

Omnidirectional Vision-based Robot Localization on Soccer Field by Particle Filter

Chia-Yang Chen, Cheng-Yao Ho, Ching-Chang Wong
Dept. of Electrical Engineering
Tamkang University
Tamsui, Taipei County, 25137, Taiwan
wong@ee.tku.edu.tw

Hisayuki Aoyama
Dept. of Mechanical Engineering and Intelligent System
The University of Electro-Communications
Tokyo, Japan
aoyama@mce.ucc.ac.jp

Abstract—An omnidirectional vision-based localization method based on the particle filter is proposed to achieve the location of robot on a soccer field in this paper. Two kinds of sensor information are considered in the method to let the robot on the soccer field can estimate its location and then to decide an appropriate strategy. One is the robot action sensor information obtained by the motor's feedback and the other is the observation sensor information obtained by the image captured by an omnidirectional vision system. The action sensor information is used to expect the robot location distribution. The location distribution is represented by particles. The omnidirectional image is used to observe the environment information. The differences between these particles location's environment information and the robot observation environment information are considered to calculate the belief values of particles. Then the posture of the particle with the highest belief is used to be the estimated posture of the robot. Some experimental results are presented to illustrate the effectiveness of the proposed method.

Keywords: *Omnidirectional Vision, Robot localization, Particle Filter localization, Monte Carlo method localization*

I. INTRODUCTION

Robot localization is a very popular topic in the robot research. In order to find the location of the robot in a known space, the robot should create a map mapping a real space to give some definitions of the space. Therefore, the robot should use the map to be a reference to find its location on this space. In general, sensors, which are used to observe the environment information, are necessary needed for the localization application. In this paper, the robot action sensor and environment observation sensor are used to estimate the location of the robot on the map. Estimation location is not kind directly to know position by object relationship, which is an iterative method.

Particle filter localization method is usually used to estimate the location of the robot. Particle filter localization method is also known as Monte Carlo localization method. The basic application is to calculate a ratio of circumference π . Define a unit square to be a domain, and a quarter unit circle in the domain to be a condition, and then random put many particles in the domain. The ratio of circumference can be calculated by the particles in the quarter circle's ratio. In this paper, the domain is viewed as a map and the environment

information is viewed as a condition. Keep the particles to locate on a similar environment and then move the particles following the robot action. Iterate the procedure can estimate the robot location and track it. Many kinds of Monte Carlo localization methods [1-4] were proposed. Their differences are the observation sensors. In this paper, the omnidirectional vision system is used to be an environment observation sensor. An omnidirectional vision-based localization method based on the particle filter is proposed to estimate the location of the robot on a soccer field.

II. PARTICLE FILTER OF ROBOT LOCALIZATION

Robot localization is the robot uses some environment information to estimate the location of the robot by itself. In general, a robot locates on a floor, so the robot's posture can be described by the following three-dimensional state vector

$$l = [x, y, \theta]^T \quad (1)$$

where x and y are respectively the coordinates of the robot's position in the x -directional axis and y -directional axis, θ is the robot's direction. Owing to the automatic movement or the influence from the exterior force, the posture of robot will be changed. The particle filter localization method is used to estimate the robot's posture based on the obtained environment information. The particle filter of localization is that estimated robot posture by the environment information, and that is an instance of Bayesian filtering problem. The estimated posture of the robot at time t can be described by the following belief function [5].

$$Bel(l_t) = p(l_t | d_{0:t}) \quad (2)$$

where l_t is the estimated posture of the robot at time t , and $d_{0:t}$ is the environment information data detected by the sensors in the robot from the beginning to the time t . The belief function is the estimated posture on state of Robot sensors' data's posterior probability distribution. According to the data types obtained by the sensors mounted in the robot, the belief function [5] can be described by

$$Bel(l_t) = \eta p(o_t|l_t) \int p(l_t|l_{t-1}, a_{t-1}) Bel(l_{t-1}) dl_{t-1} \quad (3)$$

where η is a constant, o_t is an observation data obtained from the environment, a_{t-1} is an action data obtained from the feedback of the robot, $p(o_t|l_t)$ is an observational belief function of the posture, and $p(l_t|l_{t-1}, a_{t-1})$ is a posture belief function of robot acted. In this paper, the posture belief density is represented by particles. And an action model is created to simulate robot action let particles locate on possible areas as same as function of posture belief density. An observation model is created to simulate robot observation to suppose particle's postures get virtual observation data. The observational belief value is calculated by the difference between the virtual and real observation data. The posture of the particle with the highest belief is used to be the estimated posture of the robot.

III. LOCALIZATION ENVIRONMENT

The soccer field based on the rules of RoboSot league in FIRA (Federation of International Robot-soccer Association) Cup [6] is considered to discuss the robot localization problem. This field contains both a rectangular playing field $600 \times 400 \text{ cm}^2$ in size surrounded by a bordering region of minimum width 75 cm . The playground and bordering region should be green in color. All line markings will be painted with a 3 cm thickness and must be white in color. The centre-circle will have a radius of 60 cm . The goal area will be 120 cm wide, 40 cm deep and positioned directly in front of each goal. The penalty area will be 200 cm wide, 50 cm deep and positioned centrally in front of each goal. The goal kick semi-circles will have a radius of 60 cm and are centered on the lines marking the penalty area parallel to each goal. In this paper, field lines are used to be the environment information.

IV. OMNIDIRECTIONAL VISION

An omnidirectional vision system is used to be the main observation sensor in this paper. As shown in Figure 1, a CMOS camera is set under a kind of conic or curve mirror so that the environment information around this mirror can be captured into the image.

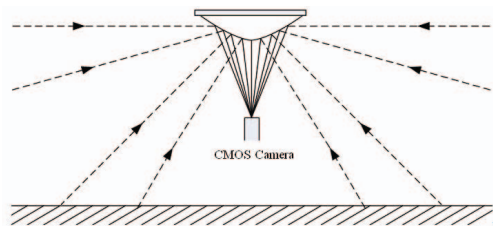


Figure 1. Omnidirectional mirror diagram.

A. Rough Characteristic

The CMOS camera is used to be a distance sensor to detect field lines and determine their distances based on the obtained omnidirectional image. The relationship between the CMOS sensor unit and the detected angle is described as shown in

Figure 2, where the positions of the CMOS camera and mirror are fixed. The angle α of incidence to the CMOS camera detected Pixel(x, y) from the mirror is also fixed [7] so that the camera detected the angle diagram can be described by Figure 3, where the mirror has two kinds of setting corresponding to the ground. According to the detected angle of camera, each pixel is fixed. In Situation A, the mirror and ground are parallel. Camera can detect same distance objects on ground from mirror angle α . Therefore in Situation B, the mirror is slanted with the angle β corresponding to the ground so that the detected angle of ground will change to be $\alpha + \beta$ and $\alpha - \beta$. In this situation, the camera detected distances on the ground are different.

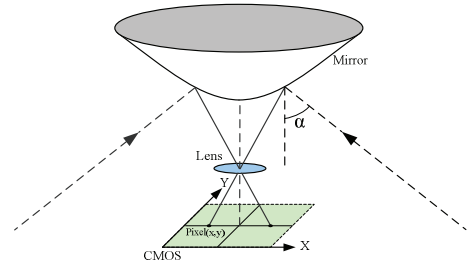


Figure 2. Relationship diagram of the omnidirectional mirror and camera.

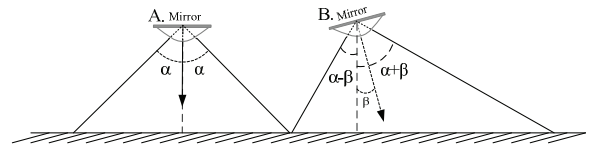


Figure 3. Omnidirectional mirror and ground.

B. Omnidirectional vision setting

The omnidirectional vision system is used to detect field lines on the ground. In order to let the omnidirectional vision system is easier to process, the mirror is set to be parallel to the ground to let the detected distance are symmetrical to the mirror center. An omnidirectional image is shown in Figure 4, where the image includes the view around the mirror in a circle image that is described by polar coordinates. And the image pixel of mirror center is set to be (0,0) of the coordinates. In order to effectively detect objects on the ground from the captured omnidirectional image, the image information in the area that is too near by or too far away from the robot is ignored. Thus, the Internal Radius and the External Radius in Figure 4 are respectively considered as a start radius and a stop radius of the image range to detect objects effectively.

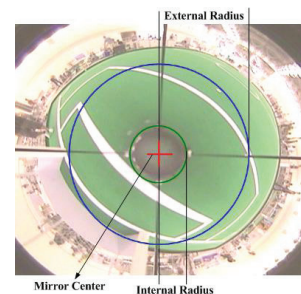


Figure 4. Omnidirectional Image and setting

C. Sample and Modeling method

In this paper, the distance and pixel radius models are created to translate a pixel radius of an object to a distance. The setting height of the robot's mirror is fixed, and the mirror is set parallel to the ground so that an image pixel radius corresponds to a distance, therefore the model can be created by sampling. A straight line can be considered as a sample and line's edges in the image are detected to be samples. The diagram of this method is shown in Figure 5, where the distance between the robot and the straight line is known by the measurement. The distance's corresponding pixels radius is obtained from the image shortest pixels between mirror center and the line. This pair is first sample of the model, and then follow the line edge can get many pixel radiuses to be samples. These pixel radiuses' distance can be calculated by

$$\text{Distance} = \frac{\text{Shortest Distance}}{\cos(\text{Angle})} \quad (4)$$

where Angle is an angle between from the pole to the first sample edge's pixel and from the pole to the detected pixel. Shortest Distance is the first sample's distance between the line and the robot. Distance is the sample's corresponding distance of the robot. The processing results are shown in Figure 6. The sample image is described in the right side of Figure 6, where the red and blue lines are edges of clockwise and counterclockwise, respectively. The modeling result is shown in the left side of Figure 6, where the red and blue lines are edge results of the clockwise and counterclockwise, respectively.

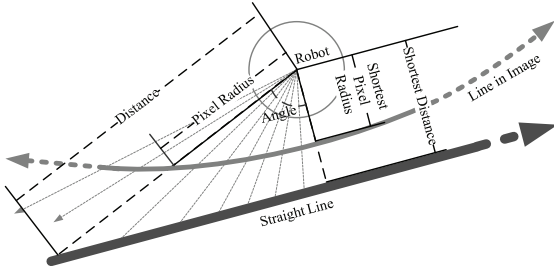


Figure 5. Distance and pixel radius model Sampling Method diagram

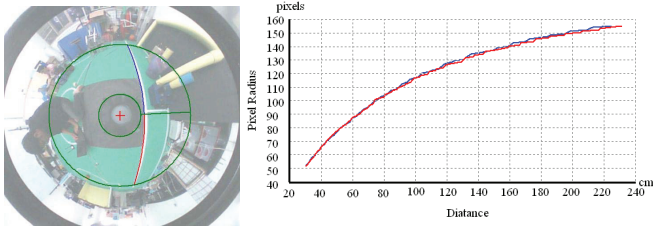


Figure 6. Pixel radius and distance relationship model modeling results

V. ROBOT SENSOR MODEL

The sensor model is used to simulate the sensor working. In general, errors will happen in any kinds of sensors. The error is considered to create the sensor model. The sensor value can be expected by the sensor model. In this paper, the sensor according to type is classified two kinds' sensors there are

action and observation sensor. The action sensor is used to expect posture of the robot and expect error happening by the sensor model. The observation sensor can detect environment information. The observation sensor model is used to simulate expectation posture's sensor values. And then belief of the posture can get by compare with real values and expectation sensor values.

A. Action sensor model

In this paper, the robot action is moved by motors. The motor feedback model is created to be action sensor model. The robot is equipped with four wheels. The setting can let robot move to any directional. The wheels setting of robot is shown in Figure 7, where ϕ is the first wheel's angle of field, the others wheels are set on each differ from ϕ angle 90° , R is the radius of the robot, Fb_i , $i=1, 2, 3, 4$, is the distance calculated by the feedback of i -th motor. The posture of the robot is estimated by

$$\begin{bmatrix} x_i \\ y_i \\ \theta_i \end{bmatrix} = \begin{bmatrix} x_{i-1} \\ y_{i-1} \\ \theta_{i-1} \end{bmatrix} + \begin{bmatrix} \Delta x_{i-1} \\ \Delta y_{i-1} \\ \Delta \theta_{i-1} \end{bmatrix} \quad (5)$$

where $[\Delta x_{i-1}, \Delta y_{i-1}, \Delta \theta_{i-1}]^T$ is the action data of a_{i-1} and described by

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \theta \end{bmatrix} = \frac{1}{4} A \times \begin{bmatrix} Fb_1' \\ Fb_2' \\ Fb_3' \\ Fb_4' \end{bmatrix} \quad (6)$$

where A is a transfer matrix from four feedback distance to moving of posture and describe by

$$A = \begin{bmatrix} \cos(\phi + \frac{\pi}{2}) & \cos(\phi + \frac{2\pi}{2}) & \cos(\phi + \frac{3\pi}{2}) & \cos(\phi + \frac{4\pi}{2}) \\ \sin(\phi + \frac{\pi}{2}) & \sin(\phi + \frac{2\pi}{2}) & \sin(\phi + \frac{3\pi}{2}) & \sin(\phi + \frac{4\pi}{2}) \\ \frac{1}{R} & \frac{1}{R} & \frac{1}{R} & \frac{1}{R} \end{bmatrix} \quad (7)$$

Fb_i' is the feedback distance before a error happen and described by

$$Fb_i' = \frac{Fb_i}{1 \pm e_{Fb_i} \%}, i=1,2,3,4 \quad (8)$$

where e_{Fb_i} is a random error rate of feedback. In this paper, the maximum error rate is determined by the experiment. The result of simulation 10,000 particles by the model is shown in Figure 8, where the accumulation error distances follow the moving distance as longer as bigger, and particles distribution

becomes wider. These particles position probably is robot's position. The simulation particles as robot moving by action model and the motor feedbacks to estimate many situations of error happening to estimate posture of the robot are called *Expectation*.

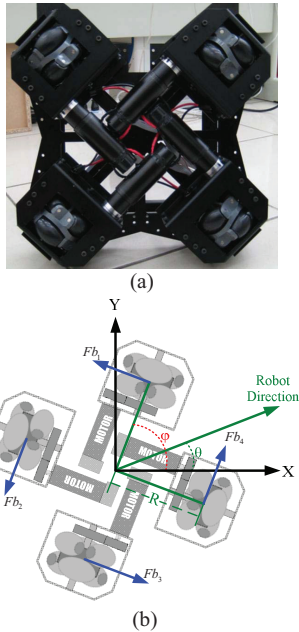


Figure 7. Robot wheels setting. (a) Real robot, and (b) diagram of the wheels setting of the robot.

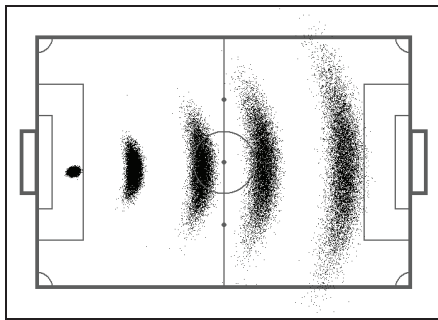


Figure 8. Particles moving by feedback.

B. Observation sensor model

The simulation observation sensor values can get by a virtual map. The simulation values are idea values of the sensor posture. According to the sensor error distribution, the probability of the sensor value and the idea sensor value can be calculated. Observation model is used to simulate estimation posture's environment information, and compare with the robot detected environment information to calculate the belief of estimation posture. In this paper, distances of the field lines around the robot are detected as the environment information [8] as shown in Figure 9, where the black lines is detected lines in the omnidirectional image from the Internal Radius of the setting radioactive to detect field lines' edges. The black lines' pixel radius can translate to distance by the pixel radius and distance relationship model. The distances are environment information.

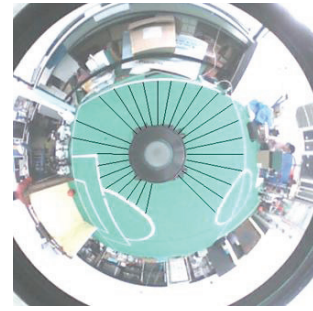


Figure 9. Environment observation by the omnidirectional vision system.

These distances sensor errors are normal distribution that is hypothesized in this paper. One of detected distance's belief is gotten by

$$p(o_i | l) = N(o_i(l) - o_i', \sigma^2) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(o_i(l) - o_i')^2}{2\sigma^2}\right) \quad (9)$$

where $o_i(l)$ is the distance obtained by the estimated posture l from the virtual map, o_i' is the distance detected by the robot sensor, σ is a standard deviation of the sensor. The belief is calculated by the difference between the estimation observation and the robot observation. The pixel's standard deviation is determined by experiment. According to the pixel radius and distance relationship is non-linear, so standard deviations are difference of each estimation distance. Pixel radius and standard deviation relationship diagram is shown in Figure 10, where the detected pixel error standard deviation σ_p is fixed, $d1$ and $d2$ is estimation distance, standard deviation σ_{d1} and σ_{d2} can be calculated by the relationship of pixel radius and distance. Difference standard deviation cause difference normal distribution function and cause the belief of near information more important than far information.

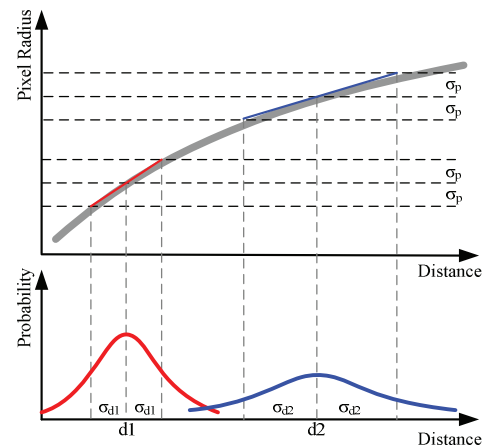


Figure 10. Relationship of the pixel length and standard deviation.

An omnidirectional image can provide many directions detected values. They are independent. Therefore particle's belief is multiplication [5] of each detected value of robot and the simulation detected value of particle's estimation posture, which can be described by

$$p(o_i | l_i) = \prod_{i=1}^n p(o_i | l_i) = \prod_{i=1}^n N(o_i(l_i) - o_i', \sigma_i(l_i)^2) \quad (10)$$

where n is number of detected number. The belief of the estimated posture is used to pick particles. According to the obtained particle's belief, some particles with a lower belief value (i.e. they locate on an impossible area) can be removed. Keeping particles to locate on a possible area is called *Correction*.

VI. EXPERIMENTAL RESULTS

A. Belief distribution

In this section, the results of belief distribution from the omnidirectional images are shown in Figure 11, where particles are distributed on the map of each 10 cm and are set the same direction as the robot to test the robot locates on the right direction's belief distribution. The obtained omnidirectional image at this posture and the detected information are shown in the left side of Figure 11. The belief distribution of particles is shown in the right side of Figure 11. These results illustrate that the peak of belief distribution is very clear when the direction of particle is the same as that of the robot. In Figure 11 (a), only one high belief area in the belief distribution. In Figure 11 (b) and (c), there can show some place of map where similar character of robot detected sensor values has. The situation may cause particle gather on multiple area.

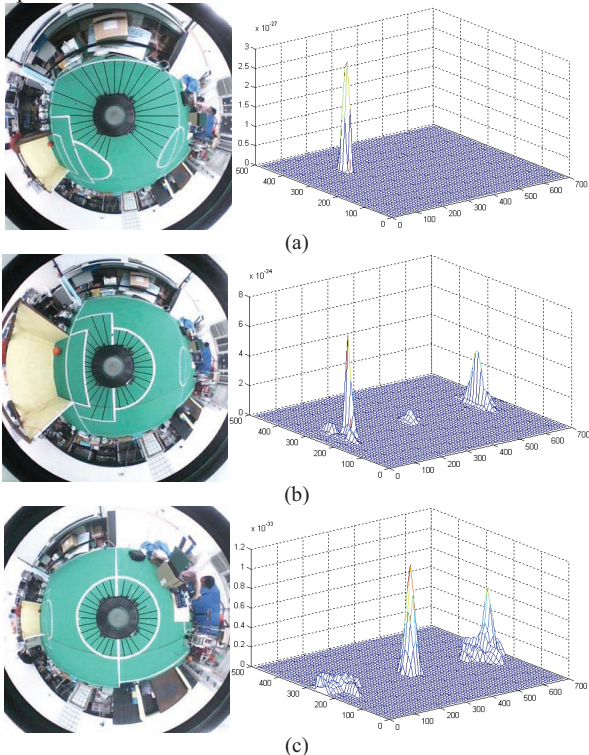


Figure 11. Belief distribution. (a) robot is on (200, 350, 0), (b) robot is on (125, 250, 0), and (c) robot is on (350, 250, 0).

B. Program Procedure

There are two states: expectation and correction in the program procedure. The state is called expectation when the motor's feedback is used to simulate the robot action by the particle. The state is called correction when the belief of particle is calculated and keep particles with a higher belief value are kept. The environment information will be changed by the robot's action, thus particles move by the expectation. Then possible particles are kept by the correction. The process procedure will be repeated many times so that the posture of these staying particles may be the robot posture. The program procedure is described in Figure 12. The Initial state is to random set particles' postures on the field. The Correction state is to keep possible particles. After that, the program will stay in Expectation until moving distance over a range that is called Correction distance. The distance is depended on the field size for the distance can cause the environment character has big change [9] or feedback error for the distance can cause a big error should to correct again. More particles to describe the action model cause the model more complete so that the removed particles will be cast back near a possible area in this program. A successful localization result of the procedure is shown in Figure 13, where the black dots represent particles' position, the red lines represent particles' direction and the white circle and the gray line represent the robot posture. First, the initial 1,000 particles are set on the virtual field as shown in Figure 13 (a). Second, particles distribution after the correction is shown in Figure 13 (b). Then particles distribution after the expectation and correction are shown in Figure 13 (c), (d), (e) and (f). These results illustrate the robot moves from the corner to in front of goal and then moves to the other side of field.

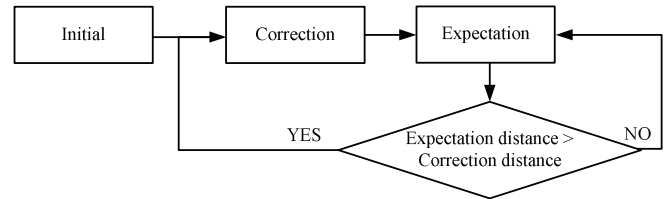
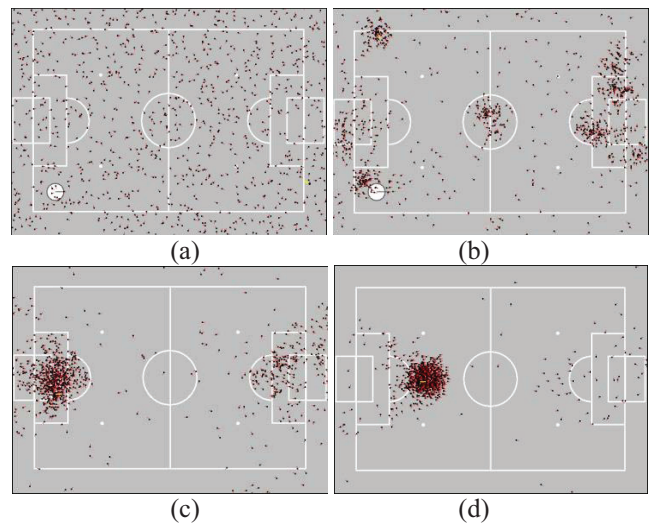


Figure 12. Program procedure of the particles filter.



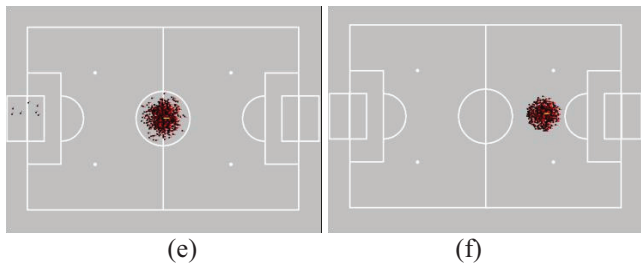


Figure 13. Particles distribution results of the program procedure.

C. Robot posture tracking.

In this section, the tracking result of the robot posture is shown in Figure 14, where the red line is the real path of the robot on the field and the blue line is the localization result. In this case, 1000 particles are used to do the correction for each 1 m movement. The localization posture and the robot posture are overlap so that this is one successful case. When particles to gather to a group, the method changes to tracking the robot posture. The soccer field is symmetrical, so a high belief area also happens in the opposite side. As shown in Figure 15, it will let the obtained result is an opposite wrong posture.

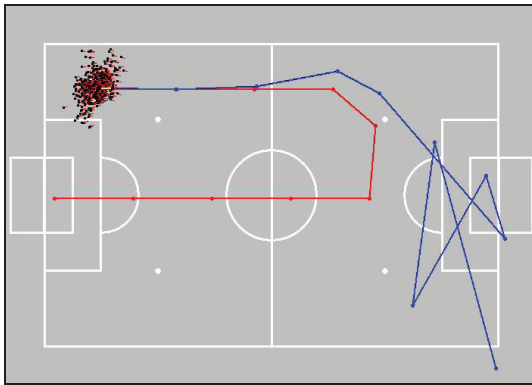


Figure 14. Posture of the robot tracking by 1,000 particles.

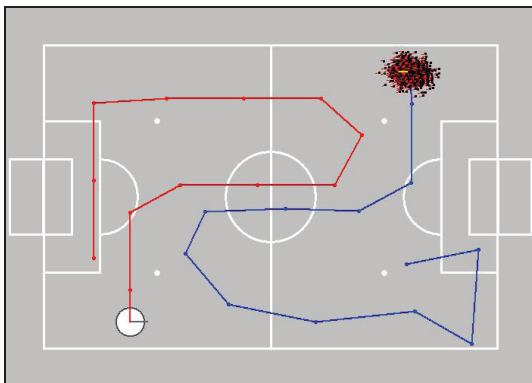


Figure 15. Symmetrical posture of the robot tracking by 1,000 particles.

In this paper, the localization has two situations: searching and tracking. When particles still gather on many areas is searching. In this situation, localization results are always wrong and unstable. Searching situation needs a large number of particles on the virtual map to let the system get many samples of posture and have a big chance to find the correct result. It also causes the system stands when the processing is

correct. When particles are convergent, correction is doing on a small area, the particle density becomes dense to let the result become more correct.

VII. CONCLUSIONS

In this paper, an omnidirectional vision-based localization method based on the particle filter is proposed to estimate the location of the robot on a soccer field. The omnidirectional vision system is used to be the observation sensor. When the object is far away from the robot, its measured distance based on the omnidirectional image will have a big error. In this paper, an adjustment method of the standard deviation is proposed to improve this problem. The proposed method will be used on a soccer game. Owing to the symmetrical posture will happen on the soccer field, it will be a very serious problem that our robots attack the own goal. In the future, two ways are planned to solve this problem. First, players only can stay on the half side of their goal in the beginning of the soccer game according to the rules. Thus particles can be set only on the half field to solve the problem. The other way is to use more environment information to provide the information of field side such as the background of goal or goal keeper.

ACKNOWLEDGMENT

This research was supported in part by the National Science Council (NSC) of the Republic of China under contract NSC 98-2218-E-032-005.

REFERENCES

- [1] J. Wolf, W. Burgard, and H. Burkhardt, "Using an image retrieval system for vision-based mobile robot localization", IEEE International Conference on Robotics and Automation, Vol. 1, pp.359-365, 2002.
- [2] S. Thrun, D. Fox, W. Burgard, and F. Dellaert, "Robust Monte Carlo localization for mobile robots", Artificial Intelligence, Vol. 128(1-2), pp. 99-141, 2001.
- [3] E. Menegatti, M. Zoccarato, E. Pagello, and H. Ishiguro, "Hierarchical image-based localization for mobile robots with monte-carlo localization", European Conference on Mobile Robots (ECMR'03), pp.13-20, Sep. 2003.
- [4] E. Menegatti, M. Zoccarato, E. Pagello, and H. Ishiguro, "Image-based monte-carlo localization with omnidirectional images", Robotics and Autonomous Systems, Vol. 48 (1), pp.17-30, 2004.
- [5] F. Dellaert, D. Fox, W. Burgard, and S. Thrun, "Monte Carlo localization for mobile robots", IEEE International Conference on Robotics and Automation, Vol. 2, pp.1322-1328, 1999.
- [6] URL: <http://www.fira.net>
- [7] Sooyeong Yi, Byoungwook Choi, and Narendra Ahuja, "Real-time Omni-directional Distance Measurement with Active Panoramic Vision", International Journal of Control, Automation, and Systems, Vol. 5, No. 2, pp. 184-191, 2007
- [8] E. Menegatti, A. Pretto, A. Scarpa, and E. Pagello, "Omnidirectional vision scan matching for robot localization in dynamic environments", IEEE Transactions on Robotics, Vol.22 (3), pp.523-535, 2006.
- [9] S. Enderle, M. Ritter, D. Fox, S. Sablatng, G. Kraetzschmar, and G. Palm, "Soccer robot localization using sporadic visual features", Proc.edings of the 6th International Conference on Intelligent Autonomous Systems (IAS-6). IOS Pr