2D Robot Localization with Image-Based Panoramic Models Using Vertical Line Features

Dana Cobzas and Hong Zhang
Dept. of Computing Science, University of Alberta
Edmonton, Alberta, Canada
{dana, zhang}@cs.ualberta.ca

Keywords: Position estimation, Localization, Panoramic model, Image-based model

Abstract

This paper presents a method for estimating the position of a mobile robot in an indoor environment. The proposed technique uses a model of the environment formed by two panoramic cylindrical images taken at different locations, and a planar image taken at the current position. The current position and orientation of the robot are then computed without any additional information. We assume that the robot is moving on a plane (floor plane), which is very common for indoor environments. We will extend our previous result on using point features and exploit vertical line features to perform the localization. Our method is attractive because it does not require an explicit 3D model of the environment, and the location of the camera is not restricted to positions close to the cylindrical models. We will describe experimental results for both synthetic and real data to demonstrate the effectiveness of the proposed method.

1 Introduction

Position estimation (localization) is important especially in the context of robot navigation. For autonomous navigation, the robot should know how to locate itself with respect to its environment in order to follow a trajectory and reach a desired destination. The destination can be specified with respect to the world model in the Cartesian space, or in terms of an image taken from that position. There have been many solutions proposed for the problem of robot localization. Some use active sensors to form an abstract world model to help localization [9]. Others represent the robot environment in terms of a set of sampled images, organized in a meaningful way. The latter approach is similar to what people do when they remember a certain place, and is referred to as *image-based model* in computer vision and graphics research communities [5, 11].

An image-based model of a route has been used for localizing a robot. In [10] the model contains a sequence of front views along the route. The robot memorizes, at each position, an image obtained from a camera facing forward, and the directional relation to the next view. The problem with this solution, however, is that it requires a large amount of memory to store the images. The localization process matches stored images with the current one, and the current position is determined incrementally using information about the previous position. In [18] a panoramic representation of the route is obtained by scanning side views along the route. The robot uses the panoramic representation recorded in a trial move and the current one for locating itself along the trial route only. Dudek *et al.* [6, 15] proposed a localization method that uses a multi-layer back-propagation neural network trained over a dense set of images of the environment. The network can determine the position of the robot by interpolating between the stored images. Another interesting system, described in [13], estimates the position of a mobile robot based on the comparison of the real images taken by the robot and images taken by a virtual camera in a virtual environment, which is built by texturing planar walls of a 3D model.

When only the goal position is of interest, in the *homming applications*, one can employ the technique in [8], where spherical panoramic images are taken at target locations, and the robot is moved towards the goal by comparing the target image with images acquired by the robot. Their method does not compute the exact distance to the target location, but a velocity vector that will guide the robot to the target location. Similarly, Barsi *et al.* [1, 2] presents a method for guiding a robot to a desired position and orientation. The target position is specified by an image taken from that position. Although they do not use a 3D model of the environment, their algorithm requires significant overlap between the current image and the target image. Working with panoramic images, which sample a wide space, our algorithm does not have this limitation.

In this paper, we propose a system that uses panoramic cylindrical models, constructed at known locations, to represent the robot environment, assuming that the robot is moving on a plane. The localization algorithm uses two panoramic models and a planar image, taken at the current location of the robot, to determine the position and orientation of the robot. In a recent paper [4] we presented a localization algorithm that uses pairs of corresponding tie points between the panoramic models and the planar image whose position is to be localized. In this paper we will present a more robust algorithm that uses triplets of corresponding vertical lines.

One advantage of our proposed method is that we do not require an explicit 3D model of the environment. In addition, we need only two panoramic images for representing an environment with a minimum storage requirement, which will not grow rapidly even if we extend the algorithm to achieve a wider coverage of the environment and reduce the problem of occlusions. Finally, the localization algorithm does not require a known starting position and is not restricted to positions close to the cylindrical models. As will be seen later, we have excellent localization results for positions between the two image-based models.

The rest of the paper will be organized as follows. Section 2 defines the problem to be solved, Section 3 presents the formation of the cylindrical panoramic model, and Section 4 describes the localization algorithm. Experimental results for both synthetic and real data are shown in Section 5.

2 Problem Formulation

The general problem of localization with two panoramic image-based models can be defined as follows. Assume two panoramic image-based models constructed at known locations in a common environment, displayed as the two cylinders with the texture map of the environment in Figure 1. When a robot with an on-board camera travels in the same environment in the proximity of the two image-based models, we would like to determine, from a single planar image captured by the robot, the position and orientation of the robot with respect to either one of the two panoramic models.

In this paper we solve the planar case of the above general problem; i.e., motion of the robot is restricted to a plane, and the vertical axes of the two panoramic models are aligned and parallel to the image plane

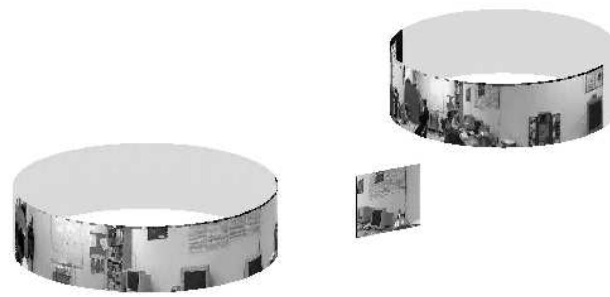


Figure 1: Two panoramic models and the planar image to be localized, each showing 180° of the view.

of the robot's camera. Our localization algorithm uses a set of corresponding vertical lines as features for the two panoramic models and the planar image. Note that the vertical lines remain vertical when projected on a cylinder. We do not consider the correspondence problem in this paper, and assume the corresponding features already exist. For the experiments, the process of detecting corresponding lines is performed manually.

3 Panoramic Model

In this section we will describe the process of building a panoramic image-based model by mosaicing. Image mosaicing means merging a collection of images into a larger image. In order to construct a mosaic from a set of images, they should be related by a 2D transformation. There are two cases when the projective model correctly describes the relationship between frames in a set of images: arbitrary scene with camera at a fixed location which is free to rotate about its center of projection, and planar scene with camera freely moving in the environment. Having these relations, different kinds of image mosaics can be constructed: spheres, planes, and cylinders.

There has been a lot of work done in mosaicing [3, 11, 14, 16]. We choose the method in [11] which can be easily implemented, and whose results can be easily stored. The cylindrical panoramic image mosaic model is the most suitable for our application of navigation and localization in 2D, and it can be acquired by panning the camera around its optical center.

3.1 Model construction

We used a CCD video camera mounted on a pan-tilt control unit on top of a tripod. The pan-tilt unit allows us to control the amount of rotation and also to keep the camera horizontal. For merging the planar images we used an algorithm similar to that of McMillan and Bishop [11]. For calibrating the camera we used the Tsai algorithm [17] with a 3D calibration pattern.

Consider the camera model,

$$C = \begin{bmatrix} \alpha_u & 0 & u_c \\ 0 & \alpha_v & v_c \\ 0 & 0 & 1 \end{bmatrix} \quad (1)$$

where α_u and α_v are the horizontal and vertical scale factors and u_c and v_c are the image coordinates of the intersection of the optical axis with the image plane [7].

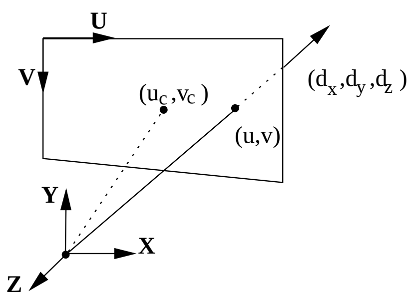


Figure 2: Image formation

Given each image pixel (u, v) taken by the camera, we first compute the direction of the ray of light (d_x, d_y, d_z) (see Figure 2) using

$$d_x = \frac{\mathrm{u} - u_c}{\alpha_u} \quad d_y = \frac{-\mathrm{v} + v_c}{\alpha_v} \quad d_z = -1 \quad (2)$$

Then, we project the planar images on a cylinder of a radius equal to the focal length f using the mapping,

$$\theta = \tan^{-1}\left(\frac{d_x}{d_z}\right) \quad v = f\frac{d_y}{\sqrt{d_x^2 + d_y^2}} \quad (3)$$

We use a correlation based technique to determine the amount of rotation between two consecutive projected images. In the cylindrical space a translation becomes a rotation, so we can easily build the cylindrical image by translating each image with respect to the previous one. To reduce discontinuities in intensity between images we weight the pixels in each image proportionally to their distance to the edge [16]. The result of this mosaicing technique will be shown in Section 5.

3.2 Calibrating two cylindrical images

For the localization algorithm we need the exact position and orientation (starting angle) of the panoramic images with respect to the world coordinate system. For simplicity and without loss of generality, we choose the center of first panoramic image as the origin of world coordinate system. We used the algorithm described in [11] to compute the relative position of the second cylinder and the rotational offsets which align the angular orientation of the two cylinders to the world coordinate system.

4 Localization

Having two calibrated panoramic image-based models we want to find the position and orientation of the robot regarding either of the two models. As mentioned before, we assume that the motion of the robot takes place in a plane (the floor). For indoor environments, this is normal because the floor is almost flat. The localization algorithm uses the position information for the panoramic models (described in Section 3) together with a set of corresponding vertical line features between the two models and the planar image to be localized. We used vertical line features because, when projected on the cylindrical surface, they remain vertical lines and are not transformed into curves like horizontal or arbitrary lines (Figure 3). For the moment we manually pick the corresponding line features, although algorithms exist for automatically generating corresponding lines [7]. The vertical line features are specified using only one coordinate - the *horizontal coordinate* (u) for the planar image and the *azimuth angle* (α) for the cylindrical models. We did not consider the length of the line feature because this might not be a robust constraint due to occlusions.

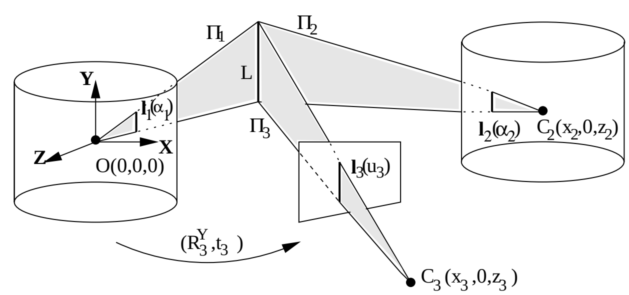


Figure 3: Matching lines in image-based models and planar image

Let us consider three corresponding lines $\mathbf{l}_1(\alpha_1)$, $\mathbf{l}_1(\alpha_2)$, $\mathbf{l}_1(u_3)$ as depicted in Figure 3. These lines together with the centers of projection O, C_2, C_3 determine three planes Π_1, Π_2, Π_3 that should intersect on the same line L. Without loss of generality, we choose the origin of the world coordinate system at the center of the first model. The normals to these planes, expressed in the normalized local coordinate systems, are given by

$$\begin{aligned} \mathbf{n}_1 &= [\cos\alpha_1, 0, \sin\alpha_1]^T \\ \mathbf{n}_2 &= [\cos\alpha_2, 0, \sin\alpha_2]^T \\ \mathbf{n}_3 &= [1, 0, u_3'] \end{aligned} \tag{4}$$

where $u_3' = \frac{u_3 - u_c}{\alpha_u}$ is the normalized coordinate of u_3 considering the camera model given by (1). The projective coordinates of the normals in the coordinate system of the first cylindrical model are [7]

$$\mathbf{n}_1^p = \begin{bmatrix} \mathbf{n}_1 \\ 0 \end{bmatrix} \quad \mathbf{n}_2^p = \begin{bmatrix} \mathbf{n}_2 \\ \mathbf{t}_2^T \mathbf{n}_2 \end{bmatrix} \quad \mathbf{n}_3^p = \begin{bmatrix} \mathbf{R}_3^Y \mathbf{n}_3 \\ \mathbf{t}_3^T \mathbf{R}_3^Y \mathbf{n}_3 \end{bmatrix}$$

where $\mathbf{t}_2 = [x_2, 0, z_2]$ is the displacement between the position of the second model and origin of the coordinate system, and $\mathbf{t}_3 = [x_3, 0, z_3]$, $\mathbf{R}_3^Y(\theta)$ are the displacement of the current position and orientation of the robot with respect to the world coordinate system.

The planes Π_1, Π_2, Π_3 should intersect on the same line. This is equivalent to the algebraic condition that the 4×3 matrix, formed with the normals $\mathbf{n}_1^p, \mathbf{n}_2^p, \mathbf{n}_3^p$ as columns, is of rank 2 [7]. This implies that all 3×3 determinants extracted from the matrix are equal to zero. In our case the second row from the matrix is zero so we have only one determinant left,

$$D(\alpha_1, \alpha_2, u_3') = \begin{bmatrix} \cos\alpha_1 & \cos\alpha_2 & \cos\theta - u_3'\sin\theta \\ \sin\alpha_1 & \sin\alpha_2 & \sin\theta + u_3'\cos\theta \\ 0 & A & B \end{bmatrix}$$

where

$$\begin{aligned} A &= -x_2\cos\alpha_2 - z_2\sin\alpha_2 \\ B &= -x_3(\cos\theta - u_3'\sin\theta) - z_3(\sin\theta + u_3'\cos\theta). \end{aligned}$$

To find θ, x_3, z_3, if we have $n \geq 3$ triplets of corresponding lines, we can solve the problem by minimizing the following function

$$\min_{x_3, z_3, \theta} \sum_{i=1}^{n} \left(D(\alpha_1^{(i)}, \alpha_2^{(i)}, u_3^{'(i)}) \right)^2$$

This minimization problem can be solved by the Levenberg-Marquardt non-linear minimization method [12]. In the next section we will present the results of this algorithm for both synthetic and real data.

5 Experimental Results

5.1 Synthetic data

To verify and evaluate the robustness of the algorithm, we test it in the presence of noise. The noise can be caused by errors in selecting corresponding lines, by image quantization error, and by errors in model construction or calibration. We select 25 space vertical lines, and project them on two cylinders. Both cylinders have a radius equal to the focal length. The first cylinder shares the origin with the world coordinate system. The second one is three meters away from the first along the X axis. We also project the space vertical lines on a planar image, using the calibrated camera model. To simulate erroneous line selection, we added noise uniformed distributed in the interval $(-w, w)$ (in pixels) to the coordinates of the line features in the cylindrical models and planar image. Then we apply the algorithm to recover the position and orientation of the third image captured by a robot between the two cylinders. For noise level $w = 3$ pixels the error was about 6.6 mm for position estimation, and 0.2° for orientation. We repeat the experiment for different numbers of lines, and the results show that the localization error decreases if the number of lines increases and, for more than 10 triplets of line features, it stabilizes (see Figure 4).

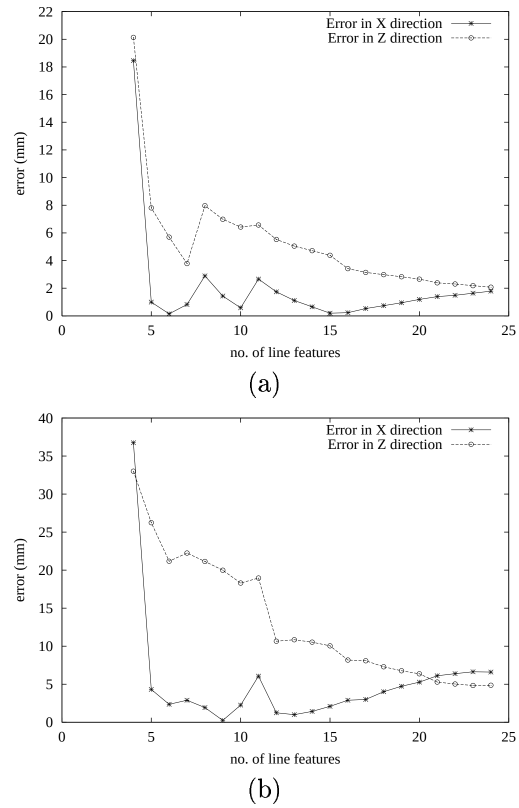


Figure 4: Variation of localization error error in X and Z directions with the number of triplets of line features; (a) without noise, (b) noise level $w = 3$.

Figure 5: Two cylindrical panoramic images (each formed using 25 planar images sampling 180°)

5.2 Real data

To evaluate the algorithm with real data we first built two panoramic cylindrical images. We used a tripod and a pan-tilt unit on top of the tripod. For simplicity we sampled only half of the environment, or 180° of each cylinder. The distance between the cylindrical models was about three meters. Then, we compute the exact location and starting angles for the cylindrical models using the method described in Section 3.2. The calibrated distance and orientation of the cylinders are $C_2 = (3000.04, 0, 3.05E-5)$(mm), and the rotational offsets $\beta_1 = -21.88°$ and $\beta_2 = -69.70°$. The results are shown in Figure 5, where the cylindrical images are unrolled.

Figure 6: Planar image from in between the two panoramic images with the line features superimposed

Then we took images at nine locations between the two cylindrical models and apply our localization algorithm for recovering the camera position and orientation. We choose the origin of the coordinate system at the origin of the first model, and the X axis along the line that connects the centers of the cylinders. Figure 6 shows one typical image.

Figure 7(a) shows the actual and the estimated position, and Figure 7(b) shows the error in estimated angle for each position. We measure the actual location of the robot using a meter stick for the position, and a pan tilt unit for the rotation angle. The position error is measured in centimeters, and the rotation error in degrees. The average position error was around 5.5 cm in X and Z directions, and 3.5° for the rotation angle.

In addition to positions between the cylinders, we also tested our algorithm with images taken from other positions. The accuracy of the proposed method depends mostly on the viewing direction, and much less on the position of the robot. When the planes Π_i generated by the line features from the current image and from one of the cylindrical models are very close to each other, the localization error increases. This happens, for example, when the robot is close to the location of one of the models and the line features belong to objects that are far away from the camera. In comparison to our previous point-based algorithm [4], however, this new line-based algorithm performs much better.

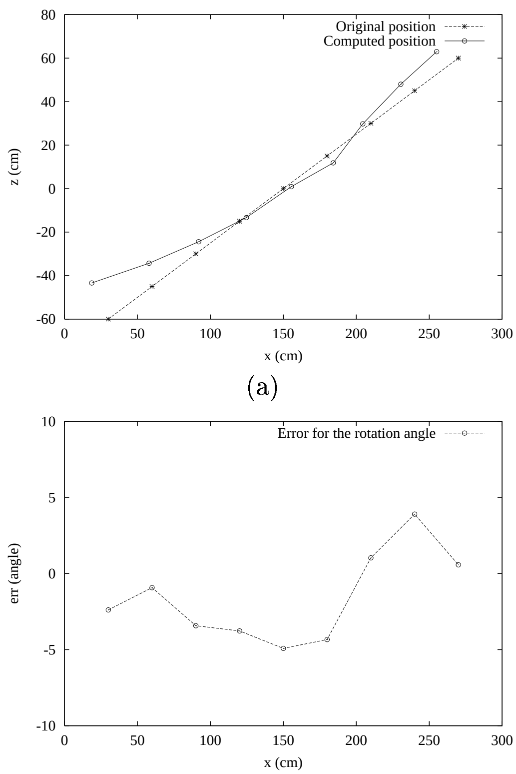
6 Conclusions and Future Work

We have presented a vision-based localization algorithm using a panoramic image-based representation of the environment. The image-based model consists of two panoramic cylindrical images taken at different known locations. The proposed algorithm estimates the position and orientation of the camera given a planar image taken at an arbitrary position in the same environment.

As a future extension we want to automatically track corresponding line features between images, to use more than two panoramic images for improving the localization accuracy, and to automatically update the model for changes in the scene.

References

[1] R. Barsi, E. Rivlin, and I. Shimshoni. Visual-homing: surfing the epipoles. In *Proc. of 1998 6th Int. Conf. on Compute Vision*, pages 863–869, 1998.



(a)

(b)

Figure 7: (a) Recovered positions from the 9 images. The first panoramic model is situated at (x,z)=(0,0) and the second one at (x,z)=(300,0); (b) Error for the recovered orientation. The x coordinates show the position of the planar image regarding the first panoramic model.

[2] R. Barsi, E. Rivlin, and I. Shimshoni. Image-based robot navigation under the perspective model. In *Proc. of 1999 IEEE Int. Conf. on Robotics and Automation*, pages 2578–2583, 1999.

[3] S. Chen. Quicktime VR - an image-based approach to virtual environment navigation. In *Computer Graphics (SIGGRAPH'95)*, pages 29–38, 1995.

[4] D. Cobzas and H. Zhang. 2D robot localization with image-based panoramic models. *Submitted to IEEE Int. Conf. on Computer Vision and Pattern Recognition (CVPR'00)*, 2000.

[5] P. E. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs. In *Computer Graphics (SIGGRAPH'96)*, 1996.

[6] G. Dudek and C. Zhang. Vision-based robot localization without explicit object models. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 76–82, 1996.

[7] O. D. Faugeras. *Three Dimensional Computer Vision: A Geometric Viewpoint.* MIT Press, Boston, 1993.

[8] J. Hong, X. Tan, B. Pinette, R. Weiss, and E. Rseman. Image-based homing. *IEEE Control Systems*, pages 38–44, 1992.

[9] L. Mathies and A. Elfes. Integration of sensor and stereo range data using a grid based representation. In *Proc of IEEE Int. Conf. on Robotics and Automation*, pages 727–733, 1988.

[10] Y. Matsumoto, M. Inaba, and H. Inoue. Visual navigation using view-sequenced route representation. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 83–88, 1996.

[11] L. McMillan and G. Bishop. Plenoptic modeling: Am image-based rendering system. In *Computer Graphics (SIGGRAPH'95)*, pages 39–46, 1995.

[12] W. H. Press, B.P. Flannery, S. A. Teukolsky, and W. T. Vetterling. *Numerical Recipies in C: The Art of Scientific Computing.* Cambridge University Press, Cambridge, England, 1992.

[13] M. Schmitt, M. Rous, A.Matsikis, and K.-F. Kraiss. Vision-based self-localization of a mobile robot using a virtual enviroment. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 2911–2916, 1989.

[14] H.-Y. Shum and R. Szeliski. *Panoramic image mosaics.* Technical Report MSR-TR-97-23, Microsoft Research, 1997.

[15] R. Sim and G. Dudek. Learning visual landmarks for pose estimation. In *Proc. of the IEEE Int. Conf. on Robotics and Automation*, pages 1972–1978, 1999.

[16] R. Szeliski and H.-Y. Shum. Creating full view panoramic image mosaics and environmental maps. In *Computer Graphics (SIGGRAPH'97)*, pages 251–258, 1997.

[17] R. Tsai. A versatile camera calibration technique for high accuracy 3d machine vision metrology using off-the-self tv cameras and lenses. *IEEE Journal of Robotics and Automation*, 3(4):323–344, 1987.

[18] J. Y. Zheng and S. Tsuji. Panoramic representation for route recognition by a mobile robot. *International Journal of Computer Vision*, 9(1):55–76, 1992.