Vision-based Navigation and Environmental Representations with an Omni-directional Camera

José Gaspar, *Member, IEEE*, Niall Winters, and
José Santos-Victor, *Member, IEEE*

Abstract— **This paper proposes a method for the visual-based navigation of a mobile robot in indoor environments, using a single omni-directional (catadioptric) camera. The geometry of the catadioptric sensor and the method used to obtain a bird's eye (orthographic) view of the ground plane are presented. This representation significantly simplifies the solution to navigation problems, by eliminating any perspective effects.**

The nature of each navigation task is taken into account when designing the required navigation skills and environmental representations. We propose two main navigation modalities: *Topological Navigation* **and** *Visual Path Following*.

Topological Navigation is used for traveling long distances and does not require knowledge of the exact position of the robot but rather, a *qualitative* **position on the topological map. The navigation process combines appearance based methods and visual servoing upon some environmental features.**

Visual Path Following is required for local, very precise navigation, for e.g. door traversal, docking. The robot is controlled to follow a pre-specified path accurately, by tracking visual landmarks in bird's eye views of the ground plane.

By clearly separating the nature of these navigation tasks, a simple and yet powerful navigation system is obtained.

Keywords— **Omni-directional Vision, Navigation, Visual Servoing, Topological Maps**

I. Introduction

We address the problem of indoor navigation of a mobile robot based on visual information provided by an omni-directional (panoramic) camera.

Most of the research on vision-based navigation has been centered on the problem of building full or partial 3D representations of the environment [1], which are then used to drive an autonomous robot. We argue that shifting the emphasis from the actual navigation problem to the process of building these 3D maps, explains why most existing systems require large computational resources, but still lack the robustness required for many real-world applications. In contrast, examples of efficiency can be drawn from biology. Insects, for instance, can solve very large and complex navigation problems in real-time [2], in spite of having limited sensory and computational resources.

One striking observation is the diversity of "ocular" geometries. Many animals eyes point laterally, which seems more suitable for navigation purposes. The majority of insects and arthropods benefit from a wide field of view and their eyes have a space-variant resolution. To some extent, the performance of these animals can be explained by their specially adapted eye-geometries. Similarly, in this work, we explore the advantages of having large fields of view by using an *omni-directional camera* with a horizontal field of view of 360°.

Studies of animal navigation [3], [4] suggest that most species utilize a very parsimonious combination of perceptual, action and representational strategies that lead to much more efficient solutions when compared to those of today's robots.

Both robustness and an efficient usage of computational and sensory resources can be achieved by using visual information in closed loop to accomplish specific navigation tasks or behaviors [5], [6]. However, this approach cannot deal with global tasks or coordinate systems (e.g. going to a distant goal), because it lacks adequate representations of the environment. Hence, a challenging problem is that of extending these local behaviors, without having to build complex 3D representations of the environment.

Another point worth discussing is the nature of the navigation requirements when covering long distances, as compared to those for short paths. Many animals, for instance, make alternate use of landmark-based navigation and (approximate) route integration methods [2]. For example, to walk along a city avenue, it is sufficient to know our position to within an accuracy of one block. However, entering our hall door would require much more precise movements.

This *path distance/accuracy* tradeoff between long-distance/low-precision and short-distance/high-accuracy mission segments plays an important role in finding efficient solutions to the robot navigation problem. In this paper, we denote these navigation modes as *Topological Navigation* versus *Visual Path Following*.

We show how omni-directional images can be used for both navigation modes and provide suitable environmental representations where *Topological Navigation* and *Visual Path Following* can be integrated in a natural way.

A. Omni-directional Vision

Omni-directional cameras provide a 360° view of the robot's environment, in a single image, and have been applied to autonomous navigation, video conferencing and surveillance [7]-[12], among others. Omni-directional images are usually obtained with Catadioptric Panoramic Cameras, which combine conventional cameras (lenses) and convex mirrors. Mirror shapes can be conic, spherical, parabolic or hyperbolic [13], [14], [15].

Visual landmarks are easier to find with omni-directional images, since they remain in the field of view much longer, than with a conventional camera. The imaging geometry has various properties that can be exploited for navigation or recognition tasks. For example, vertical lines in the environment are viewed as radial image lines. The main downfall of omni-directional images is the loss of resolution in comparison with standard images.

We describe the image formation model for an omni-directional camera with a spherical mirror. Although our sensor does not have a single projection center as in [13], [14], [15], we found that this is not a severe limitation to our approach. We show how to unwarp omni-directional images to obtain (orthographic) *Bird's eye views* of the ground plane, where perspective effects have been removed.

B. Topological Maps for Navigation

Topological Navigation is the approach used to travel long distances in the environment, without demanding accurate control of the robot position along a path. The environment is represented by a *Topological Map* [16], [17], [18], described by a graph. *Nodes* correspond to recognizable *landmarks*, where specific actions may be elicited, such as entering a door or turning left. *Links* are associated with regions where some environmental structure can be used to control the robot.

Manuscript received October 28, 1999; revised June 5, 2000. This paper was recommended for publication by Associate Editor G. Hager and Editor V. Lumelsky upon evaluation of the reviewers' comments. This work has been partially funded by the project PRAXIS 2/2.1/TPAR/2074/95, SIVA and the EU TMR network SMART II. This paper was presented in part at the IEEE Workshop on Omnidirectional Vision, Hilton Head Island, SC, June 2000.

J. Gaspar and J. Santos-Victor are with the Instituto de Sistemas e Robótica, Instituto Superior Técnico, Lisboa - Portugal (email jag@isr.ist.utl.pt; jasv@isr.ist.utl.pt).

N. Winters is with the Department of Computer Science, University of Dublin, Trinity College, Dublin 2 - Ireland (email Niall.Winters@cs.tcd.ie).

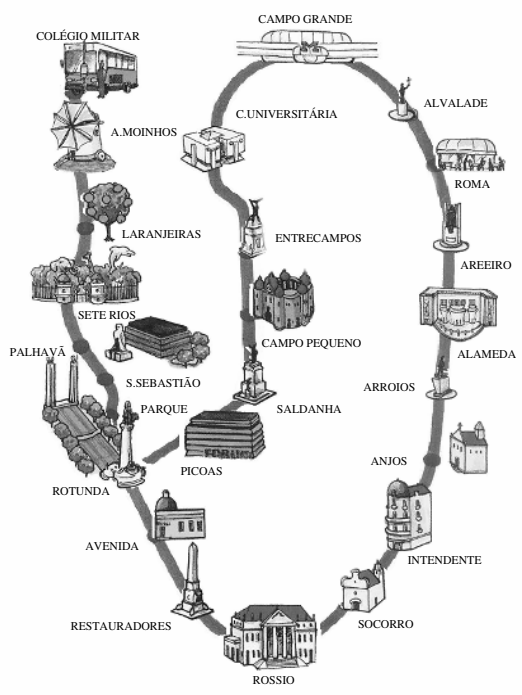

Fig. 1. A topological map of touristic landmarks in Lisbon, Portugal.

In our approach, landmarks are directly represented by (omni-directional) *images*. Links are represented by sequences of images that correspond to trajectories which the robot can follow by servoing upon some environmental features, allowing corridor following behavior.

A map is thus a collection of inter-connected images, as in the example of Figure 1, representing touristic landmarks in Lisbon, Portugal. To go from one particular locale to another, we do not have to think in precise metric terms. For example, to get from the city center, **Rossio**, to **Saldanha** square, we may *go forward* until we reach the statue in **Rotunda**, *turn right* in the direction of **Picoas** and carry on until we finally reach **Saldanha** Square. The navigation problem is decomposed into a succession of sub-goals, identified by recognizable landmarks. The required navigation skills are the ability to follow roads, make turns and recognize that we have reached a landmark.

We use omni-directional images as an implicit topological representation of the environment, and rely on appearance based methods [19]-[22] to provide a *qualitative* measurement of the robot's *global* position. Progression is assessed by comparing the current view with images acquired *a priori* [23]. Images of the topological map are encoded as a manifold in a low dimensional eigenspace obtained from Principal Components Analysis.

In [24], localization was achieved by matching images in the frequency domain, whereas Horswill [25] used the actual views as landmarks. The work most closely related to ours is that described in [26], which combined appearance based methods and visual servoing, although the image geometry, matching scheme and method of servoing were different from those detailed in this paper.

Visual servoing is applied to control *locally* the pose of the robot relative to image features, and to navigate between nodes. In this work, we control the robot heading and position by servoing upon the corridor guidelines, extracted from bird's eye views of the ground plane. Thus, the combination of appearance based methods and visual servoing, at the global level, means that we can maintain a causality constraint while traversing longer distances and sampling the environment less frequently than with previous approaches [21], [23], [24].

Fig. 2. Left: omni-directional camera. Right: camera mounted on the mobile robot.

C. Visual Path Following

Topological Maps can be used for navigating between distant environmental sites, using only qualitative and topological properties of the environment. A different approach is necessary when precise guidance or localization are required (e.g. for docking). For these precise navigation problems, we propose the approach of *Visual Path Following*, to complement topological navigation.

Visual Path Following allows a robot to follow a pre-specified path to a given location, relying on visual tracking of features (landmarks) [27], [28]. We track visual features with bird's eye views, the same image geometry as used for topological navigation, to self-localize the robot. We design a closed loop controller to drive the robot accurately along a desired trajectory, pre-specified in image coordinates, relative to a visual landmark.

The combination of omni-directional images and the Topological and Visual Path Following navigation strategies are illustrated by the complete experiments described in this paper. We believe that the complementary nature of these approaches and the use of omni-directional imaging geometries result in a very powerful solution to build efficient and robust navigation systems.

II. Omni-directional Vision

The only sensor used in this work is an omni-directional camera that provides a rich description of the environment around the robot. It is composed of a CCD camera pointed upwards looking at a spherical mirror, and is mounted on top of the mobile platform, aligned with the platform's axis of rotation (see Figure 2).

Omni-directional images contain both ground and wall regions. The spherical mirror distorts the image projection and, in general, 3D straight lines are projected as curves in the image. The horizon line is projected as an image circle and a corridor appears as an image band with a narrower width for points further away from the robot. Only 3D lines which belong to vertical planes project as straight lines.

A. A Catadioptric Panoramic Camera with a Spherical Mirror

The geometry of image formation is obtained by relating the coordinates of a 3D point, $\mathbf{P}$, to the coordinates of its projection on the mirror surface, $\mathbf{P_m}$, and finally to its image projection $\mathbf{p}$. Figure 3 shows the most relevant parameters of the geometric model for image formation.

The spherical mirror radius is denoted by R (8.35cm), and its distance to the camera projection center by L. The reflection point, $\mathbf{P_m}$, is observed from the camera at an angle β. The mirror normal at $\mathbf{P_m}$ makes an angle γ relative to the vertical

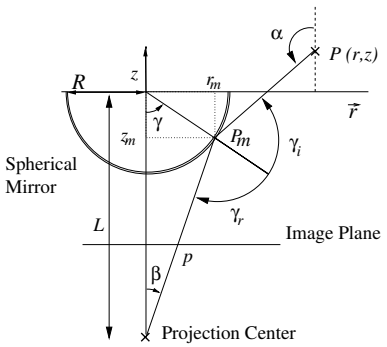

Fig. 3. Camera (spherical mirror) projection geometry. Symmetry about the z-axis simplifies the geometry.

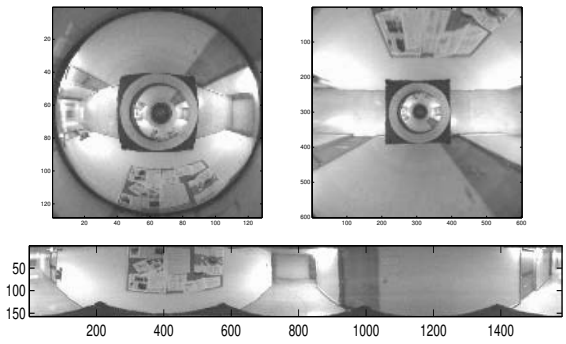

Fig. 4. Top: omni-directional image (left) and corresponding bird's eye view (right). Bottom: panoramic image.

axis, and the incident and reflection angles are denoted by γ_i, γ_r. The elevation angle, α, is that formed between the vertical axis and a ray from a 3D point, $\mathbf{P}$, to $\mathbf{P_m}$. Finally, $r = \sqrt{x^2 + y^2}$, indicates the radial distance to the optical axis and the vertical coordinate is denoted by z.

A point $\mathbf{P_m} = (r_m, z_m)$ on the mirror surface has to fulfill the following equations:

$$\begin{cases} r_m = (z_m + L)\ \tan\beta \\ z_m^2 + r_m^2 = R^2 \\ \gamma_r = \gamma_i \ \Leftrightarrow -2\arctan\left(r_m/z_m\right) = \alpha - \beta \end{cases} \tag{1}$$

These equations are reduced to a vertical plane containing the vertical z axis, since the system is rotationally symmetric around that axis. The last equation can be expressed as a function of the vertical viewing angle, α, or the coordinates of a 3D point (r, z). Some parameters involved in Equation (1) are fixed by the physical setup (R, L), whereas (α, β) depend on the coordinates of an observed 3D point.

B. Projection of a 3D Point

Let $\mathbf{P} = [x\ y\ z]^T$ denote the coordinates of a 3D point. We want to find the image projection, $\mathbf{p} = [u\ v]^T$, of $\mathbf{P}$. We first determine $\mathbf{P_m}$ on the mirror surface, and finally project this point onto the image plane. The coordinates of $\mathbf{P}$ can be expressed in cylindrical coordinates as

$$\mathbf{P} = \left[\begin{array}{ccc} \varphi & r & z \end{array}\right]^T = \left[\begin{array}{ccc} \arctan\left(y/x\right) & \sqrt{x^2+y^2} & z \end{array}\right]^T$$

The vertical viewing angle α for P can be expressed as:

$$\alpha_P = \arctan\left(\frac{z - z_m}{r - r_m}\right) + \frac{\pi}{2},$$

where (r_m, z_m) denote the coordinates of $\mathbf{P_m}$ on the mirror surface. Hence, we can replace α in Equations (1) and solve the resulting non-linear system of equations to determine (r_m, z_m). Notice that knowing (r_m, z_m) determines the value of β.

Finally, we can project the 3D point $\mathbf{P_m} = [\varphi\ r_m\ z_m]^T$ onto the image plane $\mathbf{p} = (u, v)$. Using perspective projection and taking into account the camera intrinsic parameters, we get:

$$\left[\begin{array}{c} u \\ v \end{array}\right] = \left[\begin{array}{ccc} f_u & 0 & u_0 \\ 0 & f_v & v_0 \end{array}\right] \left[\begin{array}{c} \tan\beta\cos\varphi \\ \tan\beta\sin\varphi \\ 1 \end{array}\right],$$

where f_u, f_v denote the focal length expressed in (vertical and horizontal) pixels; and u_0, v_0 is the position of the principal point in the image coordinate system.

We have derived an operator, $\mathcal{P}$ that projects a 3D point $\mathbf{P}$ onto its image projection $\mathbf{p}$. This operator depends on all the intrinsic and extrinsic parameters of the catadioptric panoramic camera, which can be estimated based on a set of image projections $\mathbf{p}^i$, with known 3D coordinates, $\mathbf{P}^i$.

C. Panoramas and Bird's Eye Views

Images acquired with our omni-directional camera are naturally distorted. For instance, a corridor appears as an image band of variable width. Knowing the image formation model, we can correct some distortions to obtain Panoramic images or Bird's Eye Views.

In a panoramic image, each scan line contains the projections of all visible points at a constant angle of elevation. Hence, the unwarping consists of mapping concentric circles to lines [29]. For example, the horizon line is actually transformed to a scan line.

Bird's eye views are obtained by radial correction around the image center[1]. The bird's eye view is a scaled orthographic projection of the ground plane, and significantly simplifies the navigation system. For example, corridors appear as image bands of constant width. Image panoramas and bird's eye views are illustrated in Figure 4.

III. Navigating using Topological Maps

We use a *topological map* to describe the robot's *global* environment. This map is used to reference the qualitative position of the robot when traveling long distances. A mission could be specified as: "*go to the third office on the left-hand side of the second corridor*".

The robot must be able to travel along a corridor, recognize the ends of a corridor, make turns, identify and count door frames. These behaviors are implemented through an appearance based system and a visual servoing strategy.

The appearance based system provides qualitative estimates of the robot position along a corridor, and recognizes distinctive places such as corners or door entrances. This is achieved by comparing current omni-directional images to previously acquired views of the corridor (landmarks).

To control the robot's trajectory along a corridor, we detect the corridor guidelines and generate adequate control signals to keep the robot on the desired trajectory. This processing is performed on bird's eye views of the ground plane, computed in real-time.

Topological maps scale easily by connecting graphs at multiple resolutions to map different regions of the environment. All

[1]Hicks [30] has demonstrated how to obtain ground plane unwarped images, directly from a custom-shaped mirror.

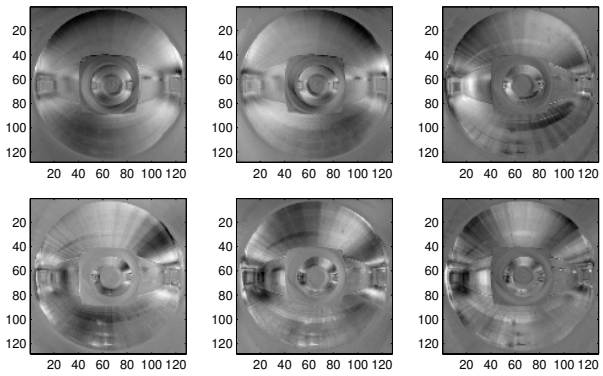

Fig. 5. The first 6 eigenimages obtained with the omni-directional vision system.

the robot needs is a specialized behavior to navigate along the *links* by using a vision-based control process and a procedure to recognize the locations/*nodes* where further actions may be undertaken.

A. Image Eigenspaces as Topological Maps

The topological map consists of a (large) set of reference images, acquired at pre-determined positions (landmarks), connected by links on a graph. Since the robot perceives the world through (omni-directional) images, these images are a natural way of represent landmarks.

During operation, the reference image that best matches the current view indicates the robot's *qualitative* position in the topological map. Hence, the reference images can be seen as a large-dimensional space where each point indicates a possible reference position of the robot.

In general, the number of images required to represent the environment is very large, and one needs to find a method to compress this information. We build a reduced-order manifold to approximate the reference images, using Principal Component Analysis (PCA), as described in [31].

The input space is composed of N images, $\mathbf{I_k}$. Using PCA, we can determine a set of $M \ll N$ *eigenimages*, $\mathbf{e}_j$, that form a low dimensional subspace which approximates the original set of reference images. These eigenimages are the eigenvectors of the covariance matrix formed by all the input images, and can be computed efficiently [32]. Each eigenvalue, λ_j, is proportional to the relevance of its associated eigenimage.

Figure 5 shows the first 6 eigenimages computed from 50 omni-directional images that represent one corridor, shown in descending order in accordance with their eigenvalues.

For our experiments, we keep the 10-12 eigenimages with the highest eigenvalues, denominated as the *Principal Components*. The reference images, $\mathbf{I}_k$ are coded by a vector of coefficients, $\mathbf{C}^k$, representing their projection along the principal components $\mathbf{e}_j$ of the reduced-order eigenspace.

Figure 6 illustrates how the reduced-order manifold, with only 10-12 eigenimages, can efficiently approximate the original input images.

Each reference image, $\mathbf{I_k}$ is associated with a *qualitative* robot position (e.g. half way along the corridor). To find the robot position in the topological map, we have to determine the reference image that best matches the current view, $\mathcal{I}$.

The distance, d_k, between the current view and the reference images can be computed directly using their projections, $\mathcal{C}$ and

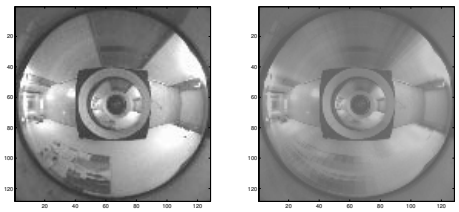

Fig. 6. The input (left) and retrieved (right) omni-directional images are very similar.

$\mathbf{C}^k$, on the lower dimensional eigenspace

$$d_k = (\mathcal{C} - \mathbf{C}^k)^T \Lambda (\mathcal{C} - \mathbf{C}^k) \tag{2}$$

where Λ is a diagonal matrix containing the (ordered) eigeinvalues which express the relative importance of the various directions in the eigenspace. Notice that d_k is computed between M-dimensional coefficient vectors (10-12, in our case), as opposed to image size vectors (128×128). The position of the robot is that associated with the reference image, $\mathbf{I}_k$ having the lowest distance, d_k.

Using omni-directional images, it is easier to deal with a relatively *dynamic* environment, where people partially occlude the robot's view. Even when a person is very close to the robot (Figure 7), the occlusion is not sufficiently large so as to cause the robot to misinterpret its topological position. Using PCA, the closest image is still correctly determined. An alternative approach is taken in [33] which used a number of small image windows (9x9 pixels) to deal with occlusions. Tolerance to illumination variations can also be improved by normalizing the brightness distribution.

An additional benefit of building the topological map using omni-directional images is that the same eigenspace can be used along both the forward and return trajectories, simply by rotating, in real-time, the acquired omni-directional images by 180^o.

Similarly, one could use the image's power spectrum, both to build the eigenspace and to represent the acquired images. The power spectra of panoramic images is invariant to image rotation and therefore, to the direction of robot motion. It offers an alternative way to travel along the topological map and recognize the various locations irrespective of the robot's orientation.

B. Corridor Following Behaviour

To navigate along the topological graph, we still have to define a suitable vision-based behavior for corridor following (*links* in the map). In different environments, one can always use simple knowledge about the scene geometry to define other behaviors. We exploit the fact that most corridors have parallel guidelines to control the robot heading direction, aiming to keep the robot centered in the corridor.

The visual feedback is provided by the omni-directional camera. We use *bird's eye views* of the floor, which simplifies the servoing task, as these images are a scaled orthographic projection of the ground plane (i.e. no perspective effects). Figure 8 shows a top view of the corridor guidelines, the robot and the trajectory to follow in the center of the corridor.

From the images we can measure the robot heading with respect to the corridor guidelines, β, and the distance to the central reference trajectory, ϵ_d.

We use a simple kinematic planner to control the robot's position and orientation in the corridor, using the angular velocity as the single degree of freedom. We consider a *look-ahead* distance, D, that defines, at each instant, the goal point that the

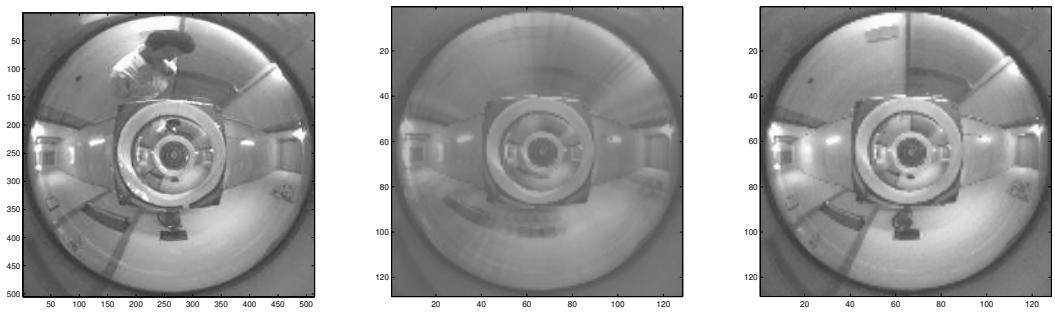

Fig. 7. (a) A person close to the robot as seen in the omni-directional image. The retrieved image (b) and the input image corresponding to the estimated position in the topological space (c). The occlusion did not affect the topological position estimation.

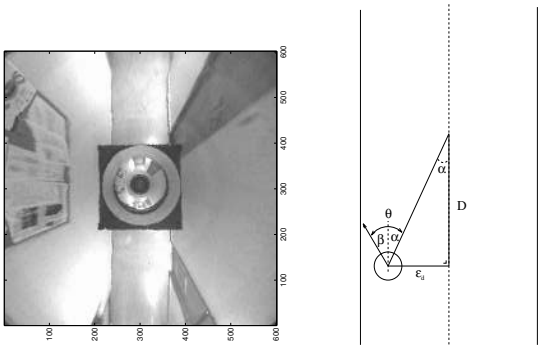

Fig. 8. Left - Bird's eye view of the corridor. Right - Measurements used in the control law: the robot heading β, the distance ϵ_d to the corridor center, and the angle, α towards a point ahead in the corridor central path. The error used for controlling the robot orientation is θ.

robot should aim for. Combining D and the robot heading and position errors, β and ϵ_d, we obtain the desired robot orientation, θ:

$$\theta = \beta + \arctan\left(\frac{\epsilon_d}{D}\right)$$

The value of D quantifies the overall influence of the displacement. It was set to 7.5% of the image width, which corresponded to about 1 meter. The value of θ is the input to a dynamic controller which controls the robot's angular velocity.

Tracking the corridor guidelines also benefits from using *bird's eye* (orthographic) views of the ground plane. We use projective-planar transformations (homographies) to predict the position of points and lines from one image to the next. These homograhies describe Euclidean image transformations and can be computed reliably using differential odometric data. Future improvements will eliminate the need for odometric readings.

To extract the corridor lines, we first find edges within predicted bounding boxes, and then use a robust line fitting procedure, based on RANSAC [34], which can deal with occlusions. The prediction is very accurate and vastly improves the probability of extracting the corridor guidelines rather than erroneous data such as door frames. Figure 9 shows a sequence of bird's eye view images acquired during tracking.

Notice that the use of bird's eye views of the ground plane simplifies both the extraction of the corridor guidelines (for e.g. the corridor has a constant width) and the computation of the robot position and orientation errors, with respect to the corridor's central path.

IV. Visual Path Following

Topological navigation is used to travel between distant places, without relying on accurate localization along a path. For local, precise navigation tasks, we rely on *Visual Path Following* for e.g. door traversal, docking and navigating in cluttered environments. In such cases, the robot must follow a reference trajectory accurately.

A. Feature tracking and self-localization

We use bird's eye views to track environmental features, estimate the robot's position/orientation and drive the robot along a pre-specified trajectory.

As features, we use corner points defined by the intersection of tracked edge segments which compare favorably, in terms of accuracy and stability, with the use of corner detection filters. Long edge segments are easier to track than corner points directly. For instance, we can continually track edge segments even when their intersection is occluded in the image, a situation where a corner detector would fail.

Edge segments are represented by 15 to 30 sampled points, that are tracked by searching the image perpendicularly to the edge segments. The search criterion is based upon the evaluation of the image gradient and the distance to the original edge position. Edge segments are obtained through a robust fitting procedure [34], and the new corner points are determined by their intersection.

We track edges lying on the ground plane as well as vertical edge segments. Corner points are defined by the intersection between ground edge segments or between vertical and ground line segments. Notice that vertical lines project as radial (or vertical) lines, in the bird's eye view (or panoramic) images. Since the robot position and orientation are estimated relative to a pre-defined coordinate system, the process of tracking is simplified by utilizing bird's eye (orthographic) views of the ground plane, thus preserving angular measurements and uniformly scaling distances.

Figure 10 illustrates tracking and self-localization while traversing a door from the corridor into a room. The tracked features (shown as black circles) are defined by vertical and ground-plane segments, tracked in bird's eye view images.

Currently, the user initializes the relevant features to track. To detect the loss of tracking during operation, the process is continuously self-evaluated by the robot, based on gradient intensities obtained within specified areas around the landmark edges. If these gradients decrease significantly compared to those expected, a recovery mechanism is launched.

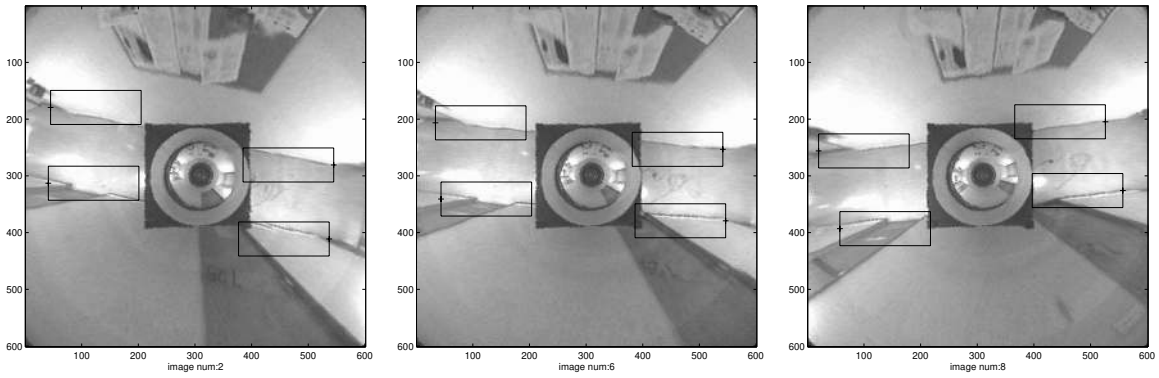

Fig. 9. Bird's eye views during tracking of the corridor guidelines.

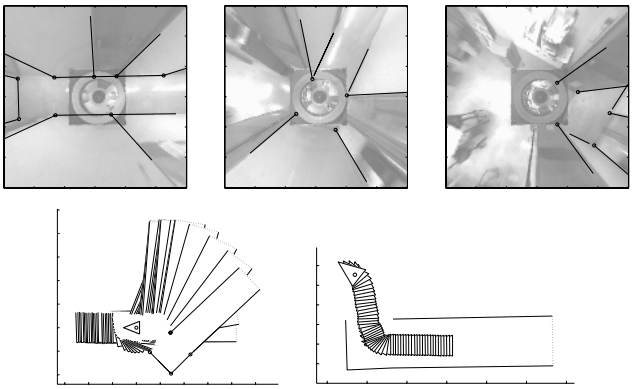

Fig. 10. Top: Feature tracking at three instants (black circles); Bottom: estimated scene model and self-localization results.

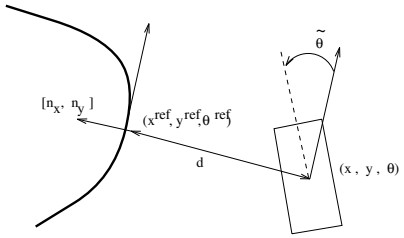

Fig. 11. Kinematic motion planner used to reference points and to define the control error for the visual path following system.

B. Robot Control

The robot state consists of a pose vector, $\tilde{\mathbf{x}} = (x, y, \theta)$, describing its position (in *pixels*) and orientation. The navigation system can modify the robot's linear and angular velocities denoted by (v, ω). The robot dynamic model is that of a wheeled unicycle mobile robot, with 2 degrees of freedom (linear and angular velocities).

We use an Extended Kalman Filter to estimate the temporal evolution of the robot's pose, (x, y, θ) and velocities, (v, ω). We assume that (v, ω) are constant and driven by white noise.

The path to follow, Ψ, is defined as a set of points $\tilde{\mathbf{x}}_\Psi = (x_\Psi, y_\Psi, \theta_\Psi)$, expressed in the same coordinate system and units as the robot state vector, $\tilde{\mathbf{x}}$.

At each time instant, the motion planning module must determine a reference point on the trajectory, $(x_\Psi^{ref}, y_\Psi^{ref})$ which is then used to determine the position and orientation errors so as to correct the robot's motion:

$$(x_\Psi^{ref}, y_\Psi^{ref}) = \underset{(x_\Psi^{ref}, y_\Psi^{ref})}{\arg\min} \left\{ \| \, (x_\Psi^{ref}, y_\Psi^{ref}) - (x, y) \, \|^2 \right\}$$

To avoid multiple solutions, we use a regularization term that selects the path point, $\tilde{\mathbf{x}}_\Psi^{ref}(k)$ closest to that at the previous time instant, $\tilde{\mathbf{x}}_\Psi^{ref}(k-1)$. A signed distance-to-path error, d and an orientation error, $\tilde{\theta}$ are defined as:

$$d = [x - x_\Psi^{ref} \quad y - y_\Psi^{ref}][n_x \quad n_y]^T, \qquad \tilde{\theta} = \theta - \theta_\Psi^{ref}$$

where $[n_x \quad n_y]$ is the normal to the path at the chosen reference point. The geometry of this kinematic motion planner is shown in Figure 11.

The dynamic controller used to generate the robot's angular velocity was proposed in [27] for path following, and shown to be stable:

$$\omega = -k_3 |v| \tilde{\theta} - k_2 v \; d \; \frac{\sin \tilde{\theta}}{\tilde{\theta}} + \frac{v \cos \tilde{\theta} c(s)}{1 - c(s) \; d} \tag{3}$$

where k_2, k_3 are constants to be tuned, s designates the path length, and $c(s)$ is the local path curvature.

Mostly, the forward velocity, v, is equal to the maximum, $V_{\max}$ but for safety reasons, we impose a maximum value on the angular velocity, $|\omega| < W_{\max}$. When this value is achieved, we saturate ω and reduce v to $V_{\max}.W_{\max}/\,|\omega|$, in order to avoid overshooting in narrow turns.

V. Experimental Results

The experiments described in this paper were undertaken at the Instituto de Sistemas e Robótica (ISR), in Lisbon, Portugal. It consists of a typical indoor environment, with corridors, offices and laboratories.

We used a TRC Labmate from HelpMate Robotics Inc., equipped with an omni-directional vision system (Figure 2) built in-house. This system contains a Cohu CCD camera pointed upwards, looking at a spherical mirror. Grayscale images were captured with a full resolution of 768x576 pixels, and sub-sampled to 128x128 images for PCA and 600x600 for visual servoing and Visual Path Following. All the processing was carried out on-board the mobile platform by a Pentium II 350MHz PC.

The results obtained illustrate the potential of our approach in a variety of different tests. First, we show separate results for *Topological Navigation* and *Visual Path Following*. Finally, we present integrated results which combine both global and local navigation methodologies.

A. Navigating using the Topological Map

The topological map was built with omni-directional images, acquired every 50 cm, along the corridors. At corresponding

distances, bird's eye views were also acquired and used for local pose control. Reference positions were ordered according to the direction of motion, thus maintaining a causality constraint.

To show that appearance based methods can provide qualitative estimates of the robot's position, we acquired a set of prior images, P, and ran the robot in the corridor to acquire a different set of run-time images, R. Figure 12 shows the distance d_k (see Equation (2)) between the prior and run-time images, P and R.

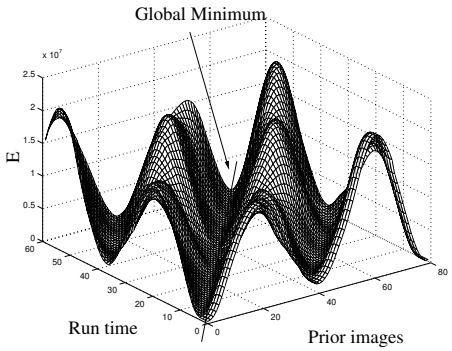

Fig. 12. A 3D plot of images acquired at run time, R versus those acquired a priori, P. This plot represents the traversal of a single corridor. The global minimum is the estimate of the robot's topological position.

The error surface presents a *global* minimum, corresponding to the correct estimate of the robot's topological position, and degrades in a piecewise smooth manner. Spurious local minima are due to distant areas of the corridor that may look similar to the robot's current position. These local minima are easily avoided by restricting the search space to images close to the previous estimated position, since images are captured sequentially according to the direction of motion.

In most cases we obtain the correct estimate for the robot position, even in the presence of occlusions. If misclassification occurs, the results are always in the vicinity of the correct answer, due to the smoothness of the error function.

Figure 13 shows results obtained when driving the robot along a corridor, using the behaviors described in Section III. The distance traveled was approximately 21 meters. Odometry was used to display the path graphically.

These results show that we can successfully drive the robot along the corridor and switch to a different behavior, when appropriate. In the example, this behavior is a 90^o turn, in order to proceed to the next corridor.

B. Visual Path Following Experiments

For *Visual Path Following*, we specified a reference trajectory in image coordinates, relative to a single landmark composed of two rectangles. The mobile robot uses the input of the omni-directional camera to move under closed loop control, as described in Section IV.

Figures 14(a,b) show estimates of self-localization. Noise is primarily due to the small size of the chosen landmark and poor image resolution. The Kalman filter can effectively reduce noise mainly along smooth paths. Figure 14(c) shows that the errors between the reference trajectory (dotted) and that resulting from visual self-localization (solid line) are very small. Figure 14(d) shows the mobile robot at the final position after completion of the desired navigation task.

The processing time was approximately 0.8sec/image, where 50% was used on image processing and the remaining 50% for displaying debugging information, image acquisition and serial communication with the mobile robot.

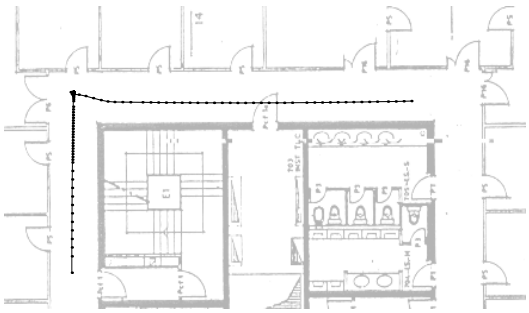

Fig. 13. One of the paths traveled by the robot at ISR.

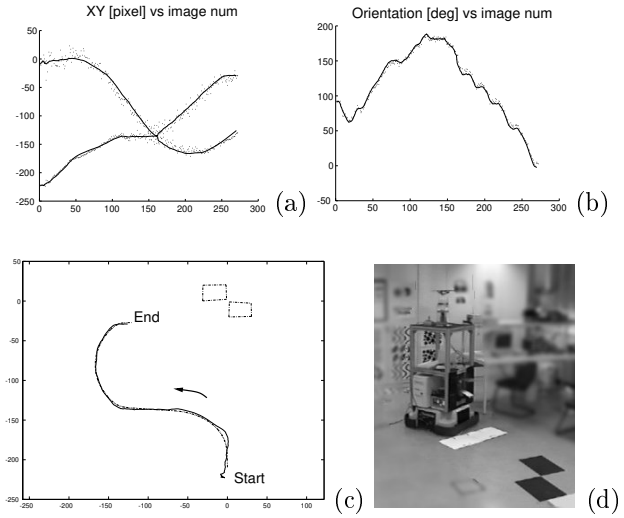

Fig. 14. Visual Path Following, with the trajectory specified in image coordinates. (a) x, y positions before (dotted line) and after filtering (solid line). (b) Orientation before (dotted line) and after filtering (solid line). (c) Dash-dotted line shows the landmark that defines the origin. The dotted line is the specified trajectory and the solid line shows the filtered position estimates. (d) Image of mobile robot at the end of path following.

C. Integrated Experiments

The concluding experiment integrates global and local navigation tasks, combining the *Topological Navigation* and *Visual Path Following* approaches.

The mission starts in the Computer Vision Lab. Visual Path Following is used to navigate inside the Lab, traverse the Lab's door and drive the robot out into the corridor. Once in the corridor, control is transferred to the topological navigation module, which drives the robot all the way to the end of the corridor. At this position a new behavior is launched, consisting of the robot executing a 180^o turn, after which the topological navigation mode drives the robot back to the Lab entry point. During this backward trajectory we use the same image eigenspaces as during the forward motion (see Section III). Finally, and once the robot is approximately located at the lab entrance, control is passed to the Visual Path Following module. Immediately it locates appropriate visual landmarks (see Section IV) and drives the robot through the door. It follows a pre-specified path until the final goal position, well inside the lab, is reached. Figure 15 shows an image sequence of the robot during this experiment.

Fig. 15. A sequence of images of an experiment combining visual path following for door traversal and topological navigation for corridor following.

Figure 16 shows the robot trajectory during one experiment, and its estimate using odometry. When returning to the laboratory, the uncertainty in odometry is approximately 0.5m. Thus, door traversal would not be possible without the use of visual control.

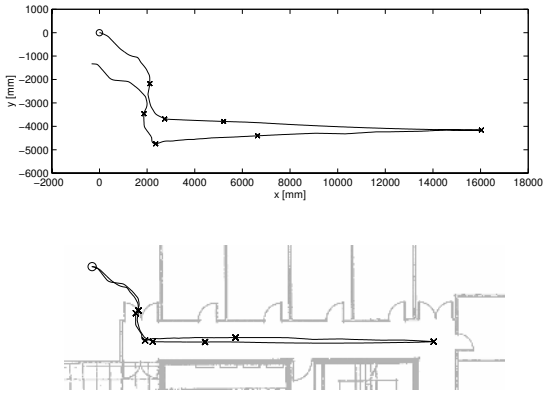

Fig. 16. The experiment combining visual path following for door traversal and topological navigation for long-distance goals. Trajectory estimate from odometry (top) and true trajectory (bottom).

This integrated experiment shows the use of Topological Maps for navigating between distant environmental points and Visual Path Following for accurate path traversal. The resulting system can robustly solve various navigation problems and makes parsimonious use of the available computational resources.

VI. Conclusions

We presented a method for the visual-based navigation of a mobile robot in indoor environments, using an omni-directional camera as the sole sensor.

Our key observation is that different navigation methods and environmental representations should be used for different problems, with distinct requirements in terms of processing, accuracy, goals, etc.

We distinguish between missions that involve traveling long distances, where the exact trajectory is unimportant (e.g corridor following), as opposed to other cases where the robot must accurately follow a pre-specified trajectory (e.g. door traversal). For these two types of missions we presented two distinct paradigms: *Topological Navigation* and *Visual Path Following*.

Topological Navigation relies on graphs that describe the topology of the environment. The qualitative position of the robot on the graph is determined efficiently by comparing the robot's current view with previously learned images, using a low-dimensional subspace representation of the input image set. At each node (landmark), a different navigation behavior can be launched, such as entering a door or turning left.

Whenever the robot needs to move in cluttered environments or follow an exact path, it resorts to *Visual Path Following*. In this case, tracked features are used in a closed loop visual controller to ensure that the robot moves according to the desired trajectory.

Omni-directional images are used in these two navigation modes to build the necessary environmental representations. For example, the *Bird's Eye Views* of the ground floor substantially simplify navigation problems by removing perspective effects.

Combining *Topological Navigation* and *Visual Path Following* is a powerful approach that leads to an overall system which exhibits improved robustness, scalability and simplicity.

References

[1] Z. Zhang and O.D. Faugeras, "Building a 3D world model with a mobile robot: 3D line segment representation and integration"," in *Proc. Int. Conf. Pattern Recognition*, 1990, pp. 38–42.

[2] R. Wehner and S. Wehner, "Insect navigation: use of maps or ariadne's thread?," *Ethology, Ecology, Evolution*, vol. 2, pp. 27–48, 1990.

[3] T. S. Collett, E. Dillmann, A. Giger, and R. Wehner, "Visual landmarks and route following in the desert ant," *J. Comp. Physiology A*, vol. 170, pp. 435–442, 1992.

[4] B. Schatz, S. Chameron, G. Beugnon, and T. S. Collett, "The use of path integration to guide route learning ants," *Nature*, vol. 399, pp. 769–772, 24 June 1999.

[5] J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi, "Divergent stereo in autonomous navigation : From bees to robots," *Int. J. Computer Vision*, vol. 14, no. 2, pp. 159–177, 1995.

[6] J. Santos-Victor and G. Sandini, "Visual behaviors for docking," *Computer Vision and Image Understanding*, vol. 67, no. 3, pp. 223–238, 1997.

[7] Y. Yagi, Y. Nishizawa, and M. Yachida, "Map-based navigation for mobile robot with omnidirectional image sensor COPIS," *IEEE Trans. Robotics and Automation*, vol. 11, no. 5, pp. 634–648, 1995.

[8] L. Delahoche, C. Pégard, B. Marhic, and P. Vasseur, "A navigation system based on an omnidirectional vision sensor," in *Proc. Int. Conf. Intelligent Robotics and Systems*, 1997, pp. 718–724.

[9] L. J. Lin, T. R. Hancock, and J. S. Judd, "A robust landmark-based system for vehicle location using low-bandwidth vision," *Robotics and Autonomous Systems*, vol. 25, pp. 19–32, 1998.

[10] K. Kato, S. Tsuji, and H. Ishiguro, "Representing environment through target-guided navigation," in *Proc. Int. Conf. Pattern Recognition*, 1998, pp. 1794–1798.

[11] V. Peri and S. K. Nayar, "Generation of perspective and panoramic video from omnidirectional video," in *Proc. DARPA Image Understanding Workshop*, 1997, pp. 243–246.

[12] S. K. Nayar, "Catadioptric omnidirectional camera," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1997, pp. 482–488.

[13] S. Baker and S. K. Nayar, "A theory of catadioptric image formation," in *Proc. Int. Conf. Computer Vision*, 1998, pp. 35–42.

[14] T. Svoboda, T. Pajdla, and V. Hlaváč, "Epipolar geometry for panoramic cameras," in *Proc. European Conf. Computer Vision*, 1998, pp. 218–231.

[15] S. C. Wei, Y. Yagi, and M. Yachida, "Building local floor map by use of ultrasonic and omni-directional vision sensor," in *Proc. of the IEEE Int. Conf. on Robotics and Automation*, 1998, pp. 2548–2553.

[16] B. Kuipers, "Modeling spatial knowledge," *Cognitive Science*, vol. 2, pp. 129–153, 1978.

[17] J. Košecká, "Visually guided navigation," in *Proc. Int. Symp. Intelligent Robotic Systems*, 1996, pp. 301–308.

[18] N. Winters and J. Santos-Victor, "Omni-directional visual navigation," in *Proc. Int. Symp. on Intelligent Robotic Systems*, 1999, pp. 109–118.

[19] J. Hong, X. Tan, B. Pinette, R. Weiss, and E. M. Riseman, "Image-based homing," in *IEEE Int. Conf. Robotics and Automation*, 1991, pp. 620–625.

[20] S. D. Jones, C. Andersen, and J. L. Crowley, "Appearance based processes for visual navigation," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 1997, pp. 551–557.

[21] N. Aihara, H. Iwasa, N. Yokoya, and H. Takemura, "Memory-based self-localization using omnidirectional images," in *Proc. Int. Conf. Pattern Recognition*, 1998, pp. 1799–1803.

[22] M. A. Turk and A. P. Pentland, "Face recognition using eigenfaces," in *Proc. IEEE Conf. Computer Vision and Pattern Recognition*, 1991, pp. 586–591.

[23] Y. Matsumoto, M. Inaba, and H. Inoue, "Visual navigation using view-sequenced route representation," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1996, pp. 83–88.

[24] H. Ishiguro and S. Tsuji, "Image-based memory of environment," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems*, 1996, pp. 634–639.
[25] I. Horswill, "Polly: A vision-based artifical agent," in *Proc. Nat. Conf. Artifical Intelligence*, 1993, pp. 824 – 829.
[26] J. Santos-Victor, R. Vassallo, and H. J. Schneebeli, "Topological maps for visual navigation," in *Proc. Int. Conf. Computer Vision Systems*, 1999, pp. 21–36.
[27] C. Canudas de Wit, H. Khennouf, C. Samson, and O. J. Sordalen, "Chap.5: Nonlinear control design for mobile robots," in *Nonlinear control for mobile robots*, Yuan F. Zheng, Ed. World Scientific series in Robotics and Intelligent Systems, 1993.
[28] J. Gaspar and J. Santos-Victor, "Visual path following with a catadioptric panoramic camera," in *Int. Symp. Intelligent Robotic Systems*, 1999, pp. 139–147.
[29] J. S. Chahl and M. V. Srinivasan, "Reflective surfaces for panoramic imaging," *Applied Optics*, vol. 36, no. 31, pp. 8275–8285, 1997.
[30] A. Hicks and R. Bajcsy, "Reflective surfaces as computational sensors," in *IEEE Workshop on Perception for Mobile Agents, CVPR 99*, 1999, pp. 82–86.
[31] H. Murase and S. K. Nayar, "Visual learning and recognition of 3D objects from appearance," *Int. J. Computer Vision*, vol. 14, no. 1, pp. 5–24, 1995.
[32] H. Murakami and V. Kumar, "Efficient calculation of primary images from a set of images," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 4, no. 5, pp. 551–515, 1982.
[33] V. Colin de Verdière and J. L. Crowley, "Local appearance space for recognition of navigation landmarks," in *Int. Symp. Intelligent Robotic Systems*, 1998, pp. 261–269.
[34] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of ACM*, vol. 24, no. 6, pp. 381–395, 1981.