Multi-robot Mapping Using Omnidirectional-Vision SLAM Based on Fisheye Images

Yun-Won Choi, Kee-Koo Kwon, Soo-In Lee, Jeong-Won Choi, and Suk-Gyu Lee

This paper proposes a global mapping algorithm for multiple robots from an omnidirectional-vision simultaneous localization and mapping (SLAM) approach based on an object extraction method using Lucas–Kanade optical flow motion detection and images obtained through fisheye lenses mounted on robots. The multi-robot mapping algorithm draws a global map by using map data obtained from all of the individual robots. Global mapping takes a long time to process because it exchanges map data from individual robots while searching all areas. An omnidirectional image sensor has many advantages for object detection and mapping because it can measure all information around a robot simultaneously. The process calculations of the correction algorithm are improved over existing methods by correcting only the object's feature points. The proposed algorithm has two steps: first, a local map is created based on an omnidirectional-vision SLAM approach for individual robots. Second, a global map is generated by merging individual maps from multiple robots. The reliability of the proposed mapping algorithm is verified through a comparison of maps based on the proposed algorithm and real maps.

Keywords: Omnidirectional-vision SLAM, fisheye lens, optical flow.

I. Introduction

With rapid advances in robotics, high-performance robots of various types have been used in diverse fields. In general, a single robot has a practical limitation because it requires high-performance hardware and software systems to perform a task by itself. As a result, using a single robot is difficult in tasks such as exploration, object transport, and cleaning. In such tasks, using multiple robots that are simple and inexpensive can be better than using a single expensive high-performance robot. Multiple robots can perform a given task efficiently through cooperation, such as exchanging information with one another. Multiple robots require various algorithms for avoiding collisions between themselves or obstacles and for facilitating formation, cooperation, and global mapping, among others [1]–[8]. Previous studies have covered such algorithms well.

A multi-robot mapping algorithm creates a global map by using the local maps of individual robots, while maintaining their formation. Formation-control refers to the technology that enables a group of robots to control the shape of their own formation according to the surrounding environment so as to enable them to complete their mission. A large number of robots change their formation to complete tasks according to their purpose and environment based on multiple maps that are generated simultaneously. Studies have examined the formation and mapping-controlled position of each robot based on its location on the global map and its controlled formation through the sharing of information on the global location of each robot [9]–[11]. Few studies on formation control and mapping have proposed methods for creating global maps using information on robots and their environments obtained through cameras attached to the robots. Previous camera-based studies have employed various algorithms. Some studies have

Manuscript received May 14, 2014; revised Sept. 15, 2014; accepted Oct. 2, 2014.

This work was supported by the Agency Specific Research Program of MSIP, Rep. of Korea (Development of Multi-Sensor Platform Technology for Context Cognitive Smart Car).

Yun-Won Choi (yunwon.choi@etri.re.kr), Kee-Koo Kwon (kwonkk@etri.re.kr), and Soo-In Lee (silee@etri.re.kr) are with the IT Convergence Technology Research Laboratory, ETRI, Daegu, Rep. of Korea.

Jeong-Won Choi (choijw@ync.ac.kr) is with the Department of Automatic Electrical Engineering, Yeungnam College of Science & Technology, Daegu, Rep. of Korea.

Suk-Gyu Lee (corresponding author, sglee@ynu.ac.kr) is with the Department of Electrical Engineering, Yeungnam University, Gyeongsan, Rep. of Korea.

http://dx.doi.org/10.4218/etrij.14.0114.0584

detected and controlled the formation of robots using their positions on a global map based on barcode information attached to the side of each robot or on a cylindrical target on top of the robot. Others have controlled robots using omnidirectional cameras by estimating the global positions based on unmanned aerial vehicle camera data, as well as controlling the robot formation by estimating the object information based on omnidirectional images reflected in a mirror [12]–[13].

Existing omnidirectional-vision simultaneous localization and mapping (SLAM) approaches create maps by estimating object locations based on images reflected on paraboloidal mirrors or by using omnidirectional information based on images from fisheye cameras installed toward the ceiling [14]–[18]. However, such methods are subject to light and are limited in that there are parts hidden from the cameras. In this paper, a fisheye camera is attached, to minimize the negative effect of light, at the top of a glass tube installed downward. This camera system is used to obtain information around all robots simultaneously. Existing omnidirectional-vision SLAM approaches correct for local positions by estimating the object locations based on corrected fisheye images. This paper extends these approaches using an object detection method based on *uncorrected* fisheye images.

This paper proposes a fusion algorithm for the global mapping of multiple robots to keep formation. In formation control, it is important to identify surrounding robots and to maintain the formation flexibly in accordance with the surrounding environment. An omnidirectional image obtained by using a fisheye camera is ideal to instantaneously estimate the location of other robots. Existing methods, in general, make use of forward cameras with a field of view of 60 degrees, which means that any robot using such a camera may need to move at least six times before it is able to fully recognize the surrounding environment [19]. In contrast, the proposed method using omnidirectional images can complete the same mission in an instant. The proposed algorithm uses local mappings based on a novel omnidirectional-vision SLAM approach for each robot and employs global mappings using formations based on motions around multiple robots. This novel omnidirectional-vision SLAM approach removes a bottom based on a histogram obtained through a fisheye camera facing downward and labels the objects (robots and obstacles). Motion vectors are obtained using the Lucas–Kanade optical flow (LKOF) method based on the corner points of an extracted object, and then the coordinate of the motion vector is corrected using a forward mapping [20]. Local maps are created using feature point (the outer point of a motion vector) connections of the same label based on the characteristics of the corrected motion vectors. Robot locations are detected by identifying those with the same movement speed on the local map of each robot, and global maps are created while robots maintain their formation. The proposed algorithm is applied to real robots, and the performance of the proposed mapping algorithm is verified by comparing real data on the robot positions with the position data generated by global maps from the proposed algorithm.

II. System Description

1. Overall System

The proposed algorithm processes data from fisheye cameras facing downward and uses the data to create local and global maps of multiple robots, while each robot with a local map maintains its formation. The proposed algorithm has two parts: local mappings for individual robots and global mappings and formation control for robots with local maps. The local mappings from individual robots produce a histogram of the ground information, which is found by analyzing the environmental and ground information obtained through fisheye lenses mounted onto the individual robots. It then clusters information using a labeling algorithm and corrects the coordinates of motion vectors (obtained through feature points in the corners of fisheye images) using forward mappings. It creates global maps (based on the shapes of the objects that disturb the path of the robot) through the motion vector information collected and cluster information. Formation-control and global mappings distinguish between obstacles and robots based on the motion vectors contained in the local maps and process the given tasks while maintaining the formation of the robots. The location of robots is estimated based on a global map generated by merging local maps.

2. Modeling

The robot system used in this paper is modeled in Fig. 1 and

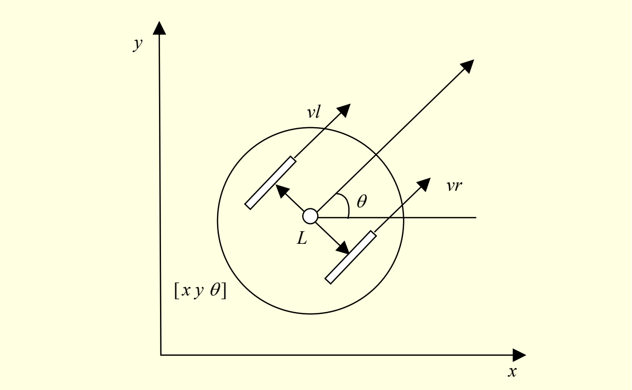


Fig. 1. Robot modeling.

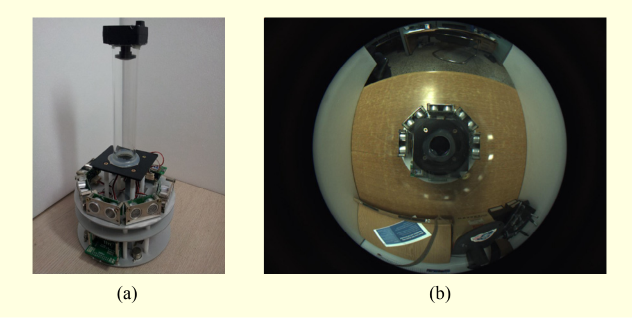


Fig. 2. (a) Robot system and (b) a fisheye image.

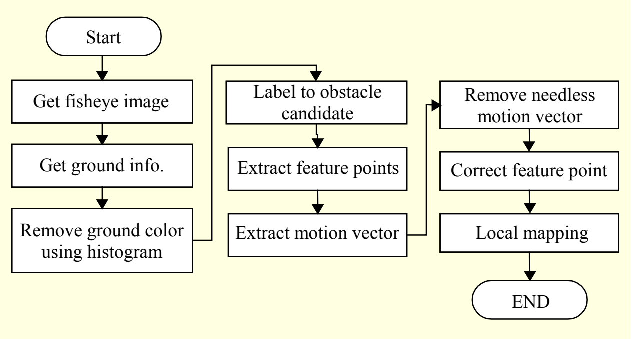


Fig. 3. Overall flow of local mapping algorithm.

the state vectors of (1), where *x*, *y*, and *θ* in Fig. 1 indicate the coordinates and angles of the robot, *L* is the distance between its wheels, *vr* and *vl* indicate the speeds of the wheels, and *v* is the speed of the robot. A real robot is made using this modeling information, as shown in Fig. 2(a). The robot has two wheels (diameter of 15 cm, and height of 30 cm) and has a cylindrical shape. A glass tube is installed on top (center) of the robot, and a fisheye camera is installed on top of the glass tube. This allows for environmental images of the robot to be obtained. Figure 2(b) shows a fisheye image (1,280 × 960 (width × height), with a viewing angle of 185°). All environmental data from the robot can be saved simultaneously because of the viewing angle.

$$\vec{x}_{(k)} = \begin{bmatrix} x_{(k+1)} \\ y_{(k+1)} \\ \theta_{(k+1)} \end{bmatrix} = \begin{bmatrix} x_{(k)} + v_{(k)}\Delta t \cos\theta_{(k)} \\ y_{(k)} + v_{(k)}\Delta t \sin\theta_{(k)} \\ \theta_{(k)} + \Delta t \dfrac{vr - vl}{L} \end{bmatrix}. \tag{1}$$

III. Mapping Algorithm for Individual Robot

1. Sequence of Local Mapping

The local mapping algorithm, used by each robot and shown in Fig. 3, produces a histogram showing environmental and ground information, which is obtained from images collected through fisheye cameras attached to the robots. It first removes unnecessary ground information using a histogram obtained from an analysis of the fisheye images, and then it labels candidate objects for clustering. It obtains motion vectors using the LKOF method based on corners as feature points in fisheye images and removes unnecessary vectors by using the magnitude and angle distribution of the motion vectors. The start and end points of available motion vectors are corrected using a correction algorithm based on a hemispherical projection model. The local mapping requires two sets of data: a dataset of the outer points of objects connected using a convex hull and one of clusters based on the labeling of objects.

2. Clustering Using the Labeling and Connection Algorithm

Previous studies have used k-means, k-neighbors, artificial neural networks, and Bayesian classifiers, among others, to extract feature points for clustering purposes. However, if the initial number of objects or groups is different from the current number, then an error prevents grouping. In this paper, ground information is removed using a histogram obtained from fisheye images. The remaining information is then labeled for clustering. The labeling results are obtained from the image that remains after the ground information has been removed, as shown in Fig. 4. As shown in Fig. 5, the labeling process is refined using a connection algorithm to merge unclear labeling regions. Algorithm 1 determines the size of the enlarged object by enclosing the overlapped or closely located objects. Algorithm 1 summarizes the connection algorithm in the form of an algorithm, and the proposed algorithm can detect objects in a static environment as well as a dynamic environment. It also improves the processing speed through a simple subtraction operation and verifies the detection reliability. The results of the connection algorithm are shown in Fig. 5.

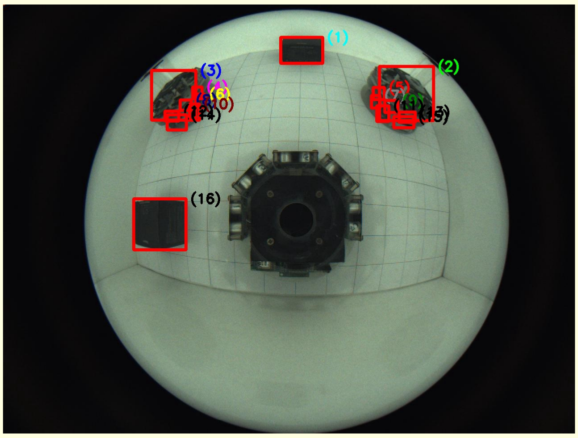


Fig. 4. Labeling before application of connection algorithm.

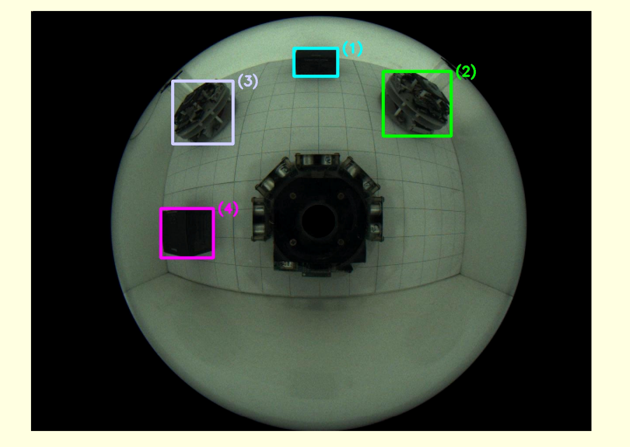


Fig. 5. Position of obstacle candidates using labeling.

```
Algorithm 1. Region connection.
  Input : corners list in object regions  $(rect_{n,x1:x4,y1:y4})$
  Input : corners of connected regions  $(Rect_{n,(x,y,\text{width},\text{height})})$
//Similar Region Connection Algorithm
for $n < blobs$ do
  if $rect_n \cap rect_{n+1}$ then
    $Rect_{n,x} = \min(rect_{n:n+1,x1:x4})$;
    $Rect_{n,y} = \min(rect_{n:n+1,y1:y4})$;
    $Rect_{n,\text{width}} = \max(rect_{n:n+1,x1:x4}) - \min(rect_{n:n+1,x1:x4})$
    $Rect_{n,\text{width}} = \max(rect_{n:n+1,y1:y4}) - \min(rect_{n:n+1,y1:y4})$
  else if $rect_n \supseteq rect_{n+1}$ then
    $Rect_{n,x} = \min(rect_{n,x1:x4})$;
    $Rect_{n,y} = \min(rect_{n,y1:y4})$;
    $Rect_{n,\text{width}} = \max(rect_{n,x1:x4}) - \min(rect_{n,x1:x4})$
    $Rect_{n,\text{width}} = \max(rect_{n,y1:y4}) - \min(rect_{n,y1:y4})$
  else
    add n step
  end if
end for
for changedRegionCounts > 0 do
  begin Override label Process
end for
for OverridedlabelCounts > 0 do
  begin Sort regions Process
end for
```

3. Extraction of Feature Points Using LKOF Method

A mono-vision or ceiling-vision SLAM approach, using images obtained from a single camera, estimates the current location of a robot based on the coordinates of the feature points obtained by comparing the previous frame with the current one [21]–[26]. This requires a comparison of the feature corners and boundaries without other information, but it is difficult to determine the correlations. In this paper, the relationship of the feature points is determined using the optical flow between the previous and current frames, and object information is estimated using the motion vectors. The basic optical flow method is a motion-estimation technique based on the differences between two consecutive frames. It assumes that the brightness of a pixel does not change even when the image frame changes and that spatially adjacent points have the same movement. Therefore, the optical flow method represents motion vectors for the direction and speed of the movement between two consecutive frames and must obtain motion vectors in a region with a distinct characteristic by extracting its edges or boundaries. In this regard, feature points are extracted in this paper using the Harris corner detection method for fisheye images [14]. That is, we have

$$C(x,y)=\begin{bmatrix} \sum_W \left(I_x(x_i,y_i)\right)^2 & \sum_W I_x(x_i,y_i)I_y(x_i,y_i) \\ \sum_W I_x(x_i,y_i)I_y(x_i,y_i) & \sum_W \left(I_y(x_i,y_i)\right)^2 \end{bmatrix},$$

$$O(x,y)=\det\left(C(x,y)\right)-k\left[\operatorname{trace}\left(C(x,y)\right)\right]^2,$$

$$HCM(x,y)=\begin{cases} 1 & \text{if } O(x,y)>0, \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

where *C*(*x*, *y*) indicates a change in the brightness of pixels calculated based on the Gaussian window *W* and brightness of each direction $I_{x \text{ or } y}$, and *O*(*x*, *y*) is the intensity of the corners obtained using a change in brightness. If the intensity of a corner exceeds the value of the constant m (0.3), then a binary image, *HCM*(*x*, *y*), can be obtained. In particular, the Harris corner detection algorithm uses 0.04 as the value for the constant k. If the algorithm applies the optical flow method using the corners as feature points, then the direction and magnitude of the motion vectors at the starting points, which are the feature points in the previous frame, can be determined. The optical flow between two frames can be expressed by the following equation based on the assumptions described earlier [15]:

$$\begin{aligned} f(x,\ y,\ t) &= I(x,\ y,\ t) \\ &= I(x+dx,\ y+dy,\ t+dt), \end{aligned} \tag{3}$$

where *dx*, *dy*, and *dt* indicate changes between two frames. Hence, using these changes for all pixels, we can obtain a matrix whose elements are the velocity components of the *x*- and *y*-axis. The least squares method is applied to this matrix to extract V_x and V_y against each axis as follows:

$$A^T Av = A^T b, \quad v=(A^T A)^{-1} A^T b,$$

$$v=\begin{bmatrix} V_x \\ V_y \end{bmatrix}=\begin{bmatrix} \sum I_{x_i}{}^2 & \sum I_{x_i} I_{y_i} \\ \sum I_{x_i} I_{y_i} & \sum I_{y_i}{}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum I_{x_i} I_{t_i} \\ -\sum I_{y_i} I_{t_i} \end{bmatrix}, \tag{4}$$

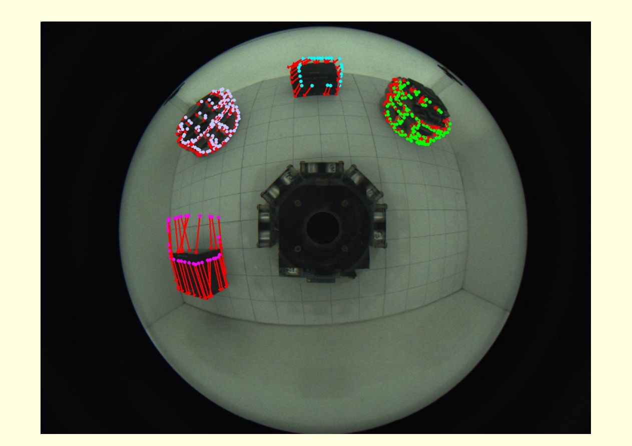


Fig. 6. Motion vectors using LKOF method.

where A and b are matrices that consist of the partial derivatives of the image I with respect to position (x, y) and time t, respectively. In addition matrix v consists of V_x and V_y. Equation (4) shows the motion vectors of the general optical flow, and the LK method uses both the addition of the Gaussian filter and the number adjustments of feature points to improve the computational load. The velocity components (V_x and V_y) of the LKOF method applied to the Gaussian function W, ω_i in (5), is as follows:

$$A^T WAv = A^T Wb, \quad v = (A^T WA)^{-1} A^T Wb,$$
$$\begin{bmatrix} V_x \\ V_y \end{bmatrix} = \begin{bmatrix} \sum \omega_i I_{x_i}{}^2 & \sum \omega_i I_{x_i} I_{y_i} \\ \sum \omega_i I_{x_i} I_{y_i} & \sum \omega_i I_{y_i}{}^2 \end{bmatrix}^{-1} \begin{bmatrix} -\sum \omega_i I_{x_i} I_{t_i} \\ -\sum \omega_i I_{y_i} I_{t_i} \end{bmatrix}. \tag{5}$$

The results of the application of the LKOF method to fisheye images are shown in Fig. 6. The points in Fig. 6 are the starting points (that is, the feature points before the robot begins to move), and the magnitude and direction of the motion vectors is shown by the red arrows. The surrounding robots have small motion vectors relative to each other because they move in a similar manner. However, the obstacles have large motion vectors relative to the moving robots because they are static.

4. Correction of Locations Based on Semispherical Modeling

Fisheye images have some distortion due to the distortion of the fisheye lens. Previous studies have used corrected images for all pixels in an image. This calibration method uses no information on the outer regions with many distortions because it only corrects small areas of a given distortion from the center of the fisheye image. In addition, it is slow because it operates for all pixels of an image. In this paper, only those coordinates of the feature points extracted using the LKOF method are corrected to improve the operating speed regardless of the size of the image.

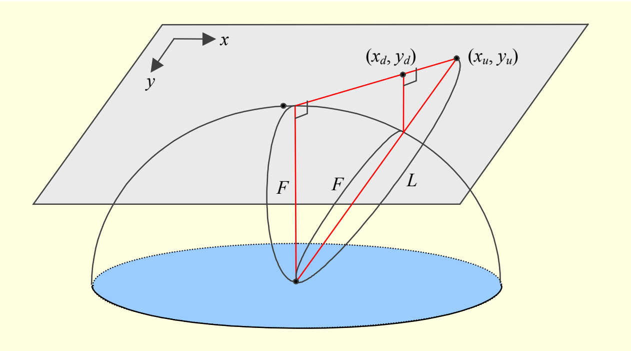


Fig. 7. Semispherical modeling.

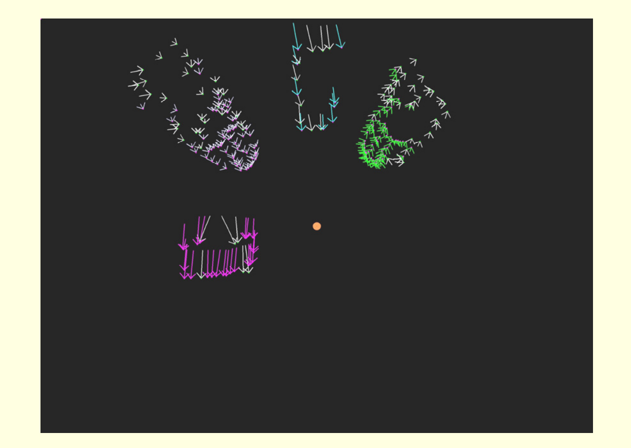


Fig. 8. Corrected motion vectors of objects.

The correction method divides the forward and backward mappings depending on how the original image is read. The forward mapping method determines the coordinates of the target image based on the original image. The backward mapping method calculates the coordinates of the original image from the target image. Forward mappings create holes that do not match because the output coordinates are calculated based on the original coordinates; thus they result in blurred parts because of the interpolation method used to remove the holes. Backward mappings do not create holes because they use the original coordinates, which are calculated based on the output coordinates, and many general corrections. The proposed algorithm uses forward mappings to correct coordinates for two reasons. First, it does not encounter any problems related with holes because it only needs to map the coordinates of objects (robots, obstacles, and so on). Second, it reduces the processing time, which is related to the access time and size of the input images. Forward mappings can be represented by (6) based on a hemispherical projection modeling, as shown in Fig. 7. Here, F is the focal length, (x_d, y_d) is the coordinate before the distortion is corrected, and (x_u, y_u) is the coordinate after the distortion correction.

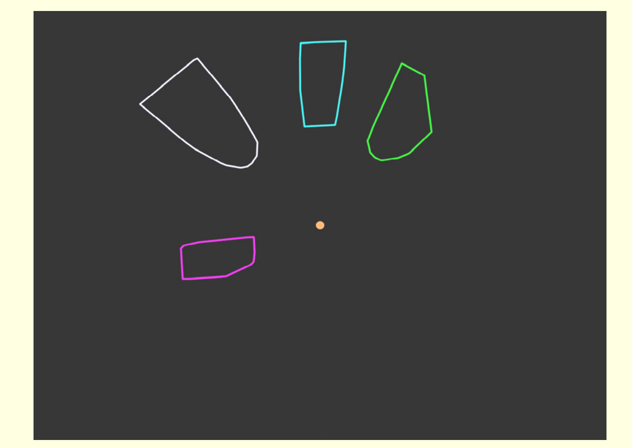


Fig. 9. Connection of outer points using a convex hull algorithm.

$$(x_u, y_u) = \left(x_d \frac{L}{F},\ y_d \frac{L}{F} \right). \qquad (6)$$

$$L = \frac{F^2}{\sqrt{F^2 - r_d^2}}. \qquad (7)$$

Equation (6) obtains the corrected coordinate using the uncorrected coordinates, and (7) obtains L, the ratio of corrected coordinates to uncorrected coordinates. In (7), r_d is the distance from the original coordinates to (x_d, y_d). The corrected motion vectors of objects, based on (6), can be seen in Fig. 8.

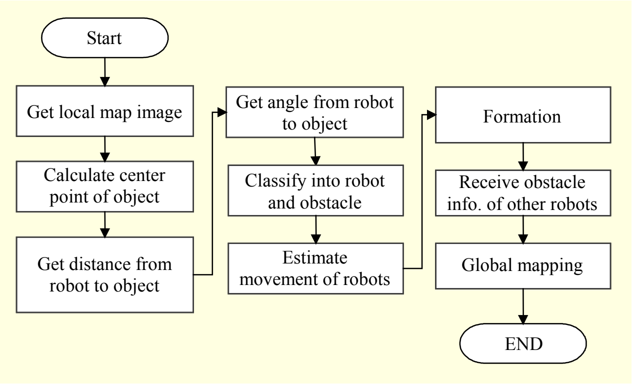
5. Local Mapping Using Corrected Points of an Object

The proposed algorithm distinguishes misrecognized vectors using the characteristics (magnitude and direction) of objects that have the same label. It then sorts all vectors according to a measured angle and filters out any misrecognized vectors in the process. Finally, using the outer-most points of an object's respective feature points, it then employs a further algorithm (convex hull) to extract the outlines of all existing objects [27]–[29].

In this paper, property information (the labelling of feature points) is carried out using the data that remains after the removal of the ground information. The resulting labelling data is then used by the convex hull algorithm to determine the outer points of objects. Once the convex hull algorithm is completed, we are then able to obtain a map that shows the locations of all the objects, as illustrated in Fig. 9.

IV. Global Mapping for Multiple Robots

1. Sequence of Global Mapping



Start
Get local map image
Calculate center point of object
Get distance from robot to object
Get angle from robot to object
Classify into robot and obstacle
Estimate movement of robots
Formation
Receive obstacle info. of other robots
Global mapping
END

Fig. 10. Overall flow of global mapping.

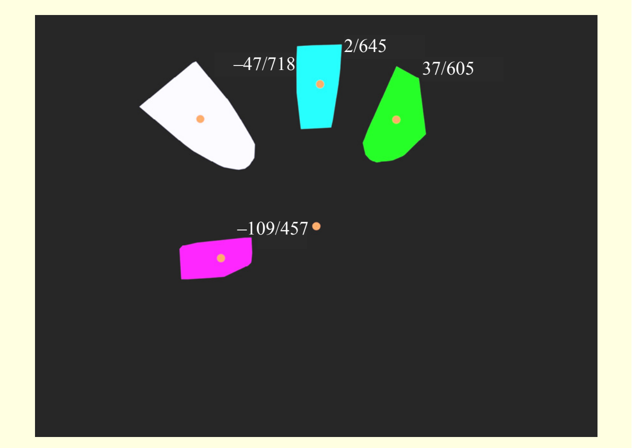


Fig. 11. Robot and object information.

The global mapping of multiple robots, shown in Fig. 10, extracts environmental information on objects using local maps created for individual robots and classifies objects. It estimates the motion of the robots found using the classified objects, and the robots move according to a predetermined formation algorithm. It maintains the formation of the robots by updating the local maps based on the object information from other robots and creates paths and global maps.

2. Object Classification and Estimation Using Vector Information

The proposed algorithm extracts the central point $O_{(x,\ y)}$ using (8) for the object region obtained from the local map and estimates the relationship between the center of the object and that of the robot. In (8), $I_{(x,\ y)}$ is the brightness of the (x, y) coordinate, and n is axially the number of bright pixels. It distinguishes between obstacles and robots based on the vector and relationship information. Because obstacles are static in comparison to robots (whose velocity is constant), the motion vectors of the obstacles have a characteristically large magnitude and are in the opposite direction relative to the motion of the robots. As shown in Fig. 11, it determines the

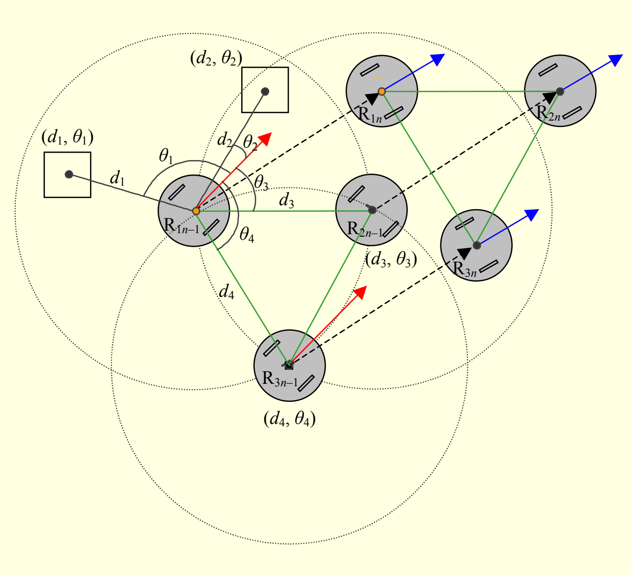


Fig. 12. Multi-robot formation for a global mapping.

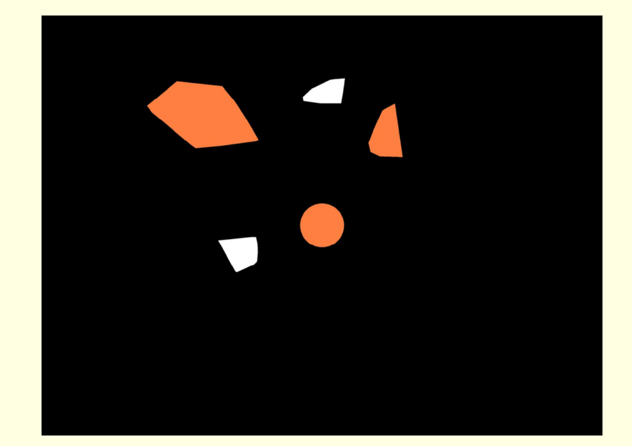


Fig. 13. Global map.

current location of the robots using information on both the robots and the obstacles, which can be distinguished based on the following characteristic:

$$O_{(x,y)} = \left(\frac{\int_0^w x dx}{n_x}, \frac{\int_0^w y dy}{n_y} \right) \quad \text{if} \quad I_{(x,y)} \neq 0. \tag{8}$$

3. Formation Control between Multiple Robots

The movement of multiple robots generates motion vectors related to the robot positions. Robots that move at a constant speed can obtain less vector information than static obstacles. The proposed algorithm checks the movement of the robots based on the vector information and retains all formations while maintaining the angles and distances between the robots. Because the robots employ an omnidirectional-vision SLAM approach, it is possible to quickly maintain their formation. In addition, formation control is possible through additional connections between the robots because the proposed algorithm can link the robots within the navigation area. As shown in Fig. 12, the proposed system consists of three robots (R_1, R_2, and R_3). If obstacles are present in the path of R_1, then R_1 changes direction. R_2 and R_3 detect the change in R_1's direction and correspondingly change their directions so as to maintain their group formation. In Fig. 12, d is the distance to an object's center point (from a robot's center point), and θ is the angle between the object's center point and the heading angle of the robot. The dotted circles, therefore, indicate regions that a robot is able to explore.

4. Global Mapping of Multiple Robots

Global mappings identify the positions of robots and obstacles to create global maps using local maps from other robots. As shown in Fig. 13, a global map is generated using overlay techniques for the same region. If it receives maps from many robots, it can then generate more precise maps.

V. Results

1. Experiment in Static Environment

A. Experiment Environment

An experiment was conducted to verify the performance of the proposed algorithm in a static environment. As shown in Fig. 14, the obstacle on the ground surface of the experiment environment, which is composed of 5 cm grids, is smaller than any of the depicted robots. Obstacles are installed to verify the generation of local maps based on an omnidirectional-vision SLAM approach, and local maps generated by individual robots are verified. When robots move in a straight line for a

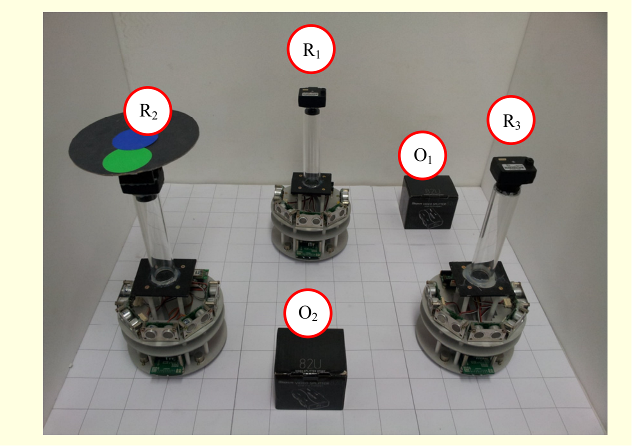


Fig. 14. Experiment environment in a static state.

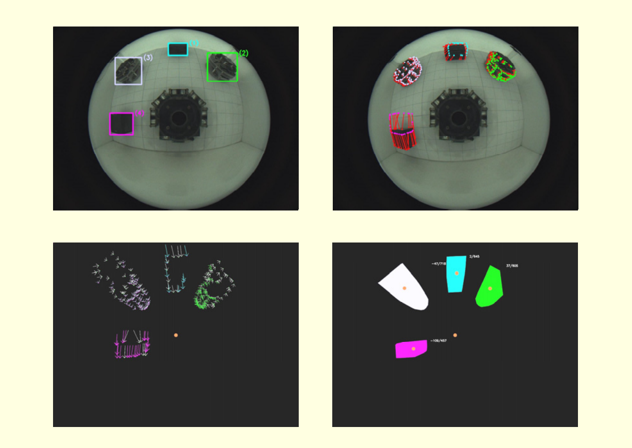


Fig. 15. Experiment result of R_1.

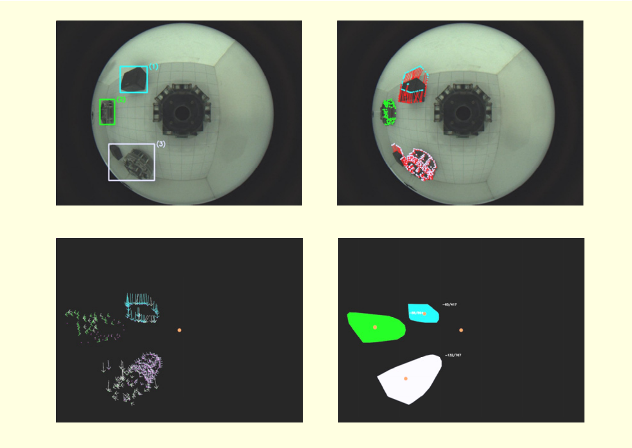


Fig. 16. Experiment result of R_2.

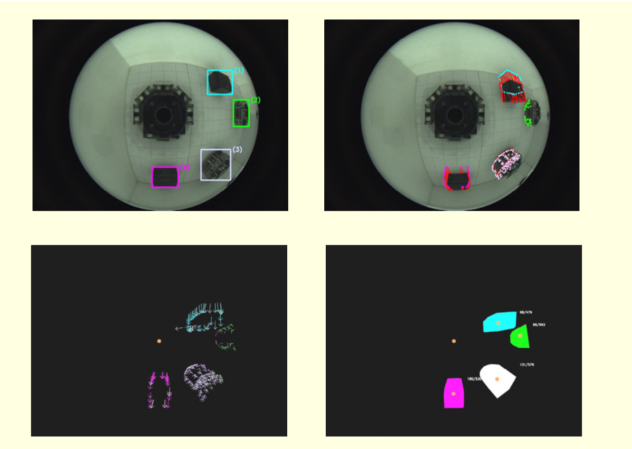


Fig.17. Experiment result of R_3.

distance of 5 cm, the proposed algorithm conducts a generation experiment for both local and global maps.

B. Experiment Results

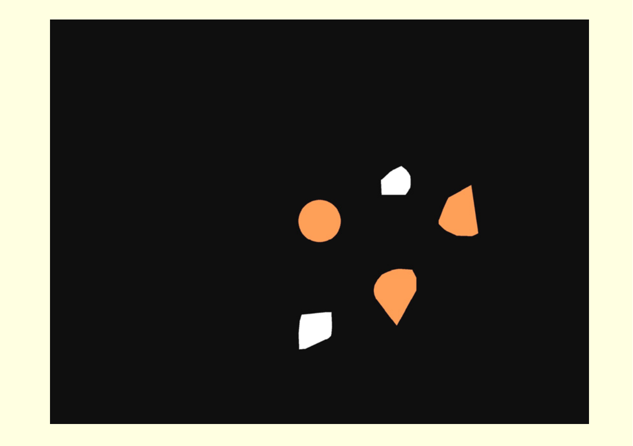


Fig. 18. Global map in a static environment.

The experimental results were obtained in a static environment, and local maps obtained from R_1 for formation control are shown in Fig. 15. Local maps obtained from R_2 are shown in Fig 16, and those from R_3 are shown in Fig. 17. The top-left images in Figs. 15, 16, and 17 are the labeling results, and the top-right images are motion vectors obtained from fisheye images. The bottom-left images are corrected motion vectors, and the bottom-right images are local maps for each robot. The results for global maps generated using local maps obtained from individual robots are shown in Fig. 18. The real distance between objects is compared with the distance based on the global maps obtained from the experimental results to verify the accuracy of the proposed algorithm. According to the results, the proposed algorithm has small errors (x-error of 2.23 cm and y-error of 1.49 cm).

2. Experiment in Dynamic Environment

A. Experiment Environment

An experiment for the generation of global maps in a dynamic environment is shown in Fig. 19(a). A color patch was attached on the top of each robot to detect its position. Global maps of the robots and obstacles were obtained using a ceiling camera. Maps based on the proposed algorithm are compared with those obtained from the ceiling camera. Here, it is a requirement that the color patches be easily recognizable and that the floor contain no colors that are similar in likeness to those used for the patches. A space painted in a non-glossy paint is used. As shown in Fig. 19(b), the position-estimation program, using global images, utilizes the various types of color patches as detected by the ceiling camera. A blue patch is the team color and represents the location of all robots. An individual color patch represents the location of individual robots. The angle between an individual robot's center line, which is the line between the center of its blue patch and that of its unique color patch, and the horizontal axis of the image

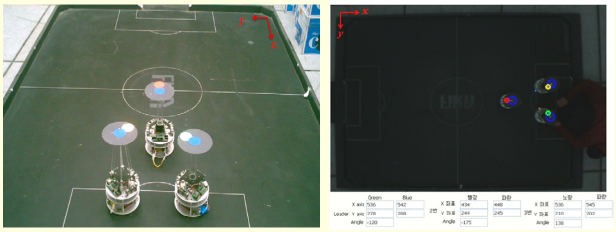

(a) (b)

Fig. 19. (a) Experimental environment in a dynamic state and (b) position-tracking program using a global vision system.

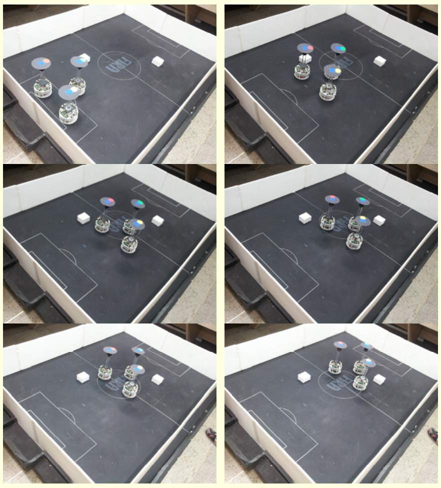


Fig. 20. Snapshots in a dynamic environment.

indicates the robot's heading. The position-estimation program implements a function that stores the position of the robot in a text file. The accuracy of experimental system experimented in previous research, and it obtained the small errors (4.16 cm and 4.46°) [30].

B. Experiment Results

As shown in Fig. 20, an experiment was conducted based on the movement of a real robot in a dynamic environment. Path data on the robots were obtained using a position-estimation program through the ceiling camera, and the performance of the proposed algorithm was verified by comparing the path data from the position-estimation program with that from the global maps based on the proposed algorithm. The results of the path comparisons are shown in Fig. 21, and those of the errors relative to the path of multiple robots are shown in Fig. 22. The proposed algorithm has relatively small errors (x = 5.65 cm and y = 4.50 cm) compared with those listed in Table 1.

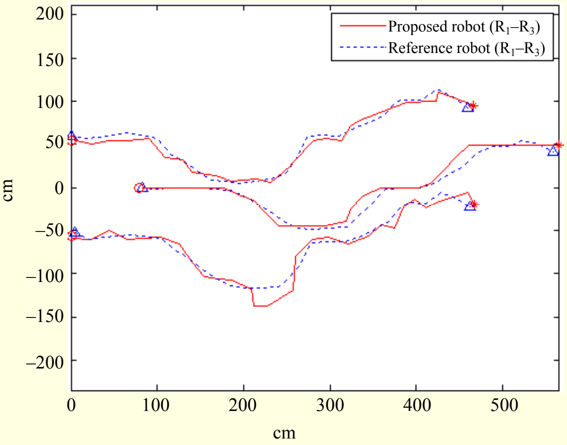


Fig. 21. Comparisons between the paths in the location-tracking system and the paths in the proposed system.

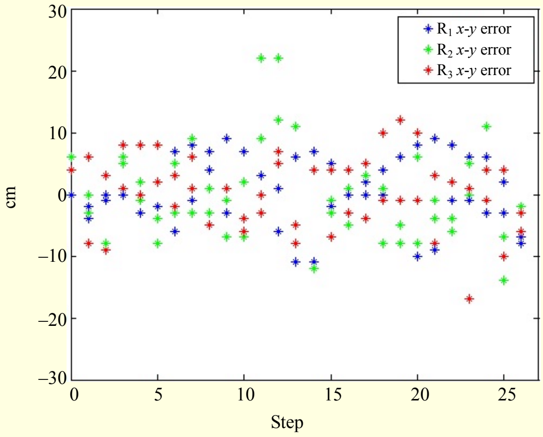


Fig. 22. Error between the paths in the location-tracking system and the paths in the proposed system.

Table 1. Comparison of experiment results for global paths with five other omnidirectional-vision SLAMs.

	Distance error (cm)
Method using the proposed algorithm	7.32
Method using Ladybug camera in 2012 [31]	8.51
Method using spherical image in 2011 [32]	7.11
Method using upward camera and CHL map in 2007 [33]	19.6
Method using topological navigation in 2007 [34]	11
Method using scan matching in 2006 [35]	10

VI. Conclusion

This paper proposed a global mapping of multiple robots based on an omnidirectional-vision SLAM approach through images obtained from fisheye cameras attached in a downward position to the robots. The proposed algorithm extracts feature points, using the LKOF method, from fisheye images and corrects the coordinates using a correction algorithm based on a hemispherical projection model. It then creates local maps by connecting the corrected outer coordinates of objects using a convex hull algorithm. Multiple robots create global maps based on the location and vector information of objects obtained from local maps while maintaining formation control. According to the experimental results, highly accurate maps are obtained from robots based on the proposed algorithm in both static and dynamic environments. In general, multiple robots must recognize all environmental data by moving extensively; however, in our proposed system, the global mapping process is significantly enhanced by simultaneously making use of all robots' local data in just a single movement.

In our future research, we plan to optimize the global mapping algorithm, in particular the way that it corrects the motion vectors. In addition, we are preparing research to improve the performance of the multi-robot system by using a fusion of laser scanner sensors and omnidirectional images. Finally, we will proceed with research on more accurate mappings through the use of a different coordinate correction algorithm.

References

[1] S. Thrun, "A Probabilistic Online Mapping Algorithm for Teams of Mobile Robots," *Int. J. Robot. Res.*, vol. 20, no. 5, May 2001, pp. 335–363.

[2] Z. Wei, G. Huang, and D. Wang, "The Research on Multi-robot Simultaneous Localization Mapping Algorithm," *IEEE Int. Conf. Autom. Logistics*, Jinan, China, Aug. 18–21, 2007, pp. 1241–1246.

[3] C. Cai et al., "Collision Avoidance in Multi-robot Systems," *Int. Conf. Mechatronics, Autom.*, Harbin, China, Aug. 5–8, 2007, pp. 2795–2800.

[4] A. Souliman et al., "Real Time Control of Multi-agent Mobile Robots with Intelligent Collision Avoidance System," *Sci. Inf. Conf.*, London, UK, Oct. 7–9, 2013, pp. 93–98.

[5] J.W. Kwon and D.K. Chwa., "Hierarchical Formation Control Based on a Vector Field Method for Wheeled Mobile Robots," *IEEE Trans. Robot.*, vol. 28, no. 6, Dec. 2012, pp. 1335–1345.

[6] J. Liu et al., "Optimal Formation of Robots by Convex Hull and Particle Swarm Optimization," *IEEE Symp. Comput. Intell. Contr. Autom.*, Apr. 16–19, 2013, pp. 104–111.

[7] H. Tan, Q. Liao, and J. Zhang, "An Improved Algorithm of Multiple Robots Cooperation in Obstacle Existing Environment," *IEEE Int. Conf. Rotot. Biomimetics*, Sanya, China, Dec. 15–18, 2007, pp. 1001–1006.

[8] C.G. Majumder, R. Janarthanan, and T.G. Majumder, "Co-operating a Blind Robot Teammate through Vision Based Real-Time Communication," *Int. Multi-conf. iMac4s*, Kottayam, India, Mar. 2013, pp. 59–64.

[9] S.R. Moon et al., "Leader-Following Approach Based Adaptive Formation Control for Mobile Robots with Unknown Parameters," *Trans. Korean Institute Electr. Eng.*, vol. 60, no. 8, Aug. 2011, pp. 1592–1598.

[10] G.H. Eoh et al., "Multi-robot Cooperative Formation for Overweight Object Transportation," *IEEE/SICE Int. Symp. Syst. Integr.*, Kyoto, Japan, Dec. 20–22, 2011, pp. 726–731.

[11] Y. Touahmi et al., "Congestion Avoidance for Multiple Micro-robots Using the Behavior of Fish Schools," *Int. J. Adv. Robot. Syst.*, vol. 9, Sept. 2012, pp. 1–12.

[12] L. Chaimowicz et al., "Experiments in Multirobot Air-ground Coordination," *IEEE Int. Conf. Robot. Autom.*, New Orleans, LA, USA, vol. 4, Apr. 26–May 1, 2004, pp. 4053–4058.

[13] O. Shakemia, R. Vidal, and S. Sastry, "Multibody Motion Estimation and Segmentation from Multiple Central Panoramic Views," *IEEE Int. Conf. ICRA*, Taipei, Taiwan, vol. 1, Sept. 2003, pp. 571–576.

[14] Y. Yagi, "Omni-directional Sensing and its Applications," *IEICE Trans.*, vol. E82-D, no. 3, Mar. 1999, pp. 568–579.

[15] T.E. Boult et al., "Frame-Rate Omnidirectional Surveillance and Tracking of Camouflaged and Occluded Targets," *IEEE Workshop Vis. Surveillance*, Fort Collins, CO, USA, July 1999, pp. 48–55.

[16] V.N. Peri and S.K. Nayar, "Generation of Perspective and Panoramic Video from Omnidirectional Video," *DARPA Image Understanding Workshop*, New Orleans, LA, USA, May 1997.

[17] K. Miyamoto, "Fish Eye Lens," *J. Opt. Soc. America*, vol. 54, no. 8, Aug. 1, 1964, pp. 1060–1061.

[18] A. Basu and S. Licardie, "Alternative Models for Fish-Eye Lenses," *Pattern Recogn.*, vol. 16, no. 4, Apr. 1995, pp. 433–441.

[19] A. Marjovi, S. Choobdar, and L. Marques, "Robotic Clusters: Multi-robot Systems as Computer Clusters: A Topological Map Merging Demonstration," *Robot. Auton. Syst.*, vol. 60, no. 9, Sept. 2012, pp. 1191–1204.

[20] T.D. Nguyen et al., "A Novel Interaction Method for Mobile Devices Using Low Complexity Global Motion Estimation," *ETRI J.*, vol. 34, no. 5, Oct. 2012, pp. 734–742.

[21] T. Fukuda et al., "Navigation System Based on Ceiling Landmark Recognition for Autonomous Mobile Robot-Landmark Detection Based on Fuzzy Template Matching (FTM)," *Proc. IEEE/RSJ Int. Conf.*, Pittsburgh, PA, USA, vol. 2, Aug. 1995, pp. 150–155.

[22] J. Gaspar, N. Winters, and J. Santos-Victor, "Vision-Based Navigation and Environmental Representations with an Omnidirectional Camera," *IEEE Trans. Robot. Autom.*, vol. 16, no. 6, Dec. 2000, pp. 890–898.

[23] A. Bakowski and G.A. Jones, "Video Surveillance Tracking

http://dx.doi.org/10.4218/etrij.14.0114.0584

Using Color Region Adjacency Graphs," *Image Process. Appl., Conf.*, Manchester, UK, vol. 2, no. 465, 1999, pp. 794–798.

[24] H. Tao, H.S. Sawhney, and R. Kumar, "Object Tracking with Bayesian Estimation of Dynamic Layer Representations," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 1, Jan. 2002, pp. 75–89.

[25] W.Y. Jeong and K.M. Lee, "CV-SLAM: A New Ceiling Vision-Based SLAM Technique," *IEEE/RSJ Int. Conf. Intell. Robot. Syst.*, Edmonton, Canada, Aug. 2–6, 2005, pp. 3195–3200.

[26] Y. Yagi et al., "Iconic Memory-Based Omnidirectional Route Panorama Navigation," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 27, no. 1, Jan. 2005, pp. 78–87.

[27] J.C. Jeong et al., "High-Quality Stereo Depth Map Generation Using Infrared Pattern Projection," *ETRI J.*, vol. 35, no. 6, Dec. 2013, pp. 1011–1020.

[28] H.J. Sohn and B.K. Kim, "An Efficient Localization Algorithm Based on Vector Matching for Mobile Robots Using Laser Range Finders," *J. Intell. Robot. Syst.*, vol. 51, no. 4, 2008, pp. 461–488.

[29] R.L. Graham, "An Efficient Algorithm for Determining the Convex Hull of a Finite Planar Set," *Inf. Process. Lett.*, vol. 1, no. 4, 1972, pp. 132–133.

[30] Y.W. Choi et al., "Localization Using Ego Motion Based on Fisheye Warping Image," *J. Institute Contr., Robot. Syst.*, vol. 20, no. 1, June 2014, pp. 70–77.

[31] G. Tong and Z. Wu., "An Omni-directional vSLAM Based on Spherical Camera Model and 3D Modeling," *Proc. World Congress Intell. Contr. Autom.*, Beijing, China, July 6–8, 2012, pp. 4551–4556.

[32] J.R. Kim, M.-S. Lim, and J.-H. Lim, "Omni Camera Vision-Based Localization for Mobile Robots Navigation Using Omni-directional Images," *J. Institute Contr., Robot. Syst.*, vol. 17, no. 3, Oct. 2011, pp. 206–210.

[33] S.J. Yoon et al., "Global Localization of Mobile Robots Using Omni-directional Images," *Trans. KSME*, vol. 31, no. 4, Apr. 2007, pp. 517–524.

[34] T. Goedeme et al., "Omnidirectional Vision Based Topological Navigation," *Int. J. Comput. Vision*, vol. 74, no. 3, Sept. 2007, pp. 219–236.

[35] E. Menegatti et al., "Omnidirectional Vision Scan Matching for Robot Localization in Dynamic Environments," *IEEE Trans. Robot.*, vol. 22, no. 3, June 2006, pp. 523–535.

Yun-Won Choi received his BS degree in electrical engineering from Yeungnam University, Daegu, Rep. of Korea, in 2009 and his MS degree in electronic engineering from Yeungnam University in 2011. He completed a doctoral course in robot engineering from Yeungnam University in 2013. Since 2013, he has been working at the Electronics and Telecommunications Research Institute, Daegu, Rep. of Korea, where he is now a researcher. His main research interests are robotics, omnidirectional image processing, and multi-robot control.

Kee-Koo Kwon received his BS degree in electronic engineering from Inje University, Gimhae, Rep. of Korea, in 1997 and his MS and PhD degrees in electronic engineering from Kyungpook National University, Daegu, Rep. of Korea, in 2000 and 2004, respectively. In 2004, he began working for the Electronics and Telecommunications Research Institute (ETRI), Daejeon, Rep. of Korea, as a researcher. In 2006, he relocated to work at ETRI's Daegu branch, where he is now a principal researcher. His main research interests are image and video signal processing; machine vision; intelligent vehicles; and automotive ADAS systems.

Soo-In Lee received his BS, MS, and PhD degrees in electronic engineering from Kyungpook National University, Daegu, Rep. of Korea, in 1985, 1989, and 1996, respectively. Since 1990, he has been working at the Electronics and Telecommunications Research Institute, Daegu, Rep. of Korea, where he is currently a principal researcher at the Daegu-Gyeongbuk Research Center. His main research interest is IT convergence technology.

Jeong-Won Choi received his BS, MS, and PhD degrees in electrical engineering from the School of Electrical Engineering, Yeungnam University, Daegu, Rep. of Korea, in 1995, 1997, and 2002, respectively. From 2003 to 2006, he worked for STX, Changwon, Rep. of Korea, and from 2006 to 2009, he worked at the Kumoh National Institute of Technology, Gumi, Rep. of Korea. Since 2009, he has been with the Department of Automation Electrical Engineering, Yeungnam University College, Daegu, Rep. of Korea, where he is now an assistant professor. His main research interests are mobile robots and automation control systems.

Suk-Gyu Lee received his BS and MS degrees in electrical engineering from Seoul National University, Seoul, Rep. of Korea, in 1979 and 1981, respectively. He received his PhD degree in electrical engineering from the University of California, Los Angeles, CA, USA, in 1990. Since 1982, he has been with the Department of Electrical Engineering, Yeungnam University, Daegu, Rep. of Korea, where he is now a professor. His research interests include robotics, SLAM, nonlinear control, and adaptive control.