Using Image-Based Panoramic Models for 2D Robot Localization

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. 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 very 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 reach a desired destination. The destination can be specified with respect to a 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 second approach is similar to what people do when they remember a certain place. It is referred to as "image-based model" in compute vision and graphics research communities [5,11].

An image-based model of a route has been used for localizing the robot along that route. 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, 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 [20] 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. The 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.

Another related work can be found in [8], where spherical panoramic images are taken at target locations, and the robot is moved to the desired position by comparing the target image with images acquired by the robot. Their method does not compute the exact direction or distance to the target location, but a velocity vector that will guide the robot to the target location. Barsi *et al.* [1,2] present a method for guiding a robot to a desired position and orientation. The target position is specified by an image taken from that position. They do not use a 3D model of the environment, but their algorithm requires significant overlap between the scene in the current image and target images. 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. 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, assuming that the robot is moving on a plane.

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 shown later, we have excellent localization results for positions between the two cylinders.

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. We assume two panoramic image-based models constructed at two known locations of a common environment, displayed in Figure 1 as the two cylinders with the texture map of the environment. 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 of the robot's camera. Our localization algorithm uses a set of corresponding points for the two panoramic models and the planar image. We do not consider the correspondence problem in this paper, and assume they already exist. For the experiments, the process of detecting tie points is performed manually.

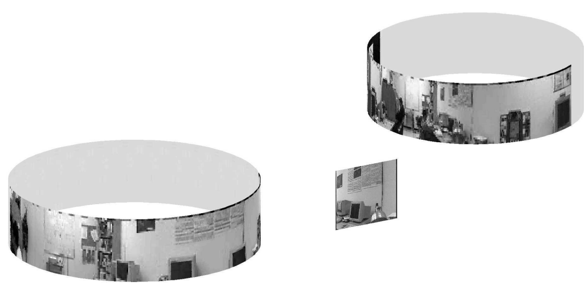


Figure 1. Two panoramic models and the planar image to be localized each sampling 180° of the view.

3 PANORAMIC MODEL

In this section we will describe the method of building the 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 an image sequence: arbitrary scene with camera at a fixed location but 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 [3,4,11,14,16,17]. 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 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 [18] 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]. 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) for the image point (u,v) (see Figure 2) using

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

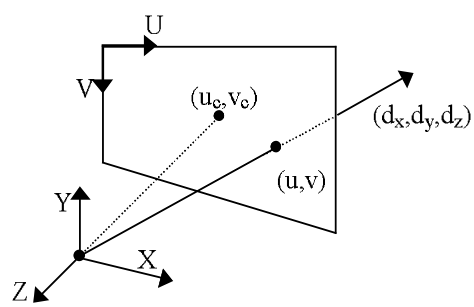


Figure 2. Image formation

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

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

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 [17]. 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 the first panoramic image as the origin of world coordinate system. We used the algorithm described in [11] to compute the relative position and orientation of the two panoramic images.

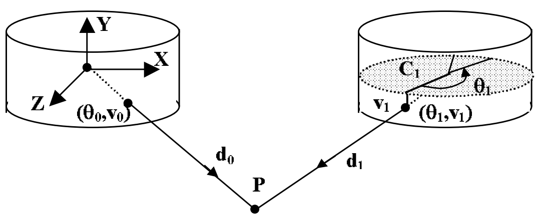


Figure 3. Two corresponding rays in the panoramic images

Assume that a set of corresponding points in the two images is known. Each pair of corresponding points determines two rays in space (see Figure 3), $\mathbf{d}_0$, $\mathbf{d}_1$, given by,

$$\mathbf{d}_0(\theta_0, v_0) = t \begin{bmatrix} f\sin(\theta_0 + \alpha_0) \\ v_0 \\ -f\cos(\theta_0 + \alpha_0) \end{bmatrix} = tD_0(\theta_0, v_0)$$
$$\mathbf{d}_1(\theta_1, v_1) = C_1 + s \begin{bmatrix} f\sin(\theta_1 + \alpha_1) \\ v_1 \\ -f\cos(\theta_1 + \alpha_1) \end{bmatrix} = C_1 + sD_1(\theta_1, v_1) \qquad (4)$$

where (θ_0, v_0), (θ_1, v_1) are the cylindrical coordinates for the corresponding points, f is the focal length of the camera (and also the radius of the cylindrical image), $C_1(x,0,z)$ is the unknown position of the second cylinder and α_0 and α_1 are the rotational offsets which align the angular orientation of the two cylinders to the world coordinate system, respectively. In total there are four unknown variables: α_0, α_1, x and z.

The pair of rays intersect at the space point that generates the two corresponding points on the cylinders. Because of the errors in model construction and feature extraction these rays do not intersect exactly. The distance between the rays is given by [11]

$$\mathbf{d}(\theta_0, v_0, \theta_1, v_1) = \mathbf{d}_0(\theta_0, v_0) - \mathbf{d}_1(\theta_1, v_1) \qquad (5)$$

where,

$$t = \frac{\begin{vmatrix} C_1 \\ D_1(\theta_1, v_1) \\ D_0(\theta_0, v_0) \times D_1(\theta_1, v_1) \end{vmatrix}}{\left\| D_0(\theta_0, v_0) \times D_1(\theta_1, v_1) \right\|^2}$$
$$s = \frac{\begin{vmatrix} C_1 \\ D_0(\theta_0, v_0) \\ D_0(\theta_0, v_0) \times D_1(\theta_1, v_1) \end{vmatrix}}{\left\| D_0(\theta_0, v_0) \times D_1(\theta_1, v_1) \right\|^2} \qquad (6)$$

Having n pairs of corresponding points on the cylindrical models we can determine the unknown parameters by minimizing the following error function, using the Powell's method [12],

$$\mathbf{Error}(\alpha_0, \alpha_1, x, z) = \sum_{i=1}^{n} \frac{\left\| \mathbf{d}(\theta_0^{(i)}, v_0^{(i)}, \theta_1^{(i)}, v_1^{(i)}) \right\|}{\left\| tD_0(\theta_0^{(i)}, v_0^{(i)}) \right\|} + \left\| x, 0, z \right\| - d \qquad (7)$$

where d is the given distance between the two cylinders. If the starting values for the unknowns are close to the real ones, the algorithm converges to the desired solution in a few iterations.

4 LOCALIZATION

As mentioned before, we assume that the motion of the camera takes place in a plane (the floor). For indoor environments this is normal because the floor is almost flat. The localization algorithm takes as input the position information of the panoramic images (described in Section 3), together with triplets of corresponding tie points between the two panoramic images and the planar image to be localized. For the moment we manually pick the corresponding points, although algorithms exist for automatically generating corresponding points [19].

The localization algorithm consists of two steps: in the first step we determine the space points $\mathbf{P}(x,y,z)$ corresponding to the pairs of cylindrical tie points (θ_0, v_0) and (θ_1, v_1); in the second step, from the space points and corresponding points in the planar image (u_2, v_2) we determine the position $C_2(x_2, 0, z_2)$ and orientation α of the robot (see Figure 4).

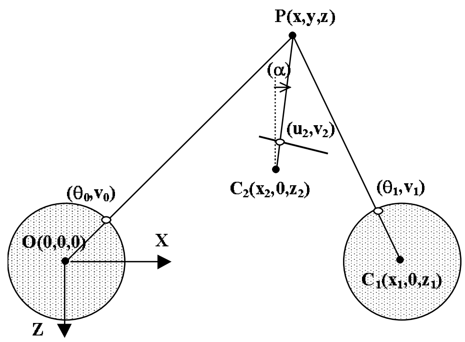


Figure 4. Corresponding rays in panoramic images and planar image

4.1 Finding the space points

Having a pair of corresponding cylinder's points (θ_0,v_0), (θ_1,v_1), we want to derive the space point that generated these tie points. We used the same algorithm described in 3.2 for calibrating the cylinders. Specifically, we compute the middle point between the light rays generated by the pair of cylindrical points using

$$\mathbf{P}(x,y,z)=\frac{\mathbf{d}_0(\theta_0,v_0)+\mathbf{d}_1(\theta_1,v_1)}{2} \qquad (8)$$

where $\mathbf{d}_0(\theta_0, v_0)$ and $\mathbf{d}_1(\theta_1, v_1)$ are given by (4), and s and t are computed using (7).

4.2 Robot localization

Having the space points $\mathbf{P}(x^{(i)}, y^{(i)}, z^{(i)})$ and the corresponding points in the planar image ($u^{(i)}$, $v^{(i)}$), we next derive the robot position and orientation with respect to the world reference frame. Again we suppose that the camera model (1) is known. The relationship between the space points and the image points is given by,

$$\begin{pmatrix} x^{(i)} \\ y^{(i)} \\ z^{(i)} \end{pmatrix} = C_2 + lR_Y(-\alpha)\begin{pmatrix} d_z^{(i)} \\ d_y^{(i)} \\ d_z^{(i)} \end{pmatrix} \qquad (9)$$

where

$$C_2=(x_2,0,z_2)^T,$$

$$R_Y(-\alpha)=\begin{pmatrix} \cos\alpha & 0 & -\sin\alpha \\ 0 & 1 & 0 \\ \sin\alpha & 0 & \cos\alpha \end{pmatrix}$$

is the rotation matrix about Y axis, and $d_x^{(i)}, d_y^{(i)}, d_z^{(i)}$ are given by (2). By expanding equation (9) and reducing l ($l = y^{(i)}/d_y^{(i)}$), we get the following equations,

$$\begin{cases} d_y^{(i)}x_2 + y^{(i)}\sin\alpha + y^{(i)}d_x^{(i)}\cos\alpha = x^{(i)} \\ d_y^{(i)}z_2 - y^{(i)}d_x^{(i)}\sin\alpha + y^{(i)}\cos\alpha = z^{(i)} \end{cases} \qquad (10)$$

with the unknowns x_2, z_2, cosα, sinα. With more than two pairs of corresponding points, we can then compute the unknown parameters using the least square approach. By considering sinα and cosα as separate unknowns, once (10) is solved, the orientation of the camera is determined by

$$\alpha = \tan^{-1}\frac{\sin\alpha}{\cos\alpha}$$

In the next section we will describe the results of this algorithm using both synthetic and real data.

5 EXPERIMENTAL RESULTS

5.1 Synthetic data

To evaluate the robustness of the algorithm, we test it in the presence of noise. The noise can be caused by errors in selecting corresponding points, by finite image resolution, and by errors in the model construction or calibration. We produce 100 space points, 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. We also project the space points on a planar image, using the camera calibration data. To simulate erroneous point selection, we added uniform noise in the interval (-w, w) to the coordinates of the points 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 the noise level w=3, the error was about 2 cm for position estimation, and 0.5° for orientation. We repeat the experiment for different number of points, and the results show that the localization error decreases if the number of points increases, and it becomes almost stable for more then 15 triplets of corresponding points.

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 results are shown in Figure 5, where the cylindrical images are unrolled.

Figure 5. Two cylindrical panoramic images (each formed using 25 planar images sampling 180 degrees)

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

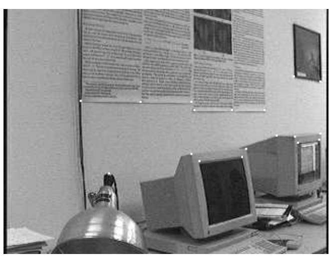
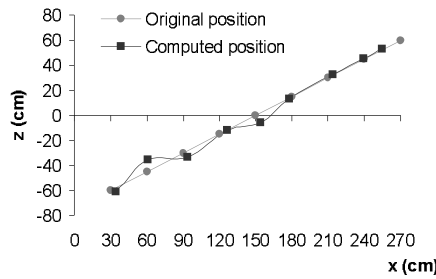
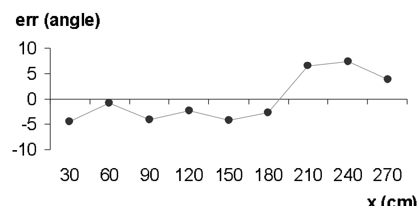


Figure 6. Planar image from in between the two panoramic images with the feature points superimposed



(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.

Figure 7(a) shows the original and the estimated position, and Figure 7(b) shows the error in estimated angle for each position. The position error is measured in centimeters, and the rotation error in degrees. We measured the actual location of the robot using a meter stick. The average position error was around 4 cm in X and Z directions, and 3.5° for the rotation angle

In addition to positions between the cylinders, we have also tested our algorithm with images taken from other positions. The accuracy of the proposed method depends mostly on the viewing direction, but not on the position of the robot. When the rays generated by the feature points 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 actual space points that generated the feature points are far away from the camera. We obtained very good results when the depth of the feature points is comparable to the relative distance between the models and the robot.

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 feature points between frames, to use more than two panoramic images for improving the localization accuracy, and to automatically update the model for changes in the scene. We are also developing an algorithm that uses line features instead of points.

REFERENCES

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

[2] R. Barsi, E. Rivlin, and I. Shimshoni. Visual-homing: surfing the epipoles. In *Proc. of the 6th Int. Conf. on Computer Vision (ICCV-98)*: 863-869, 1998.

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

[4] S. Chen and L. Williams. View interpolation for image synthesis. In *Computer Graphics (SIGGRAPH'93)*: 279-288, 1993.

[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*: 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. In *IEEE Control Systems*: 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*: 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*: 83-88, 1996.

[11] L. McMillan and G. Bishop. Plenoptic modeling: Am image-based rendering system. In *Computer Graphics (SIGGRAPH'95)*: 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, second edition, 1992.

[13] M. Schmitt, M. Rous, A.Matsikis, 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*: 2911-2916, 1999.

[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*: 1972-1978, 1999.

[16] R. Szeliski. Video mosaics for virtual environments. In *IEEE Computer Graphics and Applications*: 22-30, March 1996.

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

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

[19] Z. Zhang, R. Deriche, O. Faugeras and Q.-T. Luong. A robust technique for matching two uncalibrated images through the recovery of the unknown epipolar geometry. Rapport de Recherche 2273, INRIA Sophia-Antipolis, France, 1994

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