Robert Bączyk, MSc; Piotr Skrzypczyński, PhD
Technical University of Poznań, Department of Control, Robotics, and Computer Science

MOBILE ROBOT LOCALIZATION BY MEANS OF AN OVERHEAD CAMERA

Abstract : In this article a positioning system for mobile robots based on a ceiling–mounted camera is presented. Fish-eye lenses are used to extend the field of view of a single camera. Though fish-eye lenses provide a large field of view they introduce large geometric distortion in the image. This distortion is compensated by a correction procedure implemented in the localization system. The mobile robot has been provided with active LED markers to ensure correct detection under varying illumination conditions. Results of experiments in a laboratory environment and the evaluation of the positioning accuracy are presented.

1. INTRODUCTION

An important requirement for any mobile robot is to figure out where it is within its environment. Typical, wheeled mobile robots use odometry, which is able to provide the robot with a rough estimate of its position and orientation (i.e. the robots *pose*) at any time. The odometry is self-contained — it uses only internal sensing, what is its main advantage. The drawback of odometry is that even small position errors accumulate over the covered distance without a bound. Because of this a robot needs to re-calibrate its pose from time to time by using information from an independent source.

There are many localization techniques known from the literature. A good survey of the state-of-the-art in mobile robot positioning can be found in [2]. Among these methods the following seem to be most popular :

- systems using active landmarks (beacons),
- systems recoursing to artificial landmarks placed at known locations in the environment,
- natural landmark recognition (usually with computer vision methods),
- map matching methods,
- dense sensory data matching.

Commercial AGVs use active beacons or artificial landmarks which are read by custom laser or ultrasonic sensors [2]. These systems are reliable and thus commonly used in real-life industrial applications. But robots completely relying on artificial landmarks are tied to paths or places with engineered environment, and because of that they can not be used as flexibly as autonomous robots.

Autonomous mobile robots use external sensing to determine position and orientation with regard to environment features [5]. However in practice, methods based on range sensors (sonars, laser rangefinders) and map matching are constrained to highly structured environments like office buildings or department stores. Recognition of natural landmarks by means of computer vision or range sensors [7] imposes similar constrains. Some robots keep track on position and orientation by matching dense sensor scans with a map of the environment, without extracting features. Such techniques proved to be usable in less-structured environments, but they also cannot recover from failures caused by degraded sensory data or bad match due to environment symmetry.

The common drawbacks of positioning methods used on autonomous robots could be eliminated by recoursing from time to time to external navigation aids, in a similar manner as the odometry is re-calibrated from on-board sensors. This approach can combine advantages of positioning techniques used on AGVs (reliability) and on autonomous robots (flexibility). An *autonomous AGV* can rely mostly on its on-board sensors for continuous self-localization, and from time to time read new "initial" estimate of its pose from another system.

For mobile robots performing transportation or service tasks in a restricted area (e.g. hall of an industrial plant, warehouse, etc.) an interesting approach can be the use of global vision system to determine the position and localization of vehicles. It is especially interesting in the context of multi-robot systems because a moderate number of cameras fixed in the environment can localize many vehicles, significantly decreasing the costs of the whole system in comparison to a fleet of fully autonomous mobile robots equipped with expensive on-board sensors for localization.

For a localization system based on global vision two specific problems have to be solved :

- How to ensure large field of view of a single camera to avoid the use of large number of cameras and thus the unnecessary growth of the costs of the whole system ?
- How to ensure correct and reliable detection of mobile robots (of possibly different shapes) under varying illumination conditions ?

The goal of the article is to present solutions for these problems and to give an evidence for the viability of this kind of positioning system by presenting results of experiments with a real mobile robot. The proposed system uses a single camera mounted to the ceiling of a room. Fish-eye lenses are used with the camera to ensure as large field of view as possible. The mobile robot used in experiments is equipped with active LED markers to avoid problems with different light conditions.

2. THE LOCALIZATION SYSTEM

The localization system uses single B/W CCD camera mounted to the ceiling of the laboratory room at height h_{cam}=240 cm. The size of the room is about 6 × 6 meters. Because of the use of only one camera mounted not so high above the floor the field of view, and thus the working area of the localization system is small, about 2 × 2 meters using standard lenses with focal length f=6 mm. This area is to small to perform evaluation of localization reliability and accuracy for the mobile robot of *Labmate* type. Due to this the camera has been equipped with wide-angle fish-eye lenses. The lenses used have focal length f=2.8 mm and with the CCD matrix dimensions of 4.8 $\times$ 3.6 mm gives the viewing angle of about 145^o, and the field of view covering almost the whole laboratory floor.

Unfortunately, the fish-eye lenses introduce significant geometric distortion in the image. The distortion results in a shifting of pixels from their original positions and the effect similar to a 3D plane instead of a flat surface. Because of this distortion, known as "barrel" [6], the calibration of the fish-eye lenses camera and a correction of the image have to be performed before the image can be used for positioning. This correction method is described in the next section.

The camera is connected to a PC computer (AMD K6-2 400MHz) equipped with a low-cost PCI-bus frame-grabber DT3153 from *Data Translation Ltd.*. The frame grabber does not have its own memory (frame buffer) and uses a part of PC memory, managed by a software driver. Because of this it runs only in Win32 environment.

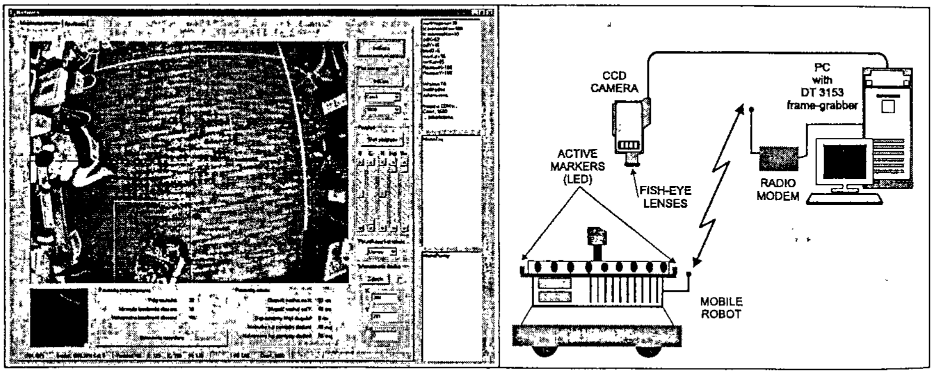

Figure 1: The main window of the image processing program (left) and the experimental set-up (right).

Because of the complex shape of the mobile robot used in experiments and the varying illumination conditions in the laboratory room the extraction of the robot from grey-level images by means of standard image processing methods was very hard and not reliable. To remedy this problem the robot has been equipped with a simple system of active markers. Four red LEDs have been attached symmetrically at the corners of the robot's body. The diodes are controlled from the on-board PC of the robot by using parallel port and a simple interface circuit that fits in the plug of a standard Centronics cable.

The detection of the robot is performed on the difference image, which is computed from a pair of images taken when the diodes are on, and then off. The difference image is binary, so the search for robot is much simpler. There are four clusters of white pixels pointing the positions of the diodes, and everything other should be black. This is of course an idealized situation, because in reality there is a lot of other clusters, caused by dynamic objects (e.g. people in the lab) or differences in illumination between the two frames used to produce the binary image. The spurious clusters can be eliminated by using the knowledge about the dimensions of the robot and the symmetry of the pattern of diodes.

To use the active markers the vision system has to communicate with the robot. So the software of the system is divided into two parts. Part one is the main program which provides the image processing and computes the pose of the robot — it runs under Windows on the stationary PC with frame-grabber. Second part is the program running under Linux on the on-board computer of the robot. This program controls the robot and the active markers. Both computers communicate through the wireless modem *Arlan 130* (Fig. 1).

The whole algorithm for image processing and computing the pose of the robot is as follows :

1. The difference between two acquired images of the scene is computed, with active markers on and off respectively.
2. In the same loop the tresholding of the difference image is performed, producing a binary image. The threshold can be set in a config file.
3. The program searches for segments (clusters of white pixels) in the binary image. It is done by using the *4-neighborhood* connectivity criteria [3] (left part of Fig. 2).
4. Too large and too small clusters (which can not be the diodes) are eliminated from the further processing.
5. For each cluster its centroid (x_c, y_c) is computed by means of the moment calculation.
6. For the points being centers of the found clusters the correction procedure is performed, in order to compensate for the distortion introduced by the fish-eye lenses. It should be noted, that performing the correction only for the few points, remained in the image after the previous steps of the algorithm, makes the whole image processing significantly shorter.
7. For the corrected points the geometric constraints imposed by the dimensions of the robot and the layout of the active markers are checked out. The algorithm must find at least three points which satisfy the constraints, and built-up a triangle with given size and angles (right part of Fig. 2, the eliminated clusters are grey, the markers are white).

The first two operations in this algorithm take the vast majority of the processing time for each localization cycle. It is purposeful to constrain all the operations to a region of interest which can be much smaller than the whole acquired frame. The localization by using the camera occurs only on occasion, and because of that it is not possible to predict the next position of the robot on the base of its previous state. However, a rough estimation of the robot's pose can be obtained from the vehicle odometry. This estimate is used to set the *region of interest* window on the image.

The position and orientation of the robot from the odometry is computed by using the method from [1]. This method takes into account the dominant role of the orientation error in the odometry model, but can not account for all non-systematic errors, e.g. wheel

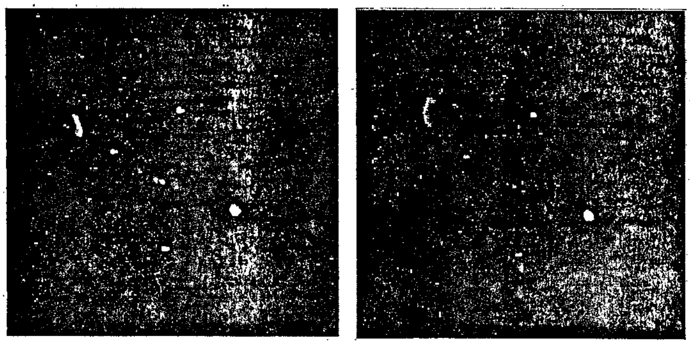

Figure 2: The clusters extracted from the binary image (left) and the found markers (right).

slippage due to a slippery floor or so called "shopping cart effect" [2] caused by the supporting wheels of the *Labmate* platform. So the odometric estimate should be used with caution, because it could be quite uncertain, especially with regard to the orientation. The window used is 200 × 200 pixels on the frame of 768 × 576 pixels. In the case when the program can not find the robot (its diodes) within this window it stops the processing and enlarges the window to the whole image. The odometric estimate is also used to resolve the ambiguity as to the orientation of the robot. The pattern of diodes and the robot itself are symmetrical, so it is not possible to compute an unambiguous orientation of the vehicle from the camera images. From the two possible orientations the system chooses this one which is closer to the orientation from the odometry.

The communication protocol between the robot and the stationary computer with the vision system is as follows :

1. The main program initializes the vision system.
2. The on-board program initializes the robot.
3. The robot executes a part of its pre-planned path (translation and/or rotation), then it estimates its own position and orientation (with the covariance matrix, being the measure of the spatial uncertainty) from the odometry.
4. The robot contacts the vision system (through the modem) and sends a request for re-localization, together with the pose estimate from the odometry.
5. The vision system sends an acknowledgement message.
6. The robot turns on the diodes.
7. The vision system takes the picture of the scene and sends an acknowledgement message.
8. The robot turns off the diodes.

9. The vision system takes the second picture and starts the image processing and localization algorithm.

10. If the vision system successfully localizes the robot it sends the estimate of the position and orientation (with uncertainty measure) to the robot. The robot uses these values to re-calibrate its odometry.

11. If the vision system can not localize the robot (e.g. due to an occlusion) it sends an appropriate message to the robot. Then the robot tries to initialize the localization in an different position.

At the initialization of the system there is no valid information from the odometry of the robot (both the position and orientation are set to zero), so the system has to perform an additional step to localize the robot. It takes the third image with only two active diodes on the robot, indicating the front of the vehicle. From these three frames two difference images can be computed, enabling the unambiguous localization of the robot without any odometric data.

3. USING THE FISH-EYE OPTICS CAMERA

The fish-eye lenses used in the proposed positioning system provide large field of view, but they introduce in the image a distortion which has to be corrected. In the image taken by a fish-eye camera each pixel in the image is shifted to a different position, what is mainly a result of the curvature of the lense which provide its large field of view (Fig. 3).

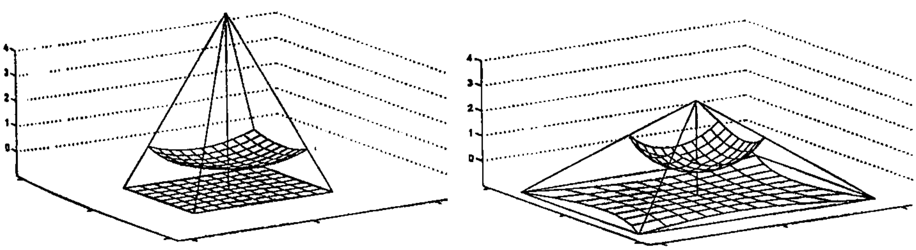

Figure 3: The model of image geometry for normal (left) and fish-eye (right) lenses.

In [6] Shah and Aggarwal presented a procedure for the calibration of such fish-eye lens camera, and a method based on polynomial transformation for correcting the distortion in the images. Their method provides good results, but the calibration is quite complicated and needs a special, precisely printed pattern which should be about the size of the view seen by the camera. It was very hard to provide such pattern for the experimental set-up described here (the pattern should cover almost the whole floor of the lab).

Because of this a new, original correction procedure has been proposed, which is based on a simple camera model where the only parameter is the viewing angle, which is determined experimentally. It is assumed that the ideal image of the scene (from a pin-hole camera model) is deformed by the wide-angle optics. The model of this deformation can be obtained by "warping" the surface of the CCD matrix on a spherical surface (see Fig. 3). In this "spherical" correction method the only corrected parameter of a pixel is the

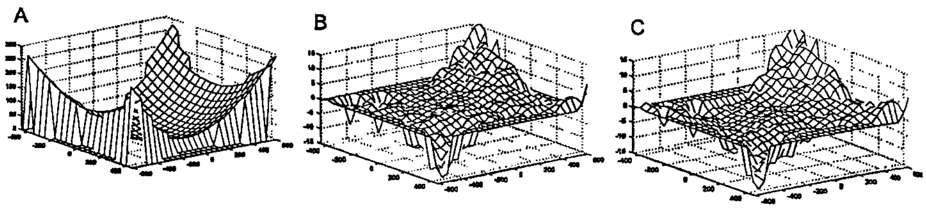

Figure 4: Radial distance errors in fish-eye images : not corrected (A), polynomial correction (B) and spherical correction (C).

distance from the center of the image.

The input of the correction procedure are the dimensions of the corrected image sh' and sv', and the angle of view α in radians ($\alpha \approx 2.16$ rad). At first, the amount of radians per pixel is computed :

$$rpp = \frac{\alpha}{\sqrt{sh^2 + sv^2}}, \tag{1}$$

where sh and sv are the dimensions of the original image. Then the distance h from the center of the sphere to the image plane is computed :

$$h = \frac{\sqrt{(\frac{sh'}{2})^2 + (\frac{sv'}{2})^2}}{\tan\frac{\alpha}{2}}. \tag{2}$$

Next, the image coordinates are transformed to the polar coordinates (r, ϕ).. The resulting r values are converted to the angle β :

$$\beta = rpp \cdot r. \tag{3}$$

The new radial coordinate r' of the given pixel is computed as :

$$r' = h \cdot \tan\beta. \tag{4}$$

The last step is the conversion of the polar coordinates of the corrected image (r', ϕ) to the Cartesian coordinates used in the further processing. Knowing the angle of view and the field of view one can easily find the proportions between the image coordinates and the real coordinates.

The spatial distribution of the distance errors in the images for both correction methods has been evaluated, by comparing the corrected image of a calibration pattern (a sheet of paper in A0 format with printed black dots) with the ground truth, i.e. the pattern itself. The radial distance errors are shown in Fig. 4. The relative error of the localization of a pixel computed in this way is 0.7% for the polynomial method from [6], and 1.1% for the spherical method presented here.

4. EXPERIMENTAL RESULTS

To validate the reported approach, several localization experiments have been carried out. The localization error defined as the distance between the center of the robot (thus the center of the LED pattern) and the reference point on the pre-planed path has been evaluated. The orientation error is the difference between the reference orientation given in the path and the actual orientation of the robot. The absolute position of the robot has been measured by hand, with help of an ordinary meter and a grid on the floor, using the walls as a reference (Fig 5).

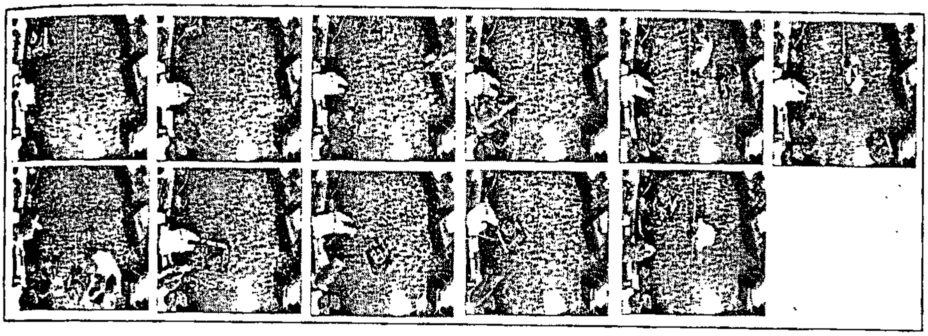

Figure 5: Experimental evaluation of the positioning method.

The firs experiment was the execution of a 2×2 meters square path in cw direction, under the odometric control and then with the re-calibration from the overhead camera. The task of the robot was to execute this path four times, and stop at its "home" position (this is the small pattern on the floor visible in the left upper part of Fig. 1). Figure 6A1 shows the error in orientation, while Fig. 6A2 the error in position of the robot. It can be seen from these figures that both errors grow without a bound when the odometry is used without any re-calibration. When the pose of the robot is updated time to time from the camera information, the position errors are bounded to about 12 cm (in the worst case of the 5th re-calibration point). Also the orientation error does not grow with the travelled distance.

The results of another experiment are shown in Fig. 6B. The path was a triangle with each edge 2 meters long. The task of the robot was again to execute this path 12 times using only the odometry data, and then again 12 times with the re-calibration from the camera. In this case the robot was not able to complete the task using only the odometry. At the 10th point of the path it was out of the area of the experiment, and it has been stopped due to the safety reasons. The errors on the path corrected from the camera information were small, with the exception of the orientation error at the point no. 8. At this point an error occurred in the execution of the pre-planned path, probably due to transmission problems with the wireless connection, and the robot executed a turn in a wrong direction, resulting in a huge orientation error. As it can bee seen from the figures the camera-based positioning system has enabled the robot to recover from such failure, and to complete the task with the position error of about 10 cm.

The localization errors measured during the experimental evaluation was much bigger than the theoretical accuracy of the fish-eye camera. It was because of such reasons as :

- the LED pattern on the robot is not perfectly aligned with the center of the platform,
- the alignment of the ceiling-mounted camera with the external coordinates system is not perfect,

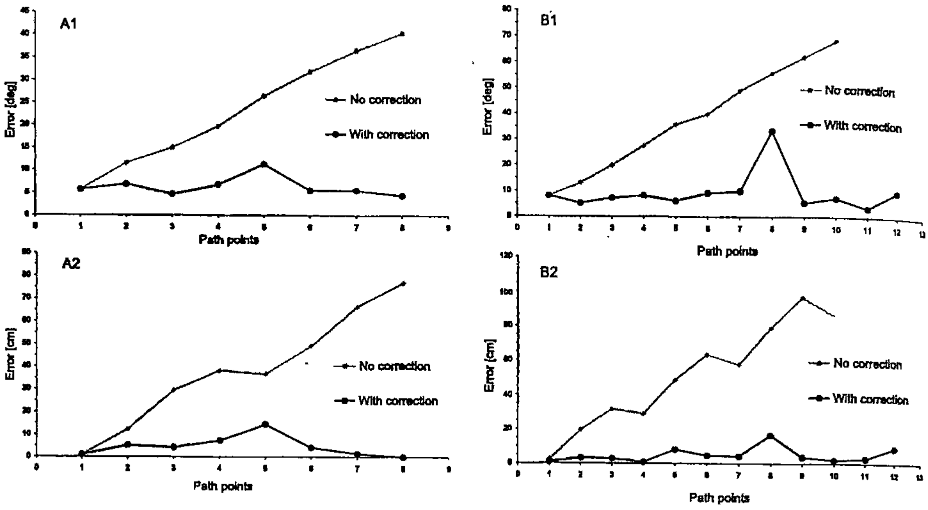

Figure 6: Results of two positioning experiments.

- the optical axis of the camera is not perfectly orthogonal to the floor plane,
- there is a pixelization error in the detection of the centers of the diodes.

Some of these errors can be suppressed by improving the experimental set-up, especially the mechanical part of the camera mounting.

6. CONCLUSIONS

This paper describes a positioning system for mobile robots which uses an overhead camera mounted to the ceiling of a room and active LED markers on robots. The novel method for correction of distorted images from the fish-eye camera has been used. The experiments performed with a real mobile robot shows the ability of the proposed method to localize the robot quite precisely and (what is very important) to recover from localization errors.

The global vision system as an alternative to active beacons or more sophisticated sensors mounted on-board of robots offers several advantages, especially in the context of multi-robot applications. Currently this system is integrated in the multi-agent perception and world-modelling architecture proposed in [4]. In this system of multiple mobile robots performing transportation tasks the vision-based positioning system will be used to re-localize mobile agents from time to time, especially in so critical areas as docking stations and narrow entrances.

Acknowledgements :
This work is sponsored by the Polish National Committee for Scientific Research under grant 8T11A 003 19.

References

[1] Crowley J. L., *Mathematical Foundations of Navigation and Perception for an Autonomous Mobile Robot*, Workshop on Reasoning with Uncertainty in Robotics, University of Amsterdam, The Netherlands, 1995.

[2] Feng L., Borenstein J., Everett H., *"Where am I ?" Sensors and Methods for Autonomous Mobile Robot Positioning*, Technical Report, University of Michigan, 1994.

[3] Jain R., Kasturi R., Schunck B., *Machine Vision*. McGraw-Hill, New York, 1995.

[4] Kasiński A., Skrzypczyński P., *Experiments and Results in Multi-modal, Distributed, Robotic Perception*, Distributed Autonomous Robotic Systems 4, (L. E. Parker *et al.*, eds.), Springer-Verlag, Tokyo, 2000, 283-292.

[5] Leonard, J., Durrant-Whyte, H. F., *Directed Sonar Sensing for Mobile Robot Navigation*, Kluwer Academic Publishers, 1992.

[6] Shah S., Aggarwal J. K., *A Simple Calibration Procedure for Fish–Eye (High Distortion) Lens Camera*, Proc. of IEEE Int. Conf. on Robotics and Automation, San Diego, 1994. 3422-3427.

[7] Skrzypczyński P., *Localization of a Mobile Robot Based on Natural Landmarks*, Proc. IFAC Symposium on Intelligent Robot Vehicles, Madrid 1998. Vol. 2, 615-620.