Real Time Indoor Robot Localization Using a Stationary Fisheye Camera

Konstantinos K. Delibasis[1], Vasilios P. Plagianakos[1], and Ilias Maglogiannis[2]

[1] Univ. of Central Greece, Dept. of Computer Science and Biomedical Informatics,
Lamia, Greece
[2] University of Piraeus, Dept. of Digital Systems, Piraeus, Greece
kdelibasis@yahoo.com, vpp@ucg.gr, imaglo@unipi.gr

Abstract. A core problem in robotics is the localization of a mobile robot (determination of the location or pose) in its environment, since the robot's behavior depends on its position. In this work, we propose the use of a stationary fisheye camera for real time robot localization in indoor environments. We employ an image formation model for the fisheye camera, which is used for accelerating the segmentation of the robot's top surface, as well as for calculating the robot's true position in the real world frame of reference. The proposed robot localization algorithm does not depend on any information from the robot's sensors and does not require visual landmarks in the indoor environment. Initial results are presented from video sequences and are compared to the ground truth position, obtained by the robot's sensors. The dependence of the average positional error with the distance from the camera is also measured.

Keywords: computer vision, indoor robot localization, stationary fisheye camera.

1 Introduction

Robot localization is fundamental for performing any task, such as route planning [1]. Most of the existing mobile robot localization and mapping algorithms are based on laser or sonar sensors, as vision is more processor intensive and good visual features are more difficult to extract and match [2]. Except for input from the sensors, these approaches also require the existence of a map, as well as the relevant software module for navigation. A number of different approaches that use data from images acquired by a camera onboard the robot have been proposed. In [3] visual memory consisting of a set of views of a number of different landmarks is used for robot simultaneous localization and map construction (SLAM). In [4] robot localization is achieved by tracking geometric beacons (visually salient points, visible from a number of locations). Similarly, a vision-based mobile robot localization and mapping algorithm is proposed in [2], which uses scale-invariant image features as natural landmarks in unmodified environments. Specifically for indoor environments, relative localization (dead-reckoning) utilizes information from odometer readings [5]. This class of algorithms, although very fast and simple, present serious drawbacks, since

H. Papadopoulos et al. (Eds.): AIAI 2013, IFIP AICT 412, pp. 245–254, 2013.

factors like slippage cause incremental error. Absolute localization is often based on laser sensors, or image processing from an onboard camera. For instance, in [6] ceiling lights and door number plates are used.

Very few approaches have been reported that employ the concept of a non fully autonomous robot. The system described in [7] uses a camera system mounted on the ceiling to track persons in the environment and to learn where the people usually walk in their workspace. The robot is tracked by the camera by means of a number of LEDs installed on the robot. In [8], the robot is tracked by a network of cameras, which use the information from a circular shape that has been installed on the robot. Fisheye cameras have been used onboard the robots for visual navigation [9].

In this work we report the use of a single stationary fisheye camera for real time robot localization, with extensions to robot navigation. We propose and utilize a computational model of the fisheye camera, which, in conjunction with the known robot shape can provide reliable location of the robot. More specifically, the top of the robot is segmented in each frame by means of color segmentation. The segmentation is assisted by precalculating all the possible pixels in the video frame that the top of the robot may occupy. The central pixel of the segmented top of the robot is used to obtain the robot's real location on the floor, using the model of the fisheye camera. Our approach is applicable to a very simple robot without any kind of sensors or map of the environment. Furthermore, the proposed approach does not require any visual landmarks in the indoor environment.

2 Methodology

2.1 Block Diagram of the Proposed Algorithm

In this work we present an algorithm for real time robot location in indoor environment, using one stationary fisheye camera. The main component of the proposed robot localization algorithm is the fisheye camera model that relates frame pixels with real world geometry. Thus, the real location of an object with known geometry (such as the robot) can be calculated. This model is also used to accelerate the segmentation of the robot from video frames. The proposed algorithm is very fast, thus the robot is localized in real time, without any additional information from sensors. The main components of the proposed algorithm are shown in Fig. 1, where the online steps are differentiated by the steps performed only once, during the calibration phase.

2.2 Forward and Inverse Fisheye Camera Model

The main characteristic of the fisheye camera is the ability to cover a field of view of 180 degrees. The objective of this subsection is to establish an analytical tool that relates a point in real world coordinates with the image pixel recorded by the fisheye camera and inversely map every frame pixel to the direction of view in the real world.

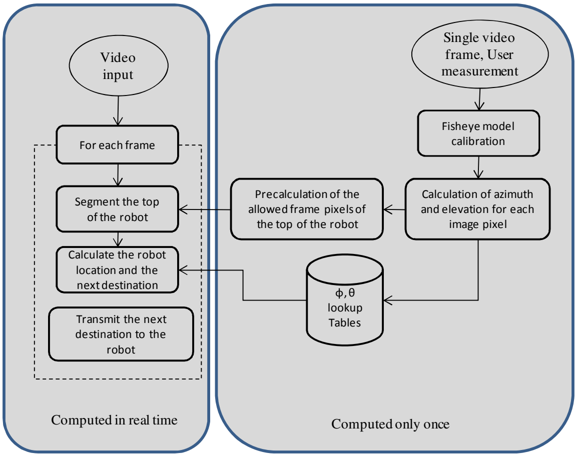

Video input
For each frame
Segment the top of the robot
Calculate the robot location and the next destination
Transmit the next destination to the robot
Computed in real time
Single video frame, User measurement
Fisheye model calibration
Calculation of azimuth and elevation for each image pixel
Precalculation of the allowed frame pixels of the top of the robot
ϕ,θ lookup Tables
Computed only once

Fig. 1. The overall architecture of the proposed algorithm

The direction of view is defined in spherical coordinates by its azimuth and elevation angles θ, φ. Thus forward fisheye modeling M can be written in the general form by

$$(j,i) = M(x,y,z) \tag{1}$$

where as inverse fisheye modeling is described as:

$$(\theta,\varphi) = M^{-1}(j,i). \tag{2}$$

Forward Fisheye Model

The definition of a model for the fisheye camera is based on the physics of image formation, as described in [9], [10]. We consider a spherical element of arbitrary radius R_0 with its center at $K(0,0,z_{sph})$. For any point P with real world coordinates (x,y,z), we determine the intersection Q of the line KP with the optical element.

The point P (as well as ny point on the KP line of view) is imaged at the central projection (x_{im}, y_{im}) of Q on the image plane with equation $z=z_{plane}$, using the $O(0,0,0)$ as center of projection. The KP line is uniquely defined by its azimuth and elevation angles, θ, φ respectively.

We set z_{plane} to an arbitrary value, less than R_0 and define $z_{sph} = pz_{plane}$, where p is the primary parameter of the fisheye model that defines the formation of the image. To account for possible lens misalignments with respect to the camera sensor that could induce imaging deformations on the imaged frame [11], we introduce two extra model parameters: the X and Y position of the center of spherical lens $K(x_{sph}, y_{sph}, z_{sph})$ with respect to the optical axis of the camera. Now the camera model parameters consist of p, x_{sph}, y_{sph}. Figure 2 shows the geometry of the fisheye camera model for $x_{sph} = 0$ and $y_{sph} = 0$.

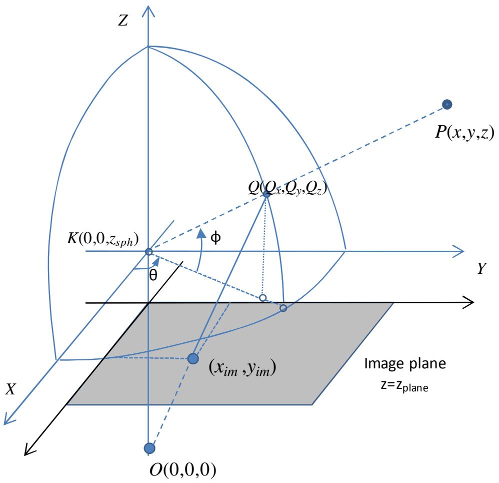


Fig. 2. The geometry of the proposed fisheye camera model

The position of Q is given by

$$\left(Q_x, Q_y, Q_z\right) = \left(\lambda\left(x - x_{sph}\right), \lambda\left(y - y_{sph}\right), \lambda\left(z - z_{sph}\right)\right) \tag{3}$$

where the parameter λ is obtained by inserting (3) this into the equation of the spherical optical element and requiring $\lambda \in$ [0,1]:

$$\left(\lambda\left(x - x_{sph}\right) - x_{sph}\right)^2 + \left(\lambda\left(y - y_{sph}\right) - y_{sph}\right)^2 + \left(\lambda\left(z - z_{sph}\right) - z_{sph}\right)^2 - R_0^2 = 0. \tag{4}$$

Finally, we calculate the central projection $\left(x_{im}, y_{im}\right)$ of Q on the image plane:

$$\left(x_{im}, y_{im}\right) = \frac{z_{plane}}{z_{sph}}\left(Q_x, Q_y\right) \tag{5}$$

Thus, any point P with real world coordinate z>z_{sph}, will be imaged on the image plane at position $\left(x_{im}, y_{im}\right)$, which is bounded by the radius of the virtual spherical optical element R_0: $-R_0 \le x_{im} - x_{sph} \le R_0$. When $x \to \infty$ then $\left(x_{im} - x_{sph}\right) \to R_0$ (the same holds for the y coordinate as well). In order to calculate the pixel of the video frame, we need to introduce the concept of the center of distortion *CoD* pixel located as the center of the circular field-of-view, (corresponding to elevation $\varphi=\pi/2$) [12] and the radius R_{FoV} of the field of view. The *CoD* and R_{FoV} are calculated only once, using user-input and a simple least squares optimization. Now, the image pixel position (i,j) that corresponds to the projection on the image plane (x_{im}, y_{im}) is calculated by a simple linear transform:

$$\left(j,i\right)=\left(x_{im},y_{im}\right)\frac{R_{FoV}}{R_0}+\left(CoD_x,CoD_y\right) \tag{6}$$

Calibration of Fisheye Camera Model

In order to utilize the fisheye camera model, we need to calibrate the model, i.e. determine the values of the unknown p, x_{sph} and y_{sph} parameters. Initially, the user provides the position of N_p=18 landmark points $\left\{\left(X^i_{im},Y^i_{im}\right)\right\}, i=1,2,\ldots,N_p$ on any video frame. The real world coordinates of these landmark points were also measured $\left\{\left(x^i_{real},y^i_{real},z^i_{real}\right)\right\}$, with respect to the reference system, (superscripts do not indicate powers). The position of the landmark points $\left(x^i_{im},y^i_{im}\right)$ on the video frame are calculated using (1). The values of the model parameters are obtained by minimizing the error between the expected and the observed frame coordinates of the landmark points. This minimization is performed using exhaustive search. If we allow p to vary from 0.5 to 1.5 with a step of 0.01 and x_{sph}, y_{sph} to vary from in the range of [–R_0/4, R_0/4] with a step of R_0/32, the model parameters are obtained in just few minutes using the Matlab programming environment in an average laptop computer. The resulting calibration of the fisheye model is shown in Figure 3, where a virtual grid of points is laid on the floor and on the two walls of the imaged room.

It has to be emphasized that this operation is only performed once after the initial installation of the fisheye camera and it does not need to be repeated in real time.

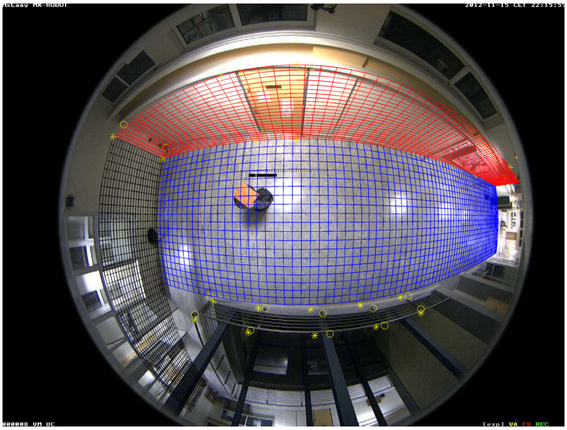


Fig. 3. Visualization of the resulting fisheye model calibration. The landmark points defined by the user are shown as circles and their rendered position on the frame marked by stars.

Inverse Model of the Fisheye Camera - Azimuth and Elevation Look-Up Tables

To use the model of the fisheye camera to refine the video segmentation, we need to utilize the elevation θ and azimuth φ of the line of view for each segmented pixel. Given the (j,i) coordinates of a pixel of the video frame, the θ and φ angles are calculated as following. Using equation (6), the position of the pixel on the camera sensor is calculated:

$$\left(x_{im}, y_{im}\right) = \frac{R_0}{R_{FOV}}\left(\left(j,i\right)-\left(CoD_x, CoD_y\right)\right) \tag{7}$$

The intersection Q of the spherical optical element with line defined by $O(0,0,0)$ and (x_{im},y_{im}) is determined, as

$$\left(Q_x, Q_y, Q_z\right) = m\left(x_{im}, y_{im}, z_{plane}\right) \tag{8}$$

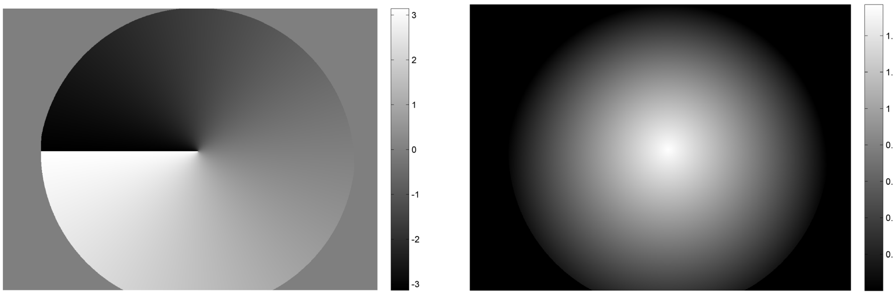
where the parameter m is determined by requiring Q to lie on the spherical optical element:

$$m^2\left(x_{im}^2 + y_{im}^2 + z_{plane}^2\right) - 2\left(x_{im}x_{sph} + y_{im}y_{sph} + z_{plane}z_{sph}\right)m + x_{sph}^2 + y_{sph}^2 + z_{sph}^2 - R_0^2 = 0. \tag{9}$$

The required θ and φ are obtained by converting the Cartesian $\left(Q_x, Q_y, Q_z\right)$ to spherical coordinates:

$$\varphi = \cos^{-1}\left(\frac{Q_z - z_{sph}}{R_0}\right), \theta = \sin^{-1}\left(\frac{Q_y - y_{sph}}{\sqrt{\left(Q_x - x_{sph}\right)^2 + \left(Q_x - x_{sph}\right)^2}}\right). \tag{10}$$

The above process is executed only once, after the calibration of the fisheye camera model and the resulting values for the θ and φ parameters for each frame pixel are stored in two look-up tables, of size equal to a single video frame. The look-up tables for the azimuth θ and the elevation φ are shown in Fig. 4(a) and 4(b), respectively. As expected, the azimuth obtains values in [-π,π], whereas the elevation obtains values in [0,π/2], with the maximum value at the CoD pixel of the frame.



(a) (b)

Fig. 4. Graphical representation of the azimuth (a) and elevation (b) look-up tables

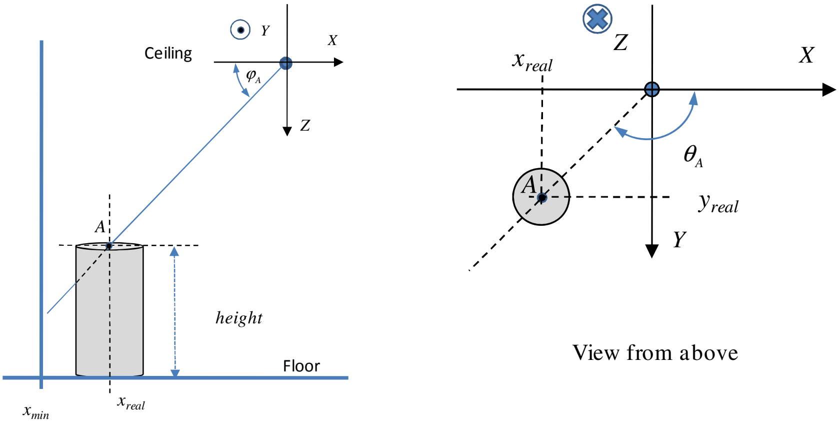
2.3 Video Acquisition and Robot Localization

Robot localization is achieved utilizing the video stream acquired by the fisheye camera. The first step is segmentation of the robot's top surface. In order to assist segmentation, we have placed a specific color on the top of the robot. Thus, a pixel (*i*,*j*) is segmented as top of the robot, according to its color in the *RGB* color system, using the following rule:

$$R(i,j) > 1.8 \cdot G(i,j) > 1.2 \cdot B(i,j) \tag{11}$$

The segmentation achieved by Eq.(11) is accurate and efficient (see Fig. 7), therefore, the use of other color spaces, such as the *HSV* was not considered necessary. Eq.(11) is specific to the color that was used to mark the top of the robot and can be modified if a different color is used. In order to increase the accuracy of the segmentation by excluding possible false pixels, as well as to accelerate its execution, we precalculate and store in a binary mask all the frame pixels in which the top of the robot is possible to be imaged by the specific camera. Since the robot has a constant height h_R, its real position can be calculated unambiguously in the frame, provided that the pixel (*i*,*j*) imaging the center of its top surface has been segmented. The azimuth θ and elevation φ of the line of view of this pixel is given by the look-up tables. Then the real position $\left(x_{real}, y_{real}\right)$ of the robot on the floor is given by

$$x_{real} = \frac{z_{\max} - h_R}{\tan\left(\varphi_{ij}\right)} \cos\left(\theta_{ij}\right), \; y_{real} = \frac{z_{\max} - h_R}{\tan\left(\varphi_{ij}\right)} \sin\left(\theta_{ij}\right) \tag{12}.$$



(a) (b)

Fig. 5. The geometry for the calculation of the robot's real location given the pixel *A* that images the center of the robot's flat top surface

The relevant geometric concept is shown in Fig. 5. Given the geometry of the indoor space being imaged by the fisheye camera, we may exclude the non accessible areas. Thus, the binary mask of the allowed pixels is set to 1 only for the pixels for which the corresponding (x_{real},y_{real}) does not lie on inaccessible areas (Fig. 6). Robot segmentation according to (11) is performed only on the non-zero pixels of the mask. The robot's real location is calculated using (12) and stored in each frame.

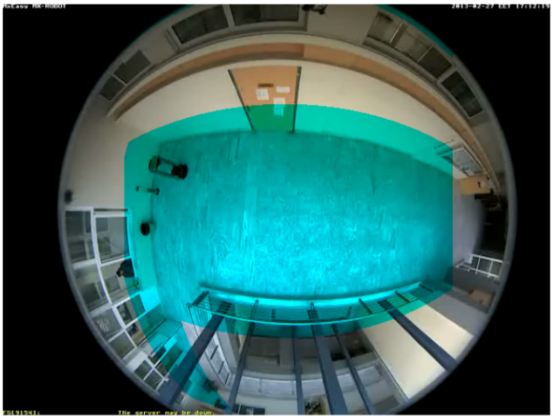


Fig. 6. A typical video frame. The pixels where it is possible for the segmented top of the robot to appear have been highlighted.

3 Results

We present results to indicate the proposed accuracy of robot localization. We acquired ten video sequences with duration between 90 and 120 sec, at 25 frames (480x640 pixels) per second (fps). The robot used in this study is a PeopleBot of the Adept MobileRobots company. We used the robot built-in sensors to record its location, to be utilized as ground truth for validating the results. No built-in robot sensor readings were used for the proposed localization algorithm.

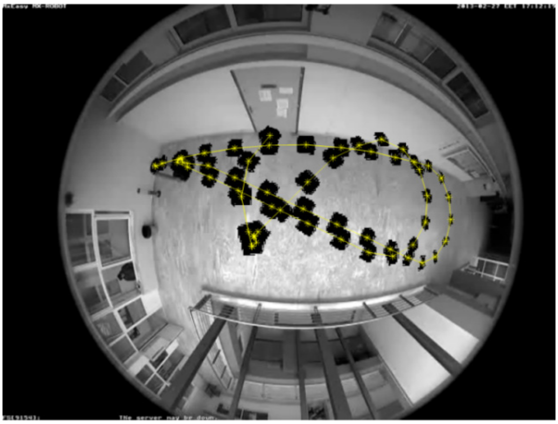
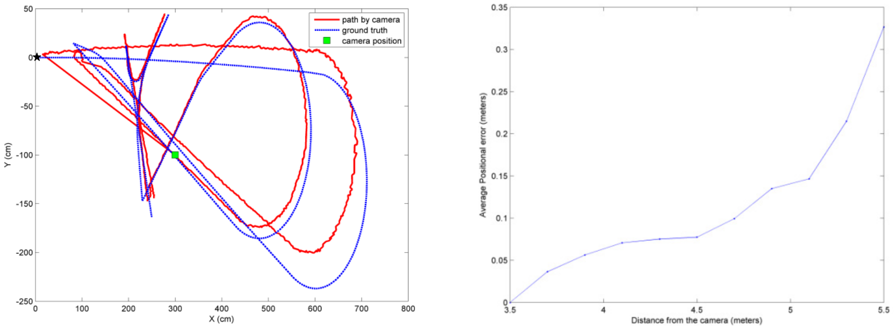


Fig. 7. The path of the robot in one of the acquired videos, the segmented top of the robot surface and the determined central point, used for the calculation of real world position

Figure 7 shows the resulting segmentation of the top of the robot from a number of frames of one of the videos, with its center of gravity marked. The path of the robot in the video frame is visible. Figure 8(a) shows the path of the robot, estimated by the proposed algorithm (continuous curve). The ground truth has also been included for comparison (dotted curve). The starting point is marked by a star and the position of the fisheye camera is also shown, since the positional error is expected to vary proportionally to the distance from the camera. As it can be observed, the position estimated by the proposed algorithm is very accurate close to the camera but the error starts to increase as the robot moves away from the camera. This can be attributed to the deterioration of the spatial resolution of the fish-eye image, as well as to inaccuracies of the calibration of the camera model. Notice that the localization error is not accumulated, as in the case of relative measurements (dead reckoning). In Fig. 8(b) the dependence of the error is presented with respect to the distance from the camera.



(a) (b)

Fig. 8. (a) The robot position obtained by the proposed algorithm in a single video, as well as the ground truth positions. (b) The average error in robot localization achieved by the proposed algorithm, as a function of the robot's distance from the fisheye camera.

4 Discussion and Further Work

An algorithm for the localization of a robot in indoor environment has been presented, which is based only on video acquired by a stationary fisheye camera, installed on the ceiling of the imaged room. The proposed algorithm has been implemented using Matlab and executed on an Intel(R) Core i5-2430 CPU @ 2.40 GHz Laptop with 4 GB Ram, under Windows 7 Home Premium. The mean execution time of the proposed localization algorithm was approximately 70 msec per frame of dimension 480x640, with no parallelization or specially optimized source code. In our experiments, the overall procedure including the acquisition of the real time video stream through a WiFi connection resulted in processing 7 frames per second, which can support real time localization because of the low speed of the robot. This rate can be increased using a different development environment.

Regarding the localization accuracy, our experiments showed that the algorithm is able to localize the robot with positional error less than 0.1 meters in distances up to 4.5 meters from the stationary camera. The localization error increases proportionally to the distance of the robot from the camera, due to inaccuracies of the fisheye camera model. For this reason we will explore further the proposed camera model, possibly by including more controlling parameters and evolutionary algorithms for its calibration. Navigation experiments based on the proposed robot localization algorithm will also be performed, to assess its usefulness in assisting environments.

Acknowledgment. The authors would like to thank the European Union (European Social Fund ESF) and Greek national funds for financially supporting this work through the Operational Program "Education and Lifelong Learning" of the National Strategic Reference Framework (NSRF) - Research Funding Program: \Thalis \ Interdisciplinary Research in Affective Computing for Biological Activity Recognition in Assistive Environments.

References

1. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics. MIT Press (2005)
2. Se, S., Lowe, D., Little, J.: Mobile Robot Localization and Mapping with Uncertainty using Scale-Invariant Visual Landmarks. The International Journal of Robotics Research 21(8), 735–758 (2002)
3. Royer, E., Lhuillier, M., Dhome, M., Lavest, J.-M.: Monocular Vision for Mobile Robot Localization and Autonomous Navigation, Intern. Journal of Computer Vision 74(3), 237–260 (2007)
4. Leonard, J., Durrant-Whyte, H.: Mobile robot localizations be tracking geometric beacons. IEEE T. on Robotics and Automation 7(3), 376–382 (1991)
5. Aider, O., Hoppenot, P., Colle, E.: A model-based method for indoor mobile robot localization using monocular vision and straight-line correspondences. Robotics and Autonomous Systems 52, 229–246 (2005)
6. Dulimarta, H.S., Jain, A.K.: Mobile robot localization in indoor environment. Pattern Recognition 30(1), 99–111 (1997)
7. Kruse, E., Wahl, F.: Camera-based monitoring system for mobile robot guidance. In: Proc. of the IEEE/RSJ Intern. Conf. on Intelligent Robots and Systems, IROS (1998)
8. Yoo, J.H., Sethi, I.K.: Sethi, Mobile robot localization with multiple stationary cameras. In: Mobile Robots VI. Proc. SPIE, vol. 1613, p. 155 (1992)
9. Courbon, J., Mezouar, Y., Eck, L., Martinet, P.: A Generic Fisheye camera model for robotic applications. In: Proceedings of the 2007 IEEE/RSJ International Conf. on Intelligent Robots and Systems, San Diego, CA, USA, October 29 - November 2 (2007)
10. Computer Generated Angular Fisheye Projections, `http://paulbourke.net/dome/fisheye/`
11. Shah, S., Aggarwal, J.: Intrinsic parameter calibration procedure for a high distortion fish-eye lens camera with distortion model and accuracy estimation. Pattern Recognition 29(11), 1775–1788 (1996)
12. Micusik, B., Pajdla, T.: Structure from Motion with Wide Circular Field of View Cameras. IEEE T. on PAMI 28(7), 1–15 (2006)