Proceedings of the 2008 IEEE
International Conference on Robotics and Biomimetics
Bangkok, Thailand, February 21 - 26, 2009

Bearing Only FastSLAM Using Vertical Line Information from an Omnidirectional Camera

Mahisorn Wongphati, Nattee Niparnan and Attawith Sudsang
Department of Computer Engineering
Faculty of Engineering
Chulalongkorn University
Bangkok, Thailand, 10330
Email: {mahisorn.w, nattee, attawith}@cp.eng.chula.ac.th

Abstract— **This paper proposes an implementation of Fast-SLAM for localization and mapping for an indoor environment. The system use only the bearing information retrieved from an omnidirectional camera via vertical line segments which are common in an indoor environment. The real world experiment is also presented using Pioneer-DX3 mobile robot moving in a room with several objects.**

Index Terms— **Omnidirectional camera, SLAM, mobile robot, localization, mapping**

I. Introduction

Mobile robots exhibit their usefulness by their abilities to work in unknown environment. To do so, the robots must be able to navigate themselves safely when the knowledge of the surroundings is limited. This raises a well known research topic in robotics : the topic of Simultaneous Localization and Mapping (SLAM) [1]. SLAM is a problem deals with two pieces of information: position of robots and information of the surrounding. Both intrinsically depend on each other. The goal of SLAM can be described by two questions: *"How can we create the surrounding map without knowledge of robot position?"* and *"How to obtain the position of the robot without knowledge of map information ?"*.

Humans are adept in these localization tasks. Human can recognize objects in the environment very easily. Moreover, human can roughly identify the distance and bearing of the object with ease, due to the stereoscopy nature of the eyes and the prior knowledge of the objects. For example, we know the distance of other humans from the size they appear in our vision. By registering and roughly identifying the position of particular objects in the environment as humans move, the objects can be used as landmarks for localization of the positions (see Section 2.1 in [2])

Obviously, researches in SLAM follow this approach. Robots are equipped with sensors that measure position of some particular landmarks in the environment and the robots keep track of such landmarks. Most of the researches focus on *laser range finders* as the primary sensor for the position of the landmark [3]–[5]. Odometers and/or compasses are also incorporated to help registering data of the landmarks acquired as the robots move. Laser range finders gain this much popularity due to its accuracy [6] and ease of use; it can acquire the position of the surroundings without further processing. On the other hands, vision system receives much less interest in SLAM even though it resembles the behavior of human. This is because of lesser accuracy when compared to laser range finder system. The accuracy of vision system depends on the resolution of the image. The other reason is that the raw data acquired from the sensor usually requires further processing which grows proportionally to the resolution of the data. The advantage of vision system is that it can give more information that can be used to recognize landmarks. However, due to the fact that processing power of computers had risen continuously every year, using vision system as a sensor in SLAM becomes more and more promising.

Another problem that plagues vision system is the limited field of view of a traditional CCD camera. This issue hinders the use of traditional camera in many robotics works. In order to overcome this limitation, an *omnidirectional camera* was introduced since the early 1990s [7]. Omnidirectional cameras allow a robot to have vision in all directions around itself. Many research works are conducted on the topic of omnidirectional cameras. The literature has been advanced in both practice [8] and theory [9].

In this work, we use an omnidirectional camera as a main and only sensor for a mobile robot to solve a SLAM problem. The objective of our work is to show that rich information from an omnidirectional camera is sufficient for a SLAM problem in an indoor environment. We propose an implementation of a SLAM technique that can accurately detects, measures and recognizes landmark features in an indoor environment and can also localize the position of the robot accurately. Though using omnidirectional cameras in SLAM is not a novel idea, the prior works rely on stereoscopic image of two omnidirectional cameras. However, we propose a method that relies only on one cameras without any additional sensor. This is very beneficial when the resource of the robot is limited.

The rest of the paper is organized as follows. The next section presents the related works that use omnidirectional cameras for localization and/or mapping in mobile robotics. Section III describes the main idea of our work in which vertical lines are use as main landmarks in a modified FastSLAM [10] implementation. Section IV elaborates the detail and the equipments of our work. The experiment and the result is presented in Section V. Finally, Section VI concludes our work.

II. Related Works

Omnidirectional cameras have been used in robotics research for more that 15 years. In the early years, most researchers focus on the development and possible usage of the device on mobile robots [7]. This stage of the research concentrates on the practical aspect. Researchers later start to focus on theories that govern the behavior of the omnidirectional cameras. Properties of different camera models (conic, parabolic, hyperbolic or other combinations of mirror shapes) are thoroughly studied in late 1990s by many works [11], [12].

With the well-established theoretical background, researchers began to study the calibration method of the cameras. Calibration or the detection of the parameters of the cameras is vital to achieving greater accuracy. A complete calibration tool set for an omnidirectional camera with comparable level of ease of use as that of a pinhole camera is not available until recently [13], [14].

One distinguishable feature of an omnidirectional camera is that some special structure in man-made indoor environment can be detected very easily. Such feature is a vertical line resulting from the edge of common object in indoor environment such as corners of a room, edges of tables, chairs, etc. Several works utilize this feature as landmark in localization and mapping problems.

There are some works that show the possibility of using an omnidirectional camera in SLAM problem. In [15] and [16], the authors use a stereo omnidirectional camera for the position measurement of landmark features. In [17], undistinguishable landmarks are used in SLAM. The work that is most related our work was introduced in 2006 by Murillo [18]. This work uses a bearing-only information extracted from vertical lines in indoor environment. However, the SLAM method in the work is not clearly described.

III. Bearing Only FastSLAM

SLAM problems have been a main topic for mobile robot for many years and the result of this long research are many solutions for the problems [19].

Particle filter is one of the most successful approaches for SLAM problems. In mobile robot problems, we usually deal with unknown errors from many sources. Most of them cannot be clearly identified or represented using an exact model. Particle filter is designed for such problem with the capability for handling multi-model noise and uncertainty [6].

In our work, we modify FastSLAM 1.0 from Thrun [10] to fit our need. Mainly, we add the support for using vertical line information as sensor data. We also use a base code and implementation guide line from [20] in our work.

In this section, we provide brief introduction to FastSLAM and describe the measurement and motion model in our work.

A. FastSLAM

Let us begin with the basis of FastSLAM, the probabilistic model of localization and mapping. The system state of our SLAM problem consists of robot position (x) and map information (M). In order to estimate the change of the system, we must have the measurement data (Z), control data of the system (U) and initial position of the robot (x_0). This probabilistic model can be written in form of *a Probability Distribution Function (PDF)* where k is current time step as follow.

$$P(x_k, M \mid Z_{0:k}, U_{0:k}, x_0) \tag{1}$$

Without simplification of (1), tremendous amount of memory and processing power is needed to estimate the system state because we must store all information from the beginning.

PDF (1) can be simplified using Bayes'rule by separating the past system state $P(x_{k-1}, M \mid Z_{0:k-1}, U_{0:k-1})$ from current system state. The present state can be estimated by previous state and current measurement (z_k) and latest control data (u_k) through measurement model and motion model. The assumption of the motion model is that the current robot position was based on the last position and control only. Hence (1) can be simplified as the follow.

$$P(x_k \mid x_{k-1}, u_k) \tag{2}$$

Similarity, the measurement model is also based on the assumption that the current measurement data relies on the current robot position and map information only, which yield the following PDF.

$$P(z_k \mid x_k, M) \tag{3}$$

This simplification of probabilistic SLAM problem can be written in well known *predict : time-update* and *correct : measurement-update* recursive form as follows.

$$P(x_k, M \mid Z_{0:k-1}, U_{0:k}, x_0) = \int P(x_k \mid x_{k-1}, u_k) P(x_{k-1}, M \mid Z_{0:k-1}, U_{0:k-1}, x_0) dx_{k-1} \tag{4}$$

$$P(x_k, M \mid Z_{0:k}, U_{0:k}, x_0) = \frac{P(z_k \mid x_k, M) P(x_{k-1}, M \mid Z_{0:k-1}, U_{0:k}, x_0)}{P(z_k \mid Z_{0:k-1}, U_{0:k})} \tag{5}$$

In FastSLAM, the method to estimate the left hand PDF of (4) and (5) is represented by the combination of *Extended Kalman Filter* (EKF) and *Particle Filter*. The robot position (governed by the motion model) is estimated by particle filter and positions for landmarks (governed by the measurement model) is estimated by EFK. This method can be written in factoring form path estimation and landmark position estimation where c represents the correspondence variable between measurement data of i^{th} landmark. For unknown correspondence problems, c is usually calculated via *Maximum Likelihood Estimation* [10] as follow.

$$P(x_{0:k}, M \mid Z_{0:k}, U_{0:k}, c_{0:k}) = P(x_{0:k} \mid Z_{0:k}, U_{0:k}, c_{0:k}) \prod_i P(m_i \mid x_{0:k}, Z_{0:k}, U_{0:k}, c_{0:k}) \tag{6}$$

The main part of FastSLAM is measurement model and motion model. Both models must be provided by the user and they should reflect actual system as accurate as possible. For the measurement model in FastSLAM, we also need the error model of the sensor for the EKF part in the algorithm. It should be noted that FastSLAM itself is robust enough for some error on system models and parameters. More information and performance of FastSLAM can be found in [10].

B. *Measurement Model*

Measurement data for FastSLAM in our work are landmarks positions. In order to obtain such information, we use angle change of the same vertical line in two consecutive image frames and moving distance of the robot for triangulation. Since the robot only has the bearing information, the robot moving distance is obtained directly from control command.

1) Sensor Information: Omnidirectional cameras have many interesting properties [21]. One property that is the foundation of our work is a property of the line parallel to the camera axis. Because vertical lines that are parallel to the axis of an omnidirectional camera will always appear as radial lines in the omnidirectional image (Fig. 1), regardless of the rotation of the robot about the camera axis.

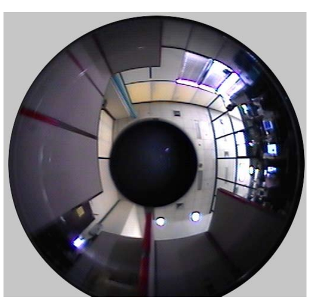

Fig. 1. Vertical line that appeared as radial line in an omnidirectional image

This rotation (about the camera axis) invariance suggests the ease of use of vertical lines in indoor environment. Moreover, vertical lines are very common in indoor structures, for example, room partitions, door frames, cabinet edges and window frames.

2) Vertical Line Detection and Bearing Information: The radial line in an omnidirectional image can be detected with a simple edge detection method by the following steps. First, we transforms an omnidirectional image into a panoramic view (Fig. 2). Second, we apply *Sobel Edge Detection* with the kernel, described in (7), that is sensitive to vertical edges.

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}, \begin{bmatrix} 1 & 0 & -1 \\ 2 & 0 & -2 \\ 1 & 0 & -1 \end{bmatrix} \tag{7}$$

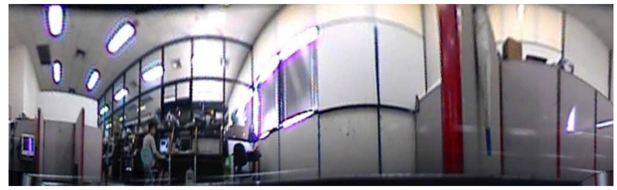

Fig. 2. Panoramic view of an omnidirectional image (radial line now appear as vertical line)

After that, we apply binary threshold to the grayscale image and locate the vertical edges by checking length, position and size. Finally, we merge the edges that are likely to be the result from the same vertical line and then calculate the position, angle with respect to the robot heading, length and neighborhood color information (Fig. 3).

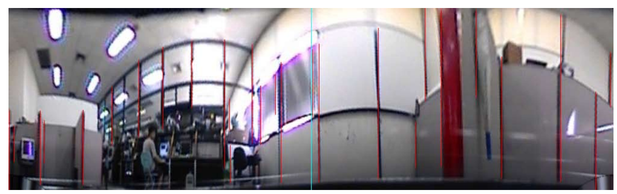

Fig. 3. Vertical lines extract using our method (red line)

3) Vertical Line Position and Error Model: With bearing only information, we calculate the position of each vertical line by triangulation using the the robot's moving distance and the angle change of the vertical line (Fig. 4).

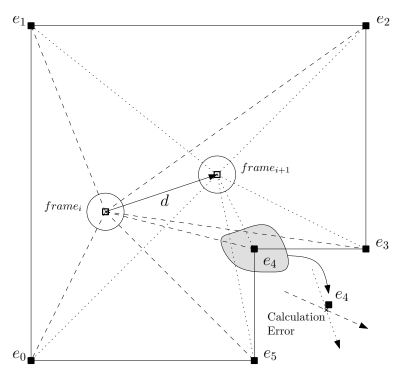

Fig. 4. Information for vertical line localization and possible error from the calculation

In order to match radial images of the same vertical line from two consecutive frames, we check the characteristic of the radial lines on moving omnidirectional (Fig. 5). We also check the neighborhood color of each line for better color matching. This color matching method and its performance is also reported by Tang [22].

This simple scheme works perfectly in an ideal environment. However, error in various part of the system, such as varying lighting condition, noise in the camera, shadows, slippage of the robot making the control command to not accurately reflect the actual position change, etc., result in the error of the resulting position of the vertical line (see Fig. 4).

This error is the most essential part in our work. We obtain an error model of our sensor by real experiment on our robot. First, we calibrate and adjust the camera mounting to reduce error. Second, we drive robot along the straight line and capture omnidirectional image at every specific distance. Finally, we calculate the vertical line position and checking the error distribution.

From our collected data, we tested various error models with FastSLAM algorithm. From our preliminary study, one of the best error model for our sensor is the two-dimensional normal distribution model. Example of triangulation error is shown In Fig. 6.

4) Omnidirectional Calibration: Error of vertical line position calculation depends on two components: moving distance of the robot and angle measurement of the vertical line. The method for handling the first component is embedded in FastSLAM and the motion model in Section III-C. For the latter component, we can improve the accuracy and decrease the error in omnidirectional camera configuration by a calibration.

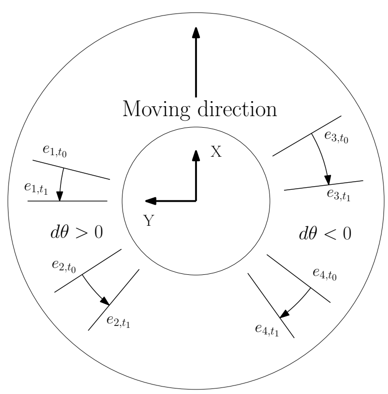

Fig. 5. Angle change of the radial lines in an omnidirectional image when robot move forward.

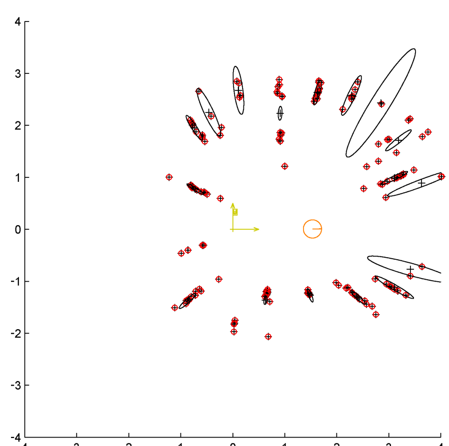

Fig. 6. Vertical line position (red circle), error covariance (ellipse) and estimated position (plus sign)

An omnidirectional camera consists of a set of curved mirrors, a lens and a CCD camera. Our camera has two parabolic mirrors: mirror-supporting a glass tube and a top mirror (see Fig. 11). Significant error comes from the misalignment between these components. These numbers of component require many joints and connections that may easily cause undesirable misalignment but it is almost impossible to rectify. The examples of the omnidirectional images with misalignment problems are shown in Fig. 7.

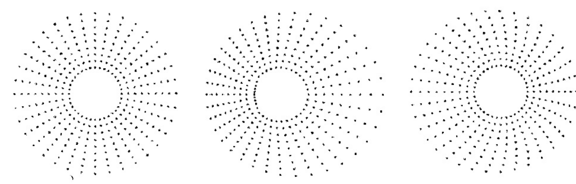

Fig. 7. Images for visualization the problem that cause by misalignment components.

To overcome this problem, we need a tool that can find an accurate mapping function between a point on an image plane and direction of the point in 3D (Fig. 8). With an accurate mapping function, we can create any accurate and precise image transformation function from an omnidirectional view to a desired view port.

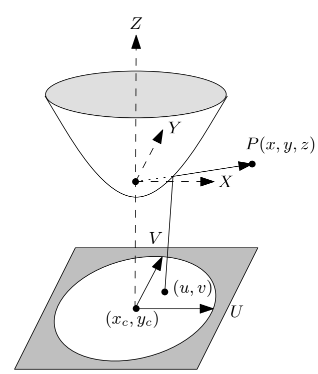

Fig. 8. This image show the relation between point in 3D (x, y, z) and point on an image plane (u, v). The center of omnidirectional was (x_c, y_c)

The software from [13] provides fully implemented Matlab toolbox for calibration and finding an accurate mapping function for an omnidirectional camera. This tool is also capable of compensating the misalignment between camera components. From our experiment, using the software from [13] for camera calibration is vital to the result of our method.

C. Motion Model

For the motion model, we adopt the probabilistic model of a differential driven mobile robot from [2]. We also simplify our model by reducing the robot motion to be rotation after translation (Fig. 9). With this model, the error from the motion of the robot will be (err_{tran}), rotation error (err_{rot}) and drift of robot heading (err_{drift}).

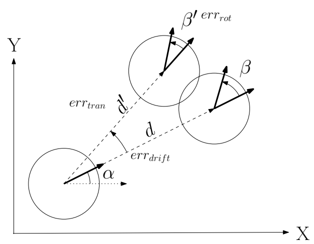

Fig. 9. Simplified robot motion step and possible error that may occur

D. Modified FastSLAM Algorithm

Traditional FastSLAM implementation assumes that there is only one measurement per each time step. However, there are multiple vertical line positions that are measured at each step. Using only one dominant measurement, even though possible, may result in poor performance. This is because the vertical line is usually very dense in the environment and we do not have explicit correspondent of measurement and the landmark data in the algorithm. The correspondent is done implicitly by maximum likelihood estimation. Hence, using one measurement might result in wrong correspondent especially when there are several landmarks situated near each other. It is better to add the support for multi-measurement to the algorithm. We add the support simply by choosing maximum likelihood outcome on every measurement at each step.

IV. Equipment and Testing Environment

A. Robot Platform

We use a Pioneer-DX3 robot for our experiment (Fig. 10). This robot uses 2 wheeled differential drive as its locomotion. There is a 1 GHz Pentium III onboard PC for local process (image capture, server for remote client and motion control). We control this robot via onboard wireless LAN using a Windows XP operating system.

Fig. 10. Pioneer-DX3 robot platform from [23]. The robot equipped with omnidirectional camera and adjust able mounting plate (pink color)

B. Omnidirectional camera

The omnidirectional camera on our robot is OmniView360° from [24]. This camera has $-5°$ to $65°$ vertical field of view. The camera was constructed from two parabolic mirrors and equipped with color NTSC CCD camera with 640×480 resolution. The outline of the camera components and the actual camera is shown in Fig. 11.

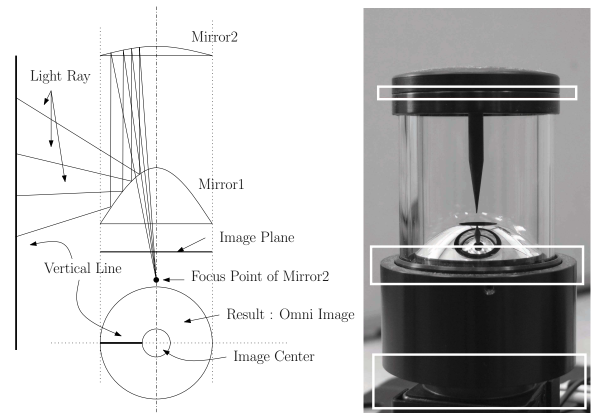

Fig. 11. Outline of camera component, property of vertical line and the camera using in our work

Our special camera mounting (pink plate in Fig. 10) provides more degree of freedom for camera adjustment to achieve a required camera configuration (perpendicular with the moving plane).

C. Testing Environment

We did our experiment on our actual laboratory. The floor plan of the experimental room was shown in Fig. 12.

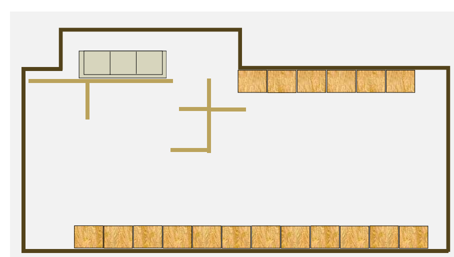

Fig. 12. Floor plan in our experiment. The room size about 4×10 m^2 and covered with plastic tiles of 30×30 cm^2 size

In our experiment, we manually drive the robot through the room from the starting point and going around the room and then back to the position near the start point as show in Fig. 13.

V. Result

With only vertical line information from an omnidirectional camera and control information from the user. We manage to get an accurate map from FastSLAM algorithm (Fig. 14). The map is compared to the floor plan that is manually measured using tape-ruler. The actual robot position was measured by the position of the robot on the tile grid on the floor at each moving interval.

The error between the real robot path and the path from our FastSLAM implementation is under $30 \times 30\ cm^2$ grid resolution. The error of the landmark position is also be under $30 \times 30\ cm^2$ resolution.

VI. Conclusion

We present an implementation of FastSLAM algorithm on a real mobile robot using only an omnidirectional camera as a sole sensor of the system. The information acquired from the vision system is bearing of vertical lines which is very easy to detect. The method is simple but efficient. The result from the experiment shows that our method has strong possibility for the real world application.

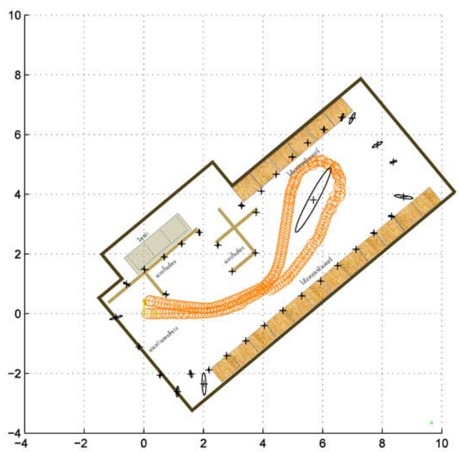

Fig. 13. Localization (orange circle) and mapping result (plus sign) overlayed on floor plan

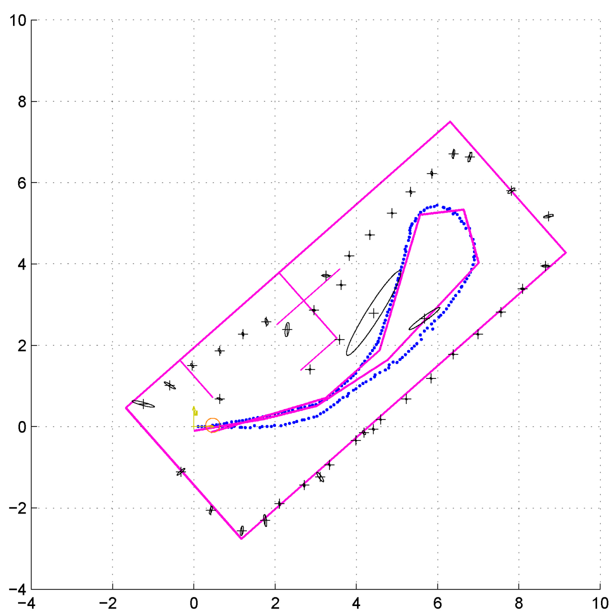

Fig. 14. Localization and mapping result from the experiment. Map information (vertical line position) was present by plus sign. The real robot path in poly line was compared to computed path from SLAM algorithm (doted line)

References

[1] H. Durrant-Whyte and T. Bailey, "Simultaneous localization and mapping: Part i, ii," *IEEE Robotics & Automation Magazine*, June June, September 2006.

[2] I. Rekleitis, "Cooperative localization and multi-robot exploration," Ph.D. dissertation, School of Computer Science, McGrill University, 2003.

[3] P. Biber, H. Andreasson, T. Duckett, and A. Schilling, "3d modeling of indoor environments by a mobile robot with a laser scanner and panoramic camera," in *IEEE/RSJ International Conference on Intelligent Robots and Systems*, Sendai, Japan, 2004.

[4] A. Diosi, G. Taylor, and L. Kleeman, "Interactive slam using laser and advanced sonar," in *IEEE International Conference on Robotics and Automation*, Barcelona, Spain, 2005.

[5] D. Fox, W. Burgard, H. Kruppa, and S. Thrun, "A monte carlo algorithm for multi-robot localization," in *AAAI 11th Conference on Innovative Applications of Artificial Intelligence*. Orlando, Florida: AAAI, 1999.

[6] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. The MIT Press, 2005.

[7] Y. Yagi and S. Kawato, "Panorama scene analysis with conic projection," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 1990.

[8] Y. Yagi and M. Yachida, "Real-time omnidirectional image sensors," *Int. Journal of Computer Vision*, vol. 58(3), pp. 172–207, 2004.

[9] S. Baker and S. K. Nayar, "A theory of catadioptric image formation," in *IEEE Int. Conf. on Computer Vision*, 1998.

[10] S. Thrun, M. Montemerlo, D. Koller, B. Wegbreit, J. Nieto, and E. Nebot, "Fastslam: An efficient solution to the simultaneous localization and mapping problem with unknown data association," *Journal of Machine Learning Research*, 2004.

[11] S. Baker and S. K. Nayar, "A theory of single-viewpoint catadioptric image formation," *International Journal of Computer Vision*, vol. 35, no. 2, pp. 1 – 22, 1999.

[12] C. Geyer and K. Daniilidis, "Catadioptric camera calibration," in *IEEE International Conference on Computer Vision*, 1999.

[13] D. Scaramuzza, A. Martinelli, and R. Siegwart, "A toolbox for easily calibrating omnidirectional cameras," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006.

[14] C. Mei and P. Rives, "Single view point omnidirectional camera calibration from planar grids," in *IEEE Int. Conf. on Robotics and Automation*, 2007.

[15] C. Drocourt, L. Delahoche, B. Marhic, and A. Clerentin, "Simultaneous localization and map construction method using omnidirectional stereoscopic information," in *IEEE Int. Conf. on Robotics and Automation*, 2002.

[16] J.-H. Kim and M. J. Chung, "Slam with omni-directional stereo vision sensor," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Las Vegas,Nevada, 2003.

[17] H. Huang, F. D. Maire, and N. Keeratipranon, "Bearing-only slam with indistinguishable landmarks." in *Proceedings The 2007 Australasian Conference on Robotics and Automation*, Brisbane, Australia, 2007.

[18] A. Murillo, J. Guerrero, and C. Sagüés, "Robot and landmark localization using scene planes and the 1d trifocal tensor," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2006.

[19] R. Siegwart and I. R. Nourbakhsh, *Introduction to Autonomous Mobile Robots (Intelligent Robotics and Autonomous Agents)*. The MIT Press, 2004.

[20] K. O. Arras, "The cas robot navigation toolbox: Users guide and reference," Center for Autonomous Systems, KTH, September 2004.

[21] S. K. Nayer, "Catadioptric omnidirectional camera," in *IEEE Conf. on Computer Vision and Pattern Recognition*, 1997.

[22] L. Tang and S. Yuta, "Indoor navigation for mobile robots using memorized omni-directional images and robot's motion," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2002.

[23] "Mobilerobots inc." in *http://www.mobilerobots.com*.

[24] "Remotereality," in *http://www.remotereality.com/*. [Online]. Available: http://www.remotereality.com/