Visual Path Following with a Catadioptric Panoramic Camera

José Gaspar and José Santos Victor⋆

Instituto de Sistemas e Robótica,
Instituto Superior Técnico,
Av. Rovisco Pais,
1049-001 Lisboa,
Portugal.

Abstract. In this paper we describe the use of a catadioptric panoramic system, built with a spherical mirror, for navigation tasks, namely visual path following. We define *Visual Path Following* as a method whereby once a robot has arrived at the start of a previously specified path, it can perform path following to a given location relying on visual tracking of features (landmarks) alone.
The geometry of the catadioptric sensor and the method of image unwarping to obtain a bird's eye view of the ground plane are presented. The ground unwarp representation significantly simplifies the solution to navigational problems, since the image coordinates differ from ground coordinates by a simple scale factor, thus eliminating any perspective effects.
Preliminary experiments with a mobile robot equiped with the described catadioptric panoramic sensor are detailed. Finally, encouraging and promising results obtained are presented.

1 Introduction

Many real world mobile robot applications, such as home assistance or office mail delivery require high levels of autonomy, but unlike industrial applications, the environment should not be altered to suit the given task. Typically, robot autonomy should cover large areas, for example several offices or an entire house. Building a complete metric description of such large areas, is expensive in computing power and in sensor allocation.

Alternative solutions [8, 9] use qualitative and topological properties of the environment for navigation, mainly when the robot has to travel rather large distances. A different approach is necessary when in regions with precise guidance or localisation requirements, e.g. a docking station. The work described here addresses such precise navigational problems, based on a method that we call *Visual Path Following*, to complement those systems based on topological maps.

We define *Visual Path Following* as a method whereby once a robot has arrived at the start of a previously specified path, it can perform path following to a given location relying on visual tracking of features (landmarks). In our work visual tracking is achieved using catadioptric panoramic images [1].

Catadioptric Panoramic Cameras are useful for visual tracking. They combine conventional cameras and mirrors primarily to obtain specific fields of view. Panoramic sensing is possible with convex mirrors such as conic mirrors, spherical mirrors or hyperbolic mirrors [1, 10, 11]. A panoramic sensor allows one to track a single feature from varying viewpoints, when it would otherwise be out of the field of view of a fixed classical camera. Another advantage, over a standard pan and tilt camera, lies in its simplicity: the system has no moving parts and given that the robot-sensor relative position remains fixed, the orientation of the sensor is related to that of the robot by a rigid transformation.

In this paper, we describe the image formation model for a catadioptric panoramic camera with a spherical mirror and the method used to obtain the model parameters. Although our catadioptric sensor does not have a single projection centre as in [1, 10, 11] we found that this is not a severe limitation to our approach. We describe a method of image unwarping to obtain bird's eye views of the ground plane. This representation allows algorithmic simplifications because perspective effects of the ground floor are eliminated. For example, position information for ground points is available without reconstruction [4] or uncalibrated reconstruction [6], and ground features move rigidly in these images, thus being easier to track. An algorithm

⋆ Email:jag@isr.ist.utl.pt, jasv@isr.ist.utl.pt

for tracking ground features (corners) is presented and utilising this, mobile robot self-localisation is shown to be achievable. Then, with the addition of a suitable controller, a *Visual Path Following* experiment in a real environment was performed and the results are detailed. Currently, the path is specified in image coordinates relative to an artificial landmark.

The paper is organised as follows: section 2 describes the geometry of a catadioptric panoramic camera with a spherical mirror, section 3 presents image unwarping to obtain bird's eye view of the ground plane, while section 4 presents the feature tracking algorithm. Section 5 describes the path following experiments and the preliminary results obtained. Finally, we present our conclusions and future work.

2 Panoramic camera with spherical mirror

In this section we detail the model for image formation with a spherical mirror. The experimental setup we used is shown in Figure 1 together with a drawing illustrating the image projection.

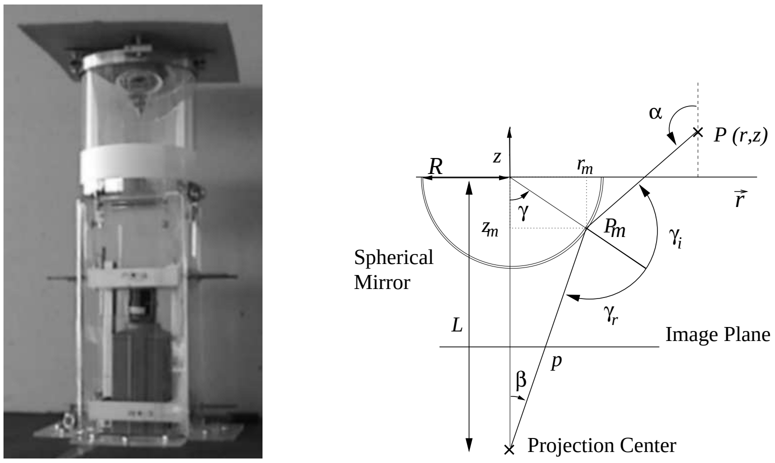

Fig. 1.: Catadioptric Panoramic Camera (spherical mirror) projection geometry. Symmetry about the z-axis simplifies the geometry to 2D (r,z).

The most relevant parameters are (see also figure 1): α the angle with vertical axis of ray from 3D point to reflection point ($max(\alpha)$ = system vertical view angle); β the angle at which camera sees the reflection point ($max(\beta)$ = camera / lens view angle) relative to system vertical axis; R the spherical mirror support sphere radius; L the distance of camera projection centre to mirror's support sphere centre; r the radial coordinate, distance to camera optical axis ($r = \sqrt{x^2 + y^2}$); z the vertical coordinate; γ_i, γ_r incidence and reflection angles (relative to local mirror normal); and finally γ identifies the reflection point.

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 p, as shown in Figure 1.

As shown on the drawing in the figure, a point $P_m = (r_m, z_m)$ on the mirror surface has to fulfill the following equations:

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

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 expressed using the coordinates of a 3D point (r, z).

Some of the 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.

2.1 Projection of a 3D Point

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

$$P = \begin{bmatrix}\varphi & r & z\end{bmatrix}^T = \begin{bmatrix}\arctan(y/x) & \sqrt{x^2+y^2} & z\end{bmatrix}^T$$

Noting that 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 P_m on the mirror surface, 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.

All that remains to be done is to project the 3D point $P_m = [\varphi\ r_m\ z_m]^T$ onto the image plane $p = (u, v)$. Using the perspective projection model and taking into account the camera intrinsic parameters, we get:

$$\begin{bmatrix}u^*\\ v^*\end{bmatrix} = f\ \tan\beta \begin{bmatrix}\cos\varphi\\ \\ \sin\varphi\end{bmatrix} \quad \text{and} \quad \begin{bmatrix}u\\ v\end{bmatrix} = \begin{bmatrix}\alpha_u & 0 & u_0\\ 0 & \alpha_v & v_0\end{bmatrix}\begin{bmatrix}u^*\\ v^*\\ 1\end{bmatrix}.$$

where $\alpha_u, \alpha_v, u_0, v_0$ denote the vertical and horizontal image scale factors and position of the principal point on the image coordinate system.

2.2 Model parameters estimation

In the previous section we derived a projection operator, $\mathcal{P}$ that given the 3D coordinates of a point $P = [X\ Y\ Z]^T$ allows us to obtain its image projections $p = [u\ v]^T$:

$$p = \mathcal{P}(P, \theta) \tag{2}$$

where θ contains all the intrinsic and extrinsic parameters of the catadioptric panoramic vision sensor:

$$\theta = [L\ R\ \alpha_u\ \alpha_v\ u_o\ v_o]^T$$

The mirror radius can be measured easily and we assume that it is known, $R = 8.35\ cm$. We further assume that the pixel aspect ratio is known. However, the camera-mirror distance, L, the overall image scale factor α and the principal point, (u_0, v_0), can only be measured up to some measurement error:

$$\begin{cases} L = 27 + \delta L\ cm\\ \alpha = \hat{\alpha} + \delta\alpha\\ u_o = 0 + \delta u_0\\ v_o = 0 + \delta v_0 \end{cases} \tag{3}$$

Hence, we define the adjustment $\delta\theta$ required to correct the nominal parameter vector θ:

$$\delta\theta = [\delta L\ \delta\alpha\ \delta u_0\ \delta v_0]^T$$

To estimate $\delta\theta$ we use a set of known 3D points, P^i, and the corresponding image projections p^i, and minimize the following cost function:

$$\delta\theta = \arg\min_{\delta\theta} \sum_i \parallel p^i - \mathcal{P}(P^i, \delta\theta) \parallel^2 \tag{4}$$

At this point we have defined the projection operator for catadioptric panoramic images with spheric mirrors, and described a procedure to estimate the model parameters, starting from initial nominal settings.

3 Unwarping to bird's eye view of ground plane

The images acquired by our panoramic sensors are naturally distorted due to the geometry of the mirror and the perspective camera projection. Different world areas are mapped with different image resolutions. In general, 3D straight lines are projected as curves in the image. For instance, the horizon line is projected as an image circle. Only 3D lines that belong to vertical planes containing camera and mirror axis project as straight lines.

In this section we present a method to unwarp a catadioptric panoramic image to a bird's eye view of the ground plane. Firstly, we rewrite the projection operator, $\mathcal{P}_\rho$ in order to map radial distances, ρ_{ground} measured on the ground plane, to radial distances, ρ_{img}, measured in the image:

$$\rho_{img} = \mathcal{P}_\rho(\rho_{ground}, \theta) \tag{5}$$

Using this information, we build a look up table that maps radial distances from the ground plane to the image coordinates. Since the inverse function cannot be expressed analytically, once we have an image point, we search the look up table to determine the corresponding radial distance on the ground plane. The image unwarping to a bird's eye view is done efficiently in this way.

Figure 2 shows an example of ground unwarping. The ground pattern shown in the original image becomes a rectangular pattern on the ground unwarped image, as desired.

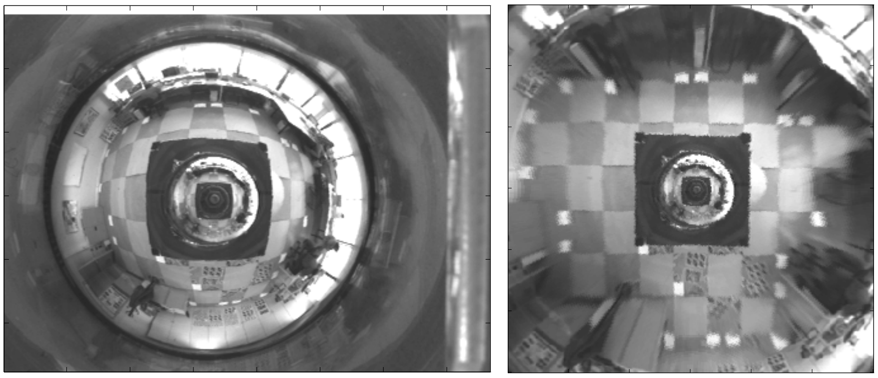

Fig. 2.: Image unwarping for bird's eye view. Left to right: original and remapped images.

4 Features tracking and self-localisation

We described already the geometric transformations required to obtain a bird's eye view of the ground floor using a catadioptric panoramic camera with a spheric mirror. The following step is to use these images in order to track environmental features to allow the estimation of the robot position/orientation or to drive the robot along a pre-specified trajectory. In this section, we describe the features used for this purpose and the method adopted for tracking.

The features we selected for tracking are image corners defined by intersections of edge segments [7]. Edge segments can usually be found in indoor environments, and their detection can benefit from a larger spatial support, as opposed to local corner detection filters. As a result, identifying corner points as intersections of tracked edge segments leads to a better accuracy and stability. Additionally, it is easier to track long edge segments than corner points directly.

Figure 3a shows the definition of corner point E, as the intersection of lines $\overline{AB}$ and $\overline{CD}$. In this way, corners do not correspond necessarily to image points of extreme brightness changes. This approach is beneficial when dealing with information loss due to occlusions or filtering (e.g. the "roundness" of corners due to image smoothing).

We track corner points by tracking the corresponding support edge lines. Edge segments are represented by a fixed number of sampled points, typically ranging from 7 to 15. We track these points by searching along

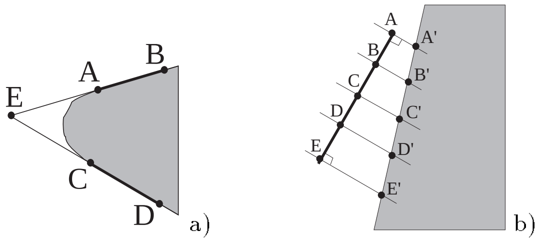

Fig. 3.: a) Definition of corner point E, as the intersection of lines $\overline{AB}$ and $\overline{CD}$; b) Segment $\overline{AE}$ is adjusted to $\overline{A'E'}$.

image directions perpendicular to the line segment guided by a local photometric and geometric criterion. Figure 3b shows how points $(A, ..., E)$ are adjusted to $(A', ..., E')$. The search criterion is based on the evaluation of local gradient information, and distance to the original edge position.

Once the correct edge points have been found, the new edge segment is obtained through a fitting procedure using all the detected points [5]. With the tracked edge segments in a new image, we can update the corner points positions.

From these new corner positions, we determine a 2D rigid transformation, $T2D$, that relates two neighboring images. At least 2 corners points are required to estimate this transformation. All line segments are then repositioned according to $T2D$, and all the corner points that do not verify this transformation are rejected as outliers. By repeating the estimation procedure, using the remaining points only, we obtain more robust estimates of $T2D$.

Figure 4 illustrates successful feature tracking results. We used a known pattern located on the ground plane as a navigation landmark, and acquired an image sequence while the robot moves. The chosen landmark is defined by eight line segments intersecting at 8 corner feature points that are tracked according to the method described above. Figure 4 contains the results of tracking in 3 different time instants, illustrating the initial and final positions of the tracked edges.

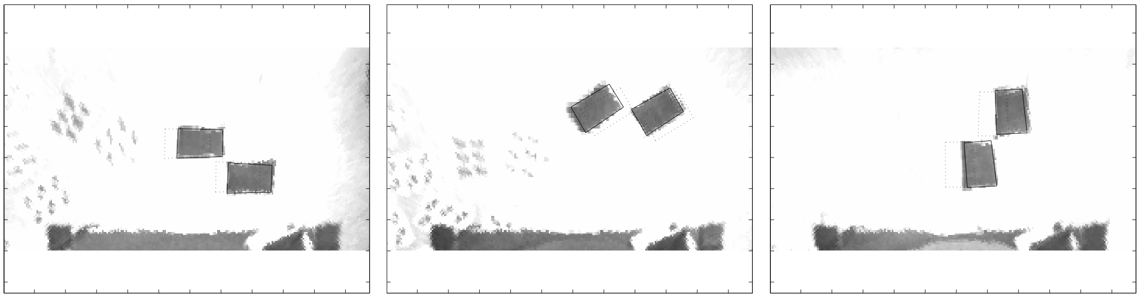

Fig. 4.: Features tracking. Left to right: Dashed and solid lines show the landmark original and final positions, respectively.

At the current stage of implementation the relevant features to track and the feature coordinate system are initialized by the user. In order to detect the loss of tracking during operation, the tracking process is continuously self-evaluated by the robot. This evaluation is based on gradient modules obtained within specified areas around the landmark edges. If these gradients decrease significantly compared to those expected, a recovery mechanism is then launched.

Having successfully tracked these visual features, we can estimate the robot position and orientation relative to some pre-defined coordinate system. This is greatly simplified by having a bird's eye image representation since it provides an orthographic projection of the ground plane, where image angles are equal to angles measured on the ground plane, and image lengths differ by a uniform scale factor from ground plane distances.

5 Visual Path following

In this section we will show how to use the information available from the tracked points so as to design a control system that drives to robot along a pre-described trajectory.

The dynamic model for our robot is that of a 2 degrees of freedom (linear and angular velocities) mobile platform:

$$\text{State vector:} \quad \tilde{\mathbf{x}} = [x\ y\ \theta]^T, \qquad\qquad \text{State Space Model:} \quad \begin{cases} \dot{x} = v\cos\theta \\ \dot{y} = v\sin\theta \\ \dot{\theta} = \omega \end{cases} \tag{6}$$

where (x, y) describe the robot position in **pixels**, θ denotes the robot orientation and (v, ω) indicate the robot linear (forward) and angular velocities.

The path, Ψ, to be followed [2] is defined as a collection 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 and depending on the robot position (x, y), the motion planning module, must determine a reference point on the trajectory $(x_\Psi^{ref}, y_\Psi^{ref})$ that will be used to determine the position and orientation errors to correct the robot motion:

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

In some cases this problem is ill-posed and may have multiple solutions. For this reason, there is a regularization term that selects the path point, $\tilde{\mathbf{x}}_\Psi^{ref}(k)$ closest to that of the previous time instant, $\tilde{\mathbf{x}}_\Psi^{ref}(k-1)$.

With current state of the robot and the desired path reference point computed, a signed distance-to-path error d and 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 normal to the path at the chosen reference point. The geometry of this simple kinematic motion planner is shown in Figure 5.

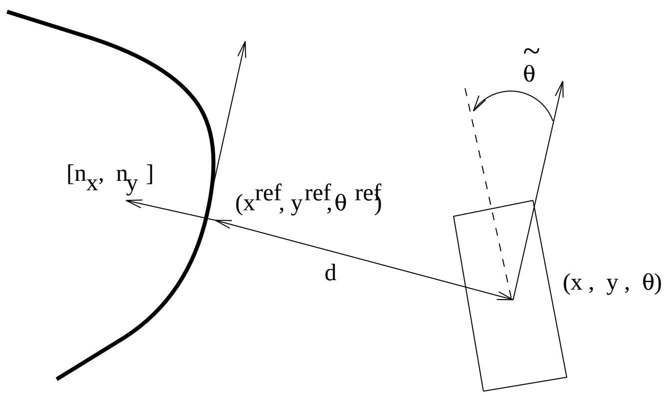

Fig. 5.: Kinematic motion planner used to reference points and define the control error for the visual path following system.

The dynamic controller to generate the robot angular velocity is that proposed by *Canudas et al* [3] for path following, for which they also presented a stability proof:

$$\omega = -k_3|v|\tilde{\theta} - k_2 v\ d\ \frac{\sin\tilde{\theta}}{\tilde{\theta}} + \frac{v\cos\tilde{\theta}c(s)}{1 - c(s)\ d} \tag{7}$$

where k_2, k_3 are constants to be tuned accordingly to specific vehicle and desired system response distance, s designates the path length, and $c(s)$ is local path curvature.

In order to operate safely we impose $|\omega| < W_{\max}$. When the control law (7) indicates larger values, instead of just thresholding ω, which would increase overshoots in narrow turns, we reduce v to $v.W_{\max}/\omega$.

For the current control law, noise in self-localisation measurements (x, y, θ) directly imply noise in control outputs (v, ω). To prevent this direct noise transmission we include temporal integration of the measurements with an Extended Kalman Filter. The inputs and outputs for the EKF are then poses (x, y, θ), and the dynamics are that of a wheeled mobile robot - unicycle type (6) with state vector augmented with velocities $\mathbf{x} = (x, y, \theta, v, \omega)$. Velocities are assumed constant and driven by white noise. Forward velocity noise covariance is assumed low due to the control characteristics.

5.1 Experimental results

Experiments were conducted using catadioptric panoramic vision system built in our institute, mounted on a TRC labmate mobile robot. Processing was carried on with an on-board PC PII-350MHz equipped with a TekRam image acquisition board.

Figure 6 shows simulation of a wheeled mobile robot performing visual path tracking. The self-positioning is simulated with mobile robot model integration.

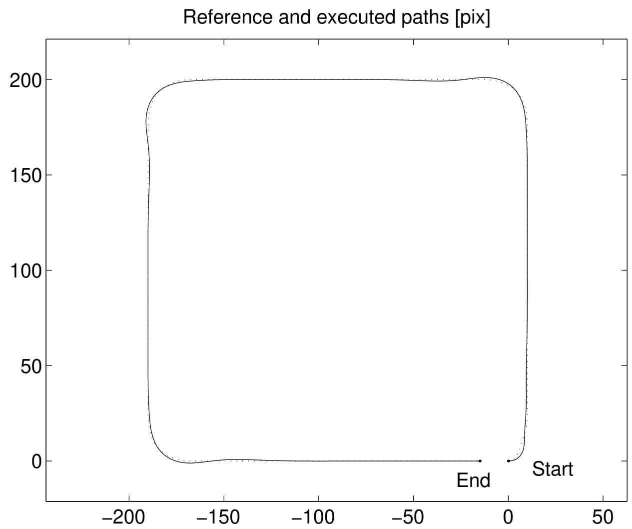

Fig. 6.: Visual path tracking shown in a simulated trajectory, with the robot moving anti-clockwise. Dotted and solid lines correspond to reference and robot trajectories respectively.

Figure 7 shows the error signals and control actions for the same input and using the kinematic planner and dynamic controller described before. This example illustrates forward velocity reduction to complement angular velocity saturation.

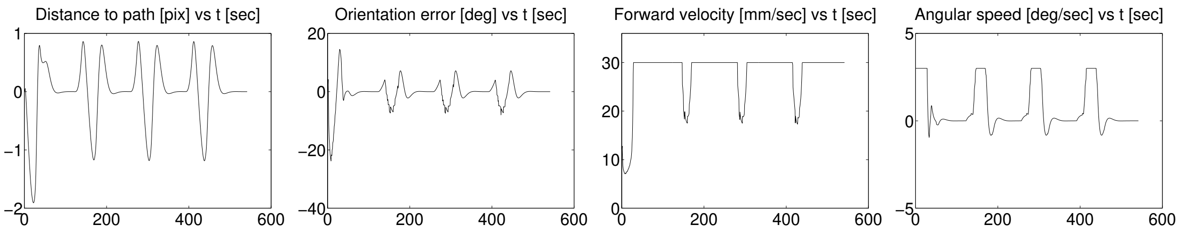

Fig. 7.: Visual path tracking in a simulated trajectory. Left to right: distance to path and orientation error (controller inputs), forward and angular velocities (controller outputs).

Finally, a real world experiment was undertaken. A reference trajectory was specified in image coordinates and the mobile platform was controlled based on input from the vision sensor. Figures 8a and 8b show estimates of self-localisation. The noise in these estimates is mainly due to the small size of the chosen landmark and poor image resolution. The implemented filtering is effective in reducing noise mainly along smooth paths. In the future, by choosing widely separated landmarks we expect to have very significant improvements in the accuracy, so filtering will be effective along sharper paths. The possibility of tracking widely separated image features is indeed one of the key advantages of using panoramic images for localisation. Figure 8c shows the reference trajectory (dots) and results of visual self-localisation (solid line).

Figure 8d 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% is used in image processing and the remaining 50% in consumed in displaying debugging information, for image acquisition and for serial communications with the mobile robot.

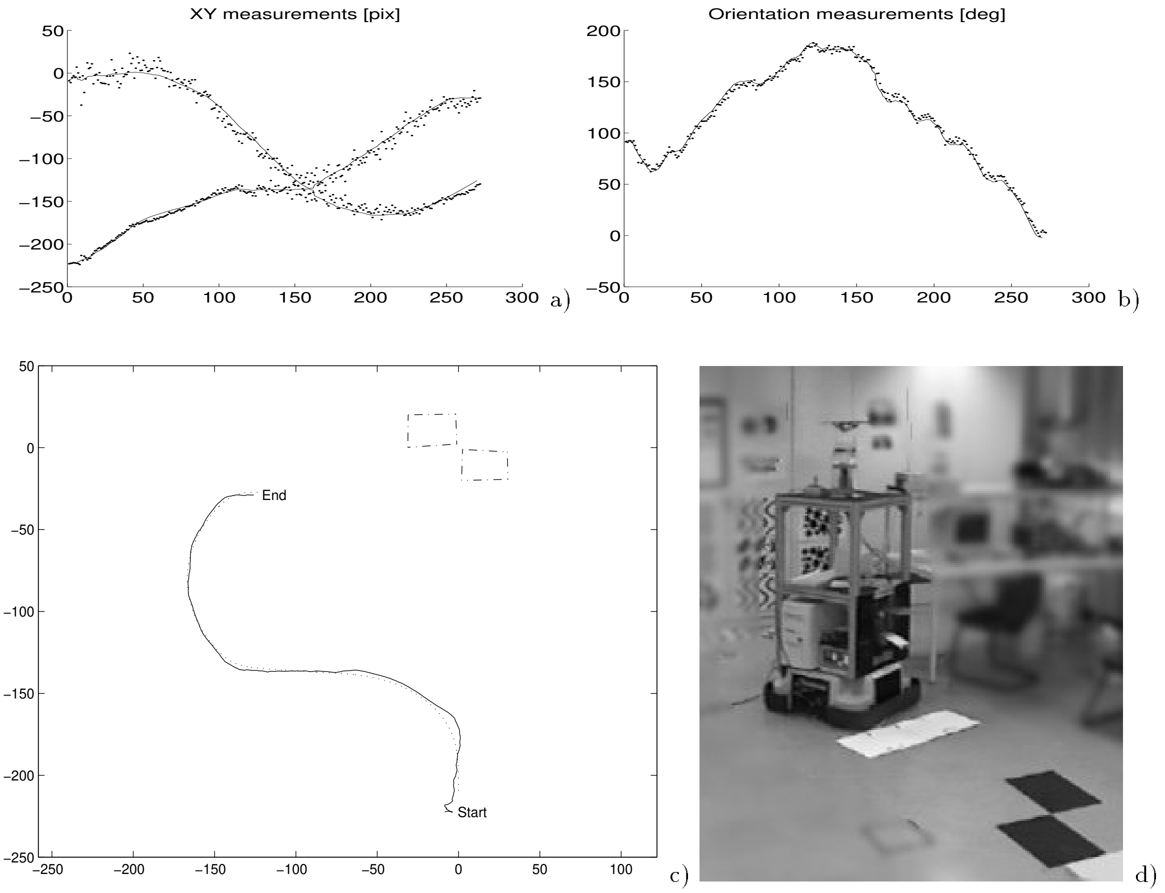

Fig. 8.: Real world visual path following, where the trajectory was specified in the image coordinates. a) x, y positions before filtering (dotted line) and after filtering (solid line). b) Orientation before filtering (dotted line) and after filtering (solid line). c) Dashed and dotted line shows the landmark that defines the origin. The dotted line is the specified trajectory. The solid line shows the filtered position estimates obtained from the visual input. d) Image of mobile robot at the end of path following.

6 Conclusions

We described the use of a catadioptric panoramic system, built with a spheric mirror, for navigation tasks, namely visually path following. One of the main advantages lies in acquiring panoramic images of the environment without any moving parts on the physical setup.

We have described the image formation model and the method adopted for estimating the model parameters. Using this projection model, we presented a method for obtaining ground unwarped images or a bird's eye view of the ground floor. This representation significantly simplifies navigation problems, since the image coordinates differ from the ground plane coordinates by a simple scale factor, thus eliminating any perspective effects.

Using this ground unwarped images we have presented a method to track image corner points defined by their support edge segments. Tracking this features is done in a robust manner and allows the estimation of the robot position and orientation relative to a known navigation landmark.

We further showed how this framework can be used to perform what we call *Visual path following*. A trajectory to follow is simply specified in the image plane and a suitable controller is used to drive the robot along that desired path. We described both the kinematic motion planner and the dynamic controller. Experiments were shown both in a simulated environment and with a real robot.

Future work will focus on ameliorating the self-localisation system, by using widely separated landmarks that remain visible in the panoramic images. The extension of ground plane landmarks to include vertical lines is straightforward, since the intersection of such vertical lines with the ground plane define ground plane feature points and so the calculations for self-localisation are the same. The implemented Extended Kalman Filter, currently only used for filtering noise of estimated self-localisations, may also be used to improve feature tracking by considering the EKF's predictions, thus further adding to the accuracy and robustness of the system.

Finally, we aim to integrate this work with a more global navigation system. Such a navigation system may rely on qualitative and topological properties of the environment for travelling relatively large distances, while our approach would fit in for precise guidance or localisation when negotiating a door entrance or other accurate positioning and navigation tasks.

Acknowledgements

This work has been partially funded by the project PRAXIS 2/2.1/TPAR/2074/95, SIVA.

References

1. Simon Baker and Shree K. Nayar. A theory of catadioptric image formation. In *ICCV'97*, pages 35–42, January 1998.
2. James L. Crowley. Mathematical foundations of navigation and perception for an autonomous mobile robot. Workshop on Reasoning with Uncertainty in Robotics 4-6 Dec. 1995; also in www-prima.imag.fr/Prima/Homepages/jlc/ navigation.tutorial.word.ps, 1995.
3. C. Canudas de Wit, H. Khennouf, C. Samson, and O. J. Sordalen. Chap.5: Nonlinear control design for mobile robots. In Yuan F. Zheng, editor, *Nonlinear control for mobile robots*. World Scientific series in Robotics and Intelligent Systems, 1993.
4. O. Faugeras. *Three-Dimensional Computer Vision - A Geometric Viewpoint*. MIT Press, 1993.
5. Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of ACM*, 24(6):381–395, June 1981.
6. G. Hager, D. J. Kriegman, A. S. Georghiades, and O. Ben-Shahar. Toward domain-independent navigation: Dynamic vision and control. In *IEEE CDC'98*, Dec. 1998.
7. Robert M. Haralick and Linda G. Shapiro. *Computer and Robot Vision (vol. 1)*. Addison-Wesley, 1992.
8. Jana Kosecka. Visually guided navigation. In *SIRS*, Lisbon Portugal, July 1996.
9. José Santos-Victor, Raquel Vassallo, and Hans-Jorg Schneebeli. Topological maps for visual navigation. In *1st International Conference on Computer Vision Systems*, pages 1799–1803, 1999.
10. T. Svoboda, T. Pajdla, and V. Hlavác. Epipolar geometry for panoramic cameras. In *ECCV'98*, pages 218–231, Freiburg Germany, July 1998
(also http://cmp.felk.cvut.cz/ svoboda/Publications/svobECCV98.html).
11. Shih-Chieh Wei, Yasushi Yagi, and Masahiko Yachida. Building local floor map by use of ultrasonic and omni-directional vision sensor. In *ICRA'98*, pages 2548–2553, Leuven, Belgium, May 1998.

This article was processed using the TEX macro package with SIRS99 style