Combining Line Segments and Points for Appearance-based Indoor Navigation by Image Based Visual Servoing

Suman Raj Bista, Paolo Robuffo Giordano and François Chaumette

Abstract— **This paper presents image-based navigation from an image memory using a combination of line segments and feature points. The environment is represented by a set of key images, which are acquired during a prior mapping phase that defines the path to be followed during the navigation. The switching of key images is done exploiting the common line segments and feature points between the current acquired image and the nearby key images. Based on the key images and the current image, a control law is derived for computing the rotational velocity of a mobile robot during its visual navigation. Using our approach, real-time navigation has been performed in real indoor environment with a Pioneer 3-DX equipped with an on-board perspective camera and the humanoid robot Pepper without the need of accurate mapping and localization nor of 3D reconstruction. We also show that the combination of points and lines increases the number of features that helps in robust and successful navigation especially in those regions where few points or lines can be detected and tracked/matched.**

I. INTRODUCTION

Visual information has already been widely used for mobile robot navigation [1]. Visual navigation can be broadly classified into two approaches: model-based and appearance-based. The first approach relies on the knowledge of an accurate and consistent 3D model of the navigation space that can be computed from different features like lines, planes, or points [2], or estimated from a learning step. Most of the Simultaneous Localization And Mapping (SLAM) based methods [3], [4], [5] fall in this category. On the other hand, the appearance-based approaches do not require a 3D model of the environment, and work directly in the sensor space. To simplify the process of such navigation schemes, the environment is generally represented as topological graphs [6], [7], [8], [9]. 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 can be built in a prior off-line learning phase, which is far less complex than what needed for metric maps. The comparison between the current view and the images in the map is generally based on global descriptors, like considering the whole image [10], [11], [12], color histograms [13], or image gradient [14]; or by using local descriptors, like photometric invariants [15] or corners, Scale-Invariant Feature Transform (SIFT)/ Speeded Up Robust Features (SURF) points or Maximally Stable Extremal Regions (MSER) [6], [7], [8], [16], or line segments [9].

The works in [6], [7], [8], [9], [16] have demonstrated navigation in indoor and outdoor environments using appearance based approaches from image memory. In [6], [7], [9], it is shown that accurate mapping and localization is not mandatory for visual navigation. The first two methods used a hybrid model for topological navigation based on visual memory in an outdoor environment where 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. Most of the techniques employed in outdoor environments typically results in a significant performance drop when applied to indoor scenarios because of windows, wiry structures, reflections and repetitions, as well as limited texture in indoor scenarios, which causes standard procedures based on image descriptors to poorly perform indoors [17]. However, our previous work [9], along with SLAM based methods like [4], [5], [18], and corridor navigation methods like [19], [20], [21], showed that line segments are actually good features for indoor navigation. Still, there are some limitations mainly due to the line matching/tracking algorithms, which are still not a mature field in computer vision unlike points, especially when very few line segments can be detected in the images. Therefore, in this work we extend the navigation framework proposed in [9] to combine line segments with point features for increasing robustness in terms of its application to the wide range of indoor scenarios with smooth motion.

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 on-line control of the rotational velocity) combining general 2D line segments with points features with the aim to coarsely follow a learned path *without* the need of accurate mapping, localization, 3D or of robot odometry. We have also performed closed loop stability analysis using line segments only, points only and combination of lines and points. Up-to our knowledge, combining general line segments with features points for the navigation has never been done before in the literature. This combination is particularly effective in those cases where there are few lines detected and matched, especially during sharp turnings and changes in the illumination conditions. Using points with lines will increase the number of features at those regions, and eventually help in

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

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

better motion control and hence more robust navigation.

The next section describes the complete framework for mapping and navigation along with closed loop stability analysis. Section III presents some experimental results with a real robotic system in different indoor scenarios, which demonstrate the validity of the proposed navigation scheme. Finally, some concluding remarks are reported in Section IV.

II. OVERVIEW

A. Constraints

We consider a non-holonomic mobile robot of unicycle type equipped with a fixed perspective camera. The intrinsic parameters of the camera are constant and coarsely known. The presented method is concerned only with a goal-directed behavior without considering obstacle avoidance, which will be considered in future works. The robot exhibits a qualitative path following behavior. 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. Features detection and tracking/matching

In this work, we combine line segments with point features. Edge Drawing Lines (EDLines) detector [22] and the line matching method proposed by [23] have been used to detect and to match the line segments, respectively. SURF points [24] and the corners detected by FAST (Features from Accelerated Segment Test) corner detector [25] have been considered as point features. Wide baseline matching of the SURF points is performed only during the initial localization. During the navigation, FAST corners detected from the key images are tracked using modified KLT (Kanade-Lucas-Tomasi Feature Tracker) algorithm as presented in [6]. Trifocal tensor with Random Sample Consensus (RANSAC) [2] has been used to remove outliers for the line segments matching similar to the process described in our previous work [9]. The outliers of point matching/tracking are removed by using 5 point algorithm with RANSAC [26].

In our method, two view correspondences are used in initial localization, switching of key images and generating three view correspondences. Three view correspondences are used in mapping, switching of key images and motion control. For generating three view correspondences, 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. In our method, we have used only those line segments and points that are geometrically consistent (2 view geometry for points and 3 view geometry for lines) and pass inlier test using RANSAC. Therefore, we have high confidence over the lines and points that have been obtained to use number of correspondences as metrics to compare.

C. Key images selection

In the proposed method, the environment is represented by a set of key images that are selected automatically. The first acquired image is always stored in the database as a key image (first node in the graph). Let I_c be the most recently acquired image and I_k be the most recent key image. The line segments and corners detected in I_k are tracked over I_c. For tracking of lines, we have used the method proposed in our previous work [9]. For tracking of points, the approach used in [6] has been used. If $n_L(I_k, I_{c-1}, I_c)$ and $n_P(I_k, I_c)$ denotes numbers of lines and points tracked after outliers rejection and ϵ is a positive integer, the new key image is selected when

$$n_L(I_k, I_{c-1}, I_c) + n_P(I_k, I_c) < \epsilon.$$

In this case I_{c-1} will be new I_k. Otherwise, the new I_c is acquired and 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. 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 and feature points as shown in Fig. 1, which makes it possible to consider multiple key images in a neighborhood for defining the heading angle of the robot.

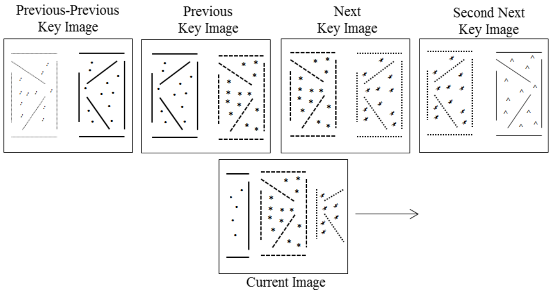

Fig. 1: The map consists of key images. Adjacent key images share some line segments and feature points with the current image. These corresponding features with the current acquired image are used for switching of key images and motion control.

D. Initial Localization in the map

The navigation starts with the initial localization, which allows determining the initial position of the robot in the map. The first image acquired (I_a) is compared with all the images in the database based upon line segment and points matching. If I_k denotes the key image, $n(...)$ is the total number of lines and points matched between the images, I_P is the previous key image and I_N is the next key image, I_P and I_N are selected as:

$$\boxed{\begin{aligned} I_{k_i} &= \arg\max_{I_k}\{n(I_a, I_{k_1}), n(I_a, I_{k_2}), ..., n(I_a, I_{k_n})\}, \\ I_{k_j} &= \arg\max_{I_k}\{n(I_a, I_{k_{i-1}}), n(I_a, I_{k_{i+1}})\}, \\ & I_P = I_{k_j} \text{ and } \quad I_N = I_{k_i} \text{if } i > j, \\ & I_P = I_{k_i} \text{ and } \quad I_N = I_{k_j} \text{if } i < j. \end{aligned}}$$

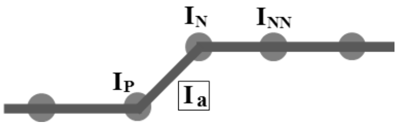

Fig. 2: Localization in the map.

E. Successive Localization (Key Images Switching)

After initial localization in the map, further localizations can be done exploiting the adjacency relationship between the key images. 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 $n_L(...)$ be the total number of lines matched and $n_P(...)$ be total number of points tracked 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}:

$$\boxed{\begin{array}{c} ((n_L(I_a, I_N, I_{NN}) + n_P(I_a, I_N, I_{NN})) > \\ (n_L(I_P, I_a, I_N) + n_P(I_P, I_a, I_N))) \\ \text{OR} \\ (((n_L(I_a, I_{NN}) + n_P(I_a, I_{NN})) > (n_L(I_a, I_N) + n_P(I_a, I_N)) \\ \text{AND} \\ ((n_L(I_a, I_{NN}) + n_P(I_a, I_{NN})) > (n_L(I_p, I_a) + n_P(I_p, I_a))). \end{array}}$$

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 that may occur in rapid turnings. 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. Therefore, the navigation needs to be stopped to prevent the robot moving outside the mapped zone.

F. Control

For the considered navigation task, the robot is not required to accurately reach each key image of the path, or to accurately follow the learned path. In practice, the translational velocity is kept constant and reduced to a smaller value when turning. The rotational velocity is derived using the key images and the current image via an image based visual servoing (IBVS) control law [27].

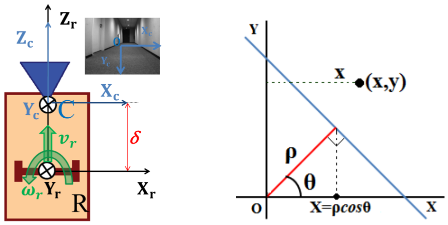

Fig. 3: Top view of robot (orange) equipped with a perspective camera (blue) with its optical axis perpendicular to axis of robot rotation (left) and representation of line and point (right).

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

$$\dot{\mathbf{s}} = \mathbf{L_s u_c}. \tag{1}$$

For the considered unicycle-like robot (Fig. 3), $\mathbf{u_c}$ can be expressed in terms of robot velocity $\mathbf{u} = (v_r, \omega_r)$ as

$$\mathbf{u_c} = \quad (-\delta\omega_r, \quad 0, \quad v_r, \quad 0, \quad -\omega_r, \quad 0), \tag{2}$$

where δ is the distance between the camera center and the robot center of rotation, v_r is the forward velocity and ω_r is the rotational velocity of the robot. Now, from (1) and (2), we obtain

$$\dot{\mathbf{s}} = \mathbf{J}_v v_r + \mathbf{J}_\omega \omega_r, \tag{3}$$

where $\mathbf{J}_v$ and $\mathbf{J}_\omega$ are the Jacobian associated with v_r and ω_r respectively. In practice, $\mathbf{J}_\omega$ and $\mathbf{J_v}$ have to be estimated or approximated because they depend on the unknown depth [27]. Let them be represented by $\hat{\mathbf{J}}_\omega$ and $\hat{\mathbf{J}}_\mathbf{v}$. In order to drive s to desired value $\mathbf{s}^*$, we set v_r as constant and control ω_r as [27]:

$$\omega_r = -\hat{\mathbf{J}}_\omega^+(\lambda(\mathbf{s} - \mathbf{s}^*) + \hat{\mathbf{J}}_v v_r), \tag{4}$$

where λ is a positive gain, $\hat{\mathbf{J}}_\omega^+$ is the pseudo-inverse of $\hat{\mathbf{J}}_\omega$ and $(\mathbf{s} - \mathbf{s}^*)$ is the error ($\mathbf{e}$). The calculation of Jacobians is presented below.

For the line segments, we use the abscissa of the centroid of the points of intersection of the n matched lines and their respective normal from the origin (Fig. 3 (right)) as in our previous work [9]. For the points, we have used x-coordinate of the centroid of the m tracked points as in [6].

The interaction matrix for line segments has been derived in [9] and it takes the expression

$$\hat{J}_{vL} = 0 \text{ and } \hat{J}_{\omega L} = \frac{1}{n}\sum_{i=1}^{n}(\cos^2\theta_i - \rho_i^2\cos(2\theta_i)), \tag{5}$$

where (ρ_i, θ_i) are the line parameters (as shown in Fig. 3 (right)) of the matched lines in I_a, and $\hat{J}_{vL}$ and $\hat{J}_{\omega L}$ are the approximated Jacobians for translational velocity and rotational velocity respectively.

From [27] and (2), the interaction matrix $\mathbf{J}_x$ of point $(x\,,\,y)$ that links the displacement of $\mathbf{x}$ w.r.t. the robot velocity $\mathbf{u}$ is

$$\mathbf{J}_x = \left[\begin{array}{cc} \frac{x}{Z} & -\left(\frac{\delta}{Z} + 1 + x^2\right) \end{array}\right], \tag{6}$$

where x and y are normalized image coordinates and Z is the depth of the point.

Using the x-coordinate of the centroid of m tracked points as feature, the Jacobians for translational velocity (J_{vP}) and rotational velocity $(J_{\omega P})$ are given as

$$\begin{aligned} J_{vP} &= \frac{1}{m}\sum_{j=1}^{m}\left(\frac{x_j}{Z_j}\right), \\ J_{\omega P} &= \frac{1}{m}\sum_{j=1}^{m}\left(\frac{\delta}{Z_j} + 1 + x_j^2\right). \end{aligned} \tag{7}$$

Neglecting δ w.r.t. Z (depth of the point), Z w.r.t x, and assuming the camera optical axis is orthogonal to the axis of robot rotation and that the centroid stays near the image center, we obtain approximated Jacobians $\hat{J}_{vP}$ and $\hat{J}_{\omega P}$ as

$$\hat{J}_{vP} = 0 \text{ and } \hat{J}_{\omega P} \simeq 1 + \frac{1}{m}\sum_{j=1}^{m}(x_j^2). \tag{8}$$

Since we only control ω_r, only one feature derived from all line segments and points is sufficient. We combine them using linear least square to derive rotational velocity.

Let X_a, X_N and X_{NN} be the abscissa of the centroid of the points of intersection of n matched lines and their respective normal from the origin of I_a, I_N and I_{NN} respectively. Then, the error term for the case of line segments is $X_a - X_N$.

Let x_a, x_n and x_{nn} be the x-coordinate of the centroid of the tracked points of I_a, I_N and I_{NN} respectively. Then, the error term for the case of points is $x_a - x_n$.

Combining lines and points together, the expression for the rotational velocity can be computed from (4), (5) and (8) as

$$\omega_r = \begin{bmatrix} \hat{J}_{\omega L} \\ \hat{J}_{\omega P} \end{bmatrix}^{+} \left(\lambda \begin{bmatrix} (X_a - X_N) \\ (x_a - x_n) \end{bmatrix} + \begin{bmatrix} \hat{J}_{vL} \\ \hat{J}_{vP} \end{bmatrix} v_r \right). \tag{9}$$

Since $\hat{J}_{vL}$ and $\hat{J}_{vP}$ are zero (from (5) and (8)), after simplification, (9) becomes

$$\omega_r = -\frac{\lambda \left(\hat{J}_{\omega L}(X_a - X_N) + \hat{J}_{\omega P}(x_a - x_n) \right)}{\hat{J}_{\omega L}^2 + \hat{J}_{\omega P}^2}. \tag{10}$$

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 and points of I_a with I_N and I_{NN}. The final expression for the rotational velocity can be computed as

$$\omega_r = -\frac{\lambda \begin{pmatrix} \hat{J}_{\omega L}(g_1(X_a - X_N) + g_2(X_a - X_{NN})) \\ +\hat{J}_{\omega P}(g_1(x_a - x_n) + g_2(x_a - x_{nn})) \end{pmatrix}}{\hat{J}_{\omega L}^2 + \hat{J}_{\omega P}^2}, \tag{11}$$

where g_1 and g_2 are positive weights such that $g_1 + g_2 = 1$. If at least one point feature can be tracked from key images to current view, no singularity can occur in (11) because of (8). Even in presence of only lines, singularities are rare. In fact in all our experiments, we never encountered any singularity.

Stability Analysis

In order to assess the stability of the closed-loop visual servo systems, we will use Lyapunov analysis. We consider the candidate Lyapunov function ($\mathcal{L}$) defined by the squared error norm [27] i.e $\mathcal{L} = \frac{1}{2}\left\|\mathbf{e}(\mathbf{t})\right\|^2$, whose derivative is given by

$$\dot{\mathcal{L}} = \mathbf{e}^{\mathbf{T}}\dot{\mathbf{s}}, \tag{12}$$

where $\mathbf{e} =\mathbf{s} - \mathbf{s}^*$. From (3), (4), and (12), we have

$$\dot{\mathcal{L}} = \mathbf{e}^T(\mathbf{J}_v v_r - \mathbf{J}_\omega \hat{\mathbf{J}}_\omega^+(\lambda\mathbf{e} + \hat{\mathbf{J}}_v v_r)). \tag{13}$$

On simplification of (13), we have

$$\dot{\mathcal{L}} = -\lambda\mathbf{e}^T\mathbf{J}_\omega\hat{\mathbf{J}}_\omega^+\mathbf{e} - \mathbf{e}^T(\mathbf{J}_\omega\hat{\mathbf{J}}_\omega^+\hat{\mathbf{J}}_v - \mathbf{J}_v)v_r. \tag{14}$$

In order to ensure global asymptotic stability, we have to show $\dot{\mathcal{L}} < 0$. In (14), the first term is always negative if $\mathbf{J}_\omega\hat{\mathbf{J}}_\omega^+ > 0$. In second term, $\mathbf{J}_v$ (hence $\hat{\mathbf{J}}_v$), which consists of depth term, can be approximated to zero as shown in (5), (7-8) and [9]. Therefore, in this case $\dot{\mathcal{L}} < 0$.

Let $J_{\omega L}$ and $J_{\omega P}$ be the true Jacobians, and $\hat{J}_{\omega L}$ and $\hat{J}_{\omega P}$ be the approximated Jacobians related to ω_r for lines (that with subscript L) and points (that with subscript P) features. From [9], we obtain

$$J_{\omega L} = \hat{J}_{\omega L} + \frac{1}{n}\sum_{i=1}^{n}\delta\cos\theta_i(\lambda_{\theta i}\rho_i\sin\theta_i - \lambda_{\rho i}\cos\theta_i)), \tag{15}$$

where $\lambda_{\theta i}$ and $\lambda_{\rho i}$ are the functions of $(1/d_i)$, with d_i being the depth of line segment from image plane. From (5) and (15), we have

$$J_{\omega L}\hat{J}_{\omega L}^+ = 1 + \frac{\frac{1}{n}\sum_{i=1}^{n}\delta\cos\theta_{ai}(\lambda_{\theta ai}\rho_{ai}\sin\theta_{ai} - \lambda_{\rho ai}\cos\theta_{ai})}{\frac{1}{n}\sum_{i=1}^{n}(\cos^2\theta_{ai} - \rho_{ai}^2\cos(2\theta_{ai}))}. \tag{16}$$

Since $\delta \ll d_i$, from (16) we have

$$J_{\omega L}\hat{J}_{\omega L}^+ > 0. \tag{17}$$

Hence, with (16-17), the global asymptotic stability is guaranteed for a feature derived from all the matched lines.

From (7-8) for points only, we have

$$J_{\omega P}\hat{J}_{\omega P}^+ = 1 + \frac{\frac{1}{m}\sum_{j=1}^{m}\left(\frac{\delta}{Z_j}\right)}{\frac{1}{m}\sum_{j=1}^{m}\left(1 + x_j^2\right)} > 0, \tag{18}$$

since $\delta \ll Z_j$ and $\frac{1}{m}\sum_{j=1}^{m}\left(1 + x_j^2\right)$ is close to 1. Hence, with (18), the global asymptotic stability is guaranteed for a feature derived from all the matched points.

For the combination of lines and points, we have

$$\begin{aligned} \mathbf{J}_\omega\hat{\mathbf{J}}_\omega^+ &= \begin{bmatrix} J_{\omega L} \\ J_{\omega P} \end{bmatrix} \begin{bmatrix} \hat{J}_{\omega L} \\ \hat{J}_{\omega P} \end{bmatrix}^+ \\ &= \frac{1}{\hat{J}_{\omega L}^2 + \hat{J}_{\omega P}^2} \begin{bmatrix} J_{\omega L}\hat{J}_{\omega L} & J_{\omega L}\hat{J}_{\omega P} \\ J_{\omega P}\hat{J}_{\omega L} & J_{\omega P}\hat{J}_{\omega P} \end{bmatrix} \geq 0, \end{aligned} \tag{19}$$

which is positive semi-definite because one of its eigen value is 0 and other is $J_{\omega L}\hat{J}_{\omega L} + J_{\omega P}\hat{J}_{\omega P} > 0$ because we are controlling one degree of freedom with two features in (9). But we have, $\hat{J}_\omega^+ J_\omega = J_{\omega L}\hat{J}_{\omega L} + J_{\omega P}\hat{J}_{\omega P} > 0$. Hence, with $\hat{J}_\omega^+ J_\omega > 0$, local asymptotic stability can be ensured using lines and points [27].

Thus, our complete framework uses only 2D information obtained from line segments and feature points. From this information, we derive the required rotational velocity using a *IBVS* control law, which makes the robot to follow the learned path successfully without any need of 3D or accurate mapping and localization.

III. EXPERIMENTAL RESULTS

The experiments in Sects. III-A-III-B were performed with a Pioneer 3-DX equipped with AVT Pike F-032C camera module, whereas experiment in Sect. III-C was performed with the humanoid Pepper. The image resolution in the experiments was 640×480 pixels. All computations, except for the low-level control, were performed on a laptop with 3-GHz Intel Core i7-3540M CPU. The mapping was performed off-line, whereas the navigation experiment was performed on-line at 5 Hz. The acquisition of images and the high-level motion control were done through the interface provided by ViSP [28]. 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. In all our experiments, we have used $g = 0.7$, $g_2 = 0.3$, and $\lambda = 1$. Even though the proposed method has been validated in a simple navigation path with linear and curved trajectories, the method can be easily extended for graphs with intersections and multiple paths. The trajectories presented below are obtained from the odometry of the Pioneer Robot. Since our navigation is qualitative rather than quantitative, the odometry obtained from Pioneer 3-DX is highly accurate enough to serve our purpose.

A. Mapping

During the mapping, the robot was driven manually to acquire the images. From the acquired images, the key images were selected automatically from the mapping algorithm described in Sect. II-C. The number of images acquired and selected key images in different experiments are presented in Table I, where $n(Acq.)$ represents the total number of acquired images and $n(Ref.)$ represents the total number of key images selected. The trajectory obtained from the odometry is shown by a red curve in Figs. 4, 6, 7, 8, and 9, where the red symbol $*$ represents the location of the key images. There are more key images over a small distance in case of quick displacements of features like during turnings or when line segments and points cannot be successively matched/tracked over the sequence because of, e.g., changes in the illumination.

	$Experiment$	$n(Acq.)$	$n(Ref.)$
I	Inside Robotics room	1260	19
II	Room-Corridor-Robotics room	4555	105
III	Robotics Room-Corridor	4100	52
IV	Corridor Out-In	10297	132
V	Corridor In1	7021	140
VI	Corridor In2	6530	122

TABLE I: Table showing the number of acquired images and selected key images in the different experimental scenarios.

B. 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 0.15m/s and reduced to 0.08m/s when turning, whereas the rotational velocity was controlled by the navigation algorithm (11). Such turnings are automatically detected by observing the commanded rotational velocity. During navigation, the robot has been able to follow the learned trajectory as shown by the blue curve in Fig. 4, and Figs. 6-9, where green dot (●) represents the starting location.

1) Navigation inside the robotics room: The navigation has been performed with some changes in environment, for instance the chair and table have been added to the environment after the mapping phase. Besides, the chair was moving from one point to another and a person was walking during the navigation. These changes can be clearly seen in Fig. 4 (middle and right). Fig. 4 (left) shows the successful navigation of the Pioneer in the environment despite the various changes. After turning, the majority of lines detected and matched only belong to the floor. But thanks to the points, the robot was able to obtain features from the walls and from other objects. With more and better distribution of obtained features from points and lines, the motion was smoother especially after turning, which was not the case using line segments only (refer Fig 5).

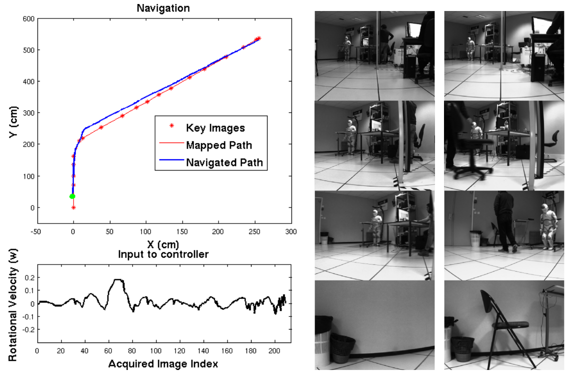

Fig. 4: Navigation inside the robotics room (left), some key images (middle), and some changes in environment during navigation (right).

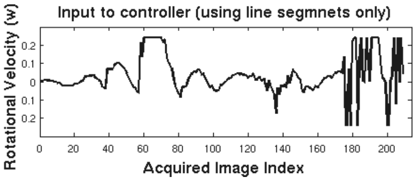

Fig. 5: Rotational velocity input with line segments only in the same environment as in Fig. 4

2) Navigation from a room to another room via a corridor: The navigation between rooms via a corridor is shown in Fig. 6. The robot followed the learned path of 22m with turnings whenever it was required. Right angle turning, especially the second one (in the corridor), was a challenging task. This turning was very difficult using line segments only because of few lines matched and illumination conditions, which were not always sufficient to get a large enough rotational velocity for this sharp turning. However, together with the points, the number of features in this region increased significantly, which allowed the system to be successful during the sharp turning. The lateral drift was within 5 cm from the mapped position, thus confirming the accuracy of the visual servoing control law.

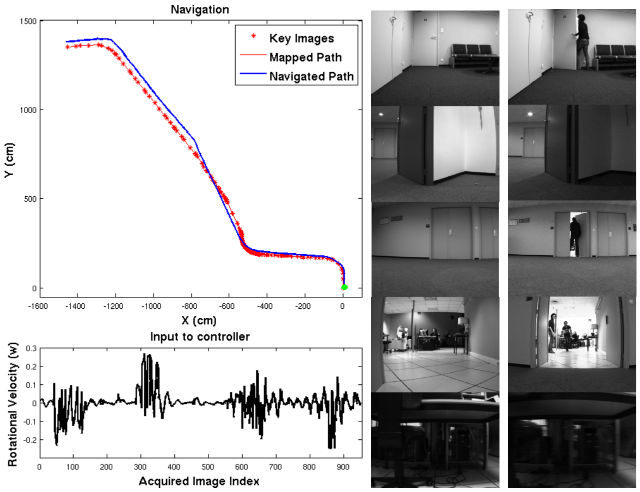

Fig. 6: Navigation between the rooms via corridor (left), some key images (middle), and some images during navigation (right).

3) Navigation from a room to the corridor: In this case, we performed the navigation in the 23m path that starts from the robotics room and navigates into the corridor. The path consists of multiple turns and different levels of illumination as shown in Fig. 7. Even in presence of moving people and objects, the robot was able to follow the learned trajectory. With the addition of point features, the motion was relatively smooth because of the increased number and better distribution of features to compute the rotational velocity.

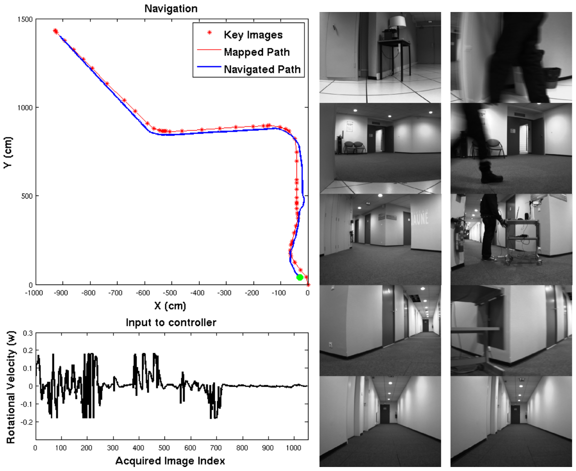

Fig. 7: Navigation in between the room and the corridor (left), some key images (middle), and some images during navigation (right).

4) Navigation in a corridor (Out-In): In this case, we performed the navigation of 47m in a corridor (mapped length is 51m). The beginning of the navigation zone consists of large windows that allow seeing outdoor. So, the robot undergoes large changes in the illumination in the mapped path as we enter inside the corridor. Also, there are two small inclined planes in the path. Even in presence of changing lightning condition (cloudy when mapped and sunny during navigation) and walking people, the robot was able to follow the learned trajectory as shown in Fig. 8. Especially due to illumination change, the number of lines matched decreased. However, the addition of point features allowed for a smooth navigation. For this experiment we started the robot 4m in front the mapped location. The robot successfully navigated the remaining path by automatically selecting the initial key images, thanks to the initial localization.

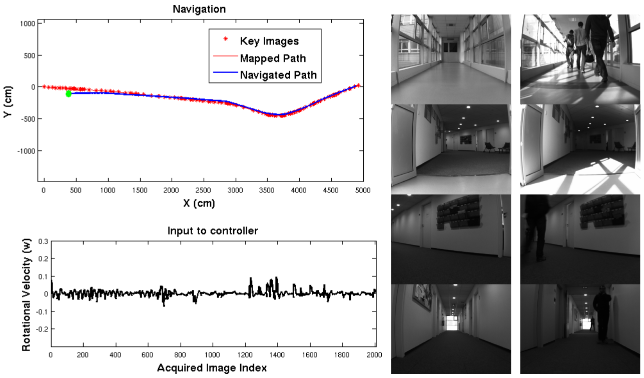

Fig. 8: Navigation in the corridor (left), some key images (middle), and some images during navigation (right).

5) Navigation in indoor corridor (In1, In2): Here, we performed navigation in two corridors that are completely indoor. The corridors have shiny floor and walls are mostly white. The illumination in first corridor (In1, length 32m) totally depends upon the bulbs on the top of the ceilings where as in second corridor (In2, length 34m) there is also light from outside. Navigation in both cases were difficult using line segments only because of the type of surface of walls, very few line segments can be detected and matched in some regions and in turnings. With lines only as in [9] or points only as in [6], navigation was not successful. Using points in addition to lines, more number of features are present which makes it possible for successful navigation as shown in Fig. 9.

C. Navigation with the humanoid robot Pepper

Here, we performed a simple experiment to validate our approach in more sophisticated platform like Pepper. Pepper, as shown in Fig. 10, is a humanoid robot developed by SoftBank Robotics [29]. It consists of two High Definition 2D cameras at its head and three multi-directional wheels that enables it to move around freely through 360°. Out of two cameras, we have used upper camera only because the lower camera can see the floor only. The images grabbed by Pepper

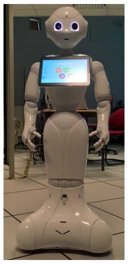

Fig. 10: Humanoid Robot Pepper.

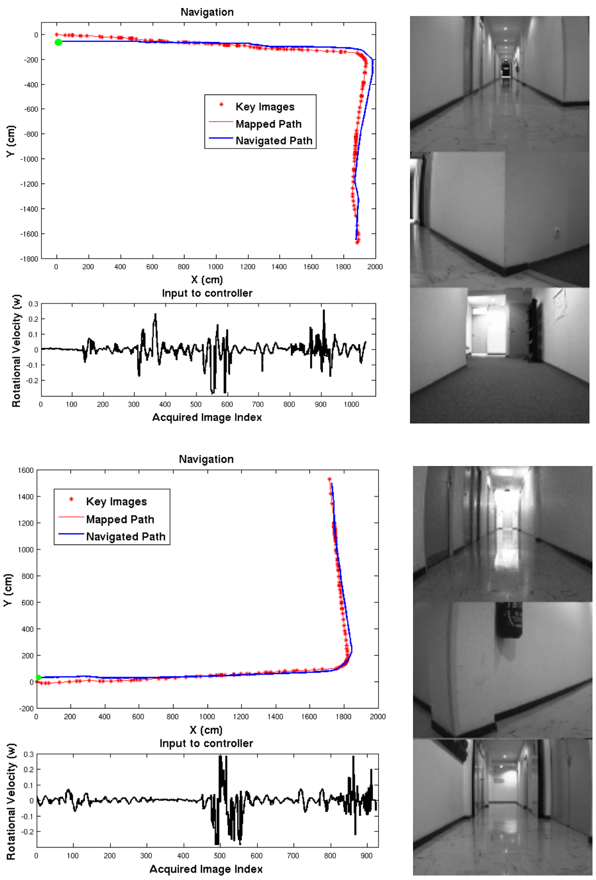

Fig. 9: Navigation in the corridor : In1 (upper) and In2 (lower).

in VGA resolution is send to laptop by Wifi for processing. With wheel base, and one camera keeping head pitch and roll angle constant, the control scheme as discussed in Sect. II-F can be used as motion control. Therefore, we control the rotational velocity of Pepper using (11) via NaoQi API [29]. The navigation environment is represented by 12 keyframes as shown in Fig. 11. Some images during navigation are shown in Fig. 12. Despite the changes in the navigation environment from mapping one that are clearly seen in Figs. 11-12, Pepper is able to navigate inside the room.

D. Discussion

The presented results show the viability of our approach in different scenarios and constraints. The robot has been able to follow autonomously the learned path from the start position. Our framework does not depend upon any particular type of line segment or points but it is just based on generic lines or points that are detected and matched/tracked from key images. The key images selected by our approach proved to be good enough for the navigation. Based upon the line matching and tracking of the points from neighboring key frames, the key images are switched automatically and an appropriate rotational velocity can be computed for allowing the robot to follow the learned path. The IBVS law has been able to keep the error within small bounds. The deviation from the learned path is within the range of 5 cm. We

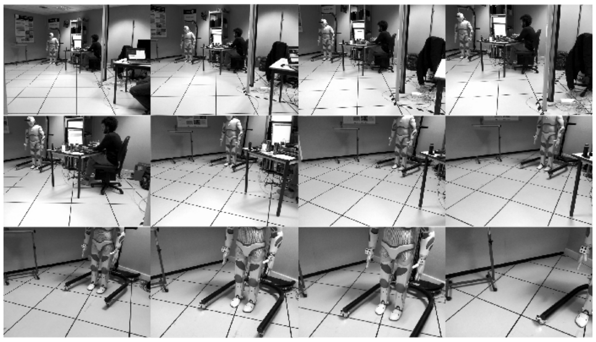

Fig. 11: Reference Images of Robotics Room.

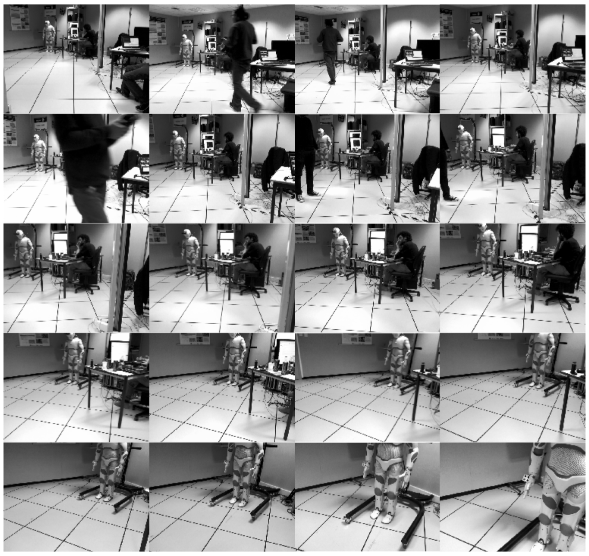

Fig. 12: Navigation inside Robotics Room.

have also shown that our closed loop control scheme is asymptotically stable.

Our framework takes advantages from both lines and points. Line segments are abundant in a structured indoor environment, and they are also more resilient to motion blur and partial occlusions. Even if the performance of point features can degrade in indoor environments, they can still be tracked locally to some extent, which is sufficient to supplement the line segments. This then helps in better switching the key images and in computing the rotational velocity. The robust sharp turning, smooth motion of the robots, and successful navigation in wide range of environment are the consequences of using multiple features. Besides that, the point tracking and the line matching can be operated in parallel without any significant increase in the computational time. We still manage to operate at 5Hz with the improved performance as in the case of line segments only [9]. However, our framework has also some limitations that are mainly due to the line matching and the point tracking algorithm especially in the cases where very few line segments are detected in the images and the tracking of points does not perform well. Initial localization might produce false results when there are few matches that make the geometrical verification of matched points/lines not possible. However, most of above problems can be greatly avoided by selecting a proper trajectory during the mapping.

IV. CONCLUSIONS

We have presented a framework for indoor qualitative mapping and navigation based on image memory using a combination of line segments and points, by expanding our previous work [9]. Our navigation is exclusively based on 2D image information without relying on any 3D reconstruction or pose estimation, and also without accurately tracking the trajectory used in the learning phase. This is possible because of the topological representation of the environment and of the robustness of the adopted IBVS law. We validated our approach with a mobile robot and the humanoid robot Pepper. Combination of points and line features increases the number of features in the image which results in a more robust navigation and smoother control. Increased number of features plays an important role especially during sharp turnings where less features can be detected, which changes rapidly also. We showed a successful navigation in different indoor scenarios while being robust to some level of occlusions and blur in the image, moderate changes in lighting conditions and presence of new objects in the environment. Difficult situations include featureless areas like smooth/texture-less walls. Apart for the initial localization, we have used only the corners detected by FAST algorithm for the navigation. However, we can easily incorporate other interest point detectors like SIFT, SURF, BRISK (Binary Robust Invariant Scalable Keypoints [30]) etc. in our framework so as to increase its robustness. Besides, more precise image processing and/or robot control strategies, like using vehicle kinematics and filtering the velocity, could be incorporated to overcome uncertainties in the navigation. Finally, incorporating obstacle avoidance as in [31] will constitute our future work.

ACKNOWLEDGMENTS

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

REFERENCES

[1] F. Bonin-Font, A. Ortiz, and G. Oliver, “Visual navigation for mobile robots: A survey,” *Journal of Intelligent and Robotic Systems*, vol. 53, no. 3, pp. 263–296, 2008.

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

[3] A. Davison, I. Reid, N. Molton, and O. Stasse, “Monoslam: Real-time single camera slam,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1052–1067, June 2007.

[4] A. P. Gee and W. Mayol-Cuevas, “Real-time Model-based SLAM using Line Segments,” in *Proc. of the 2nd. Int. Conf. on Advances in Visual Computing*, pp. 354–363, Springer-Verlag, 2006.

[5] 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.

[6] 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, Sept. 2011.

[7] 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.

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

[9] S. R. Bista, P. Robuffo Giordano, and F. Chaumette, “Appearance-based indoor navigation by IBVS using line segments,” *IEEE Robotics and Automation Letters*, vol. 1, pp. 423–430, Jan. 2016.

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

[11] 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.

[12] S. R. Bista, P. Robuffo Giordano, and F. Chaumette, “Appearance-based indoor navigation by IBVS using mutual information,” in *IEEE Int. Conf. on Control, Automation, Robotics and Vision*, pp. 1–6, 2016.

[13] F. Werner, J. Sitte, and F. Maire, “Visual topological mapping and localisation using colour histograms,” in *IEEE Int. Conf. on Control, Automation, Robotics and Vision*, pp. 341–346, 2008.

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

[15] 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.

[16] 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.

[17] B. Micusik and H. Wildenauer, “Descriptor free visual indoor localization with line segments,” in *IEEE Conf. on Computer Vision and Pattern Recognition*, pp. 3165–3173, 2015.

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

[19] A. Faragasso, G. Oriolo, A. Paolillo, and M. Vendittelli, “Vision-based corridor navigation for humanoid robots,” in *IEEE Int. Conf. on Robotics and Automation*, pp. 3190–3195, 2013.

[20] F. Pasteau, A. Krupa, and M. Babel, “Vision-based assistance for wheelchair navigation along corridors,” in *IEEE Int. Conf. on Robotics and Automation*, pp. 4430–4435, 2014.

[21] 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.

[22] 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.

[23] 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.

[24] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, “Speeded-up robust features (surf),” *Computer Vision and Image Understanding*, vol. 110, no. 3, pp. 346–359, 2008.

[25] E. Rosten, R. Porter, and T. Drummond, “Faster and better: A machine learning approach to corner detection,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 32, pp. 105–119, Jan 2010.

[26] D. Nistér, “An efficient solution to the five-point relative pose problem,” *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 26, pp. 756–770, June 2004.

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

[28] 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 Magazine*, vol. 12, pp. 40–52, Dec. 2005.

[29] Aldebaran Robotics and SoftBank, “Pepper - Documentation.” http://doc.aldebaran.com/2-4/home_pepper.html Accessed: 27-06-17.

[30] S. Leutenegger, M. Chli, and R. Y. Siegwart, “Brisk: Binary robust invariant scalable keypoints,” in *IEEE Int. Conf. on Computer Vision*, pp. 2548–2555, 2011.

[31] A. Cherubini and F. Chaumette, “Visual navigation of a mobile robot with laser-based collision avoidance,” *The Int. Journal of Robotics Research*, vol. 32, pp. 189–205, Feb. 2013.