mathbf{J}_\omega^+(\lambda(\mathbf{s} - \mathbf{s}^*) + \mathbf{J}_v v_r), \tag{5}$$

where λ is a positive gain, and $\mathbf{J}_\omega^+$ is the pseudo-inverse of $\mathbf{J}_\omega$. The expression of $\mathbf{J_v}$ and $\mathbf{J}_\omega$ can be obtained as follows. In 3D, a straight line can be represented by the intersection of two planes:

$$a_i x + b_i y + c_i z + d_i = 0, \; i = 1, 2. \tag{6}$$

Except for the degenerate cases ($d_1 = d_2 = 0$), a 3D line in a scene projects onto the image plane as a 2D line. As in [28], we choose to parameterize line segments with parameters (ρ, θ) as

$$X\cos\theta + Y\sin\theta - \rho = 0. \tag{7}$$

The interaction matrix related to ρ and θ is given by [28]

$$\begin{aligned} L_\rho &= [\lambda_\rho C_\theta \;\; \lambda_\rho S_\theta \;\; -\lambda_\rho \rho \;\; (1+\rho^2)S_\theta \;\; -(1+\rho^2)C_\theta \;\; 0] \\ L_\theta &= [\lambda_\theta C_\theta \;\; \lambda_\theta S_\theta \;\; -\lambda_\theta \rho \;\; -\rho C_\theta \;\; -\rho S_\theta \;\; -1], \end{aligned} \tag{8}$$

where $C_\theta = \cos\theta$, $S_\theta = \sin\theta$, $\lambda_\rho = (a_i\rho\cos\theta + b_i\rho\sin\theta + c_i)/d_i$ and $\lambda_\theta = (a_i\sin\theta - b_i\cos\theta)/d_i$. Since we only control ω_r, only one feature derived from all line segments is sufficient. We have chosen the abscissa of the centroid of the points of intersection of the matched lines and their respective normal from the origin. For a given line as shown in Fig. 4 (right), $X = \rho\cos\theta$ gives the abscissa of such a point.

For a set of n matched lines between I_a, I_N and I_{NN}, we define:

$$\begin{aligned} X_{ai} &= \rho_{ai}\cos\theta_{ai}, & X_a &= \tfrac{1}{n}\sum_{i=1}^{n} X_{ai}, \\ X_{Ni} &= \rho_{Ni}\cos\theta_{Ni}, & X_N &= \tfrac{1}{n}\sum_{i=1}^{n} X_{Ni}, \\ X_{NNi} &= \rho_{NNi}\cos\theta_{NNi}, \text{ and } & X_{NN} &= \tfrac{1}{n}\sum_{i=1}^{n} X_{NNi}. \end{aligned} \tag{9}$$

Hence, our visual feature is $s = X_a$ and the desired feature is $s^* = X_N$. We then have

$$\dot{s} = \dot{X}_a = \frac{1}{n}\sum_{i=1}^{n}(\dot{\rho}_{ai}\cos\theta_{ai} - \rho_{ai}\sin\theta_{ai}\dot{\theta}_{ai}). \tag{10}$$

From (4), (8) and (10), we obtain

$$\begin{aligned} J_v = \;& \tfrac{1}{n}\sum_{i=1}^{n}(\lambda_{\theta ai}\rho_{ai}^2\sin\theta_{ai} - \lambda_{\rho ai}\rho_{ai}\cos\theta_{ai}). \\ J_\omega = \;& \tfrac{1}{n}\sum_{i=1}^{n}((1+\rho_{ai}^2)\cos^2\theta_{ai} - \rho_{ai}^2\sin^2\theta_{ai}) \\ & +\delta(\lambda_{\theta ai}\rho_{ai}\sin\theta_{ai}\cos\theta_{ai} - \lambda_{\rho ai}\cos^2\theta_{ai})). \end{aligned} \tag{11}$$

Neglecting δ with respect to d_i (distance of line from image plane), and assuming the camera optical axis is orthogonal to the axis of robot rotation and that the centroid stays near the image plane center, (11) can be approximated as

$$J_v \simeq 0 \text{ and } \; J_\omega \simeq \frac{1}{n}\sum_{i=1}^{n}(\cos^2\theta_{ai} - \rho_{ai}^2\cos(2\theta_{ai})) = J_a. \tag{12}$$

Since visual servoing is known to be robust against modeling errors [29], such approximations are reasonable. Thus, from (5) and (12) we finally obtain the following expression for the rotational velocity:

$$\omega_r = -\frac{\lambda}{J_a \pm \epsilon}(X_a - X_N), \tag{13}$$

where ϵ is a small constant to prevent possible division by zero. In order to smooth the rapid steering actions when switching between frames, a feed-forward command is also added to ω_r. The calculation of the feed-forward term is based on the difference of the centroids between the shared lines of I_a with I_N and I_{NN}. The final equation is given as follows

$$\omega_r = -\frac{\lambda}{J_a \pm \epsilon}(h_1(X_a - X_N) + h_2(X_a - X_{NN})), \tag{14}$$

where h_1 and h_2 are positive weights such that $h_1 + h_2 = 1$.

Thus, our complete framework uses only the *2D* information obtained from the *line segments* matching, without requiring any 3D information, which was not the case in previous works. From this 2D information, we derive the required rotational velocity using *IBVS*, which makes the robot to follow the learned path successfully without any need of the accurate mapping or localization.

III. EXPERIMENTAL RESULTS

The experiments were performed with a Pioneer 3DX equipped with an AVT Pike 032C camera module. All computations, except for the low-level control, were performed on a laptop with 3-GHz Intel Core i7-3540M CPU. The image resolution in the experiments was 640×480. The mapping was done offline, whereas the navigation experiment was performed online at 6 Hz. The acquisition of images and the high-level motion control for the Pioneer were done through the interface provided by ViSP [30]. For line segments detection and matching, the implementation provided by the MIP group [1], University of Kiel, has been used with some modifications as per our requirements. The image coordinates have been normalized by the camera intrinsic parameters before deriving the rotational velocity. The experiments have been performed in an indoor environment, i.e., inside a room and a corridor. Even though simple navigation path with linear and curved trajectories have been used in the experiment, the method can be easily extended for the graphs with intersections and multiple paths. The qualitative results of mapping and navigation using different trajectories in corridor and inside the room are now presented.

A. Experiment I: Inside a room

1) Mapping: 617 images have been acquired as the learning sequence. 18 images shown in Fig. 5 have been selected automatically from the mapping algorithm described in Sect. II-C as key frames. The trajectory obtained from the odometry is shown by a red curve in Figs. 6 and 7, where the red symbol $*$ represents the location of the key images. The obtained key images are able to represent the learned path. There are more key images over a small distance in case of quick displacements of features like in turnings or when line segments cannot be successively matched over the sequence due to changes in illumination.

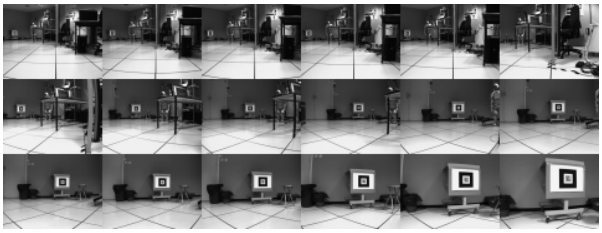

Fig. 5. Key images of the robotics room.

2) Navigation: The robot was placed inside the mapped environment with the camera facing towards the mapped direction (initial position shown by green dot). The forward velocity was set to 0.2 m/s. During navigation, the robot was able to follow the learned trajectory as shown by the blue curve in Figs. 6 and 7, with automatic switching of the reference images. Figure 6 shows the navigation of the Pioneer in the map without any change in environment from the time of mapping.The navigation in presence of obstacles is shown in Fig. 7 (left). Even during a continuously obstructed view by walking in front of the camera (as shown in Fig. 8 (left)), the robot was still able to follow the desired path. Fig. 7 (middle) shows the navigation with some changes in the room as shown in Fig. 8 (third column), where the table was moved from the

[1]http://www.mip.informatik.uni-kiel.de/tiki-download_file.php?fileId=1965 [Accessed: August 24,2015].

end to the middle of the room and replaced by a chair and the stool was pushed further from the time of mapping. Fig.

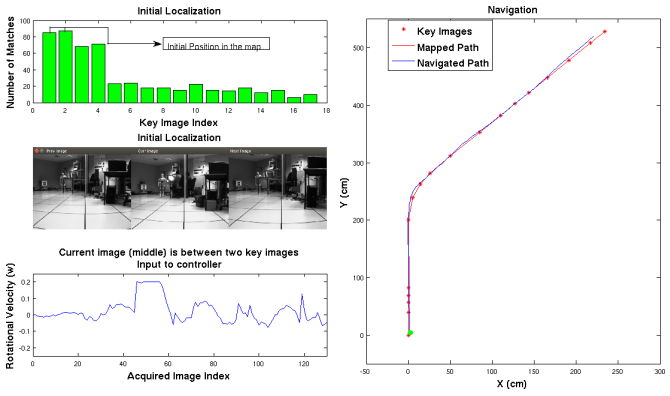

Fig. 6. Initial localization and navigation inside the robotics room.

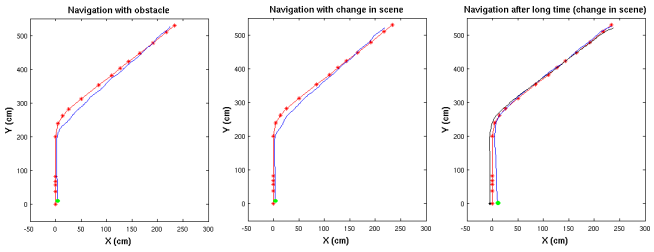

Fig. 7. Navigation inside the robotics room in presence of people (left) and changes in mapped scene (middle and right).

Fig. 8. Obstruction in view (left), mapped scene (second column) and changes in scene (third column, right).

7 (right) shows navigation in the room performed 6 months later than the mapping stage with many changes and dynamic objects in the scene like a table, chairs, boxes, etc (Fig. 8 (right)). The successful navigation in these latter cases was possible due to the presence of sufficiently large number of line matches from the static objects like ceilings, floor tiles, posters, and pillars. In all cases, the drift was within 3cm from mapped position.

B. Experiment 2: In a Corridor

1) Mapping: Out of 1208 images acquired in the corridor, 45 have been selected automatically as the key images (Fig. 9). Similarly, 53 key images have been obtained from 1083 images of the same corridor taken from a reverse direction (Fig. 10). The mapping has been done with all doors closed except one. This was meant to ensure that illumination from the room and outside windows has negligible effects. The obtained key images represent a path of length 32 meters. The distribution of key images concentrated at the turnings and when line segments of key images cannot be successively matched over the sequence.

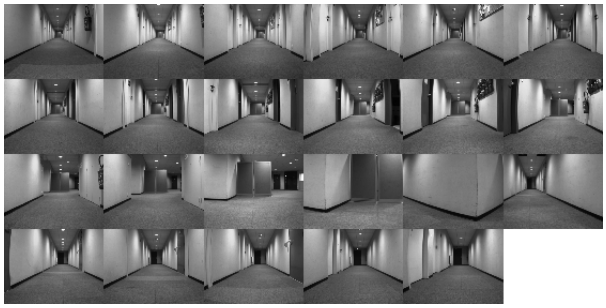

Fig. 9. Odd key images (1^{st},3^{rd},5^{th},..) of the corridor.

Fig. 10. Odd key images (1^{st},3^{rd},5^{th},..) of the corridor from reverse direction.

2) Navigation: Figures 11 and 12 show navigation in the corridor. The robot was placed inside the mapped location (initial position shown by green dot). The forward velocity was set to 0.15m/s and reduced to 0.075m/s when turning, whereas the rotational velocity was controlled by the navigation algorithm. Even with the open doors, people walking in the corridor and blur in some images as shown in Fig. 13, the robot was still able to navigate successfully with turning whenever it was required. Right angle turning is a challenging task, in the

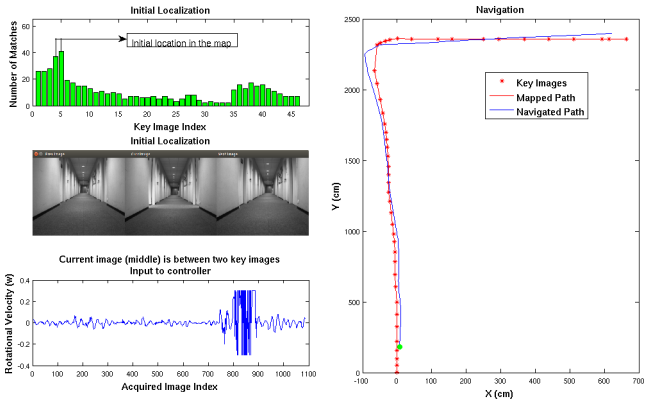

Fig. 11. Navigation in the corridor.

sense that there are few lines with fast changes. However, the key images obtained from the mapping part were still able to handle such situations. The lateral drift when navigating through 32 meters in the corridor is within 5 cm from the mapped position, thus confirming the accuracy of the visual servoing control law.

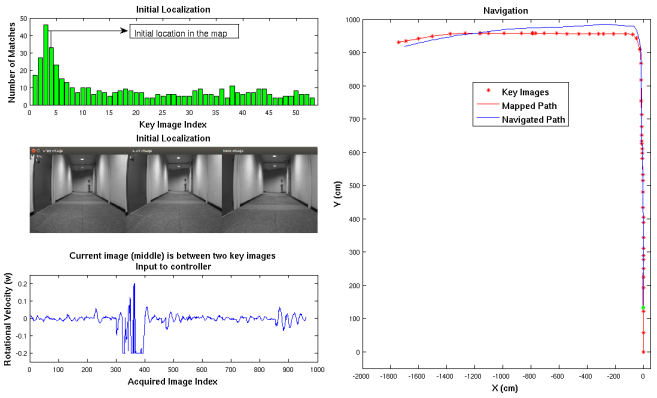

Fig. 12. Navigation in the corridor in reverse direction.

Fig. 13. Changes in the navigation environment from the mapped one: people passing by, opening of the doors, change in the local illumination due to a dead bulb and blur in the image.

C. Experiment 3: Navigation in room and corridor

Two experiments have been performed in which the robot moves between a corridor and a room. Some key images of the learned paths are shown in Fig. 14. In the first experiment, the robot navigated a 40m path from corridor to inside the room. Out of 7745 images acquired during mapping, 105 images have been automatically selected as reference images. In the second experiment, the robot navigated inside the room and then into the corridor in a 22m path. Out of 4291 images acquired, 73 images were selected as reference images. The navigation path consists of straight line and multiple turns as shown in Fig. 15.

Fig. 15 presents the navigation of the robot. The robot successfully followed the learned path with turning whenever required. There are more deviations in turnings especially in case of turning in large angle (semi-circular turnings) because of approximation of the arcs as straight lines and few lines detected with fast changes between the frames. Even though a large drift is present while doing circular turn in the second experiment, the navigation was still successful.

Fig. 14. Some key images from the navigation paths. Top row represents the 40m navigation path. Bottom row represents the 22m navigation path.

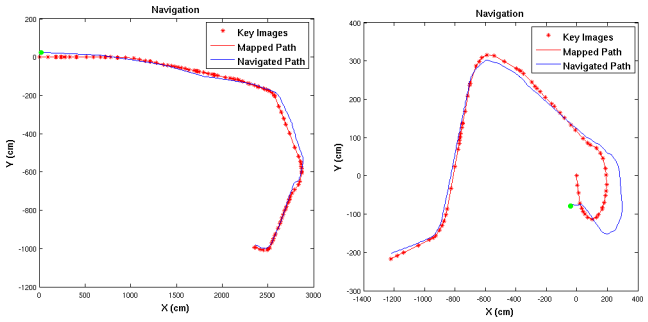

Fig. 15. Navigation in the corridor and the room.

D. Discussions

The presented results show the viability of our approach in many different scenarios and constraints. The robot has been able to navigate autonomously in the learned path from the start position. Our framework does not depend upon any particular type of line segment, and the key images selected by our approach proved to be good enough for the navigation. Our navigation algorithm is based on the idea that the key images around the immediate neighborhood of the robot have more matches than others. The bar graphs in Figs. 6, 11 and 12 confirm this idea. The adjacent key images that have maximum common lines give the initial location in the map. Based upon line matching results, the key images are switched automatically and the appropriate rotational velocity is set that allows the robot to follow the learned path. IBVS has been able to keep the error within small bounds. The robot did not exactly follow the learned path because neither 3D information nor any 3D motion estimation to correct the pose was used, as this is not our objective of the navigation to be accurate, but to be successful and robust. The other reason is also due to approximation of the path by straight lines. However, neglecting 3D information also results in some limitations like more lateral deviation especially after sharp turnings. Nevertheless, based on 2D information only, a useful navigation could be performed in the corridors and inside the room as visual servoing is robust enough to handle such errors.

Comparing with point based method [5], [6], our method performs better especially in low textured environment as in tunings of corridor and in-presence of motion blur due to rapid camera motion as shown in Fig. 16, where reliable point based features cannot be detected, resulting in failure in tracking and 3D reconstruction (without using external informational from sensors like IMU). Our framework performs better in the environments like those in Figs. 13 and 16 because line segments are abundant in a structured indoor environment, and they are also more resilient to motion blur and partial occlusions. However, our framework also has some limitations that are mainly due to the line matching algorithm, which is not still a mature field in computer vision unlike points, especially in the cases where there are very few line segments detected in the images. Initial localization might produce false results when there are few matches (say less than 10) by using just two view matching. In such cases, 3 view

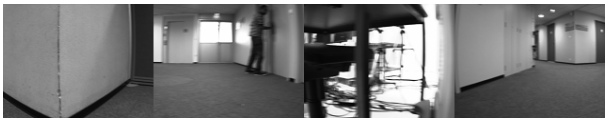

Fig. 16. Some cases where point based methods [5], [6] failed and our approach succeed.

matching can be used for verification to select the two nodes in the map. However, most of these problems can be greatly avoided by selecting proper trajectory during the mapping. The other problem that might be encountered in our framework is a singularity of J_a (of (14)). During all our experiments, singularity condition never occurred. The smallest value of J_a encountered during our all experiments was -0.0018. In most cases, the value of J_a is always greater than 1. The possible strategy that can be used during the singularity (if occurred) is to only use fewer lines that have errors around the median value.

IV. CONCLUSIONS

We have presented a method for indoor qualitative mapping and navigation based on a topological representation of the environment using only line segments extracted from a perspective camera. Our navigation is exclusively based on 2D image measurement without relying on any 3D reconstruction process as in most existing literature. This is possible due to a topological representation of the environment and the use of image based visual servoing for motion control. We also showed that image-based navigation could be performed without accurately tracking the trajectory used in the learning phase. Using line segments as features, instead of points as classically done, makes navigation possible despite some level of occlusions and blur in the image. The experiments showed that the method is able to handle moderate changes in lighting conditions and new objects in the environment. Difficult situations include featureless areas like smooth/texture-less walls (especially for sharp turnings), photometric variations like strong shadows, rapid and sharp turnings. More elaborate image processing and/or robot control strategies can address these issues. Obstacles avoidance during the navigation will be handled using an approach similar to [31] in a future work.

REFERENCES

[1] R. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*. Cambridge Univ. Press, ISBN: 0521540518, second ed., 2004.

[2] A. Davison, I. Reid, N. Molton, and O. Stasse, "Monoslam: Real-time single camera slam," *Pattern Anal. and Machine Intl., IEEE Trans. on*, vol. 29, pp. 1052–1067, June 2007.

[3] A. P. Gee and W. Mayol-Cuevas, "Real-time Model-based SLAM using Line Segments," in *Proc. of the Second Int. Conf. on Advances in Visual Computing*, ISVC'06, pp. 354–363, Springer-Verlag, 2006.

[4] L. Zhang and R. Koch, "Hand-held monocular SLAM based on line segments," *Int. Machine Vision and Image Processing Conf.*, vol. 0, pp. 7–14, 2011.

[5] A. Diosi, S. Segvic, A. Remazeilles, and F. Chaumette, "Experimental evaluation of autonomous driving based on visual memory and image based visual servoing," *IEEE Trans. on Intelligent Transportation Systems*, vol. 12, pp. 870–883, September 2011.

[6] S. Segvic, A. Remazeilles, A. Diosi, and F. Chaumette, "A mapping and localization framework for scalable appearance-based navigation," *Computer Vision and Image Understanding*, vol. 113, pp. 172–187, 2009.

[7] J. Courbon, Y. Mezouar, and P. Martinet, "Indoor navigation of a non-holonomic mobile robot using a visual memory," *Autonomous Robots*, vol. 25, no. 3, pp. 253–266, 2008.

[8] F. Labrosse, "Short and long-range visual navigation using warped panoramic images," *Robotics and Autonomous Systems*, vol. 55, no. 9, pp. 675 – 684, 2007.

[9] A. Dame and E. Marchand, "Using mutual information for appearance-based visual path following," *Robotics and Autonomous Systems*, vol. 61, no. 3, pp. 259 – 270, 2013.

[10] F. Werner, J. Sitte, and F. Maire, "Visual topological mapping and localisation using colour histograms," in *Control, Automation, Robotics and Vision, 2008. 10th Int. Conf. on*, pp. 341–346, Dec 2008.

[11] J. Kosecka, L. Zhou, P. Barber, and Z. Duric, "Qualitative image based localization in indoors environments," in *Computer Vision and Pattern Recognition, 2003. Proc.. 2003 IEEE Computer Society Conf. on*, vol. 2, pp. II–3–II–8 vol.2, June 2003.

[12] T. Goedeme, M. Nuttin, T. Tuytelaars, and L. Van Gool, "Omnidirectional vision based topological navigation," *Int. Journal of Computer Vision*, vol. 74, no. 3, pp. 219–236, 2007.

[13] O. Booij, B. Terwijn, Z. Zivkovic, and B. Krose, "Navigation using an appearance based topological map," in *Robotics and Automation, 2007 IEEE Int. Conf. on*, pp. 3927–3932, April 2007.

[14] A. Remazeilles, N. Mansard, and F. Chaumette, "A qualitative visual servoing to ensure the visibility constraint," in *Intelligent Robots and Systems, 2006 IEEE/RSJ Int. Conf. on*, pp. 4297–4303, October 2006.

[15] N. X. Dao, B.-J. You, and S.-R. Oh, "Visual navigation for indoor mobile robots using a single camera," in *Intelligent Robots and Systems, 2005, IEEE/RSJ Int. Conf. on*, pp. 1992–1997, Aug 2005.

[16] R. von Gioi, J. Jakubowicz, J. M. Morel, and G. Randall, "LSD: A fast line segment detector with a false detection control," *Pattern Anal. and Machine Intl., IEEE Trans. on*, vol. 32, pp. 722–732, April 2010.

[17] C. Akinlar and C. Topal, "EDlines: A real-time line segment detector with a false detection control," *Pattern Recognition Letters*, vol. 32, no. 13, pp. 1633 – 1642, 2011.

[18] C. Schmid and A. Zisserman, "The geometry and matching of lines and curves over multiple views," *Int. Journal of Computer Vision*, vol. 40, no. 3, pp. 199–233, 2000.

[19] B. Fan, F. Wu, and Z. Hu, "Line matching leveraged by point correspondences," in *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conf. on*, pp. 390–397, June 2010.

[20] L. Zhang and R. Koch, "An efficient and robust line segment matching approach based on LBD descriptor and pairwise geometric consistency," *Journal of Visual Communication and Image Representation*, vol. 24, no. 7, pp. 794 – 805, 2013.

[21] P. Smith, I. Reid, and A. Davison, "Real-time monocular SLAM with straight lines," in *Proc. British Machine Vision Conf.*, pp. 17–26, 2006.

[22] A. Faragasso, G. Oriolo, A. Paolillo, and M. Vendittelli, "Vision-based corridor navigation for humanoid robots," in *Robotics and Automation (ICRA), 2013 IEEE Int. Conf. on*, pp. 3190–3195, May 2013.

[23] F. Pasteau, A. Krupa, and M. Babel, "Vision-based assistance for wheelchair navigation along corridors," in *IEEE Int. Conf. on Robotics and Automation, ICRA'14*, (Hong Kong, China), June 2014.

[24] R. F. Vassallo, H. J. Schneebeli, and J. Santos-Victor, "Visual servoing and appearance for navigation," *Robotics and Autonomous Systems*, vol. 31, no. 1, pp. 87–97, 2000.

[25] L. Wang, U. Neumann, and S. You, "Wide-baseline image matching using line signatures," in *Computer Vision, 2009 IEEE 12th Int. Conf. on*, pp. 1311–1318, September 2009.

[26] Z. Wang, F. Wu, and Z. Hu, "MSLD: A robust descriptor for line matching," *Pattern Recognition*, vol. 42, no. 5, pp. 941 – 953, 2009.

[27] F. Chaumette and S. Hutchinson, "Visual servo control, part i: Basic approaches," *IEEE Robotics and Automation Mag.*, vol. 13, pp. 82–90, December 2006.

[28] B. Espiau, F. Chaumette, and P. Rives, "A new approach to visual servoing in robotics," *IEEE Trans. on Robotics and Automation*, vol. 8, pp. 313–326, June 1992.

[29] F. Chaumette and S. Hutchinson, "Visual servo control, part ii: Advanced approaches," *IEEE Robotics and Automation Mag.*, vol. 14, pp. 109–118, March 2007.

[30] E. Marchand, F. Spindler, and F. Chaumette, "Visp for visual servoing: a generic software platform with a wide class of robot control skills," *IEEE Robotics and Automation Mag.*, vol. 12, pp. 40–52, Dec. 2005.

[31] A. Cherubini and F. Chaumette, "Visual navigation of a mobile robot with laser-based collision avoidance," *The Int. Journal of Robotics Research*, 2012.