

Self-localization based on Image Features of Omni-directional Image

メタデータ	言語: eng 出版者: 公開日: 2012-12-19 キーワード (Ja): キーワード (En): 作成者: HIBINO, Fuminori, II, Yuta, TAKAHASHI, Yasutake, MAEDA, Yoichiro メールアドレス: 所属:
URL	http://hdl.handle.net/10098/6961

= 2012 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, creating new collective works, for resale or redistribution to servers or lists, or reuse of any copyrighted component of this work in other works.

Self-localization based on Image Features of Omni-directional Image

Fuminori HIBINO, Yuta II*, Yasutake TAKAHASHI, and Yoichiro MAEDA

*Dept. of Human & Artificial Intelligent Systems, Faculty of Engineering, University of Fukui

Dept. of Human & Artificial Intelligent Systems, Graduate School of Engineering, University of Fukui

3-9-1, Bunkyo, Fukui, Fukui, 910-8507, Japan

Email: {fhibino,yii,yasutake,maeda}@ir.his.u-fukui.ac.jp

Abstract—Omni-vision system using an omni-mirror is popular to acquire environment information around an autonomous mobile robot. In RoboCup soccer middle size robot league in particular, self-localization methods based on white line extraction on the soccer field are popular. We have studied a self-localization method based on image features, for example, SIFT and SURF, so far. Comparative studies with a conventional self-localization method based on white line extraction are conducted. Compared to the self-localization method based on white line extraction, the method based on image feature can be applied to a general environment with a compact database.

I. INTRODUCTION

Omni-vision system using an omni-mirror is popular to acquire environment information around an autonomous mobile robot. In RoboCup soccer middle size robot league in particular, self-localization methods based on white line extraction on the soccer field are popular[1], [2], [3]. White line points in an omnidirectional camera image are identified as points if a sequence of green-white-green segments on any line in the image. These points on the omnidirectional camera image are transformed into robot centered coordinates. A Monte-Carlo method or its derivatives are widely used for the localization on the soccer field.

On the other hand, more general image features, for example, SIFT and SURF[4], are used for visual localization or simultaneously localization and mapping (SLAM) ([5], for example). We have studied a self-localization method based on image features in omnidirectional and ceiling camera images, so far. An omnidirectional camera image is mapped into robot centered coordinate image. Image features detected on the omnidirectional robot centered image are collected as a template set of the image features into a map database with the robot position in the field coordinate system. Image features on a query image is matched with the templates and the camera position in the field coordinate system. The idea of self-localization based on the image features in omnidirectional camera images is similar to [6]. Their method needs two reference templates from a database with position information because they use only image feature's orientation information on the omnidirectional image. Our method needs only one reference template because it utilizes not only the orientation but also distance of each image feature based on a camera model.

Comparative studies with a conventional self-localization method based on white line extraction are conducted in this paper. Compared to the self-localization method based on white line extraction, the method based on image feature can be applied to a general environment with a compact database. We focus on matching error distributions of two methods in order to evaluate them from the view points of computational efficiency and memory for maintaining a world model, in this paper.

II. SELF LOCALIZATION BASED ON OMNIDIRECTIONAL IMAGE

This section describes details of model matching error calculation of two localization methods. One is based on white-line point extraction and the other is based on image feature extraction.

A. Localization based on extracted white line points

The method based on white-line point extraction follows a procedure below:

- 1) Generate a model of white-line points on the field, lookup tables of positions of white-line points at regular sampling places on the field.
- 2) Capture a omnidirectional image.
- 3) Extract white-line points on the image
- 4) Compare the extracted white-line points to the model, lookup tables, and calculate matching error.
- 5) Estimate the position and posture of the omnidirectional camera by finding the minimum matching error with the model.
- 6) goto 2).

It maintains a model of white lines on the field. For computational efficiency, it maintains lookup tables of positions of white-line points $D^m(x, y, \theta)$ in the omnidirectional-camera polar coordinate system (r_i^m, ϕ_i^m) $i = 1, \dots$ when the robot locates itself at position (x, y) and orientation θ in the field coordinate system. Our experiments with the white-line-point based localization use the look up tables at 0.1m by 0.1m intervals.

D^q is a set of white-line points (r_j^q, ϕ_j^q) $j = 1, \dots$ that the robot captures on the field.

$$\begin{aligned} D^q &= \{(r_1^q, \phi_1^q), (r_2^q, \phi_2^q), \dots, (r_Q^q, \phi_Q^q)\} \\ D_{(x,y,\theta)}^m &= \{(r_1^m, \phi_1^m), (r_2^m, \phi_2^m), \dots, (r_M^m, \phi_M^m)\} \\ E_{(x,y,\theta)} &= \sum_{i=1}^Q \begin{cases} e_{none} & \text{if } \forall \phi_i^q \neq \phi_j^m \\ \min_{j|\phi_i^q=\phi_j^m} |r_i^q - r_j^m| & \text{else} \end{cases} \quad (1) \end{aligned}$$

Matching error E between a lookup table $D^m(x, y, \theta)$ and a query D^q is calculated by Eq. 1 where M and Q are number of white-line points of a lookup table at (x, y, θ) and the query point, and e_{none} is a penalty when there is no white-line point in the lookup table.

B. Localization based on image features

The method based on image follows a procedure below:

- 1) Load a database of sets of image features, templates, on the images captured at regular sampling position on the field.
- 2) Capture a omnidirectional image.
- 3) Transform the image into robot centered coordinate system.
- 4) Extract image features on the transformed image.
- 5) Ignore the image features outside of the field.
- 6) Compare the extracted image features to all templates in the database, and calculate matching errors.
- 7) Estimate the position and posture of the omnidirectional camera by finding the minimum matching error with the template corresponding to the position on the field.
- 8) Eliminate wrong matching pair based on distribution of distance of matching pairs.
- 9) Calculate the translation distance and orientation angles from the position.
- 10) goto 2).

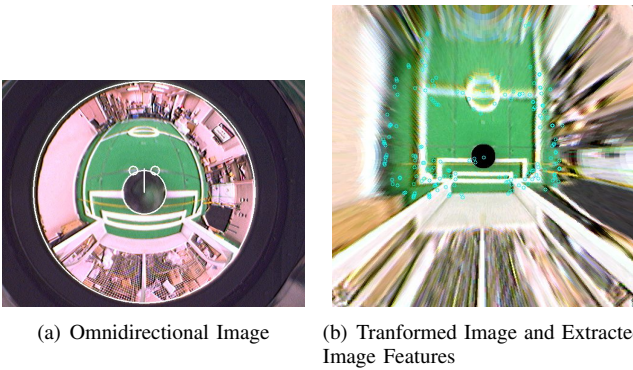


Fig. 1. Input image captured by an omnidirectional camera (a) and extracted image features on the image translated into robot centered coordinates (b)

Fig.1(a) shows a sample image captured by a omnidirectional camera. This image is transformed into robot centered coordinate system on the field. Fig.2 shows relationship between the distances from the center of camera and the point on the field on camera image and field. The relationship

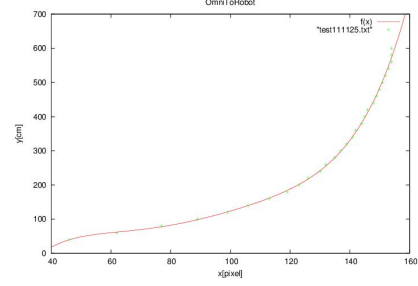


Fig. 2. Relationship between the distances from the center of camera and the point on the field on camera image and field: cross points and a curve indicates sampling points and estimated relation function.

is approximated with a fifth-order polynomial function $f(r)$ shown in Eq.(2) while r is distance from the center of camera to the point on the omnidirectional camera image.

$$f(r) = ar^5 + br^4 + cr^3 + dr^2 + er + g \quad (2)$$

The parameters, a , b , c , d , e and g , are estimated by the least-square method. A sample transformed image is shown in Fig.1(b).

Image features, SURF in this paper, are extracted from the transformed image. The image features out of the ground are ignored from the set of the image features. The ground region on the image is extracted based on the color of the field, that is, we assume that the green region is the ground. Image features on a sample transformed image are shown in Fig.1(b).

A set of image features on the image captured at a position corresponding to a position in the field coordinate system is called a template in this paper. Database is generated by collecting the templates at regular sampling position on the field.

Flann based descriptor matcher provided by OpenCV library [7] is used to find corresponded pair of the image features. Then, a translation matrix and the matching error between the image feature sets of database and query image are calculated as follows. We define the translation matrix A with parallel translation vector $(\Delta x, \Delta y)$ and orientation angle $\Delta\theta$ as below:

$$A(\Delta x, \Delta y, \Delta\theta) = \begin{pmatrix} \cos\Delta\theta & -\sin\Delta\theta & \Delta x \\ \sin\Delta\theta & \cos\Delta\theta & \Delta y \\ 0 & 0 & 1 \end{pmatrix} \quad (3)$$

Matching error is defined with Eq.(4)

$$E(\Delta x, \Delta y, \Delta\theta) = \sum_{i=0}^M \| {}^t\mathbf{p}_i - A(\Delta x, \Delta y, \Delta\theta) {}^q\mathbf{p}_i \|^2 \quad (4)$$

where i and M are index and number of matching pair of image features between the template and the query image, respectively. ${}^t\mathbf{p}_i$ and ${}^q\mathbf{p}_i$ are position of image features in the robot centered transformed coordinate system of template and query image defined as below:

$${}^t\mathbf{p}_i = ({}^tx_i, {}^ty_i, 1)^T \quad (5)$$

$${}^q\mathbf{p}_i = ({}^qx_i, {}^qy_i, 1)^T \quad (6)$$

where $(^t x_i, ^t y_i)$, $(^q x_i, ^q y_i)$, and T are position of i th matched image feature of the template, position of i th matched image feature of the query image, and transposition of matrix, respectively.

In order to find the best parallel translation vector $(\Delta x, \Delta y)$ and orientation angle $\Delta \theta$, we introduce a method similar to Gibbs sampling as follows:

- Step i Set $(\Delta x_t, \Delta y_t, \Delta \theta_t) = (0, 0, 0)$ at time $t = 0$, and calculate E by Eq.(4).
- Step ii Find the best $\Delta \theta_{t+1}$ to minimize E while Δx_t and Δy_t are fixed.
- Step iii Find the best Δy_{t+1} to minimize E while Δx_t and $\Delta \theta_{t+1}$ are fixed.
- Step iv Find the best Δx_{t+1} to minimize E while Δy_{t+1} and $\Delta \theta_{t+1}$ are fixed.
- Step v If $\Delta x_{t+1} = \Delta x_t \wedge \Delta y_{t+1} = \Delta y_t \wedge \Delta \theta_{t+1} = \Delta \theta_t$, stop searching.
- Step vi Increment t and goto Step ii.

After the best parallel translation vector $(\Delta x, \Delta y)$ and orientation angle $\Delta \theta$ against to the template is calculated, variance of distance of matching pairs is calculated and eliminate wrong matching pairs based on the variance. Then, $(\Delta x, \Delta y, \Delta \theta)$ is calculated again. Finally, estimated position and orientation (x, y, θ) in the field coordinate system is derived from the $(\Delta x, \Delta y, \Delta \theta)$ and the position of the template in the field coordinate system.

III. EXPERIMENTS

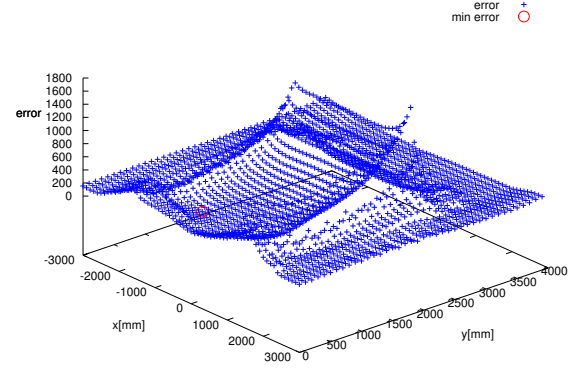
Comparative studies with a conventional self-localization method based on white line extraction are conducted to evaluate the method based on image features of omnidirectional camera image. We set up a soccer field which is smaller than the official field of RoboCup Middle Size league and a soccer robot with an omnidirectional camera on the field.

A. Matching error distributions between one model/template and a query point

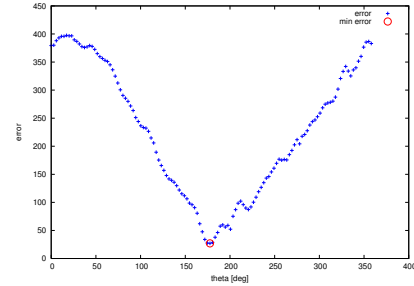
Fig.3 shows an example matching error distribution between a query and a white line model of translation and orientation error to the query. In this experiment, the query is fixed on the field and the virtual position and orientation on the field of the white line model varies. Red circle indicates the minimum point where the position of the model is same to the query on the field.

This figure shows that the matching error distribution has steeply local minima so that full search is needed to find the global minimum. This figure indicates that it needs a computational high cost method to find the best parameters. Monte-Carlo methods, include particle filter, are widely used to estimate the position with limited computational cost.

Fig.4 shows an example of matching error distribution between a template and query one that has parallel translation and orientation difference from the template. In this experiment, the template is fixed and the query has parallel translation and orientation difference $(\Delta x, \Delta y, \Delta \theta)$ from the template. Red circle indicates the minimum point where the



(a) Matching error distribution between a query and white line model at (x, y) in parallel translation error from the query



(b) Matching error distribution between a query and a white line model of $\Delta \theta$ orientation error from the query

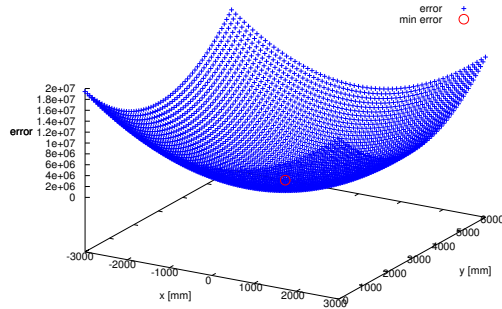
Fig. 3. Matching error distribution between a query and a white line model of translation and orientation error to the query: Red circle indicates the minimum point.

query is at same position and orientation to the template on the field.

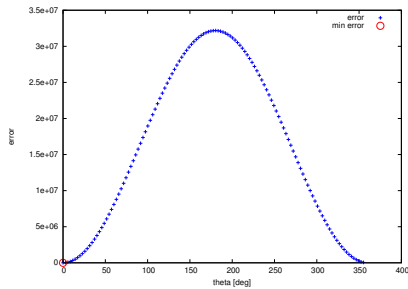
This figure shows that the matching error distribution is smooth and there is only one local minimum in global. This insight suggests that a simple gradient method, as we proposed in the previous section, is enough to find the best parameter values $(\Delta x, \Delta y, \Delta \theta)$.

B. Estimation error distribution between one template and query points

Fig.5 shows estimated position and orientation error distribution. Query points are sampled at 0.2m by 0.2m intervals on the field. The estimated position error is $\sqrt{(\hat{\Delta x} - \Delta x)^2 + (\hat{\Delta y} - \Delta y)^2}$ while $(\hat{\Delta x}, \hat{\Delta y})$ is the estimated parallel translation difference and $(\Delta x, \Delta y)$ is the ground truth. Fig.5(a) shows that estimation error of query position depends on the distance from the template. The error is small enough to estimate the parallel translation difference $(\Delta x, \Delta y)$ though it becomes bigger if the distance from the template is over 1500 [mm]. This experimental result suggests that template sampling at 1000 [mm] by 1000 [mm] is enough for self-localization on the field. Fig. 5(b) shows the estimated orientation error at a query point. The query



(a)
with $\Delta x, \Delta y$



(b) Matching error distribution between a template and query one with θ orientation error from the ground truth

Fig. 4. Matching error distribution between one template and the query one with translation and rotation error from the template: Red circle indicates the minimum error point.

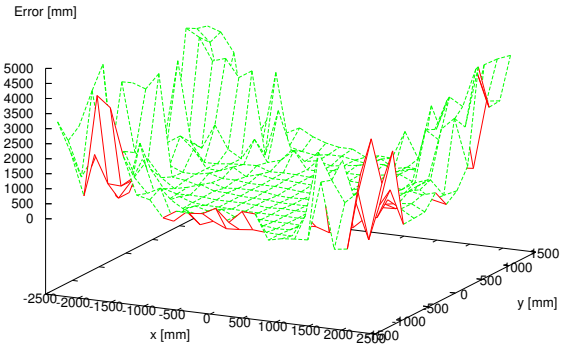
orientation are sampled at 10 degree intervals at the query point. The error is less than ± 15 degree in all orientation.

Our primitive experiments use a set of look-up tables of white line points model at 0.1[m] by 0.1[m] interval. The density of the models is necessary for adequate self-localization. Image feature based localization needs 1 [m] by 1 [m] interval-template database. The image feature based localization needs less computational memory to store the database than the white line point based localization.

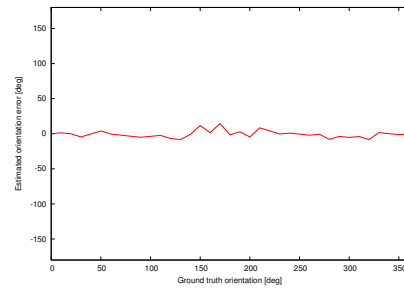
IV. CONCLUSION

We conducted comparative studies of our proposed self-localization method based on image features in omnidirectional camera images with a conventional self-localization method based on white line extraction. Our method maps an omnidirectional camera image into robot centered coordinate image, then, the image features detected on the omnidirectional robot centered image are collected as a template set of the image features into a map database with the robot position in the field coordinate system. Image features on a query image is matched with the templates and the camera position in the field coordinate system.

The comparative studies show that our method has a good property on matching error distribution, therefore, less-cost searching algorithm is applicable to find the best parameter



(a) Position estimation error



(b) Orientation estimation error

Fig. 5. Estimation error distribution between one template and query points

for localization while conventional one needs high cost search methods. Compared to the self-localization method based on white line extraction, the method based on image feature can be applied to a general environment with a compact database.

REFERENCES

- [1] A. Merke, S. Welker, and M. Riedmiller, "Line based robot localization under natural light conditions," in *In ECAI 2004 Workshop on Agents in Dynamic and Real Time Environments*, 2004.
- [2] M. Lauer, S. Lange, and M. Riedmiller, "Calculating the perfect match: An efficient and accurate approach for robot self-localization," in *RoboCup 2005: Robot Soccer World Cup IX*, A. Bredenfeld, A. Jacoff, I. Noda, and Y. Takahashi, Eds. Springer, 2006, pp. 142–153. [Online]. Available: <http://www.springerlink.com/content/v59m05j034n40228>
- [3] P. Heinemann, J. Haase, and A. Zell, "A combined monte-carlo localization and tracking algorithm for robocup," in *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems*, Oct 2006, pp. 1535–1540.
- [4] H. Bay, A. Ess, T. Tuytelaars, and L. V. Gool, "Surf: Speeded up robust features," *Computer Vision and Image Understanding (CVIU)*, vol. 110, no. 3, pp. 346–359, 2008.
- [5] W. Jeong and K. M. Lee, "CV-SLAM: A new ceiling vision-based SLAM technique," in *2005 IEEE/RSJ International Conference on Intelligent Robots and Systems*, 2005, pp. 3195–3200.
- [6] A. C. Murillo, J. J. Guerrero, and C. Sagales, "Surf features for efficient robot localization with omnidirectional images," in *Proceedings of 2007 IEEE International Conference on Robotics and Automation*, 2007, pp. 3901–3907.
- [7] OpenCV. Open source computer vision library,"

<http://sourceforge.net/projects/opencvlibrary/>.