2022 International Seminar on Intelligent Technology and Its Applications (ISITIA)

Soccer Robot Localization Based on Sensor Fusion From Odometry and Omnivision

Muhammad Azhar Ismail
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
azhar.206022@mhs.its.ac.id

Djoko Purwanto
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
djoko@ee.its.ac.id

Achmad Arifin
Department of Electrical Engineering
Institut Teknologi Sepuluh Nopember
Surabaya, Indonesia
arifin@bme.its.ac.id

2022 International Seminar on Intelligent Technology and Its Applications (ISITIA) | 978-1-6654-6081-1/22/$31.00 ©2022 IEEE | DOI: 10.1109/ISITIA56226.2022.9855313

***Abstract*— Mobile robots have ability to move their entire body and doing some tasks automatically. One of them is RoboCup Middle Size League (MSL) soccer robot. Main systems in the MSL robots are a self-localization. Localization system is very important because it can help some important aspects such as ability to navigate and control. The common localization method of mobile robots is odometry because the process is fast, but the weakness is that the error from odometry will increase over time due to gyro drifting and encoder wheels are slipping. Meanwhile, MSL robots generally use omnivision with the particle filter method for localization. Localization error with omnivision does not increase over time but requires heavy computational processing. Therefore, a sensor fusion system was designed to combine odometry and omnivision. Thus, they can cover each other's disadvantages, and make localization become more accurate. From the experimental results on the soccer field with size 9 m × 6 m, sensor fusion can provide good localization data. The localization error results $x = 10.5 \pm 7.8\ cm$, $y = 7.6 \pm 6.8\ cm$, and $\theta = 1.9 \pm 1.2\ °$, with average time response 1.6 ms . This system is expected to give soccer robot more accurate localization in Robocup MSL matches and help robot navigation when avoiding obstacles.**

Keywords*—*soccer robot, self-localization, odometry, omnivision, particle filter, sensor fusion

I. Introduction

Mobile Robot is a robot that has ability to move places and do certain jobs automatically. Besides being used to help human work, mobile robots can also be used in various competitions. One of the most popular mobile robot competitions today is RoboCup Middle Size League (MSL). MSL competition uses a mobile robot to play soccer like a human soccer game. One important feature that MSL robot must have in the competition is self-localization, to estimate position and orientation automatically and independently.

There are many methods commonly used in mobile robots to estimate its position. One of the commonly used methods is odometry. Odometry system usually use combination of encoder sensor and gyro or compass sensor to provide better location results. In the MSL soccer robot, this method become less effective because the soccer robot often moves around soccer field during the game, therefore the errors accumulate throughout the game because encoder wheels are slipping and gyro drifts.

Most MSL soccer robots use omnivision cameras to recognize condition of their surroundings. With an omnivision camera, the robot is able to get images with a 360° viewing angle. Particle filters are usually used to estimate the robot's position and orientation based on image data from omnivision. However, filter particles require a heavy computational process.

In research [1], the robot uses Convolutional Neural Network (CNN) with 360° omnivision sensor for self-localization on soccer field. The soccer field divided into equally sized. Every section has identical shape, but some sections usually have similar images that can make localization mistake. To overcome that problem, this research uses visual modelling of the gyrocompass line mark and omnivision image as CNN main feature. The accuracy of this system is quite low, the best accuracy results only 0.6827.

Research [2] proposed a particle filter or also known as Monte Carlo localization to estimate the location of a soccer robot using an omnivision camera sensor. This method requires a map database to estimate its location. The main feature used by this method is the white line in field image captured by an omnivision camera. The distance obtained from each line is then compared with the location of the particles that have been spread previously. The value of the highest degree of confidence for a particle is used as the robot's reference location. This method also uses motor speed as motion model of particle to estimate robot location. But this method is also prone to symmetrical field shapes, where the two sides of the field have a similar pattern and can cause reverse localization result.

The method proposed in the research [3] is pattern match system. This system uses the distance between robot and field line. This system also requires a database containing the distance from each scan line at each position and orientation of the robot in the field. So, to estimate location of robot in the field, that is by calculating the magnitude of the error from the distance obtained from omnivision with distance data stored in the database. The result of the smallest error will be used to estimate the location of the robot in the field. This system has a computation time of about 5ms.

While in research [4], The proposed method is matching optimization. This system requires the white line distance to the robot and a database of field line distance errors to calculate the probability of the scattered particles. In addition, the location data from the particles are also combined with the encoder data from the robotic motor. The disadvantage of this system is that when omnivision gets a little line distance data, the error in the estimation of the robot's location becomes large because the robot's motor wheel slips easily, therefore the motor encoder becomes less accurate.

In this research, localization system designed using combination data from odometry and omnivision through sensor fusion. Thus, disadvantages of odometry and omnivision can cover each other. Then, make localization data become more accurate.

II. Method

Overall, the designed system can be seen in Figure 1. There are several sensors that are used to recognize the surrounding conditions. The encoder and gyro sensors are used to recognize the displacement of the robot, two encoders

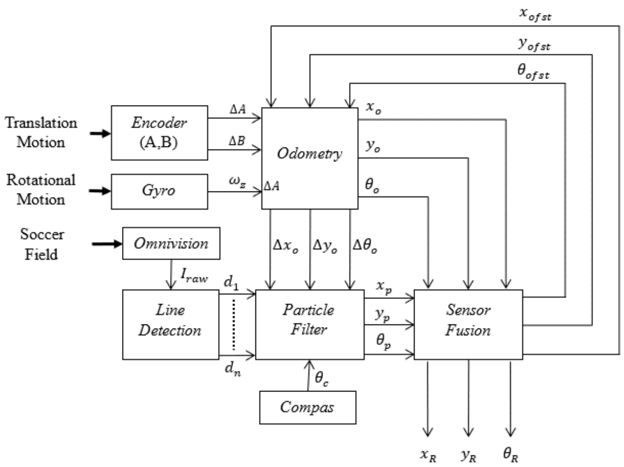


Figure 1. Block Diagram Overall System

Use to sense translation motion and gyro to sense rotational motion of robot body. Omnivision use to capture images of the environmental conditions around the robot, main task of this sensor for localization is to detect white lines in soccer field, so there is an image processing in omnivision raw image to generate white lines data before used in particle filter process, and compass sensor is used to assist particle filter process. Each of these sensors has a different data output, so it takes a different computational process or method so that the resulting data can be the same before being received by sensor fusion and offset data from sensor fusion is used to correct errors from odometry. Odometry and particle filter methods have the same output data with different inputs. The two methods have their advantages and disadvantages. Therefore, to cover each other's weaknesses, the output data from the two methods are then combined in the sensor fusion process. The location of the sensor used in the robot can be seen in Figure 2. Omnivision is placed at the top to captured image from entire field. The encoders are at the bottom of the robot to measure the displacement of robot.

A. Odometry System

Odometry uses data from actuators or motion sensors to estimate changes in the position and orientation of an object. Odometry is often used in mobile robots for localization. The sensors used in odometry generally consist of several sensor data which are combined and calculated to estimate the location, the sensors used can also be different. Currently there are many odometry methods that have been developed. In omni-directional mobile robots usually use an odometry system from a combination of a pair of encoders and a gyro sensors [5] [6]. The configuration of a pair of encoders on the omni-directional mobile robot forms an angle of 90° to provide position displacement data on the x-axis and y-axis as shown in Figure 3. Meanwhile, the gyro sensor serves to provide angular velocity data on the z-axis.

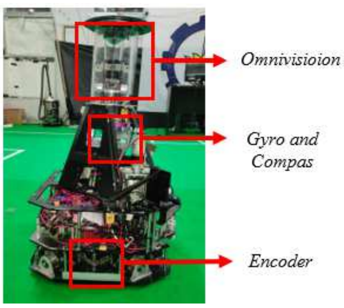


Figure 2. IRIS Soccer Robot

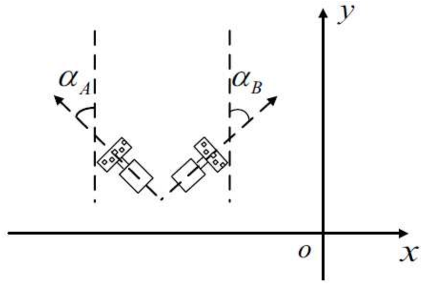


Figure 3. Odometry System with 2 Encoders [5]

To obtain angular displacement data based on the z-axis angular velocity given by the gyro, it can be calculated as equation (1).

$$\Delta\theta_o = \int_0^t \omega_z \,.\, dt \tag{1}$$

Where ω_z is z-axis angular velocity data from the gyro. Meanwhile, to calculate the displacement on the x-axis and y-axis can be stated as in equations (2) and (3).

$$\Delta x_o = \frac{\Delta A.Cos(\alpha_B + \theta_0) + \Delta B.Cos(\alpha_A + \theta_0)}{Sin(\alpha_B - \alpha_A)} \tag{2}$$

$$\Delta y_o = \frac{\Delta A.Sin(\alpha_B + \theta_0) + \Delta B.Sin(\alpha_A + \theta_0)}{Sin(\alpha_B - \alpha_A)} \tag{3}$$

Where ΔA and ΔB are rotational change from each encoder wheel at certain period of time. α_A and α_B are wheel mounting angle characteristics, while θ_0 is heading angle of robot at field based on odometry system. In addition, the process to estimate position and orientation of the robot from the odometry system can be seen in Figure 4. Odometry requires an offset value to determine the robot's initial location in the field and to compensate for errors from the odometry. For initial location, offset value are get from sensor fusion system, which come from particle filter system. Thus, the location data in the next process will not wrong.

Encoder sensors are used to measure the distance the robot moves on the x and y axis, while the gyro is used to measure the angular displacement. Data from the gyro also cannot be directly processed by odometry. The data must first be converted into the facing angle of the robot. Thus, the displacement data provided by the encoders can be translated into the correct location. To estimate the localization of the robot with the odometry system, it can be seen in equations (4), (5), and (6).

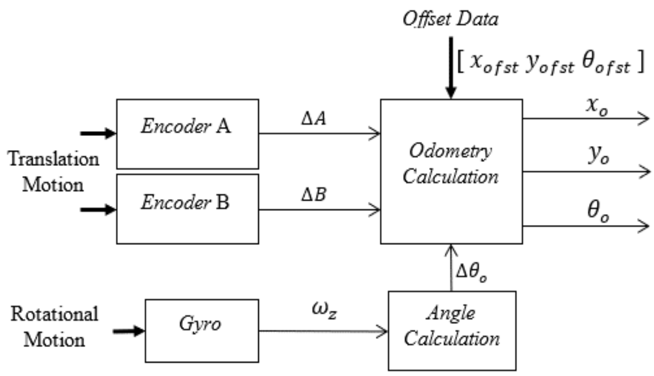


Figure 4. Block Diagram Odometry System

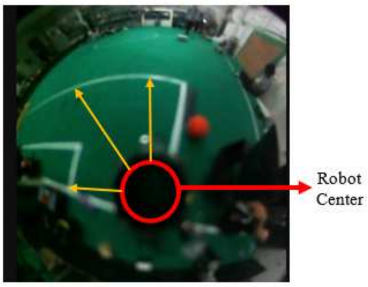


Figure 5. Captured Image from Omnivision

Where Δx_o, Δy_o, and $\Delta\theta_o$ are robot location displacement data, while x_{ofst} , y_{ofst} , and θ_{ofst} are offset data to compensate initial location of robot and error from odometry.

$$x_o = \left(\sum \Delta x_o\right) + x_{ofst} \quad (4)$$

$$y_o = \left(\sum \Delta y_o\right) + y_{ofst} \quad (5)$$

$$\theta_o = \left(\sum \Delta\theta_o\right) + \theta_{ofst} \quad (6)$$

Δx_o, Δy_o, and $\Delta\theta_o$ always added in every odometry data update. While x_{ofst}, y_{ofst}, and θ_{ofst} are constant and the value can be changed if there is an offset update.

B. Line Detection System

One of the important data from the designed localization system is captured image from omnivision. Omnivision is typically made up of a reflective 360° omni-directional mirror, lens, and digital camera [7]. Figure 5. is captured image from omnivision, the captured image is an image of the field conditions around the robot. When viewed in the omnivision image capture, there is a black dot in the center of the image. The black dot is the position of the robot body or also can be called the center of the robot. Before the image can generate an estimation of robot location, it is necessary to have an image processing process first. This process aims to detect and measure how far the white lines of the field from the center of the robot.

To determine white lines of the field from an omnivision image, it takes some image processing first. So that, field lines data obtained to be more accurate, it is necessary to preprocess the raw images obtained by omnivision. This preprocessing stage can be seen in Figure 6. To detect white line features from the field, the raw image from omnivision is divided into two processes. The first is the HSV Threshold and Convex Hull processes which function to detect green fields from raw omnivision image [1], while the HSL Threshold functions to detect white color in raw omnivision images. The results of each process are combined with the Bitwise AND method, this helpful to eliminate the white color on the outside green field.

After the feature of white lines in the field are obtained, then continues calculating pixel distance from robot center and line that has been detected at each predetermined angle.

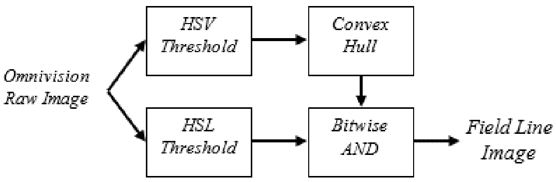


Figure 6. Block Diagram Omnivision Preprocessing

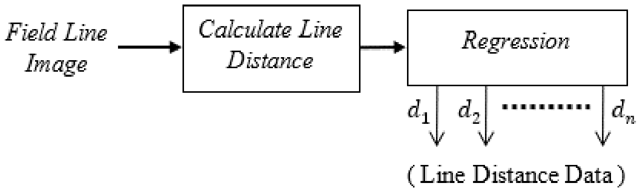


Figure 7. Block Diagram Line Distance Measurement

Each pixel distance obtained are regressed. Thus, the lines distance data obtained are similar to real situation. This process can be seen in Figure 7.

C. Particel Filter System

Particle Filter Localization or commonly known as Monte Carlo localization (MCL) is an algorithm or method used to estimate the location of a robot using particles. MCL has ability to estimate location of the robot initial position data in its working environment. However, this localization requires a map of the working environment of the robot to estimate the position and orientation of the robot in that environment. The particle filter uses particles that are randomly distributed at each coordinate point on the map. MCL is divided into 3 important parts, namely motion model, observation model, resampling [8]. To estimate the location of robot with particle filter, it can be seen in Figure 8. This process uses particles that are initialized randomly at several locations in the field. Each particle that are distributed will be calculated its probability to estimate the location of the robot.

The number of particles used are 600 with a scan radius every 2.5°. Angle data from Compass is only used to help initialize particles randomly, Therefore, the location of the final estimation result is not reversed, after the particles are spread out, then the particles are moved based on the robot movement data from odometry. After that, each particle's locations are combined with the obtained lines distance data to estimate the global coordinates of the detected line distance. These coordinates are used to retrieve the error value on the database map that has been created as in the research [4]. When the database map is drawn, it will look like in Figure 9. Figure 9(b) is error distribution map from soccer field 9(a) with horizontal is y-axis and vertical is x-axis. The black color equal to 0 and white equal to 255. Color between black and white have value between 0-255. Black color indicates coordinates of white lines in soccer field. Farther the coordinates from black color, the color will change to white gradually. If detected line point is on white, it means that detected line has big error.

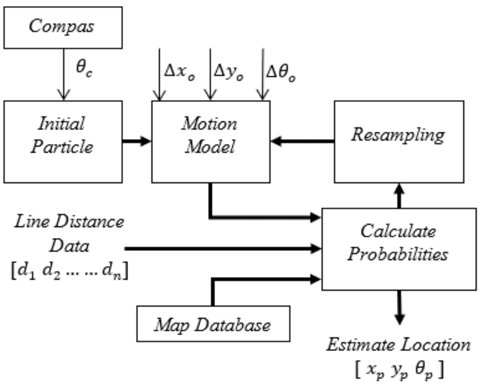
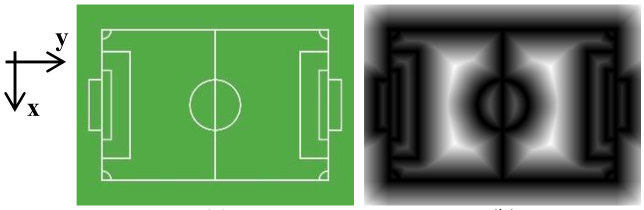


Figure 8. Block Diagram Particle Filter System



(a) (b)

Figure 9. Soccer Field Map(a), Error Distribution Map(b)

If o_i is global coordinate from detected line point and $[o_i^x \ o_i^y]$ is the relative coordinate of line distance data to the robot, so that o_i can be calculated with equation (7).

$$o_i = \begin{bmatrix} x_j \\ y_j \end{bmatrix} + \begin{bmatrix} cos\theta_j & -sin\theta_j \\ sin\theta_j & cos\theta_j \end{bmatrix} \begin{bmatrix} o_i^x \\ o_i^y \end{bmatrix} \quad (7)$$

Where $[x_j \ y_j]$ is robot position, while θ_j is the orientation from scattered particles. After the coordinates of the database error obtained, then calculate weight from error value, if f_i is detected line distance and L_j is location of particle, then the weight of the line distance data at a particular particle location can be calculated according to equation (8).

$$P(f_i|\, L_j) = exp\left(\frac{-d(o_i)^2}{2\sigma^2}\right) \quad (8)$$

Where $d(o_i)$ is database error value based o_i coordinate and σ^2 is constants that can be set to get optimal calculation results. Meanwhile, to calculate the probability of each particles from lines distance data, can be seen in equation (9).

$$P(o|L_j) = P(f_1|\, L_j)\,.P(f_2|\, L_j) \dots\dots\dots . P(f_n|\, L_j) \quad (9)$$

Probability of each particle is normalized to get value between 0 to 1. Next step is select several high probability particles using low variance resampling method. This method can be seen in Figure 10. where r is random number between 0 until N^{-1} with N is number of particles used and w_k^i is weight from each particle. Particle with weight less than u will be eliminated, while the larger weight will be duplicate [9], the value of u obtained from the equation (10).

$$u = r + \frac{n}{N} \quad (10)$$

where n = 1, 2, …., N. Meanwhile, to estimate position of the x and y axis can be seen in equations (11) and (12).

$$x_p = \frac{1}{N}\sum_{j=0}^{N} L_j^x \quad (11)$$

$$y_p = \frac{1}{N}\sum_{j=0}^{N} L_j^y \quad (12)$$

Meanwhile, to estimate the optimal orientation of the particle filter system, it can't be calculated like $x_p$and y_p. Therefore, the angle can be written as equations (13) and (14).

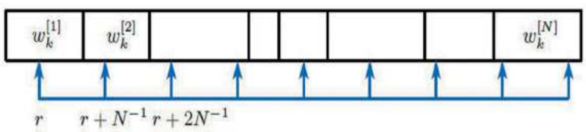


Figure 10. Low Variance Resampling Method [9]

$$f(y,x) = \begin{cases} tan^{-1}\left(\frac{y}{x}\right), & x > 0 \\ tan^{-1}\left(\frac{y}{x}\right) + \pi, & x < 0 \wedge y \geq 0 \\ tan^{-1}\left(\frac{y}{x}\right) - \pi, & x < 0 \wedge y < 0 \\ \frac{\pi}{2}, & x = 0 \wedge y > 0 \\ -\frac{\pi}{2}, & x = 0 \wedge y < 0 \\ undefined, & x = 0 \wedge y = 0 \end{cases} \quad (13)$$

$$\theta_p = f\left(\frac{1}{N}\sum_{j=0}^{N} Sin(L_j^\theta), \frac{1}{N}\sum_{j=0}^{N} Cos(L_j^\theta)\right) \quad (14)$$

Where $[\,L_j^x \ L_j^y \ L_j^\theta\,]$ is location from particles with high probability and $[x_p \ y_p \ \theta_p]$ is robot location estimation based on omnivision. The particles are then resampled for reuse in the next process.

D. Sensor Fusion System

In order to create optimal localization data from odometry and omnivision, sensor fusion will play an important role to combining position and orientation data that has been obtained from the odometry localization system and particle filter localization system. There are several types of sensor fusion techniques such as comparison, merging, and smart voting between sensors [10]. By considering the advantage dan disadvantage of each system, this research used a method of competitive fusion and complementary fusion which can overcome the disadvantage of each system. The algorithm of the sensor fusion used can be seen in Figure 11. For omnivision mode, data $[\,x_R \ y_R \ \theta_R\,] = [\,x_p \ y_p \ \theta_p\,]$, and for mode odometry the localization data become $[\,x_R \ y_R \ \theta_R\,] = [\,x_o \ y_o \ \theta_o]$, these two modes are the result of competitive sensor fusion because the data is taken independently. While for complementary sensor fusion is on omnivision + odometry mode.

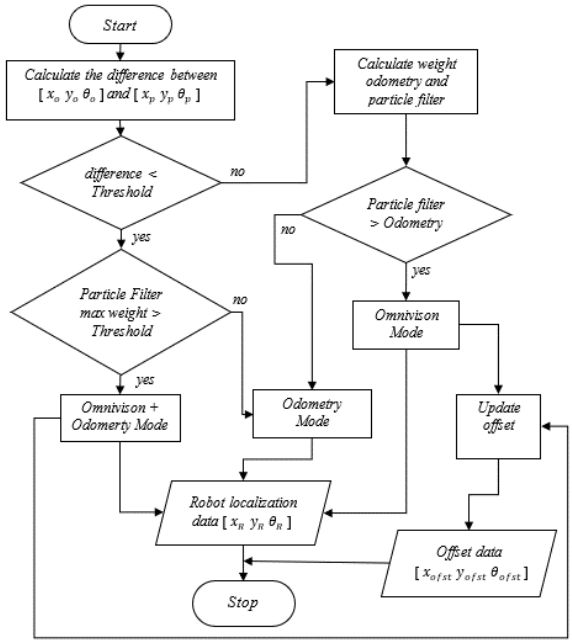


Figure 11. Sensor Fusion Algorithm

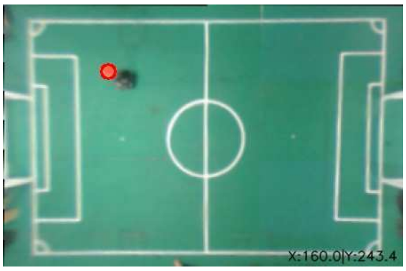


Figure 12. Field Image from Global Camera

In omnivision + odometry mode, robot localization data $[\, x_R \; y_R \; \theta_R \,]$ can be calculated like equation (15), (16), and (19).

$$x_R = \beta x_o + (1 - \beta) x_p \tag{15}$$

$$y_R = \beta y_o + (1 - \beta) y_p \tag{16}$$

$$\theta_R^S = \beta Sin(\theta_o) + (1 - \beta) Sin(\theta_p) \tag{17}$$

$$\theta_R^C = \beta Cos(\theta_o) + (1 - \beta) Cos(\theta_p) \tag{18}$$

$$\theta_R = f(\theta_R^S, \theta_R^C) \tag{19}$$

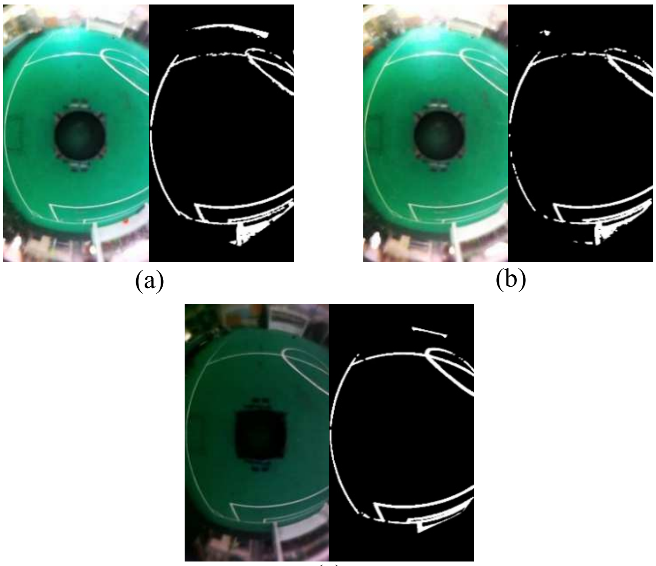
Where β is constant between 0 and 1, and function f can be seen in equation (13).

III. Experimental Results

There are several experiments that have been carried out on the design of this localization system start from experiments in line detection results to experiments for estimate global coordinates of the robot in the field, the soccer field used has a size of 9 m × 6 m as shown in Figure 12. For the robot localization experiment, the final data generated by the system will be compared with the data obtained from the global camera, as shown in Figure 12.

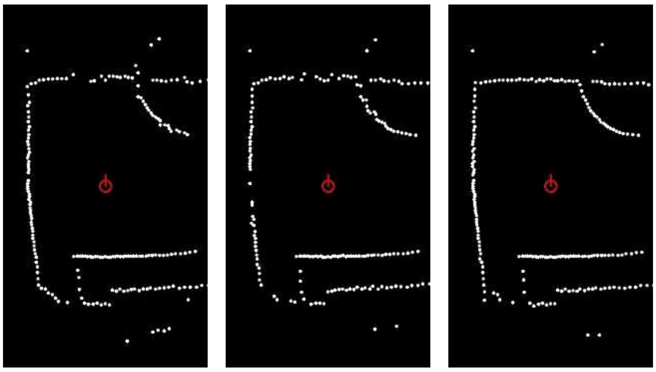
A. Line Detection Result

Experiments for line detection were carried out in 3 different lighting conditions with the same robot location, the optimal results of the designed detection system can be seen in Figure 13.



(a) (b) (c)

Figure 13. Line Detection Result at Condition 1(a), Condition2(b), Condition 3(c)



(a) (b) (c)

Figure 14. Regression Result Condition 1(a), Condition 2(b), Condition 3(c)

The regression results based on the line detection results from Figure 13 can be seen in Figure 14. Based on Figure 13 and Figure 14, it can be seen, that line detection and regression results from condition 3 are smoother, this happens because the environmental light is dim and low white noise, making it easier to detect lines, while in conditions 1 and 2, bright environmental lighting makes line detection little difficult.

B. Robot Global Localization Result

In this experiment the robot is moved manually to 20 coordinate points in a randomly determined field, the location of the 20 coordinate points can be seen in Figure 15. In addition, the movement of the robot from points 1 to 20 is also tracked to compare with data from global camera. After the localization results were measured at these 20 coordinates, then the mean error value, standard deviation error, and maximum error from the results of the odometry system, particle filter system, and sensor fusion system were calculated at all test points. The results of these calculations can be seen in Table I.

TABEL I. THE STATISTICS OF LOCALIZATION ERRORS

		$x(cm)$	$y(cm)$	$\theta(°)$
Odometry System	Mean error	24.1	35.6	2.7
	Standard dev	18.4	17.1	2.0
	Max error	53.5	73.3	6.2
Particle Filter System	Mean error	5.7	8.3	2.2
	Standard dev	5.8	6.3	1.7
	Max error	25.1	19.9	5.5
Sensor Fusion System	Mean error	10.5	7.6	1.9
	Standard dev	7.8	6.8	1.2
	Max error	26.5	21.8	4.4

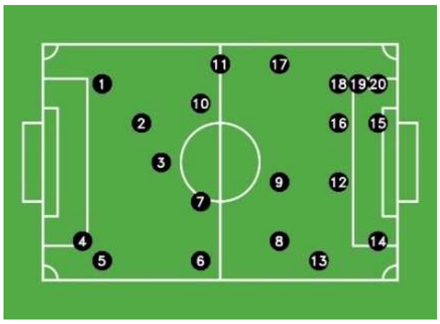
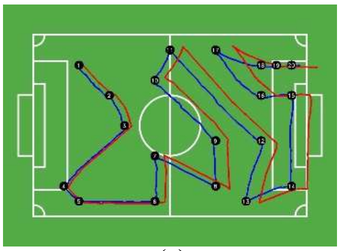
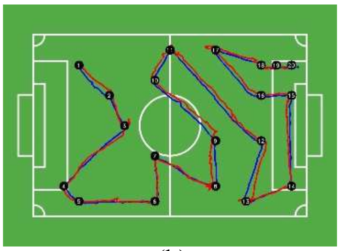


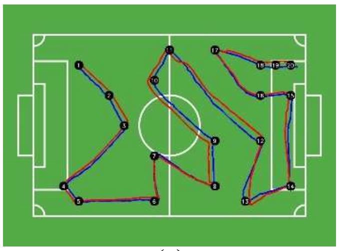
Figure 15. 20 Coordinates Location for Test



(a)



(b)



(c)

Figure 16. Localization Data from Odometry System (a), Particle Filter System (b), and Sensor Fusion System(c)

Based on the data from Table I, sensor fusion system can provide accurate localization data by combining odometry and the particle filter. While the results of localization and movement of the robot based on the global camera can be seen in Figure 16. The blue line is data from the global camera while the red line is the localization data. It is seen that the error of the odometry system will increase. While the particle filter and sensor fusion system errors do not increase, but in sensor fusion data are smoother.

C. *Sensor Fusion Result*

The results of the sensor fusion system algorithm can be seen in Figure 17. The robot is moved manually from point A to point B, the cyan color is omnivision or omnivision + odometry mode and the magenta color is odometry mode, while the black line is robot movement data from the global camera. From the results obtained, it appears that the odometry mode is more dominant. This system has average time respone 1.6 ms for each localization data update.

IV. Conclusion and Discussions

Based on the research results obtained, sensor fusion can produce optimal data by combining odometry and omnivision with particle filter system. From experiment results on $9\ m \times 6\ m$ soccer field, it can be said that this system is able to provide optimal localization. The localization errors result $x = 10.5 \pm 7.8\ cm$, $y = 7.6 \pm 6.8\ cm$, and $\theta = 1.9 \pm 1.2\ °$. with average time response 1.6 ms. These results are better than just using odometry system only and the localization data are smoother than particle filter system only.

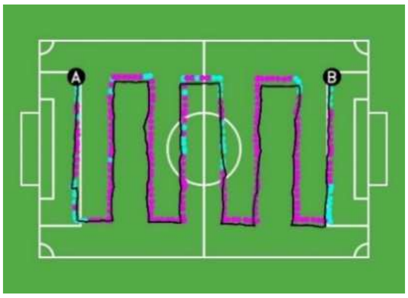


Figure 17. Sensor Fusion Algorithm Results

This system also can compensate odometry errors. Therefore, the errors do not increase every time and expected can give accurate localization in Robocup MSL. But this system still need improvement for line detection because threshold parameters need to be set manually when robot placed under different lighting condition to get good detection. In addition, line regression results are still less accurate and sometimes can make particle system less accurate too. Furthermore, this localization system can be used to help robot navigation when avoiding obstacle in the field.

Acknowledgment

This research was supported by IRIS team from Sepuluh Nopember Institute of Technology (ITS). IRIS team is one of the ITS robotics teams that focuses on developing Middle Size League (MSL) wheeled soccer robot since 2017.

References

[1] R. Dikairono, S. Setiawardhana, D. Purwanto, and T. Sardjono, "CNN-Based Self Localization Using Visual Modelling of a Gyrocompass Line Mark and Omni-Vision Image for a Wheeled Soccer Robot Application," *IJIES*, vol. 13, no. 6, pp. 442–453, Dec. 2020, doi: 10.22266/ijies2020.1231.39.

[2] C.-Y. Chen, C.-Y. Ho, C.-C. Wong, and H. Aoyama, "Omnidirectional vision-based robot localization on soccer field by particle filter," p. 6.

[3] S.-Y. Chiang, X. Guo, and H.-W. Hu, "Real time self-localization of omni-vision robot by pattern match system," in *2014 International Conference on Advanced Robotics and Intelligent Systems (ARIS)*, Taipei, Jun. 2014, pp. 46–50. doi: 10.1109/ARIS.2014.6871525.

[4] H. Lu, X. Li, H. Zhang, M. Hu, and Z. Zheng, "Robust and real-time self-localization based on omnidirectional vision for soccer robots," *Advanced Robotics*, vol. 27, no. 10, pp. 799–811, Jul. 2013, doi: 10.1080/01691864.2013.785473.

[5] M. Chuwei, H. Ju, and Z. Zhanyu, "Localization and Navigation Method for Omni-directional Mobile Robot Based on Odometry," Guangzhou, Jul. 2019, p. 6. doi: 10.23919/ChiCC.2019.8865214.

[6] T. Chaudhari, M. Jha, R. Bangal, and G. Chincholkar, "Path Planning and Controlling of Omni-Directional Robot Using Cartesian Odometry and PID Algorithm," p. 6.

[7] B. Liu, J. Fan, J. Zhou, K. Li, and Y. Xie, "A self-localization method through pose point matching for autonomous soccer robot based on omni-vision," in *2009 9th International Conference on Electronic Measurement & Instruments*, Beijing, China, Aug. 2009, pp. 4-246-4–249. doi: 10.1109/ICEMI.2009.5274077.

[8] A. C. Almeida, S. R. J. Neto, and R. A. C. Bianchi, "Comparing Vision-Based Monte-Carlo Localization Methods," in *2018 Latin American Robotic Symposium, 2018 Brazilian Symposium on Robotics (SBR) and 2018 Workshop on Robotics in Education (WRE)*, Joao Pessoa, Nov. 2018, pp. 437–442. doi: 10.1109/LARS/SBR/WRE.2018.00084.

[9] C. Real Ehrlich and J. Blankenbach, "Indoor localization for pedestrians with real-time capability using multi-sensor smartphones," *Geo-spatial Information Science*, vol. 22, no. 2, pp. 73–88, Apr. 2019, doi: 10.1080/10095020.2019.1613778.

[10] G. Jesus, A. Oliveira, A. Azevedo, and A. Casimiro, "Improving sensor-fusion with environmental models," in *2015 IEEE SENSORS*, Busan, Nov. 2015, pp. 1–4. doi: 10.1109/ICSENS.2015.7370654.