Visual Navigation: Combining visual servoing and appearance based methods*

Raquel Frizera Vassallo, Hans Jorg Schneebeli

Departamento de Engenharia Elétrica
Universidade Federal do Espírito Santo
Av. Fernando Ferrari, s/n 29060-900
Vitoria/ES - Brasil
{raquel,hans}@ele.ufes.br

José Santos-Victor

Instituto de Sistemas e Robótica
Instituto Superior Técnico
Av. Rovisco Pais, 1,
1096 Lisboa Codex - Portugal
jasv@isr.ist.utl.pt

Abstract. We address the problem of visual-based navigation of a mobile robot in indoors environments. The robot control system is based on a single camera that observes the navigation area, and provides the required visual feedback information. The control strategy merges two distinct paradigms that appeared recently in the technical literature, in order to provide the robustness and computation speed needed for closed loop control. On one hand, we servo on the vanishing point defined by the intersection of the corridor guidelines. This mode is used for the heading control and ensures that the vehicle moves along corridors. On the other hand, we use appearance based processes to monitor the robot position along the path and to launch different navigation tasks (e.g. turn left, enter door, etc). The combination of visual servoing techniques that provide stable control loops for specific *local* tasks, and appearance based methods that embed a representation of the environment at a *larger* scale, results in extended autonomy even with modest computational resources. Preliminary tests have shown encouraging results, as discussed in the paper.

1 Introduction

Increasing the autonomy of a robot is intimately connected to its ability to perceive the surrounding environment and react to dynamic changes while performing a specified task. Examples in navigation include controlling the robot orientation and velocity, detecting obstacles, going to a point, etc.

Although vision can provide a rich description of the working space of a robot, earlier research was focused on partial or full reconstruction of the 3D structure of the scene before taking any action. In spite the fact that we have witnessed an impressive increase of the available computational power in the past few years, together with the advance of new vision algorithms and techniques, the reconstruction approach is still a difficult, demanding problem, hardly suitable for real-time navigation.

Since the early days of active vision [1, 3] it became clear that for some visual tasks one could profit from exploiting the structure of the specific problem to handle, together with the definition of *customized* algorithms to increase both the system robustness and speed. Typically it corresponds to sacrificing the generality of the reconstruction paradigm by specific visual measurements (in some cases imprecise or qualitative) relevant to a precise problem. Additionally, one would use vision in a more *continuous* way [13] to control the actions of a mobile agent.

This approach has led to successful working systems (see [15] for example) and a suitable framework was defined to study some of these problems [5]. However, it often encountered difficulties to deal with problems involving global tasks or coordinate systems, like going to a distant destination.

In this paper we use a priori knowledge about the 3D scene where the robot operates together with a purposive definition of the navigation task to be solved. We use a vision-based control system whenever a continuous stream of image features can be extracted from the video sequence. This is done by servoing on the vanishing point defined by the intersection of the corridor guidelines as they are observed by the robot camera [11].

In terms of projective geometry, these lines intersect at a point in the line at infinity belonging to the ground plane [8, 6]. If the camera orientation is kept constant relative to the guidelines, this point at infinity should be remain fixed. Otherwise a control action must be taken to keep the vanishing point under fixation.

* This research has been partially funded by projects PRAXIS/3/3.1/TPR/23/94 and JNICT-PBIC/TPR/2550/95.

As desirable, this technique does not require any knowledge of the camera calibration parameters. Figure 1 illustrates the view of a corridor captured by the on-board camera.

Fig. 1.: The corridor guidelines intersect at the vanishing point at the horizon line.

As discussed earlier, visual servoing alone can hardly cope with more global navigation tasks that require the existence of some kind of map or internal representation of the environment. Appearance based techniques offer an alternative solution for transforming pre-stored images in motion commands [12, 2]. Here we use such techniques as an implicit topological description of the environment, that is used to monitor the robot progress along the task.

In a simple office environment, one can then build a graph like structure representing the hallways, office doors, lifts, and other relevant environmental features. Links in the graph correspond to trajectory segments where visual features are available for visual servoing (as the corridor guidelines). Progression can be monitored using appearance based methods, simply by comparing previously stored images against acquired views. Nodes in the graph occur whenever decisions need to be taken like turning left, entering a door or picking an object.

Similar methods have been proposed in the past. In some cases it was assumed that image features were **always** available thus providing the necessary input to the visual servoing system (see [14, 10, 4], for example). In other cases, the use of appearance methods alone does not exploit the full geometric structure available for servoing [2] and may impose heavier computational requirements on the whole system.

While the robot is moving along a corridor, with the camera pointing forward, the navigation system must control the angular velocity based on the images acquired by the vision system. Meanwhile, the appearance based system provides *qualitative* mesaurements of the position of the robot, thus monitoring the progress of the overall mission. Once certain *relevant* positions are attained, other navigation behaviors are launched. This is done by an onboard computer, that runs the tasks of image processing and determines the changes of the robot angular velocity. The control sequence consists of the following operations:

- Image acquisition.
- Edge detection for finding the corridor guidelines.
- Calculation of the vanishing point and lines orientation.
- Determine the angular velocity command using a controller driven by the image based error signals.
- Monitor the robot progress along the task relying on appearance based methods and determine if special decisions need to be taken.

Because of the simplicity of the line detection process and the vanishing point determination, the image processing is quite fast and robust, as explained in the next section.

The experiments described in the paper were made with a TRC Labmate platform from HelpMate Robotics Inc. The processing was done in 192x144 grayscale images captured from a Sony pan-tilt camera installed on the robot, and a DT3852 frame grabber. Communication between the base motion controller and the main computer is done via a serial link.

This paper is organized as follows: Section 2 describes the geometric foundations related to the vanishing point determination and the servoing strategy. In Section 3, we show how appearance based methods can be used for extending the autonomy achieved by servoing methods alone. The the main experimental results are also included in this section. Finally, in Section 5, we draw some conclusions and final remarks.

2 Servoing on the vanishing point

In an indoors environment one has often to navigate along corridors to go to a different room to perform a task. For this reason, the ability to extract the visual information required to navigate along corridors is quite important [11]. Our system uses the information on the corridor guidelines for motion control.

We consider the robot and ground coordinate frames as shown in Figure 2. The robot has 3 degrees of freedom defined by its position on the ground plane and the heading direction.

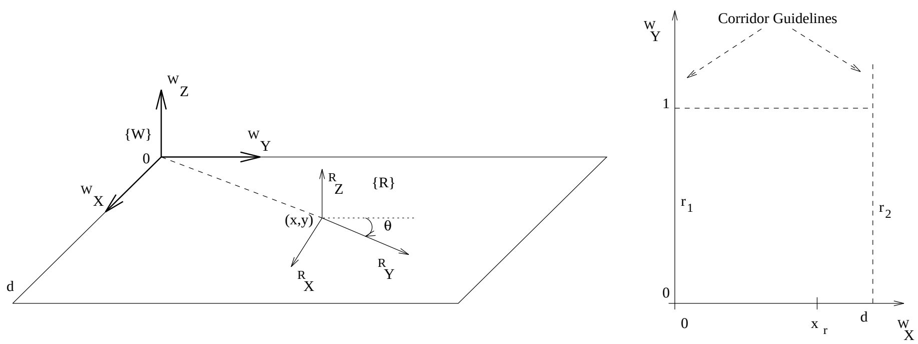

Fig. 2.: Robot and World (ground) coordinate frames, and the degrees of freedom identifying the robot position and orientation relative to the world frame.

As shown in Figure 2, the corridor guidelines can be defined by the points:

$$\begin{aligned} &\text{line } r_1: \quad (x_1, y_1) = (0, 0); \quad (x_2, y_2) = (0, 1) \\ &\text{line } r_2: \quad (x_3, y_3) = (d, 0); \quad (x_4, y_4) = (d, 1) \end{aligned} \tag{1}$$

where d denotes the corridor width. Any point ${}^W P$ expressed in the world coordinate frame $\{W\}$ can be expressed in the robot coordinate frame $\{C\}$ through a rotation matrix ${}^R R_W$ and a translation ${}^W P_{OR}$:

$${}^R P = {}^R R_W ({}^W P - {}^W P_{OR}) \quad \text{with} \quad {}^R R_W = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \tag{2}$$

where θ is the robot heading direction. We can then define a collineation on the projective space relating points expressed in the world coordinate frame and the same points expressed in the robot coordinate frame[2]:

$${}^R T_W = \begin{bmatrix} c_\theta & s_\theta & 0 & -x_r c_\theta \\ -s_\theta & c_\theta & 0 & x_r s_\theta \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{3}$$

where we denote the position of robot coordinate frame along the corridor x direction by x_r. The position along the y axis is set to zero, as it is irrelevant for this analysis.

We further assume that the camera coordinate frame $\{C\}$ differs from the robot coordinate frame by a *pitch*-angle, p, measured downwards, and a vertical translation by h (height), as shown in Figure 3. In this way ,we can now introduce another coordinate transformation from the robot frame to the camera frame, ${}^C T_R$:

$${}^C T_R = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -s_p & -c_p & h c_p \\ 0 & c_p & -s_p & h s_p \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{4}$$

[2] To simplify the notation we use (s_α, c_α) to denote $(\sin\alpha, \cos\alpha)$

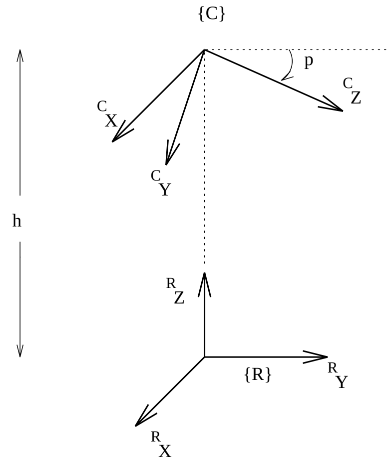

Fig. 3.: Robot and Camera coordinate frames differ solely by a rotation about that camera x axis - pitch, p, and a vertical translation of h.

Now that we have all the information to express the points on the ground plane on the camera coordinate frame, we can determine the image projections using the pin-hole camera model. The projection is achieved by the following camera matrix ${}^{i}T_C$:

$$
{}^{i}T_C = \begin{bmatrix} S_u & 0 & u_0 & 0 \\ 0 & S_v & v_0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{5}
$$

where S_u, S_v, u_o and v_o are the usual camera intrinsic parameters. On the remaining part of the paper, we assume that the image coordinate system is centered on the principal point, i.e. $(u_0, v_0) = (0, 0)$.

Using all the coordinate transformations defined above, we are now ready to express the overall coordinate transformation from the ground plane points $({}^{w}x,\ {}^{w}y,\ {}^{w}z)$ to the image projections $({}^{i}x,\ {}^{i}y)$:

$$
\begin{bmatrix} \lambda\, {}^{i}x \\ \lambda\, {}^{i}y \\ \lambda \end{bmatrix} = {}^{i}T_W \begin{bmatrix} {}^{w}x \\ {}^{w}y \\ {}^{w}z \\ 1 \end{bmatrix} \qquad \text{with} \quad {}^{i}T_W = {}^{i}T_C\, {}^{C}T_R\, {}^{R}T_W
$$

Noting that all the points of interest lie on the ground plane and therefore ${}^{w}z = 0$, we can define a homography ${}^{i}H_W$ relating ground plane points and the corresponding image projections:

$$
\begin{bmatrix} \lambda\, {}^{i}x \\ \lambda\, {}^{i}y \\ \lambda \end{bmatrix} = {}^{i}H_W \begin{bmatrix} {}^{w}x \\ {}^{w}y \\ 1 \end{bmatrix} \qquad \text{with} \quad {}^{i}H_W = \begin{bmatrix} S_u c_\theta & S_u s_\theta & -S_u c_\theta x_r \\ -S_v s_\theta s_p & S_v s_p c_\theta & S_v (s_p s_\theta x_r - h c_p) \\ -s_\theta c_p & c_p c_\theta & c_p s_\theta x_r + h s_p \end{bmatrix} \tag{6}
$$

The transformation ${}^{i}H_W$ depends on the robot heading and position, (θ, x_r), and on the camera intrinsic and extrinsic parameters. The corridor guidelines equations can now be determined and the corresponding image projections found. The projective coordinates of both lines $\tilde{r}_1, \tilde{r}_2$ are given by the vector product of the supporting points:

$$
\begin{aligned} \tilde{r}_1 &\sim \tilde{u}_1 \times \tilde{u}_2 = [1\ \ 0\ \ 0]^T \\ \tilde{r}_2 &\sim \tilde{u}_3 \times \tilde{u}_4 = [1\ \ 0\ \ d]^T \end{aligned}
$$

where $\sim$ denotes equality up to scale, and $\tilde{u}_1$ to $\tilde{u}_4$ are the homogeneous coordinates of the points defined in equation (1). Similarly, the point at infinity $\tilde{u}_\infty$ where these lines intersect can be determined in projective coordinates by the vector product of the lines representation:

$$
\tilde{u}_\infty \sim \tilde{r}_1 \times \tilde{r}_2 \sim [0\ \ 1\ \ 0]^T
$$

We can project $\tilde{u}_\infty$ on the image plane using equation (6) to get the vanishing point coordinates $({}^ix_v, {}^iy_v)$:

$$({}^ix_v, {}^iy_v) = (S_u \frac{\tan\theta}{c_p}, \quad S_v \tan p) \tag{7}$$

Equation (7) shows that the horizontal coordinate of the vanishing point depends on the robot heading, θ, on the horizontal pixel size and on the camera pitch angle. It does not depend on the robot position, x_r. Therefore, it can be extracted from images and when servoed to zero, we ensure that the vehicle heading is parallel to the corridor direction. To extract information regarding the robot position, x_r, we need to look at the image projection of corridor guidelines again and use information about the lines slope, m_i, and intercept b_i. In particular the image point where the lines cross the row $y = 0$ are quite meaningful:

$$\frac{b_1}{m_1} = S_u \frac{s_p x_r - s_\theta c_p h}{h c_\theta} \qquad \frac{b_2}{m_2} = S_u \frac{s_p(x_r - d) - s_\theta c_p h}{h c_\theta}$$

If we take the average between these two coordinates, it yields:

$$\begin{aligned} \delta_x &= \frac{1}{2}(\frac{b_1}{m_1} + \frac{b_2}{m_2}) \\ &= S_u \frac{s_p(x_r - d/2)}{h c_\theta} - S_u \tan\theta c_p \\ &= S_u \frac{s_p(x_r - d/2)}{h c_\theta} - {}^ix_v \cos^2 p \end{aligned}$$

The equations above show that the signal δ_x, that can be extracted from the image projection of the corridor guidelines, conveys information regarding the robot position error relative to the corridor center $(x_r - d/2)$. However, it is also dependent on a second term influenced by the robot heading. This coupled effect can be removed either if the camera pitch angle p or the camera parameter S_v, are known. The parameter S_v can be used to determine the camera pitch angle, p, from ${}^iy_v$ defined in equation (7).

Figure 4 shows the evolution of the vanishing point coordinate ${}^ix_v$, when the robot is centered in the corridor and the orientation changes. Similarly, the same figure shows how δ_x varies when the robot is suitably oriented in the corridor (i.e. $\theta = 0$) and its transversal position changes (when the robot is centered in the corridor, we have $x_r = 0.5$).

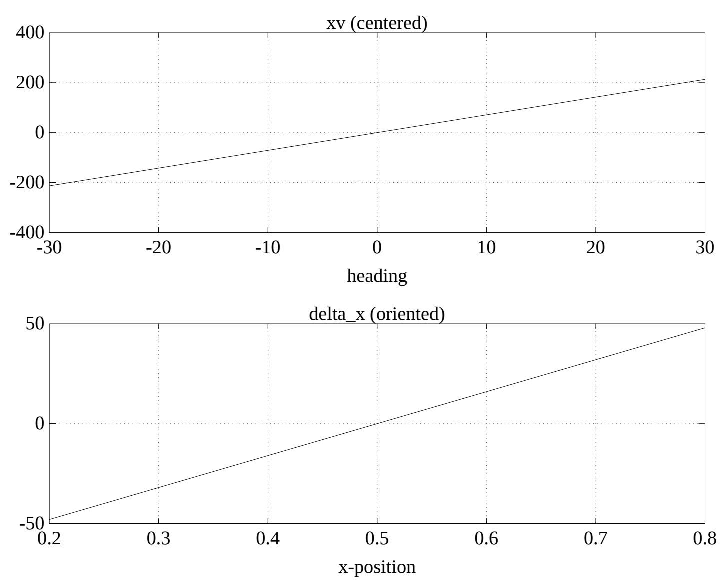

Fig. 4.: Top: change of ${}^ix_v$ when the robot orientation changes. Bottom : δ_x as a function of the robot position.

In some cases, when the robot is located away from the central trajectory in the corridor, using the vanishing point alone to control the robot heading can result in further deviating the robot from the central path. This is due to the fact that we need to control both the robot *orientation* **and** *position* with only a single error signal. To overcome this problem we use a combination of ${}^{i}x_v$ and δ_x to determine the control signal θ_c:

$$\theta_c(k) = \theta_c(k-1) - K_p({}^{i}x_v + \delta_x), \text{with} \quad K_p \sim 0.3 \tag{8}$$

Figure 5 shows a simulation of applying this control law when the robot is started off the corridor center and with an incorrect orientation. The left plot shows the evolution of the robot position and orientation when only the vanishing point information is considered. The robot corrects the orientation but fails to follow a central path in the corridor. On the right plot, we show the results of introducing δ_x. The robot corrects both the orientation and the position in the corridor.

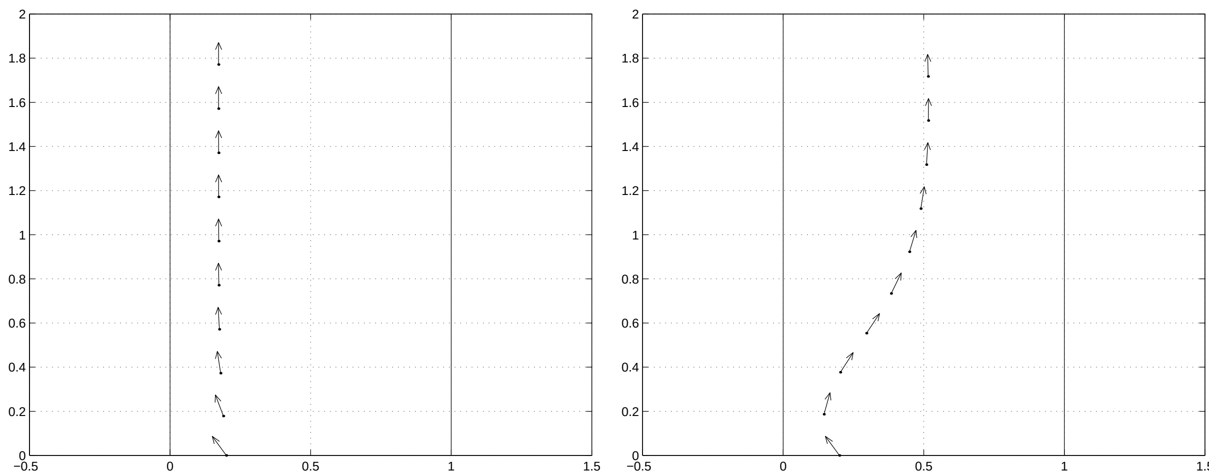

Fig. 5.: Simulation of the robot control when only the vanishing point is used (left) or when δ_x is also included.

2.1 Extracting the vanishing point data

The corridor guidelines are extracted using the Sobel gradient operator [8, 9] and by imposing a set of constraints to select the points of interest belonging to the guidelines. The procedure used is as follows:

- Extract the vertical adn horizontal image gradient components;
- Select the points that have significant horizontal **and** vertical gradient information. This procedure keeps only edge points that have diagonal directions.
- Separate this point set in the subsets of positive gradient orientation and negative gradient orientation. Each set corresponds to one of the corridor lines.
- Erode each image with a mask corresponding to diagonal lines, to further eliminate data outliers in the previous point sets.
- Use the selected points to robustly estimate the corridor lines, using Ransac [7], as the fitting procedure.

Once the two lines are determined we calculate the vanishing point. The deviation of the vanishing point from the central column of the image defines the error signal, ${}^{i}x_v$, to drive the controller of the heading direction. The result of the image processing described above is shown in Figure 6, where the lines detected and the selected points are superimposed on the original image.

So far we use a PID for the heading control, while the linear velocity is kept constant independently of the visual information. The current control frequency is about 1 Hz. Another relevant aspect is that we propagate the uncertainty information from the edge localization errors all the way till the determination of the vanishing point coordinates. This uncertainty measurement, $\sigma^2_{{}^{i}x_v}$, can be used not only to validate the extraction of the corridor guidelines but also to modulate the control gains as a function of the reliability of the visual measurements.

Fig. 6.: Results of the line detection process.

3 Extending the autonomy: appearance based navigation

As mentioned earlier, by servoing on image features alone it is hard to tackle navigation problems involving a more global representation of the environment.

We extend the robot autonomy by implicitly embedding a topological representation of the environment based on appearance based methods. The representation can be seen as a graph. Links correspond to trajectory segments where visual servoing can be used. Nodes correspond to locations where special actions need to take place like turning, entering a door, launching another navigation behavior, etc.

The progress along the links is monitored by using a set of reference images acquired at *meaningful* positions along the path. During normal operation, the current image is compared to the reference images, the best matching indicating the approximate position of the robot. Figure 7 shows a reference image set acquired along a corridor in our research institute.

Fig. 7.: Reference images $\{1^{st}, 3^{rd}, 5^{th}, 7^{th}, 9^{th}, 11^{th}\}$ used for the appearance based method.

Figure 8 shows an image acquired during the robot operation and the result of the comparison (using the SSD - *sum of squared differences* metric) against the reference image set.

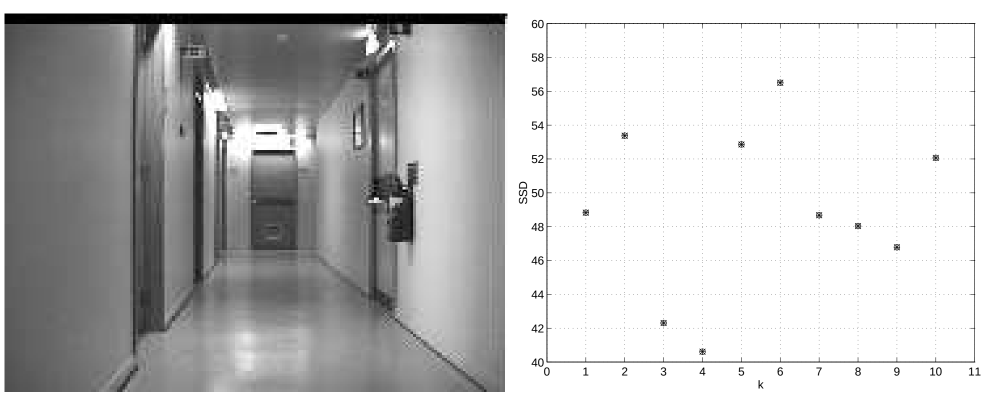

Fig. 8.: Left: Image acquired along the trajectory. Right: Result of the SSD computed between the candidate images and the reference image set. The best match is obtained against the 4^{th} image.

The comparison of the current image and the reference set is supported by the causality constraint, i.e the reference positions are ordered according to the direction of motion. The aim of the appearance based method is not to obtain accurate position or orientation estimates. Instead, we need *qualitative* information regarding the robot status along the task that is sufficient to trigger further actions. As an example, the final images in the corridor are used to realize that the robot is located in a corner and perform and adequate turn to proceed. Similarly, one of such images could indicate that the robot is close to the final destination, thus eliciting another operating mode.

Figure 9 shows some results obtained while maneuvering in a corridor. We can observe the robot trajectory recovered from odometry measurements. As mentioned before, we defined the control error as the horizontal deviation of the vanishing point to the image center, ${}^{i}x_v$. The figure also shows the temporal evolution of ${}^{i}x_v$ and the corresponding standard deviation estimate $\sigma_{{}^{i}x_v}$; and the heading angle over time.

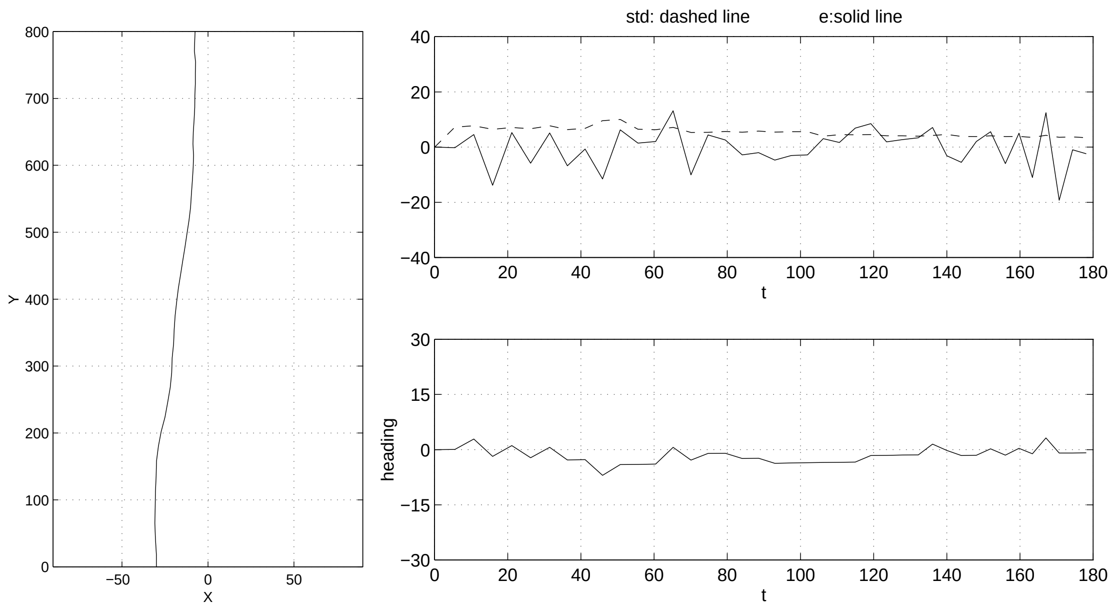

Fig. 9.: Left: Robot trajectory during the maneuver (cm). Top right : horizontal coordinate of the vanishing point and propagated standard variation estimate (both in pixels). Bottom right: robot heading (degrees) during the experiment.

We have tested the system in a simple indoor office environment to show the potential of integrating these two methodologies. The environment consists in a set of corridors with office doors on the sides. We have set the navigation goal to making a complete tour of all corridors. Hence, the topological map can be represented as in Figure 10. There are 5 nodes in the graph: the start and end positions and 3 intermediate corners of the corridors. In each corridor we use the servoing strategy as defined in the previous section. Simultaneously, the *indicative* robot position is inferred from the appearance based sub-system. The corners (nodes) correspond to special sites where a specific action must be taken. In this case these actions correspond to a clock-wise turn of 90 degrees.

This idea can be extended to consider missions like entering a given office. It would only require the appearance based system to identify the door and then a suitable servoing strategy would be launched. A complex environment can be mapped in such a way, considering various levels of the graph. A node could then be expanded to another sub-graph and the robot pursue to the following goal.

Figure 11 shows the result of the integrated navigation control system on our office floor. The images identifying the corners are also shown. The full path length is about 50 meters and the mission was accomplished without requiring accurate positioning or odometry information. For visualization purposes alone we plot the robot trajectory recovered from odometry. We have used a simple linear model to correct odometry errors along the corridors. This information is only need for displaying.

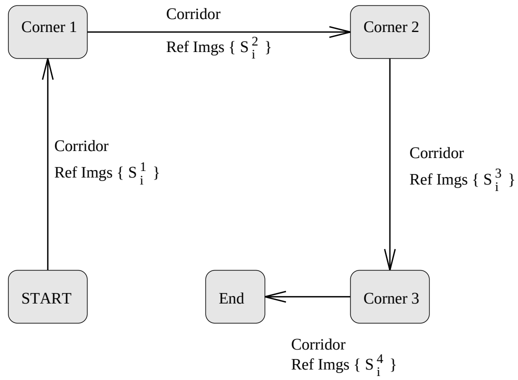

Fig. 10.: Topological map of the office environment. Nodes correspond to points where the robot must turn to another corridor. Navigation along the corridors is done using the servoing strategy and progress monitored by the appearance based approach.

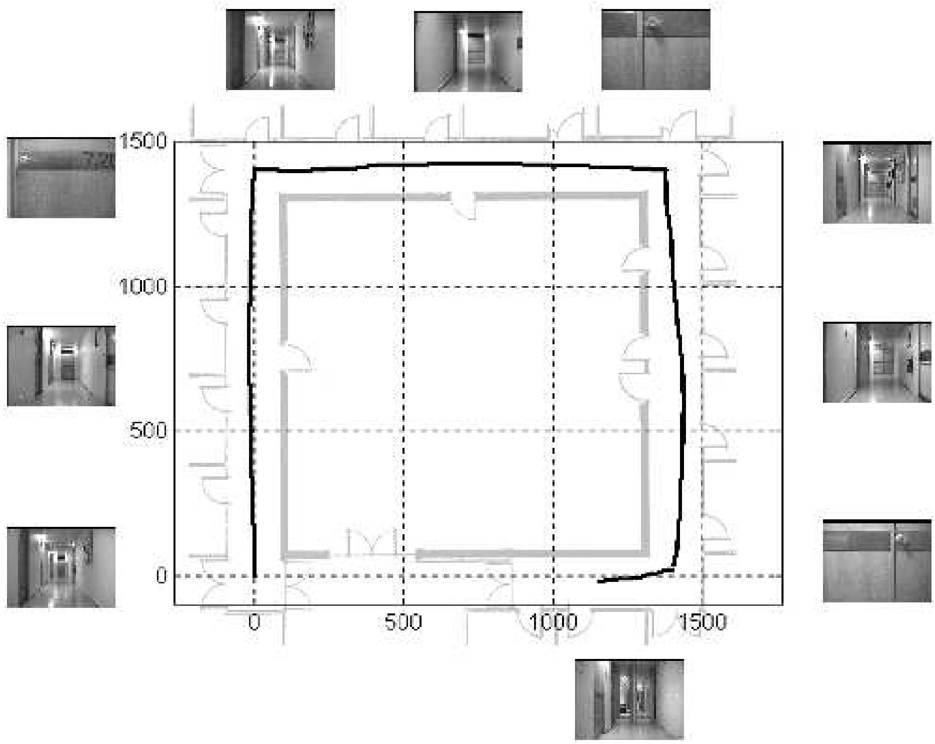

Fig. 11.: Navigation results on an office floor. The full path consists of about 50 meters. The integrated navigation system does not require accurate position information or odometry information.

4 Concluding Remarks

We have addressed in this paper the problem of visual-based mobile robot navigation using a single camera and processing a sequence of grayscale images captured during the robot motion.

The control strategy combines two different approaches. On one hand we servo on the vanishing point defined by the corridor guidelines as captured in the image. This mode is suitable when a continuous stream of image features can be extracted from the video sequence, but it does not provide a means with dealing with environment representation or global tasks.

A qualitative representation of the environment is implicitly embedded in the system by using appearance based methods for navigation. Images acquired in relevant positions along the path are stored initially. Then, during normal operation the system compares the current image with the reference set to approximately determine its position.

The environment “map” can then be seen as a graph structure whose links correspond to paths where visual servoing can be applied. Nodes indicate sites where decisions are to be taken: turning, entering a door, grasping an object, etc. The task execution is monitored by means of the appearance based methods. Visual servoing provides a robust way to perform *local* tasks while appearance based methods embed the environment representation required for more general tasks. Preliminary tests have shown encouraging results. We believe that by combining these two powerful approaches, one can significantly extend the robot autonomy without requiring accurate measurements of the robot position and orientation.

Some problems remain to be solved and are out of the scope of this paper. The most challenging idea is to define a method to automatically extract the reference images that will be used as landmarks. Right now, these images are acquired off-line at pre-defined positions. Another aspect for future research is to find more efficient image representations and associated comparison metrics to improve both the speed and robustness of the appearance based method.

References

1. Y. Aloimonos, I. Weiss, and A. Banddophaday. Active vision. *International Journal of Computer Vision*, 1(4):333–356, January 1988.
2. C. Andersen, S.D. Jones, and J.L. Crowley. Appearance based processes for navigation. In *Proc. of the 5th International Symposium on Intelligent Robotic Systems - SIRS97*, Stockholm, Sweden, July 1997.
3. R. Bajcsy. Active perception. *Proceedings of the IEEE*, 76(8):996–1005, August 1988.
4. C. Carreira and J. Santos-Victor. Vision based teleoperated cellular robots. In *Proc. of the 5th International Symposium on Intelligent Robotic Systems - SIRS97*, Stockholm, Sweden, July 1997.
5. B. Espiau, F. Chaumette, and Patrick Rives. A new approach to visual servoing in robotics. *IEEE Transactions on Robotics and Automation*, 8(3):313–326, June 1992.
6. O. Faugeras. *Three-Dimensional Computer Vision- A Geometric Viewpoint.* MIT Press, 1993.
7. 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*, 24(6), 1981.
8. B. Horn. *Robot Vision.* MIT Press, 1986.
9. R. Jain, R. Kastwi, and B.G. Schunck. *Machine Vision.* McGraw-Hill, 1995.
10. J. Kosecka. Visually guided navigation. *Robotics and Autonomous Systems*, 21(1):37–50, July 1997.
11. X. Lebegue and J.K. Aggarwal. Significant line segments for an indoor mobile robot. *IEEE Transactions on Robotics and Automation*, 9(6):801–806, December 1993.
12. J. Nielsen and G. Sandini. Learning mobile robot navigation: A behavior-based approach. In *IEEE Int. Conf. on Systems, Men and Cybernetics*, San Antonio, Texas, USA, October 1994.
13. G. Sandini, F. Gandolfo, E. Grosso, and M. Tistarelli. Vision during action. In Y. Aloimonos, editor, *Active Perception.* Lawrence Erlbaum Associates, 1993.
14. J. Santos-Victor and G. Sandini. Embedded visual behaviors for navigation. *Robotics and Autonomous Systems*, 19(3-4):299–313, March 1997.
15. J. Santos-Victor, G. Sandini, F. Curotto, and S. Garibaldi. Divergent stereo in autonomous navigation : From bees to robots. *International Journal of computer Vision*, 14:159–177, 1995.

This article was processed using the TEX macro package with SIRS98 style