Localization for Self-Assembly Robot Using Indoor GPS and Omni-directional Mirror with Landmark

Yong Min Tai, Yeon Geol Ryu, Si Jong Kim, Ji Sung Kim and Myung Jin Chung

Department of Electrical Engineering, KAIST, Daejeon, Republic of Korea
(Tel: 82-42-350-5429; E-mail: {neotym, ygryu, terre, jisung}@rr.kaist.ac.kr, mjchung@ee.kaist.ac.kr)

Abstract — In this paper, we present the method of localization system that uses iGPS (indoor GPS) sensor and Omni-directional vision sensor in the environment involving landmark. Self-assembly robots need to know its own position to assemble to other robot. The indoor GPS (iGPS) is a simple and cheap localization sensor to adapt self-assembly robots in the indoor environment. However, the error range of iGPS is too wide to be used in a small size robot. We assume that every robot have landmark to recognize. Then, we propose a more accurate method of localization system that uses the iGPS sensor and Omni-directional vision sensor. We use the relative angle of robot estimation method and the linear position estimation. Then, we fuse the data from two sensors to get the accurate position.

Keywords — Localization, Self-assembly robot, iGPS, Omni-Mirror, linear position estimation.

1. Introduction

An accurate localization has been an important part of mobile robotics. It is a prerequisite for building a good map and an important factor for a mobile robot to move to a wanted place. Therefore, there has been considerable effort on the problem of mobile robot localization. Simultaneous localization and mapping (SLAM) is one method of localization and it has been especially successful in indoor structured environments [1, 2, and 3]. However, SLAM is not efficient localization method in outdoor environments. Although various types of sensors can be used for localization, including sonar, lasers, and cameras [6, 7], GPS is used for localization in outdoor because it is affordable and convenient [4, 5].

We have researched about a self-assembly robot. A self-assembly robot needs to know its own position to assemble to other robot. The indoor GPS (iGPS) is a
simple and cheap localization sensor in the indoor environment. It is a profit sensor to do experiment a localization system before making one for outdoor system. In this paper, we assume that an indoor GPS (iGPS) can be expanded to GPS because both principles are same. Thus, we can expand our indoor localization system to outdoor environment with GPS.

However, the error range of iGPS is too wide to be used in a small size self-assembly robot. An error range of iGPS is usually about 10cm~15cm in 5m×5m environment. It can be a serious problem to assemble to other robot. A self-assembly robot has to know more accurate position than iGPS system to assemble.

Accordingly, this paper presents a more accurate method of localization system that uses the iGPS sensor and an omni-directional vision sensor. An omni-directional vision sensor can see around the robot. If the robots have marks to detect each other, we can calculate more accurate position using other robot's marks. Thus, we made landmarks on each self-assembly robots to be detected by omni-directional vision sensor. The omni-directional vision sensor is more accurate than the iGPS in short distance. Because the omni-directional vision sensor uses a camera that is accurate sensor in the near range.

We know angles of other robots using an image from the omni-directional vision sensor. We use the specific color to distinguish the landmark. This system uses angles between landmark and the main robot, \( \theta_l \), mean value of landmark color (Red, Green, Blue), \( m_{k, \text{color}} \), and covariance of color, \( \sum_{k, \text{color}} \). To calculate accurate and fast relative angle, the system needs a process that finds regions of landmarks before calculating the angle.

2. ESTIMATING RELATIVE ANGLES OF ROBOTS

Angles of other robots can be known by using an image from the omni-directional vision sensor. We can calculate relative angles of landmarks (which can be other robots). We use the specific color to distinguish the landmark. This system uses angles between landmark and the main robot, \( \theta_l \), mean value of landmark color (Red, Green, Blue), \( m_{k, \text{color}} \), and covariance of color, \( \sum_{k, \text{color}} \). To calculate accurate and fast relative angle, the system needs a process that finds regions of landmarks before calculating the angle.

2.1 Finding landmarks in the Omni-vision image

The first step for finding regions of landmarks is the classification of landmark in the Omni-vision image. We search possible regions where landmarks are existed using the iGPS (The every landmark has its location by the iGPS. Because they are self-assembly robots). Calculating the positions from iGPS help to
find possible regions of landmarks roughly. To solving real time problem for mobile robot, it is necessary to decrease the serching time in image.

Then, we use the Mahalanobis distance\cite{8} that is calculated from $m_{k,\text{color}}$ and $\Sigma_{k,\text{color}}$ on possible regions. Because color of landmark has three channel, red, green, and blue. If we used the Euclidean distance, we cannot consider the correlation and the scale among three color channels. On the other hand, we can calculate scale invariant distance using the Mahalanobis distance. It makes more robust classification of landmark in the Omni-vision image. Mahalanobis distance is described in Eq. (1).

$$r^2 = (p_i - m_{k,\text{color}})^T \Sigma^{-1}_{k,\text{color}} (p_i - m_{k,\text{color}})$$

\textit{r$^2$}: Mahalanobis Distance

\textit{where}, $m_{k,\text{color}} = (m_1, m_2, m_3)^T$: mean

$\Sigma_{k,\text{color}}$: covariance matrix

$p_i = [R \ G \ B]^T$: i-th pixel value vector

After calculating Mahalanobis distances of pixels for possible regions, choose the nearest value with landmark. It is used for finding real landmark regions.

2.2 Relative angle of robot

After obtaining the nearest point of pixel from Mahalanobis distances, save the point of pixels for expected region. We select the expected regions where landmarks are existed using that point of pixels. Then we can extract the accurate landmark regions from the expected regions on Omni-vision image.

The saved point of pixels are calculated to obtain angles between robot and landmark. If this angle exist in the reasonable angle variation, it is valid point. We can find the reasonable angle variation by using $\theta'_{l,k}$ that is obtained from the iGPS. expected region is described in Eq(2) and “Fig.2”.

$$\theta'_{l,k} = \tan^{-1}(x_i - c)$$

$$\theta'_{l,k} = \lambda < \theta'_{l,k} < \theta'_{l,k} + \lambda$$

\textit{where}, $\lambda$ : angel variation

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{diagram.png}
\caption{Estimate the relative angle of robot.}
\end{figure}
Then, we can calculate the relative angles of robot from center of landmark region. It is described in Eq. (3).

\[
\theta^\prime_{k,vision} = \tan^{-1}\left(\frac{1}{n} \sum_{i=1}^{n} x_{i,true} - c \right)
\]

(3)

\(\theta^\prime_{k,vision}\): robot angle  
\(x_{i,true}\): robot pixels (Landmark pixels)  
\(c\): image center

The whole process of estimate the relative angle of robot follows "Fig. 1".

When the robot and landmarks are located like "Fig. 3.", the position of main robot is represented \((x, y)\) using positions of landmarks, \((x_k, y_k)\) and angle, \(\theta_k\). It is described in Eq. (4).

\[
\sin \theta_k \cdot x - \cos \theta_k \cdot y = \sin \theta_k \cdot x_k - \cos \theta_k \cdot y_k
\]

(4)

Eq. (4) can be expanded as following,

\[
\begin{pmatrix}
\sin \theta_1 & -\cos \theta_1 \\
\sin \theta_2 & -\cos \theta_2 \\
\vdots & \vdots \\
\sin \theta_n & -\cos \theta_n
\end{pmatrix}
\begin{pmatrix}
x \\
y
\end{pmatrix}
= 
\begin{pmatrix}
\sin \theta_1 \cdot x_1 - \cos \theta_1 \cdot y_1 \\
\sin \theta_2 \cdot x_2 - \cos \theta_2 \cdot y_2 \\
\vdots \\
\sin \theta_n \cdot x_n - \cos \theta_n \cdot y_n
\end{pmatrix}
\]

(5)

estimates robot position using positions of other robots (or landmarks) and angles. We’ve already assumed that every self-assembly robots have landmark. Thus, we can also get the position \((x, y)\) of other robots (landmarks) by iGPS. However, these positions are not accurate to assemble each other. We need to obtain accurate position from inaccurate positions by iGPS system.

3. LINEAR POSITION ESTIMATION

The linear position estimation is the algorithm that
Then, Eq. (5) can be represented the first order linear equation like Eq. (6).

\[ Ax = b \]  \hspace{1cm} (6)

\[
A_{(m2)} = \begin{pmatrix}
\sin \theta_1 & -\cos \theta_1 \\
\sin \theta_2 & -\cos \theta_2 \\
\vdots & \vdots \\
\sin \theta_n & -\cos \theta_n
\end{pmatrix},
\]

where, 

\[
x_{(m1)} = \begin{pmatrix}
x \\
y
\end{pmatrix},
\]

\[
b_{(m1)} = \begin{pmatrix}
\sin \theta_1 \cdot x_1 - \cos \theta_1 \cdot y_1 \\
\sin \theta_2 \cdot x_2 - \cos \theta_2 \cdot y_2 \\
\vdots \\
\sin \theta_n \cdot x_n - \cos \theta_n \cdot y_n
\end{pmatrix}
\]

Eq. (6) could be solved using Eq. (7) when the rank of matrix \( A \) is two.

\[
\arg_x \left( \min \left\| A x - b \right\|^2 \right)
\]  \hspace{1cm} (7)

Thus, position of main robot \((x, y)\) can be estimated as following equation if rank of \( A \) is 2.

\[
x^* = (A^T A)^{-1} A b
\]  \hspace{1cm} (8)

The global coordinate angle of other robots (landmarks), \( \theta_k \) and local coordinate angle, \( \theta_k^l \) are calculated by iGPS. \( \theta_k^l \) can modified by omni-directional vision system. The position of robot \((x, y)\) is calculated using modified local angle and global angle. The whole process of localization of robot follows “Fig. 5”.

4. LOCALIZATION USING SENSOR FUSION

The error range of iGPS is too wide to be used in a small size self-assembly robot. It can be a serious problem to assemble to other robot. A self-assembly robot has to know more accurate position than iGPS system to assemble. Thus, we fuse iGPS sensor and Omni-directional vision sensor to estimate position more accurate as “Fig. 4.”. We use the accurate angle data from iGPS and estimated position data from the iGPS and the omni-directional vision sensor.
Next, calculating local angle of other robots. A coordinate of robot is changed when it moves. Because of an input angle from omni-directional vision sensor is changed, we have to change from a global angle of other robots $\theta^*_{\text{g}k}$ to a local angle of other robots $\theta_{\text{l}k}$. We use a heading angle of main robot $\theta_{\text{g}m}$ to calculate a local angle of other robots $\theta_{\text{l}k}$.

Then, we compensate a local angle of other robots $\theta_{\text{l}k}$ with an image from omni-directional vision sensor. An accurate local angle of other robots $\hat{\theta}_{\text{l}k}$ are obtained by algorithm of “Fig.1”.

A heading angle of main robot $\theta_{\text{g}m}$ is renewed by difference between a global angle of other robots $\theta^*_{\text{g}k}$ and an accurate local angle of other robots $\hat{\theta}_{\text{l}k}$. After changing an accurate local angle of other robots $\hat{\theta}_{\text{l}k}$ to an accurate global angle of other robots $\hat{\theta}_{\text{g}k}$, a position of main robot $\hat{x}_{\text{g}m}$ can be calculated by linear position estimation algorithm. If a rank of matrix A wasn’t two, we can use my position $(x_{\text{g}m}, y_{\text{g}m})$ of first step. Because the result of linear position estimation is invalid when a rank of matrix A wasn’t two. And we use position from iGPS $x_{\text{g}m}$ than estimated position $\hat{x}_{\text{g}m}$ when a difference between $x_{\text{g}m}$ and $\hat{x}_{\text{g}m}$ are larger than error of iGPS range. We also use low pass filter for robust position estimation.

5. EXPERIMENT AND RESULT

First, calculating a global angles $\theta^*_{\text{g}k}$ between main robot and other robots using my position $(x_{\text{g}m}, y_{\text{g}m})$ and positions of other robots $(x^*_k, y^*_k)$ from the iGPS.

Before doing the experiment, we simulate different number of landmarks case using MATLAB. We add 100mm random noise to the position of landmark.
same as iGPS has. We can notice that this system has less error than iGPS, and we can also know that the more landmarks make more accurate localization. After confirming our system by simulation, we've done experiment by the iGPS and the Omni-vision sensor.

5.1 Simulation Results

Before doing the experiment, we do experiment by simulation system using MATLAB. We add 100mm random noise to the every position of robots and landmarks same as iGPS has to obtain same result as real experiment. We do the experiment for different number of robots. We can find that a position from proposed algorithm is more accurate than a position from iGPS in every case. They are shown at “Fig.6”.

Number of landmarks = 3

Number of landmarks = 4
A linear position estimation algorithm is affected by number of robots. Because it calculates a main robot position from other robots. We can find the more robots the lower error position in “Fig. 6”. “Fig. 7” is described error distance for number of landmarks.
5.1 Experiment Results

After confirming our system by simulation, we've done experiment by the iGPS and the Omni-vision sensor with a mobile robot. iGS-U by ninety system co. which is iGPS, L-type (74mm) Omni-mirror by Accowle, VX-6000 camera by Microsoft and Pioneer 3 by Active Media co. which is mobile robot are used in our experiment. We used 4 landmarks for experiment. However, this system can be expanded to a multi mobile robot system if landmarks are alternated with other mobile robots. It is enough to use a mobile robot to prove our system is more accurate than only iGPS system. “Fig. 8” shows that our algorithm can estimate a position in real time.

We obtained data over the time and moved robot to parallel direction of axis. Our localization algorithm is robust to estimate position over the time if omni-directional vision sensor lost one landmark. It could calculate position with 3 landmarks, but it needs more than 2 landmarks. When the robot moved along the x-axis or y-axis, we could find our algorithm is more robust and accurate than the iGPS localization. It is described in “Fig. 9” and “Fig. 10”.

The maximum error of the localization system we proposed show better performance as Table 1. The maximum error is 150mm when iGS uses only. However the maximum error is 40mm in our system on 5m x 5m.

We can know the ground truth because of doing experiment on the grid. We made grid on the ground and measured real position of robot. The speed of mobile robot was about 50mm/sec and it moved randomly.

![Fig. 8. Localization experiment with mobile robot](image)

![Table 1 Comparison between iGPS and Sensor Fusion.](image)

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Maximum Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>iGPS only</td>
<td>15 cm</td>
</tr>
<tr>
<td>Sensor Fusion</td>
<td>4cm</td>
</tr>
</tbody>
</table>
6. CONCLUSION

We have researched about a self-assembly robot. A self-assembly robot needs to know its own position to assemble to other robot. In this paper, we assume that an indoor GPS (iGPS) can be expanded to GPS because both principles are same. Thus, we can expand our indoor localization system to outdoor environment with GPS. However, the error range of iGPS is too wide to be used in a small size self-assembly robot. It can be a serious problem to assemble to other robot. A self-assembly robot has to know more accurate position than iGPS system to assemble.

We propose a more accurate method of localization system that uses iGPS sensor and Omni-directional vision sensor. It is more accurate and robust than using only iGPS system. Moreover, this system uses iGPS localization to find expected region in Omni-vision image. It doesn’t need to search the whole image to find landmarks in the Omni-vision image. It makes the system fast and accurate.

We used a mobile robot and several landmarks in the experiment. However, this system can be expanded to a multi mobile robot system if landmarks are alternated with other robots. And we proved this localization algorithm run in real time through the experiment. This algorithm can be used for localization of self-assembly robot.

Acknowledgement

This work was supported by the IT R&D program of MKE/IITA. [2008-F-041-01, Development of S/W and H/W Re-engineering Technology for Intelligent Robotics based on Learning and Evolution Techniques]

References


