

# Parallel Projection Based Self Localization Method for Mobile Navigation Applications

Shung Han Cho, Yuntai Kyong, Yunyoung Nam and Sangjin Hong  
*Stony Brook University*

USA

We-Duke Cho

*Ajou University*

KOREA

## 1. Introduction

In navigation application of mobile robot, position is estimated based on odometry and the measurement obtained by peripheral devices such as ultra-sonar, range finder, and visual sensor. Odometry is usually unreliable due to the internal factor of encoder device and the external factor of slippage. These kinds of errors accumulated in navigation continuously affect the estimation accuracy (Borenstein & Feng, 1996b; Martinelli et al., 2007). Range finder and sonar are often used in navigation (Borenstein et al., 1996a; Borenstein et al., 1997; Großmann & Poli, 2001; Demirli & Molhim, 2004). However such measurement is not reliable in highly-dynamic environments where radar or sonar beams can be frequently blocked or confused by moving objects such as people. They are not applicable to localization in large area either because of their limited range. Also, passive sensor requires active landmarks such as beacon, which requires modification of environment and is not practical especially in an outdoor environment. Moreover, interference among several mobile sensors causes inability to properly localize their locations.

Visual-information-based self-localization has advantages over above methods in two-folds. First, it does not require active landmarks (i.e., reference objects) such as beacon and also natural objects can serve as landmarks. The other advantage is that it is more effective and reliable in dynamic environment as the sensible range is not limited by the line-of-sight. These advantages have fostered much effort in research community of navigation application (Jang et al., 2002; Hayet et al., 2007; Se et al., 2002).

Reference objects can be either artificial or natural objects (Betke & Gurvits, 1997; Briggs et al., 2000; Thrun, 1998; Dao et al., 2004). We assume that reference objects can be uniquely identified by a mobile robot. The global coordinates of landmarks or reference objects are known to mobile robot. There have been extensive researches about robot localization with known reference objects. The conventional self localization algorithms use the bearings of the landmarks relative to each other. This is called "visual angle" formed by the rays from a mobile robot to each reference point (Sutherland & Thompson, 1993). The location of a mobile robot is estimated by finding the intersection point of the circles passing the

reference point. In the ideal case, the mobile robot is localized with three landmarks. However, although the matched landmarks from the image are found in the known map, the visual angle is usually distorted by the nonlinear property of a camera lens. Thus, the solution is estimated by minimizing the error of all possible landmarks pairs and the estimation error is minimized in proportion to the number of landmarks pairs (Betke & Gurvits, 1997; Cohen & Koss, 1993). Moreover, multiple solutions exist when all landmarks form a circle. Another approach uses perspective projection model to identify the relationship between the view point and the landmark. Although this method is simple in calculation, the performance is also worsened by the projection method without calibrating the camera nonlinearity (Dao et al., 2004; Liu & Zhou, 2007).

In this chapter, we propose a self localization method using a single visual image with the simple iteration technique. We assume that reference objects can be reliably extracted and identified. The proposed method identifies the relationship between the landmarks on the image and the known global reference points by the parallel projection model. The parallel projection model calibrates the non-linearity of optical lens distortion without computational complexity. The coordinates and the orientations are estimated with minimum relation equations by the simple iteration method. Our method can be used in large area with artificial or natural references as long as their coordinates are known and they can be reliably identified. The possible error source of the self localization method is explained and analyzed in terms of the performance of the self localization.

The rest of this chapter is organized as follows. Section 2 discusses background of self localization and problem description. In Section 3, a parallel projection model and its basic concept are discussed. Section 4 proposes a self localization algorithm for determining the coordinates and the orientation from external reference points. In Section 5, we present an experiment and the analysis of simulation results with extensive analysis of the error effect of the measurement on the performance of the algorithm. Section 6 concludes the chapter.

## 2. Background and Problem Description

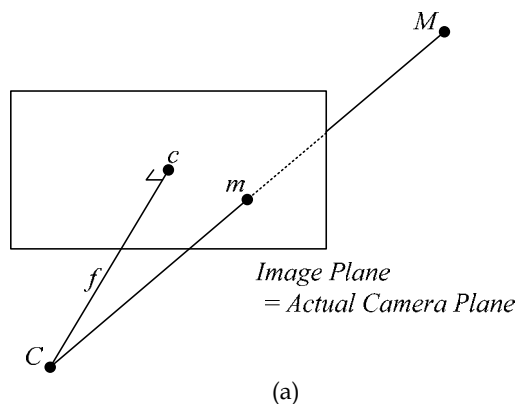
### 2.1 Related Work on Self-Localization

In general, self localization with visual information is related to photometry. (Yuan, 1989) presents a general method for determining the three-dimensional position and orientation of an object relative to a camera based on a two-dimensional image of known feature points located on the object. (Horaud et al., 1989) analytically deals with the perspective  $n$ -point ( $PnP$ ) problems with four correspondences of scene objects. Our approach does not analytically solve matrix transformation, but calculates the orientation and the location using a computationally efficient iterative algorithm.

A simple method for localization which allows a robot to determine its absolute position with a view of single landmark in one image is presented in (Dao et al., 2004). The landmark is chosen as the intersection of natural lines easily found in indoor environment such as edge of doors or walls. Another localization algorithm which is based on comparing the images taken in advance and taken during navigation is discussed in (Betke & Gurvits, 1997). In this scheme, the shape and the coordinate of images are stored in memory efficient format for quick retrieval and comparison. A similar method is presented where planar landmarks such as posters are used in visual localization of a mobile robot in indoor environment (Ayala et al., 2000). This algorithm has a restriction on the shape of landmark

and is not suitable in an open area. Similar to these approaches, our assumption requires a map of global coordinates of reference objects that can either be natural or artificial objects. Scale Invariant Feature Transform (SIFT) developed for image feature generation in object recognition application is used for robot localization in (Se et al., 2001). The invariant characteristics of SIFT are captured by three images and stereo-matched to elect landmark that is later used to compute 3-D world coordinates relative to the robot. This algorithm does not require modification of the environment or map of reference objects, but needs more than two cameras with expensive computation to compare the features in the image. In this chapter, we use a parallel projection model to simplify the computational complexity in determining the location and the orientation of a mobile robot. In the parallel projection model, a *virtual viewable plane* is defined to formulate the relationship between a real object and an actual camera. By using this model, the minimum of relation equations between the detected landmarks on the image and known corresponding reference points are obtained. This enables decrease of the computational complexity in finding the solution from all possible landmark pairs with the minimum mean square errors.

The parallel projection model is similar to the pinhole camera model in perspective projection model since the camera angle is relatively narrow and the size of the image obtained by the sensor is much smaller than real object area (Swaminathan et al., 2003). However, in the perspective projection model, the calibration process usually uses a flat plate with a regular pattern or the known several reference points (Lenz & Tsai, 1988; Heikkila & Silven, 1997; Lv et al., 2006). In addition, the calibration information should be updated for projection accuracy whenever the camera status is changed. This is not appropriate for highly dynamic application such as the robot navigation. On the other hand, the parallel projection model uses zoom factor instead of focal length and scale factor. This model simplifies the calibration process of the camera non-linear property by using the pre-obtained calibration table. Therefore, it is easily applied to the reference projection in the robot navigation. As a result, the relation equations are less affected by the camera model. This minimizes the self localization error due to the projection model. Fig.1 illustrates the difference of our parallel projection model and the perspective projection model (Lepetit & Fua, 2006) where  $d_1$  and  $d_2$  denote the distance of the camera and *Object Plane*. *Object Plane* contains the object and is parallel to *Actual Camera Plane*.



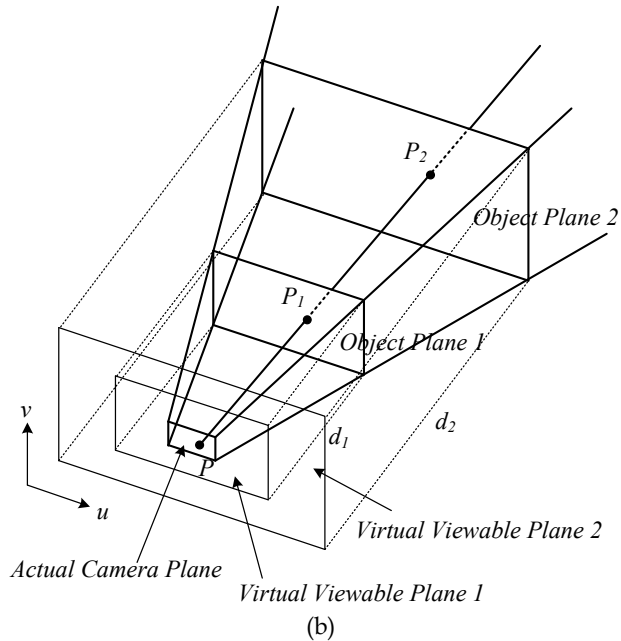


Fig. 1. Difference of perspective projection model and parallel projection model. (a) Perspective projection model. (b) Parallel projection model.

## 2.2 Problem Overview

In this chapter, we define the term “mobile sensor” to describe a mobile robot with visual sensor. We assume that mobile sensor can identify the reference objects and their coordinates are known (i.e., stored in a database). We limit our discussion to the self localization problem, and the method of how to identify such objects is not considered. The mobile sensor navigates by itself and visual image is obtained periodically. Based on the captured image data, the self localization comprises of determining both the orientation and the location of the mobile sensor. We use the global coordinate system, the origin of which is arbitrarily chosen but used to represent the coordinates of the reference points and the location of the mobile sensor. The objective is to utilize the projected reference points to determine the location of the mobile sensor. In this chapter, we focus on two aspects of the proposed method. The first aspect is to maintain the accuracy of the self localization and the second aspect is to maintain computational efficiency.

Since our method uses captured image data through the application of a typical digital imaging device, several sources of error are possible. Since the proposed approach relies on the point that is chosen from an area of pixels which is the projected image of the reference object, there can be inherent errors from image processing that selects one point from the area of an object image. This error can vary depending on many factors such as distance from mobile sensor to reference objects, distance between reference objects, etc. In addition, the non-linearity of the lens of imaging device causes shifting of projected point when the

distance to reference points is not known. This shifting also affects the fidelity of self localization if compensation is not done.

Since the mobile sensor changes its location and orientation continuously, the reference points may be changed accordingly. The self location method should be computationally efficient by effectively utilizing available reference points. As we will show later in this chapter, the selection of reference points affects the self localization errors. When more than three reference points are inside the viewable range of the mobile sensor at the same time, the mobile sensor has freedom to choose the reference objects in such a way that can minimize such errors. Therefore, multiple reference objects should be strategically distributed to harness self localization of individual mobile sensor. A computationally efficient iterative algorithm using the relationship between the locations of reference points is proposed.

### 3. Characterization of Viewable Images

#### 3.1 Basic Concept of Parallel Projection Model

In this subsection, we introduce the *parallel projection model*. In order to simplify the process of projected image on the camera device, we define three planes: the *object plane*, the *virtual viewable plane*, and the *actual camera plane* as shown in Fig.2. An object  $P$ , which is in the viewable area of a mobile sensor, is considered to be on the *object plane*. As opposed to the traditional model of a camera, in a parallel projection model, the object  $P$  is projected in parallel onto *virtual viewable plane* and the projected point is denoted as  $P_p$ . The *virtual viewable plane* is parallel to the *object plane* with distance  $d_p$ .  $L_c$  denotes the length of the *object plane*, which is the length of viewable area at distance  $d_p$ .  $L_s$  denotes the length of the *actual camera plane* on which the measurement of projected image is done. The position of the projected object on a *virtual viewable plane* and an *actual camera plane* is denoted as  $u_{pp}$  and  $u_p$ , respectively.

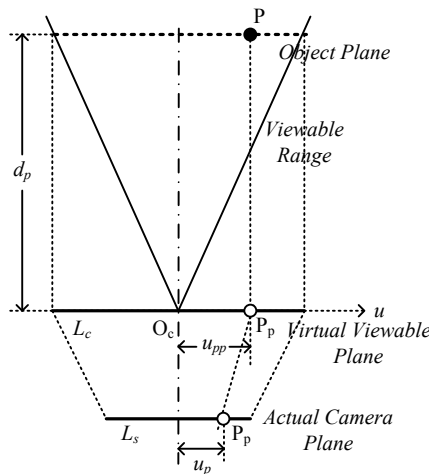


Fig. 2. Parallel projection model and relationship with actual image projected on visual sensor (Camera).

In the parallel projection model, a real object located on an *object plane* is projected to an *actual camera plane* through a *virtual viewable plane*. Hence, as formulated in (1),  $u_{pp}$  is expressed as  $L_c$ ,  $L_s$  and  $u_p$  using the proportionality of the length of a *virtual viewable plane* and an *actual camera plane*.

$$u_{pp} = u_p \left( \frac{L_c}{L_s} \right) \quad (1)$$

The position of the real object can be obtained from  $u_{pp}$  and  $d_p$ , once the ratio of  $L_c$  and  $d_p$  is known. This ratio is defined to be a zoom factor,  $z$ , which is the property of the image device.

$$z = \frac{d_p}{L_c} \quad (2)$$

### 3.2 Relationship of Reference Points on Different Planes

Given the parameters of visual sensor,  $z$  and  $L_s$ , we can derive the relationship between a projected reference point on the *virtual viewable plane* and one on the *actual camera plane* with their distance to the origin of each plane. The origin of each plane is defined to be the cross point between a plane and its perpendicular line, the view axis, which also crosses the location of the mobile sensor. Specifically, the origin of the *actual camera plane* is the axis of panning. In Fig. 3, the origin on the actual camera plane is denoted as  $O_c$  and the origins of the virtual viewable planes are denoted as  $O_{v1}$  and  $O_{v2}$ , respectively. Even though the planes are rotated as visual sensor is panned, the relationship based on the distance on each plane remains unchanged. When  $p_1$  and  $p_2$  denote the distance to the view axis on each virtual plane, and  $i_1$  and  $i_2$  denote the corresponding measurement on the *actual image plane*, using (1) and (2), we can derive

$$\begin{aligned} \frac{p_1}{i_1} &= \frac{L_{p1}}{L_s} = \frac{D_1}{z_1 L_s}, \\ \frac{p_2}{i_2} &= \frac{L_{p2}}{L_s} = \frac{D_2}{z_2 L_s}, \end{aligned} \quad (3)$$

where  $z_1$  and  $z_2$  are the zoom factors of the mobile sensor corresponding to distance,  $D_1$  and  $D_2$ , from the *actual camera plane* to the *object plane* for each reference point.  $L_s$  is the size of the image on which  $i_1$  and  $i_2$  are measured.

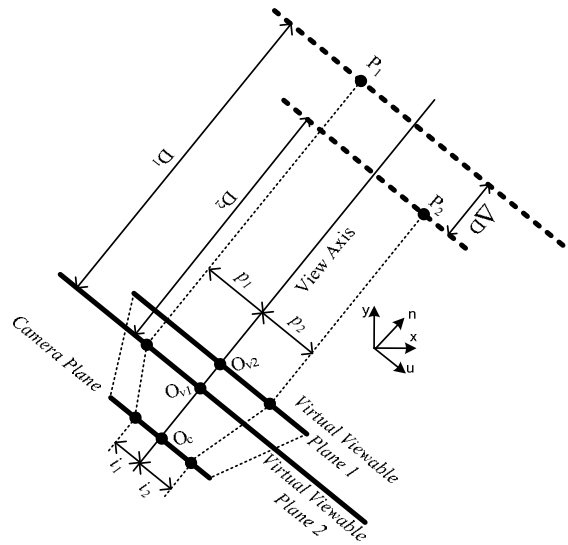


Fig. 3. Self localization with two reference points.

In practice, the location of a projected point on the image device is obtained from the image processing of the target objects such as edge detection and/or feature extraction. Thus, the projected point on the image device usually contains some uncertainty. In later sections, how this uncertainty affects self localization algorithm is discussed in detail.

### 4. Self Localization Algorithm

#### 4.1 Self Localization with Known Orientation

In this subsection, we introduce self localization when two reference points and the orientation of visual sensor are known. We define  $\theta$  as the angle formed between the camera plane and global  $x$  axis. We define a unit vector  $\hat{u}$  to have the direction of the camera plane and  $\hat{n}$  to be the unit vector along the view axis, the direction to which the visual sensor is facing. Therefore,  $\theta$  is the angle between the  $x$  axis and  $\hat{u}$ . Using the following equations, we can obtain values  $p_1, p_2, D_1$  and  $D_2$  in (3). In the previous section, we described  $i_1$  and  $i_2$  as the distance to the view axis on the camera plane, but, from now on, they are considered to be able to have negative values when the projected reference point is in the left side of the view axis. It does not change the distance relationship described in the previous section by allowing  $p_1$  and  $p_2$  to have negative values as well when they are also in the left side of the view axis.

For  $p_1$  and  $p_2$ ,

$$\begin{aligned} p_1 &= \overline{CP_1} \cdot \hat{u}, \\ p_2 &= \overline{CP_2} \cdot \hat{u}. \end{aligned} \tag{4}$$

For  $D_1$  and  $D_2$ ,

$$\begin{aligned} D_1 &= \overline{CP_1} \cdot \hat{n}, \\ D_2 &= \overline{CP_2} \cdot \hat{n}. \end{aligned} \quad (5)$$

Here,  $\overline{CP_1}$  and  $\overline{CP_2}$  denote the vectors from the location of the mobile sensor,  $O_c$ , to each reference point. Above the two sets of equations are simply the decomposition of  $\overline{CP_1}$  and  $\overline{CP_2}$  to the  $\hat{u}$  and  $\hat{n}$  components.

Then, from (3), we have

$$\begin{aligned} \frac{p_1}{i_1} &= \frac{\overline{CP_1} \cdot \hat{u}}{i_1} = \frac{\overline{CP_1} \cdot \hat{n}}{zL_s}, \\ \frac{p_2}{i_2} &= \frac{\overline{CP_2} \cdot \hat{u}}{i_2} = \frac{\overline{CP_2} \cdot \hat{n}}{zL_s}. \end{aligned} \quad (6)$$

or

$$\begin{aligned} \overline{CP_1} \cdot \left( \hat{u} - \frac{i_1}{zL_s} \hat{n} \right) &= 0, \\ \overline{CP_2} \cdot \left( \hat{u} - \frac{i_2}{zL_s} \hat{n} \right) &= 0. \end{aligned} \quad (7)$$

In (6),  $\overline{CP_1}$  and  $\overline{CP_2}$  can be expressed with their  $x$  and  $y$  components in global  $x$ - $y$  coordinate system as

$$\begin{aligned} \overline{CP_1} &= (P_{1x} - x_c) \hat{x} + (P_{1y} - y_c) \hat{y}, \\ \overline{CP_2} &= (P_{2x} - x_c) \hat{x} + (P_{2y} - y_c) \hat{y}, \end{aligned} \quad (8)$$

where  $P_{1x}$  and  $P_{2x}$  are the  $x$  components of  $P_1$  and  $P_2$ , respectively, and  $P_{1y}$  and  $P_{2y}$  are the  $y$  components of  $P_1$  and  $P_2$ , respectively.

The  $x$  and  $y$  components of the dot products are expressed by



$$\begin{aligned}
 (\hat{u} - \frac{i_1}{zL_s} \hat{n}) \cdot \hat{x} &= \cos \theta + \frac{i_1}{zL_s} \sin \theta, \\
 (\hat{u} - \frac{i_1}{zL_s} \hat{n}) \cdot \hat{y} &= \sin \theta - \frac{i_1}{zL_s} \cos \theta, \\
 (\hat{u} - \frac{i_2}{zL_s} \hat{n}) \cdot \hat{x} &= \cos \theta + \frac{i_2}{zL_s} \sin \theta, \\
 (\hat{u} - \frac{i_2}{zL_s} \hat{n}) \cdot \hat{y} &= \sin \theta - \frac{i_2}{zL_s} \cos \theta.
 \end{aligned}
 \tag{9}$$

Then, (7) are equivalent to

$$\begin{aligned}
 (P_{1x} - x_c)(\cos \theta + \frac{i_1}{z_1L_s} \sin \theta) + (P_{1y} - y_c)(\sin \theta - \frac{i_1}{z_1L_s} \cos \theta) &= 0, \\
 (P_{2x} - x_c)(\cos \theta + \frac{i_2}{z_2L_s} \sin \theta) + (P_{2y} - y_c)(\sin \theta - \frac{i_2}{z_2L_s} \cos \theta) &= 0.
 \end{aligned}$$

Let us introduce intermediate variables to simplify the final equations for  $x_c$  and  $y_c$ . They are

$$\begin{aligned}
 \alpha_1 &= \cos \theta + \frac{i_1 \sin \theta}{zL_s}, \\
 \beta_1 &= \sin \theta - \frac{i_1 \cos \theta}{zL_s}, \\
 \alpha_2 &= \cos \theta + \frac{i_2 \sin \theta}{zL_s}, \\
 \beta_2 &= \sin \theta - \frac{i_2 \cos \theta}{zL_s}.
 \end{aligned}$$

Thus, we can obtain the coordinate of mobile sensor expressed as

$$\begin{aligned}
 x_c &= -\frac{\alpha_1\beta_2P_{1x} + \beta_1\beta_2P_{1y} - \beta_1\alpha_2P_{2x} - \beta_1\beta_2P_{2y}}{\beta_1\alpha_2 - \alpha_1\beta_2}, \\
 y_c &= -\frac{\alpha_1\alpha_2P_{1x} - \alpha_2\beta_1P_{1y} + \alpha_1\alpha_2P_{2x} + \alpha_1\beta_2P_{2y}}{\beta_1\alpha_2 - \alpha_1\beta_2}.
 \end{aligned}
 \tag{10}$$

Since the reference object is projected onto the camera plane, the coordinates of reference objects correspond to the centroid of the reference objects. Then, we can obtain the coordinate of mobile sensor using (10). However, even though the coordinates of reference points are accurately known in advance, the measurement  $i_1$  and  $i_2$  on the image may not be corresponding to the true reference points. Possible sources of the uncertainties may arise from the pixel resolution of the image planes as well as incorrect determination of the centroid of the detected reference shape. This uncertainty is evident even with perfect lens view characteristics. We will introduce the effect of non-linearity of camera lens in the later section.

#### 4.2 Orientation Determination

Thus far, we have considered determining the position of the mobile sensor when its orientation is given. However, it is necessary to determine the orientation of the mobile sensor as well as its position. Determining both position and orientation concurrently requires a third reference point. From the parallel projection model, using (3), we can obtain the angle of the line that crosses the center of the camera plane and the reference point, where the angle is formed between the line and the camera plane. With two reference points, we have two lines with known angle respect to the camera plane, and we know each reference point is on one of them. Since there are infinite numbers of ways to position a line segment having two reference points as vertexes sitting on those lines, we cannot determine the position and the orientation of the mobile sensor with two reference points. With one more reference point, the problem becomes to position three vertexes of a triangle with known length onto three lines with a known angle. There is only one way to position the triangle in such way if we limit the orientation of the mobile sensor to  $180^\circ$  range. From the above, we can conclude that three reference points are enough for determining both the orientation and the location of the mobile sensor when the general directions of the reference points are assumed in the following discussion.

We can find a solution by solving three simultaneous solutions using (10), but its non-linearity requires large computational complexity to be implemented on resource limited devices, such as mobile robot. Instead, we developed an effective iteration algorithm which involves solving only two simultaneous equations and the solution is given in (10). In our iteration approach, we determine the orientation of the mobile sensor. Once we found the orientation, we obtain the location of the mobile sensor using (10).

For a given orientation angle,  $\theta$ , using (10), we can obtain two sets of coordinates,  $(x_{c1}, y_{c1})$  and  $(x_{c2}, y_{c2})$  using two pairs of reference points out of three. When three reference points,  $P_1$ ,  $P_2$  and  $P_3$  are chosen for self-localization, using  $P_1$  and  $P_2$ , we have

$$\begin{aligned} x_{c1} &= -\frac{\alpha_1\beta_2P_{1x} + \beta_1\beta_2P_{1y} - \beta_1\alpha_2P_{2x} - \beta_1\beta_2P_{2y}}{\beta_1\alpha_2 - \alpha_1\beta_2}, \\ y_{c1} &= -\frac{\alpha_1\beta_2P_{1x} + \beta_1\beta_2P_{1y} - \beta_1\alpha_2P_{2x} - \beta_1\beta_2P_{2y}}{\beta_1\alpha_2 - \alpha_1\beta_2}, \end{aligned} \quad (11)$$

and by using another pair,  $P_2$  and  $P_3$ , we have

$$\begin{aligned}
 x_{c2} &= -\frac{\alpha_2\beta_3P_{2x} + \beta_2\beta_3P_{2y} - \beta_2\alpha_3P_{3x} - \beta_2\beta_3P_{3y}}{\beta_2\alpha_3 - \alpha_2\beta_3}, \\
 y_{c2} &= -\frac{\alpha_2\beta_3P_{2x} + \beta_2\beta_3P_{2y} - \beta_2\alpha_3P_{3x} - \beta_2\beta_3P_{3y}}{\beta_2\alpha_3 - \alpha_2\beta_3}.
 \end{aligned}
 \tag{12}$$

In order to develop an effective iterative strategy, we investigate the behavior of the differences of the two coordinates,  $d_{cx} = x_{c1} - x_{c2}$  and  $d_{cy} = y_{c1} - y_{c2}$  while varying the angle of orientation. We define *error\_distance* as

$$error\_distance(\theta) = \sqrt{(x1 - x2)^2 + (y1 - y2)^2}
 \tag{13}$$

where  $\theta$  is the angle of the mobile sensor's orientation. Fig. 4 shows the behavior of this function, as  $\theta$  varies from  $0^\circ$  to  $180^\circ$ . The figure shows a case when the true orientation of the mobile sensor is  $80^\circ$ , and, at this angle, *error\_distance*( $\theta$ ) becomes zero. We call this angle *solution point*. Around this point, the function is found to be symmetric and periodical with  $180^\circ$ . Based on the characteristics of the simulated distance error function, there is always one global minimum between  $0^\circ$  to  $180^\circ$ . Thus, there will be two possible orientations between  $0^\circ$  to  $360^\circ$ . The good initial point is approximated by the direction which a robot moves toward.

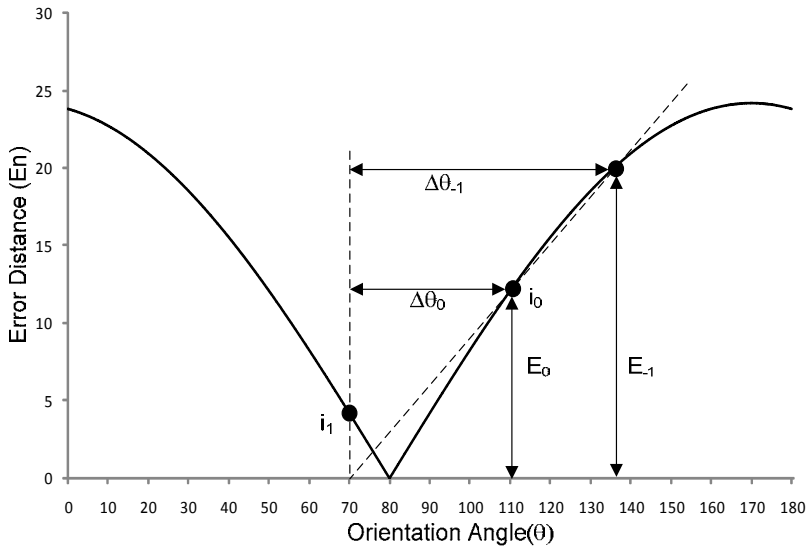


Fig. 4. Distance error function as a function of orientation error. The slope estimations of initial iteration points are shown.

If we start iteration inside  $45^\circ$  range from *solution point*, and if we follow down the slope, it is guaranteed to find the solution. In order to find such an initial iteration point,  $i_0$ , inside the range, we arbitrarily choose two angles separated with  $90^\circ$ . Since one of them will be inside  $45^\circ$  range from the *solution point* and the other one will be outside, we simply choose the angle that gives smaller  $error\_distance(\theta)$  as our initial iteration angle,  $\theta_0$ .

Once we choose an initial point, we have the initial iteration point,  $i_0$ , determined by  $\theta$  and  $error\_distance(\theta)$ . In order to estimate the slope at that point, another  $error\_distance$  function is evaluated using  $\theta_{-1}$  which is chosen to be very close to  $\theta_0$  such as  $0^\circ$ . We call this estimated slope as  $slope_0$  and the relationship of the initial iteration variables are

$$\begin{aligned}\Delta\theta_0 &= \theta_0 - \theta_{-1}, \\ \Delta E_0 &= E_0 - E_{-1}, \\ slope_0 &= \frac{\Delta E_0}{\Delta\theta_0},\end{aligned}\tag{14}$$

where  $E_n = error\_distance(\theta_n)$ .

Depending on the sign of the estimated slope, we choose the direction of the iteration,  $dir$ . If  $slope_0 > 0$ , we set  $dir_0 = -1$ , and, swap  $\theta_0$  with  $\theta_{-1}$ , and,  $E_0$  with  $E_{-1}$ . Otherwise,  $dir_0 = 1$ .

First, by assuming the slope at our initial point being close to be linear, we choose the next angle where the slope line crosses  $x$ -axis. Since the slope increases as approaching to the *solution point*, the next iteration step will overshoot albeit very close to *the solution point*. As shown in Fig. 4, the  $error\_distance$  function evaluated at  $\theta_1$  is the other side of the *solution point*.

From the second step, instead of using the slope, we choose the angle of next iteration step based on the two previous angle and  $error\_distance$  evaluated with them. In this case, since the two triangles shown in Fig. 5 are approximately proportional near to the solution point, the angle for the next step is evaluated by the following equations. From

$$\frac{\Delta\theta_n - \Delta\theta_{n+1}}{E_{n-1}} = \frac{\Delta\theta_{n+1}}{E_n},\tag{15}$$

the next iteration angle is calculated as

$$\Delta\theta_{n+1} = dir \times \frac{E_n}{E_n + E_{n-1}} \times \Delta\theta_n.\tag{16}$$

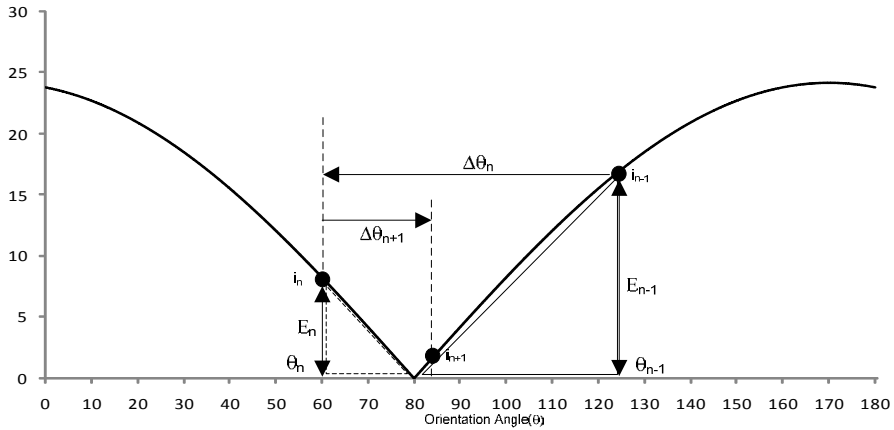


Fig. 5. Convergence steps of the iteration algorithm.

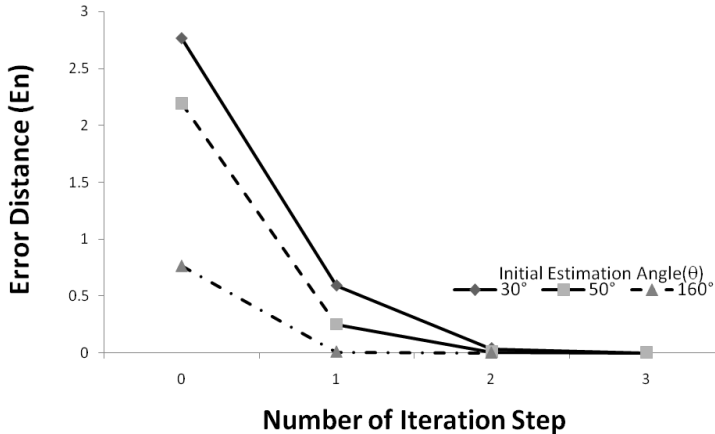


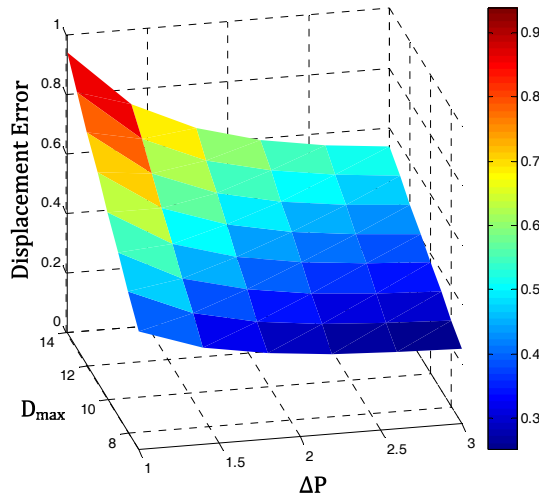
Fig. 6. Convergence of the iteration algorithm as a function of the number of iterations.

The iteration continues until, the change of estimated orientation,  $\Delta\theta_n$ , becomes smaller than the threshold value,  $\epsilon$ . Otherwise, we change the direction,  $dir = dir \times (-1)$  and continue. Fig. 6 shows that the algorithm is converging very rapidly. The figure shows the iteration algorithm is applied when three initial estimation angles are used,  $10^\circ$ ,  $40^\circ$  and  $80^\circ$ . The value of  $error\_distance(\theta)$  is plotted at each iteration step. Note that the iteration starts with two initial iterations (i.e., as shown in the figure, the iteration starts at  $-1$  index).

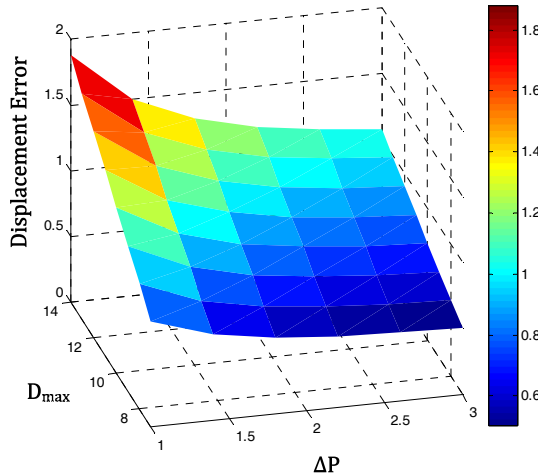
Fig. 7 illustrates the importance of the orientation error on the localization. The plot shows the displacement error for several orientation errors. Throughout this chapter, the displacement error is defined as

$$\sqrt{(x_{c,true} - x_{c,est})^2 + (y_{c,true} - x_{y,est})^2}, \tag{18}$$

where  $(x_{c,true}, y_{c,true})$  is the true coordinate and  $(x_{c,est}, y_{c,est})$  is the estimated coordinate. The results are plotted as a function of  $\Delta P$  and  $D_{max}$  where  $\Delta P$  represents the separation (in parallel to the projected plane) between the reference points and the  $D_{max}$  represents the largest distance (perpendicular to the projected plane) of the reference points. The angle determination is very critical since the displacement is computed after the orientation is determined. Thus, if the orientation computation is not accurate, the localization may not successfully estimate the coordinates.



(a)



(b)

Fig. 7. Displacement error as a function of the orientation error  $\Delta\theta$ . (a) Orientation error,  $\Delta\theta = 2$ . (b) Orientation error,  $\Delta\theta = 4$ .

### 4.3 Lens Distortion

The non-linear distortion of non-ideal lens affects the scale in the parallel projection model. Since the distances between the mobile sensor and the references are not known, we compute the coordinate of the mobile sensor using the value of  $z$  corresponding to the value when the distance is large (i.e., the value of  $z$  converges to a specific value). Once initial value of the coordinate is obtained, we use specific values of  $z$  (i.e., the one for the first reference and the other for the second reference) to compensate for the non-linearity to obtain the final coordinate of the mobile sensor. Note that the zoom factor depends on the distance from the imaging device, as well as the distance from the axis of the lens. The calibration table for the zoom factor is constructed according to the distance from the imaging device and the distance from the axis of the lens at the interval of 0.3m (Park et al., 2008).

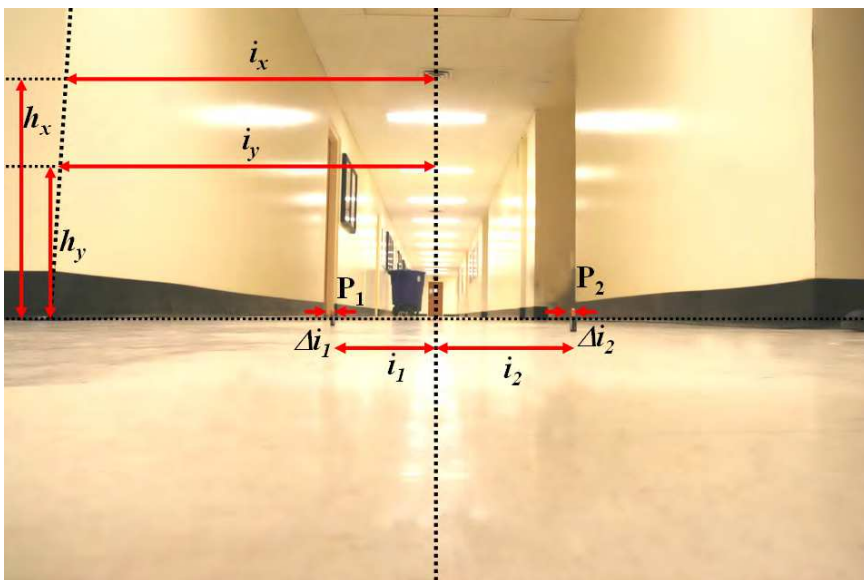


Fig. 8. Nonlinear effect of the lens on estimation error.

Fig. 8 illustrates the nonlinear effect on the coordinate determination. In the figure, the orientation is chosen to be 0 (i.e., view axis is perpendicular to the  $x$  axis). The camera is located at (4.8m, 3.0m) with respect to the global coordinate (0, 0) and the first reference  $P_1$  is located at (4.5m, 4.5m) and the second reference  $P_2$  is located at (5.1m, 5.1m). The lens is set at 17mm zoom range. The value of  $L_s$  is 18.4cm (i.e., the image size in  $x$  direction). The projected position of the first reference  $i_1$  is at 3.85cm and the projected position of the second reference  $i_2$  is at 2.45cm from the center of the image. These positions are from the center of the reference objects to the center of the image. The reference objects both have finite widths of 0.16cm and 0.12cm corresponding to  $\Delta i_1 = 0.0087$  and  $\Delta i_2 = 0.0065$ , respectively. In this chapter,  $\Delta i$  is defined as the uncertainty range or the measurement error with the respect to the overall image size (i.e., 18.4cm in the example). Since the centroid of the reference points are determined from the image, potential measurement errors will be

within  $\Delta i$ . The actual zoom factor corresponding to the first reference  $z_1$  is 0.8238 and the zoom factor corresponding to the second reference  $z_2$  is 0.8119. In the initial estimation, the zoom factor corresponding to the infinite distance of 0.8 is used. The estimated coordinate without compensation is (4.8017m, 3.03815m) which is 3.87cm off from the true position of the mobile sensor.

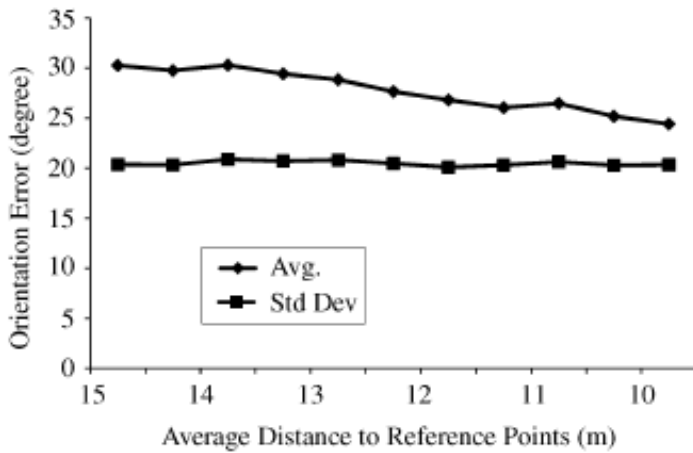
#### 4.4 Effects of Reference Measurement Uncertainty

The measurement error directly affects the accuracy of the localization including the orientation. Since the reference object is usually projected as an area on the image, for applying the parallel projection model, one point should be determined from the area. In the parallel projection model, we only take the horizontal component of the determined point. If we designate the coordinate of the reference object as its centroid, we can choose its position on the camera plane,  $i$ , as the center of the projected area. However, if the shape of reference object is not symmetrically round, there is always certain amount of error in determining  $i$ . This type of error is usually influenced by the image processing in finding the boundary of the projected references. Quantization error due to limited resolution of the visual sensor may affect the accuracy but it is not the biggest source of error.

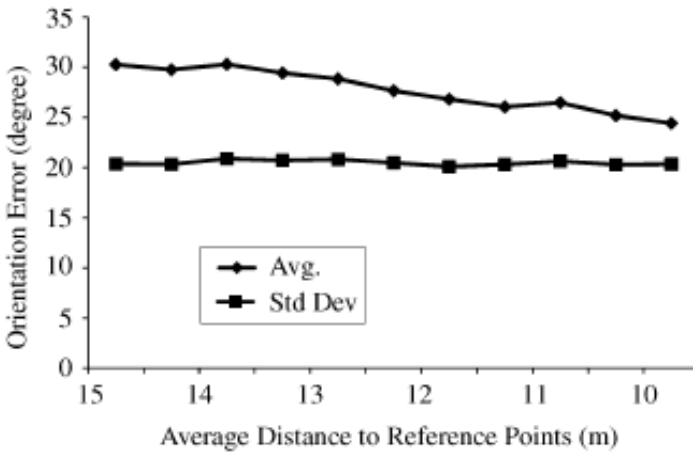
Another source of measurement error is that the reference object is not projected in the center of the horizontal line (i.e., illustrated as dotted line in Fig. 8). This is because the lens distorts the projected objects. In the figure, the edge of a wall has one coordinate value. However, multiple reference values can be obtained for the edge of a wall. For example, both  $i_x$  measured at  $h_x$  and  $i_y$  measured at  $h_y$  should represent the same coordinate, but the projected values are different. The difference between  $i_x$  and  $i_y$  contributes as  $\Delta i$  in the localization. However, (Park et al., 2008) shows that the projection error is compensated by using the parallel projection model if the height of a landmark is known. Since the height of a landmark can be known, the localization performance is not worsened in this case.

The orientation errors due to the incorrect determination of  $i_1$  and  $i_2$  are illustrated in Fig. 9. Figs. 9(a) and 9 (b) show the results for two different values of the measurement errors. The average of orientation error is measured using all the possible combinations of the reference points located on 5m by 5m grid. Each reference point is located at 50cm interval in the grid with a small amount of additional random variation. Due to the variation of  $i_1$  and  $i_2$ , the estimated orientation can be different from the true orientation of a mobile sensor. The figures show that overall range of error and standard deviation are larger when  $\Delta i = 0.03$  than when  $\Delta i = 0.05$ . Also, when the mobile sensor is closer to the reference points, the orientation error is very critical since the actual coordinates are obtained by computing the orientation first.





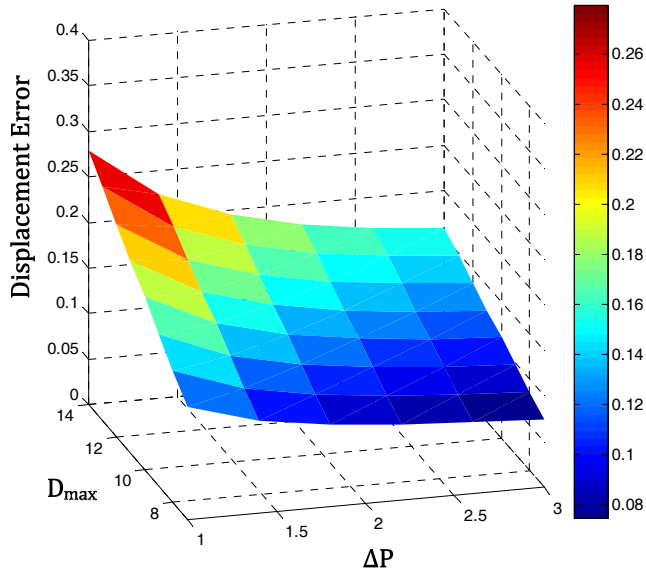
(a)



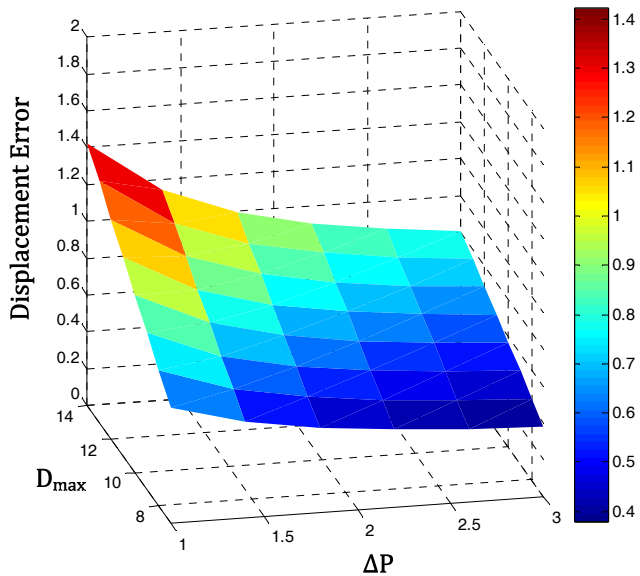
(b)

Fig. 9. Orientation error,  $\Delta\theta$ , as a function of the projected measurement error  $\Delta i$ . (a) Orientation error,  $\Delta i = 0.03$ . (b) Orientation error,  $\Delta i = 0.005$ .

The similar results are obtained for the displacement errors in Fig. 10. Similarly, Figs. 10(a) and 10(b) show the results for two different values of the measurement errors. The displacement error is plotted as a function of two variables,  $D_{max}$ , the largest distance from the mobile sensor to the reference points, and  $\Delta P$ , the distance between two most separated reference points measured from the *view axis* of the mobile sensor. As the figure shows, the overall error range increases as  $\Delta i$  increases.



(a)



(b)

Fig. 10. Displacement error as a function of projected measurement error  $\Delta i$ . (a) Measurement error,  $\Delta i = 0.01$ . (b) Measurement error,  $\Delta i = 0.05$ .

Both results show that the algorithm is more prone to error when distance from the mobile sensor to the reference points is larger, and when the references are closer to one another. From Fig. 9 and Fig. 10, we know that estimation error is smaller when the distance between the reference objects along the camera plane is larger. Since our iteration algorithm uses two pairs of reference objects out of three pairs that can be made from three reference objects, given three reference points,  $R_1$ ,  $R_2$  and  $R_3$ , we can choose two pairs that give maximum distance on the camera plane to minimize error. This selection criterion can be applied also when there are more than three reference objects viewable and three of them need to be selected for self localization.

### 5. Analysis and Simulation

#### 5.1 Experimental Setup

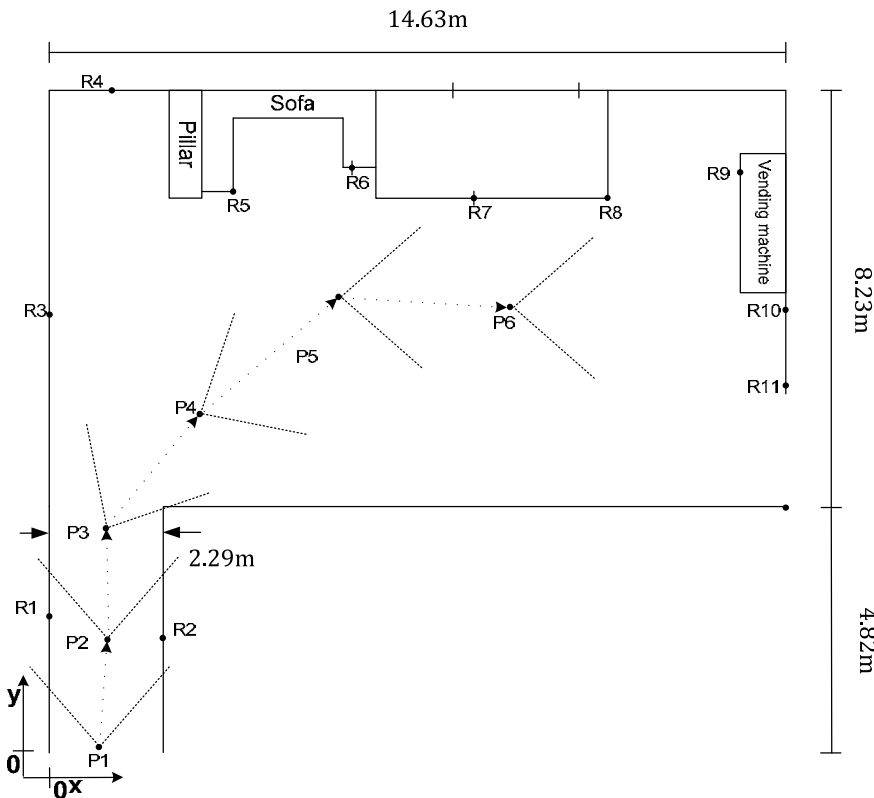
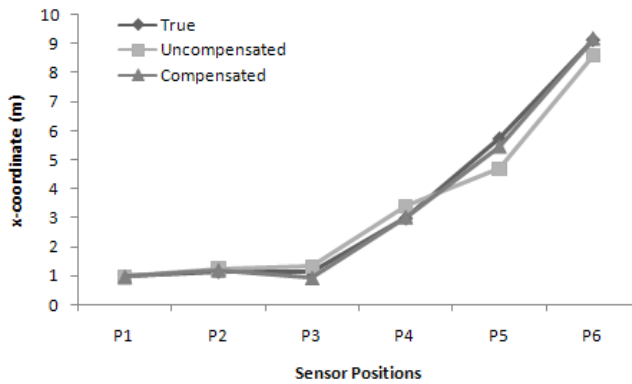


Fig. 11. Experimental setup used in the self localization illustration. 10 reference points are used by the mobile sensor located at 6 distinct coordinates.

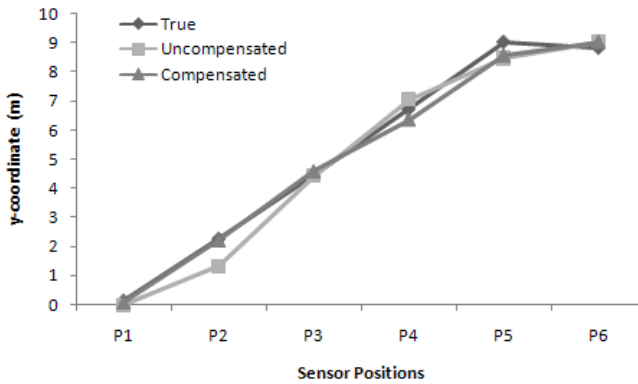
Fig. 11 shows the experimental setup for the verification of the proposed method. A mobile sensor is placed at several positions indicated by  $P_1, P_2, \dots, P_6$  with  $P_1$  as the starting

position. Several reference points,  $R_1, R_2, \dots, R_{11}$ , are designated and measured beforehand. At each position, the mobile sensor estimates the position and orientation before moving on to the next position. Navigation application utilizing our method can employ two strategies as to when to run the algorithm. In the first case, the mobile sensor can move until three reference points are in its sight before running the algorithm. In the other case, the mobile sensor can move to a designated position directed by a navigation algorithm, and search for three reference points by rotating before running the localization algorithm. Three reference points are extracted using edge detection and color matching. We evaluate two cases. The first case assumes the orientation is known and the displacement errors are obtained. In the second case, the orientation is computed first, and the displacement errors are obtained from the orientation.

### 5.2 Localization Performance with Known Orientation



(a)

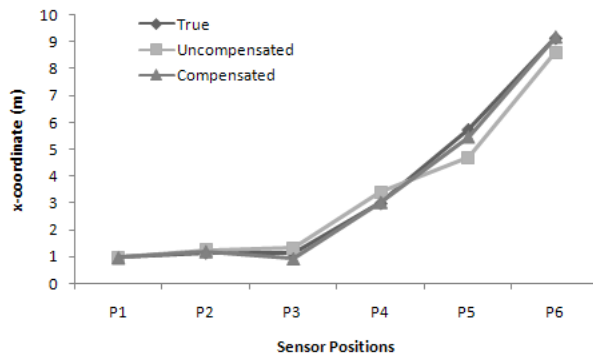


(b)

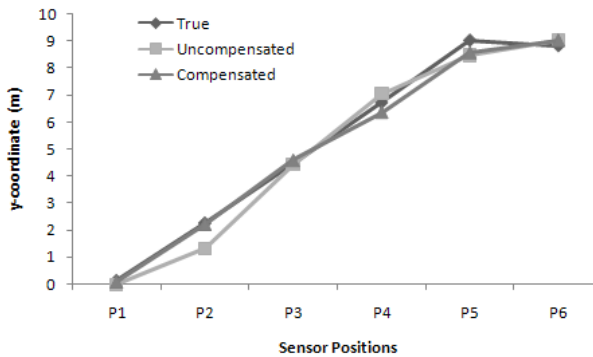
Fig. 12. Displacement error at each mobile sensor location with known orientation. Both compensated and uncompensated coordinates are compared to those of true coordinates. (a)  $x$ - direction. (b)  $y$ -direction.

We first assume that mobile visual sensor knows its orientation. Figs. 12(a) and 12(b) show the true position of the mobile sensor, in  $x$ -direction and  $y$ -direction separately, and their estimated trajectory obtained from using the algorithm. The deviation from the true position is shown as the distance error from the true position to the estimated position of the mobile sensor. Two estimated positions are plotted to show the effect of Zoom factor compensation. For the uncompensated estimation, the average value of the zoom factors is used. While displacement error in  $x$ -direction as shown in Fig. 12(a) is negligible, the displacement error in  $y$ -direction as shown in Fig. 12(b) illustrates that the uncompensated estimation deviates from the true position as much as 0.5m. It indicates the zooming factor is very sensitive to the distance from the visual sensor to the reference points. It is because the zoom factor has non-linear property only along the  $y$ -direction or the distance from the mobile sensor to the reference objects. However, when the zoom factors are compensated for within the algorithm, the distance error in  $y$ -direction disappears.

### 5.3 Localization Performance with Unknown Orientation



(a)

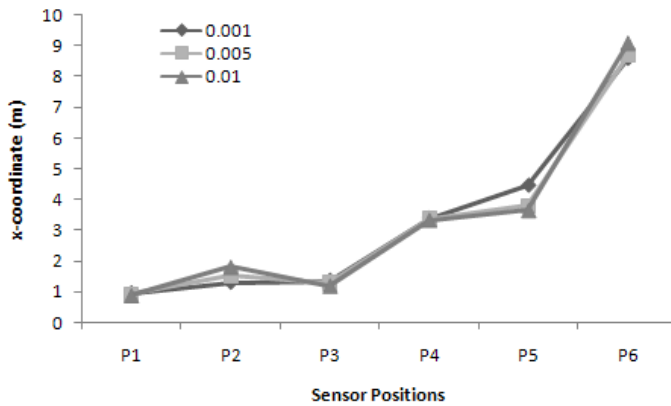


(b)

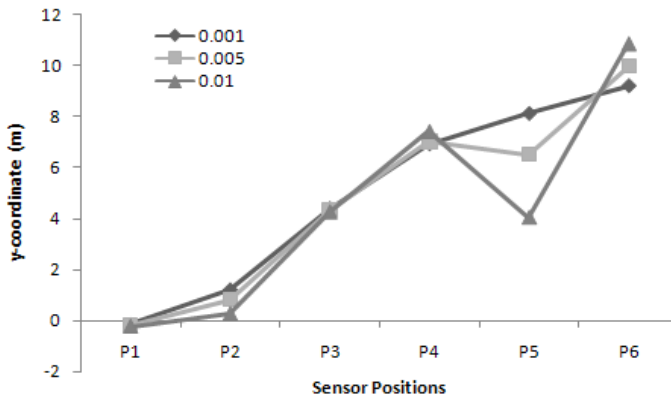
Fig. 13. Displacement error at each mobile sensor location with unknown orientation. Both compensated and uncompensated coordinates are compared to those of true coordinates. (a)  $x$ -direction. (b)  $y$ -direction.

In practice, the mobile sensor estimates the orientation as well as the coordinates. The proposed localization algorithm can determine the orientation of the sensor from the reference points. Since the coordinate of the mobile sensor is determined from the estimated orientation, the minimization of the error in estimating the orientation is very critical.

Fig.13(a) and Fig.13(b) illustrate the displacement error in  $x$ -direction and  $y$ -direction, respectively. As before, the displacement error in  $x$ -direction is negligible even without the compensation. Similar result is also shown in  $y$ -direction. The difference between these two figures and the previous figures obtained for the known orientation is that the displacement error is computed after the orientation is determined.



(a)



(b)

Fig. 14. Displacement error for different measurement errors for unknown orientation. The displacement error is obtained after the orientation estimation with the measurement errors. (a)  $x$ - direction. (b)  $y$ -direction.

Fig. 14 shows the effect of  $\Delta i$  on the error of coordinates of the mobile sensor. For this simulation, we used maximal separation criterion for reference objects selection. As shown in the figure, when the mobile sensor is farther from the reference objects, the coordinate error is more sensitive to  $\Delta i$ . In our experimental setup, the mobile sensor is the closest to its selected reference points at the position  $P_4$ . In the figure, at  $P_4$ , the coordinate error is the least sensitive to  $\Delta i$ . When the mobile sensor at  $P_5$ , where its distance to the reference objects is the farthest, the coordinate error is very sensitive to  $\Delta i$ . Especially, the  $y$ -direction error at  $P_5$ , in Fig. 14(b) shows the large sensitivity to  $\Delta i$  of the coordinate error in  $y$ -direction. It is because the captured image does not contain any depth information, the variation to  $i$  can be mapped to large range of location of the mobile sensor in  $y$ -direction. 0.01 as  $\Delta i$  value, is unrealistic considering the real dimension of the image.

## 6. Conclusion

In this chapter, we present a novel self localization method using parallel projection model for mobile sensor in navigation applications. The algorithm estimates the coordinate and the orientation of mobile sensor using projected references on single visual image. The camera lens non-linearity is compensated for using lens specific calibration table. The method utilizes a simple iterative algorithm, which is very accurate with low computational demand. We identify various sources of measurement error that affect the estimation accuracy. We demonstrate with an example that the algorithm can be utilized in robot navigation as well as positioning application where accurate self localization is necessary.

## 7. References

- Ayala, V.; Hayet, J. B.; Lerasle, F.; *et al.* (2000). Visual localization of a mobile robot in indoor environments using planar landmarks, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.275-280, 0-7803-6348-5, Takamatsu, Japan, Oct. 31-Nov. 5, 2000.
- Betke, M. & Gurvits, L. (1997). Mobile robot localization using landmarks, *Transactions on Robotics and Automation*, Vol. 13, No. 2, 251-263, 1042-296X.
- Borenstein, J.; Everett, B. & Feng, L. (1996a). *Navigating Mobile Robots: Systems and Techniques*, AK Peters, 1-5688-1058-X, MA.
- Borenstein, J. & Feng, L. (1996b). Measurement and correction of systematic odometry errors in mobile robots, *IEEE Transactions on Robotics and Automation*, Vol. 12, No. 6, 869-880, 1042-296X.
- Borenstein, J.; Everett, H.; Feng, L. *et al.* (1997). Mobile robot positioning: Sensors and techniques, *Journal of Robotic Systems*, Vol. 14, No.4, 231-249.
- Briggs, A.; Scharstein, D.; Braziunas, D. *et al.* (2000). Mobile robot navigation using self-similar landmarks, *Proceedings of IEEE International Conference on Robotics and Automation*, pp.1428-1434, 0-7803-5886-4, San Francisco, USA, April 24-28, 2000.
- Cohen, C. & Koss, F. V. (1993). A comprehensive study of three object triangulation, *Proceedings of SPIE Conference on Mobile Robots*, pp.95-106, Boston, USA, May 4, 1993.

- Dao, N. X.; You, B. J.; Oh, S. R. *et al.* (2004). Simple visual self-localization for indoor mobile robots using single video camera, *Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp.3767-3772, 0-7803-8463-6, Sendai, Japan, Sept. 28-Oct. 2, 2004.
- Demirli, K. & Molhim, M. (2004). Fuzzy dynamic localization for mobile robots, *Fuzzy Sets and Systems*, Vol. 144, No. 2, 251-283.
- Großmann, A. & Poli, R. (2001). Robust mobile robot localisation from sparse and noisy proximity readings using Hough transform and probability grids abstract, *Robotics and Autonomous Systems*, Vol. 37, No. 1, 1-18.
- Hayet, J. B.; Lerasle, F. & Devy, M. (2007). A visual landmark framework for mobile robot navigation, *Image and Vision Computing*, Vol. 25, No. 8, 1341-1351.
- Heikkila, J. & Silven, O. (1997). A four-step camera calibration procedure with implicit image correlation, *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp.1106-1112, San Juan, Puerto Rico, June 17-19, 1997.
- Horaud, R.; Conio, B.; Leboulloux, O.; *et al.* (1989). An analytic solution for the perspective 4-point problem. *Computer Vision, Graphics, and Image Processing*, Vol. 47, No. 1, 33-44, 0734-189X.
- Jang, G.; Kim, S.; Lee, W. *et al.* (2002). Color landmark-based self-localization for indoor mobile robots, *Proceedings of IEEE International Conference on Robotics and Automation*, pp. 1037-1042, 0-7803-7272-7, Washington, USA, May 11-15, 2002.
- Lenz, R. & Tsai, R. (1988). Techniques for calibration of the scale factor and image center for high accuracy 3-D machine vision metrology, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 5, 713-720, 0162-8828.
- Lepetit, V. & Fua, P. (2006). Monocular model-based 3D tracking of rigid objects: A survey. *Foundations and Trends in Computer Graphics and Vision*, Vol. 1, No. 1, 1-89.
- Liu, W. & Zhou, Y. (2007). Robot self-localization based on a single image of identified landmarks, *Proceedings of IEEE International Symposium on Computational Intelligence in Robotics and Automation*, pp.248-253, 1-4244-0790-7, Jacksonville, USA, June 20-23, 2007.
- Lv, F.; Zhao, T. & Nevatia, R. (2006). Camera calibration from video of a walking human, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 28, No. 9, 1513-1518, 1051-4651.
- Martinelli, A.; Tomatis, N. & Siegwart, R. (2007). Simultaneous localization and odometry self calibration for mobile robot, *Autonomous Robots*, Vol. 22, No. 1, 75-85, 0929-5593.
- Park, K.; Lee, J.; Stanačević, M. *et al.* (2008). Iterative object localization algorithm using visual images with a reference coordinate, *EURASIP Journal on Image and Video Processing*, Article ID 256896.
- Se, S.; Owe D. & Little, J. (2001). Vision-based Mobile robot localization and mapping using scale-invariant features, *Proceedings of IEEE International Conference on Robotics and Automation*, pp.2051-2058, 0-7803-6576-3, Seoul, Korea, May 21-26, 2001.
- Se, S.; Lowe, D. G. & Little, J. (2002). Global localization using distinctive visual features, *Proceedings of International Conference on Intelligent Robots and Systems*, pp. 226-231, Switzerland, Sept. 30-Oct. 4, 2002.



- Sutherland, K. T. & Thompson, W. B. (1993). Inexact navigation, *Proceedings of IEEE International Conference on Robotics and Automation*, pp.1-7, 0-8186-3450-2, Atlanta, USA, May 2-6, 1993.
- Swaminathan, R.; Grossberg, M. D. & Nayar, S. K. (2003). A perspective on distortions, *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pp.594-602, 0-7695-1900-8, Madison, Wisconsin, USA, June 18-20, 2003.
- Thrun, S. (1998). Finding landmarks for mobile robot navigation, *Proceedings of IEEE International Conference on Robotics and Automation*, pp.958-963, 0-7803-4300-X, Leuven, Belgium, May 16-20, 1998.
- Yuan, J. (1989). A general photogrammetric method for determining object position and orientation, *IEEE Transactions on Robotics and Automation*, Vol. 5, No. 2, 129-142, 1042-296X.





## **Mobile Robots Navigation**

Edited by Alejandra Barrera

ISBN 978-953-307-076-6

Hard cover, 666 pages

**Publisher** InTech

**Published online** 01, March, 2010

**Published in print edition** March, 2010

Mobile robots navigation includes different interrelated activities: (i) perception, as obtaining and interpreting sensory information; (ii) exploration, as the strategy that guides the robot to select the next direction to go; (iii) mapping, involving the construction of a spatial representation by using the sensory information perceived; (iv) localization, as the strategy to estimate the robot position within the spatial map; (v) path planning, as the strategy to find a path towards a goal location being optimal or not; and (vi) path execution, where motor actions are determined and adapted to environmental changes. The book addresses those activities by integrating results from the research work of several authors all over the world. Research cases are documented in 32 chapters organized within 7 categories next described.

### **How to reference**

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Shung Han Cho, Yuntai Kyong, Yunyoung Nam and Sangjin Hong (2010). Parallel Projection Based Self Localization Method for Mobile Navigation Applications, Mobile Robots Navigation, Alejandra Barrera (Ed.), ISBN: 978-953-307-076-6, InTech, Available from: <http://www.intechopen.com/books/mobile-robots-navigation/parallel-projection-based-self-localization-method-for-mobile-navigation-applications>

**INTECH**  
open science | open minds

### **InTech Europe**

University Campus STeP Ri  
Slavka Krautzeka 83/A  
51000 Rijeka, Croatia  
Phone: +385 (51) 770 447  
Fax: +385 (51) 686 166  
[www.intechopen.com](http://www.intechopen.com)

### **InTech China**

Unit 405, Office Block, Hotel Equatorial Shanghai  
No.65, Yan An Road (West), Shanghai, 200040, China  
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元  
Phone: +86-21-62489820  
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.