

Visualization of Obstacles on Bird's-eye View Using Depth Sensor for Remote Controlled Robot

Yasuyuki Awashima, Ren Komatsu, Hiromitsu Fujii, Yusuke Tamura, Atsushi Yamashita, Hajime Asama
Department of Precision Engineering, The University of Tokyo, Tokyo, Japan
Email: {awashima, komatsu, fujii, tamura, yamashita, asama}@robot.t.u-tokyo.ac.jp

Abstract—The bird's-eye view system is one of the image presentation systems for remote control of robots. However, when there are obstacles that are higher than the floor, there is a problem that the appearances of them are incorrectly shown on the bird's-eye view image due to the image distortion. That is because the image is generated under the assumption that all objects are present on the floor. In this paper, we propose a method to visualize correctly the areas where obstacles are present on the bird's-eye view image by using 3D measurement data. Specifically, the information of the robot's distance from obstacles measured by a depth sensor is superimposed onto the bird's-eye view image as points with colors obtained by fish-eye cameras. The bird's-eye view image generated by the proposed method can improve the visibility of the surrounding environment.

I. INTRODUCTION

There are a lot of disaster prone countries where eruptions of various disasters occur frequently, and Japan is also one of such countries. When disasters occur, prompt surveying and recovery work in disaster sites are very important to limit the extent of the damage. However, there are some disaster sites that is exposed to the risks of the collapse of buildings, and people cannot enter such disaster sites. Therefore, in such disaster sites, surveying and recovery work by using remote controlled robots are required [1], [2]. To minimize the disaster damages, it is important to do surveying and recovery work by using remote controlled robots efficiently. Nowadays, there are a lot of researches on visualization techniques related to remote controlled robots to improve the efficiency of the operations.

An example of good visualization techniques for remote controlled robots is the bird's-eye view system [3–7]. In the bird's-eye view system, the image of the robot and the surrounding environment that are seen from above is generated artificially using wide-angle cameras such as fish-eye cameras. By using this bird's-eye view system, it becomes easy to operate remote controlled robots because of the ease of grasping the relative positional relationship between the robot and the surrounding environment.

However, in the conventional approach, when there are obstacles, there is a problem that their positional appearance with respect to the robot is incorrectly shown. This is because the bird's-eye view is generated under the assumption that all objects are present on the floor. An example of this problem is the situation in which an obstacle exists as shown in Fig. 1. In this situation, the positional appearance with respect to the robot is incorrect on the bird's-eye view image generated by the conventional method as shown in Fig. 2. In actual field, this problem can cause collisions between robots and obstacles because operators misidentify the surrounding environment, and this problem is serious in indoor environments where

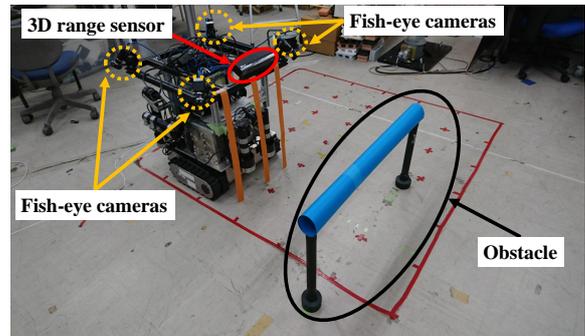


Fig. 1. An example situation in which there is the obstacle that is higher than the floor. In this case, the appearance of the obstacle cannot be shown correctly on the conventional bird's-eye view (Fig. 2).

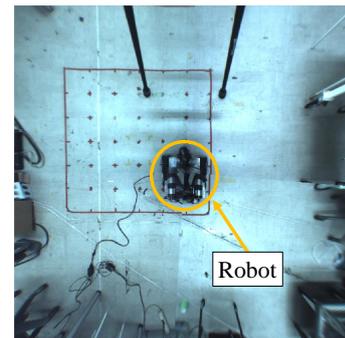


Fig. 2. The bird's-eye view image generated by the conventional method [5] in the situation of Fig. 1. With regard to the upper side of the obstacle, the positional appearance with respect to the robot is incorrectly shown on the image.

objects are scattered due to an accident such as explosion. For this problem, a method for detecting obstacles using stereo bird's-eye images is proposed [4]. In this method, obstacles are captured by multiple cameras on the ceiling, and the obstacles are detected by the showing the positional relationship on the image. By presenting the information of obstacles, operators can easily grasp the existence of obstacles in surrounding environments. However, it is difficult to apply this method to our study because indoor disaster sites where people cannot enter are supposed, but cameras cannot be installed on the ceiling in such sites. In the previous research [5], another method to detect obstacles is proposed. Specifically, the information of obstacles is superimposed onto the bird's-eye view image by using a depth sensor. In this method, the information of obstacles regarding a particular plane is presented to operators. However, when obstacles don't have intersection points with the plane, it is difficult to show accurately the positional appearance of the obstacles with respect to the robot.

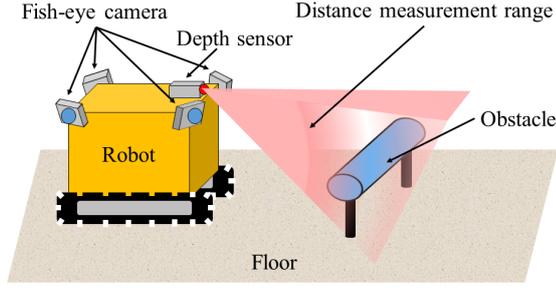


Fig. 3. The conceptual figure of the proposed method in this study

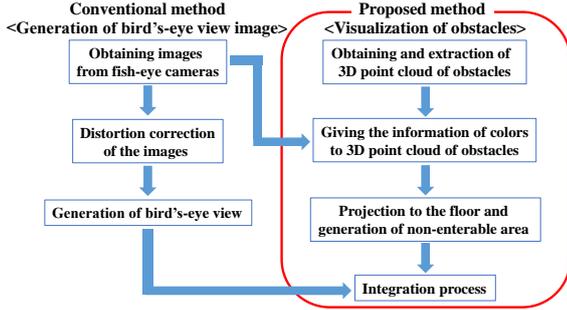


Fig. 4. The schematic of the proposed method algorithm

In our research, we propose a method to visualize areas where obstacles are present on the conventional bird's-eye view image by using 3D measurement data of obstacles. As shown in Fig. 3, indoor sites are assumed as the environments of the operation of robots, and by using a depth sensor in addition to fish-eye cameras, the information of the robot's distance to the obstacle is measured. Then, the information of the obstacle positions generated on the basis of the distance information is superimposed onto the conventional bird's-eye view image as points with the appropriate colors. Thus, the correct positional relationship with respect to the robot and colors of the obstacles can be shown, which is necessary for safe remote control.

II. PROPOSED METHOD

The algorithm of the proposed method is shown in Fig. 4. First, in this research, let objects that are higher than the floor be obstacles, and they are detected from 3D measurement data obtained by a depth sensor. The obstacles are projected to the floor by orthogonal projection onto the image. The regions beyond the obstacle, where the robot cannot enter, are also shown remarkably by using a vivid color on the image. Also, the obstacles' colors are given to their points by corresponding the fish-eye cameras' images and the 3D points of the obstacles. Following this, the two images are integrated with the bird's-eye view image generated by fish-eye cameras.

A. Definition of coordinate systems

In this research, fish-eye cameras and a depth sensor are mounted on the robot. A point $\mathbf{P}_w = [x_w, y_w, z_w]^T$ in the

robot coordinate system is expressed as $\mathbf{m}_f = [u_f, v_f]^T$ in the fish-eye camera image coordinate system. The same point is expressed as $\mathbf{P}_f = [x_f, y_f, z_f]^T$ and $\mathbf{P}_d = [x_d, y_d, z_d]^T$ in the fish-eye camera coordinate system and the depth sensor coordinate system, respectively. These points are converted to a point $\mathbf{m}_v = [u_v, v_v]^T$ in the image coordinate system of the virtual camera at the bird's-eye point of view, where x_w - y_w plane in the robot coordinate system is the same plane as the floor, and the positive direction of z_w axis is the vertical upward direction. Additionally, let the homogeneous coordinate expression of a point \mathbf{N} in any coordinate system be expressed as $\tilde{\mathbf{N}}$.

B. Distortion correction of fish-eye camera images

Although fish-eye cameras have wide angles of view, there is particular distortion in the fish-eye camera images because of their projection mechanism. In this research, the relationship between a point \mathbf{P}_f and a point \mathbf{m}_f is found by the method proposed by Scaramuzza *et al.* [8], [9]. The relationship is expressed as follows;

$$\mathbf{P}_f = \begin{bmatrix} x_f \\ y_f \\ z_f \end{bmatrix} = a \begin{bmatrix} u_f \\ v_f \\ f(u_f, v_f) \end{bmatrix}, \quad (1)$$

where $f(u_f, v_f)$ is the function regarding the fish-eye camera coordinate (u_f, v_f) , and a is a constant. From this formula, the one-to-one relationship between each point in the fish-eye camera image coordinate system and each 3D vector in the fish-eye camera coordinate system is found, and the distortion is removed.

C. Generation of bird's-eye view image

After removing the distortion of fish-eye camera images, they are converted to images that are seen from above, namely, bird's-eye view images. First, the relationship between the robot coordinate system and the fish-eye camera image coordinate system is expressed as follows:

$$\tilde{\mathbf{m}}_f = \mathbf{H}_{w \rightarrow f} \tilde{\mathbf{P}}_w, \quad (2)$$

where $\mathbf{H}_{w \rightarrow f}$ is 3×4 perspective projection matrix that relates these coordinate systems. The projection plane is the floor in the robot coordinate system, and the equation $z_w = 0$ holds on the floor. Therefore, in Eq. (2), the components of the third column of $\mathbf{H}_{w \rightarrow f}$ can be omitted as follows:

$$\begin{aligned} \tilde{\mathbf{m}}_f &= \mathbf{H}'_{w \rightarrow f} \tilde{\mathbf{P}}'_w \\ &= \begin{bmatrix} h_{11} & h_{12} & h_{14} \\ h_{21} & h_{22} & h_{24} \\ h_{31} & h_{32} & h_{34} \end{bmatrix} \begin{bmatrix} x_w \\ y_w \\ 1 \end{bmatrix}, \end{aligned} \quad (3)$$

where $\mathbf{H}'_{w \rightarrow f}$ and $\tilde{\mathbf{P}}'_w$ are the omitted matrixes when the components of the third column of $\mathbf{H}_{w \rightarrow f}$, and the components of the third row of $\tilde{\mathbf{P}}_w$ are omitted, respectively. In the same way, the relationship between the robot coordinate system and the virtual camera image coordinate system is expressed as follows:

$$\tilde{\mathbf{m}}_v = \mathbf{H}'_{w \rightarrow v} \tilde{\mathbf{P}}'_w, \quad (4)$$

where $\mathbf{H}'_{w \rightarrow v}$ is the omitted matrix when the components of the third column of $\mathbf{H}_{w \rightarrow v}$ are omitted, and $\mathbf{H}_{w \rightarrow v}$ is the perspective projection matrix that relates these coordinate systems. By using Eqs. (3) and (4), the following equation

that relates the robot coordinate system and the virtual camera image coordinate system is obtained.

$$\tilde{\mathbf{m}}_v = \mathbf{H}'_{w \rightarrow v} \mathbf{H}'_{f \rightarrow w} \tilde{\mathbf{m}}_f, \quad (5)$$

where $\mathbf{H}'_{f \rightarrow w} = (\mathbf{H}'_{w \rightarrow f})^{-1}$ is completed. From Eq. (5), the bird's-eye view images can be generated from the fish-eye camera images. Also, it is necessary to obtain $\mathbf{H}'_{w \rightarrow v}$ and $\mathbf{H}'_{f \rightarrow w}$. In this research, the one-to-one relationship of each pixel between the fish-eye camera image and the virtual camera image is obtained by a non-linear optimization method. To reduce computational costs, the relationship is listed in a lookup table, and the virtual image is composed by referring to the table.

D. Obtaining 3D point cloud and obstacle extraction

In this research, by using a depth sensor, the 3D point cloud in the surrounding environment is obtained. Therefore, the coordinate system of the depth sensor is converted to the robot coordinate system to extract the 3D point cloud of obstacles. By capturing a checker board in front of the robot, and by detecting the feature points, 4×4 extrinsic parameter matrix \mathbf{M} that shows the depth sensor's position and posture in the robot coordinate system is calculated. By using \mathbf{M} , the relationship between the robot coordinate system and the depth sensor coordinate system is expressed as follows:

$$\tilde{\mathbf{P}}_w = \mathbf{M}^{-1} \tilde{\mathbf{P}}_d. \quad (6)$$

With regard to the 3D point cloud in the robot coordinate system obtained by Eq. (6), the only points that are present at higher position than the floor, namely the points whose coordinates of height $z_w > 0$, are extracted. Thus, the 3D point cloud of obstacles can be extracted.

E. Mapping color information onto 3D point cloud of obstacle

The information of textures is given to the 3D point cloud of obstacles from the fish-eye camera images. First, 4×4 extrinsic parameter matrix \mathbf{K} that shows the fish-eye camera's position and posture in the robot coordinate system is found by the method proposed by Scaramuzza *et al.* [8], [9]. By using \mathbf{K} , the relationship between the robot coordinate system and the fish-eye camera coordinate system is expressed as follows:

$$\tilde{\mathbf{P}}_f = \mathbf{K} \tilde{\mathbf{P}}_w. \quad (7)$$

By using Eqs. (6) and (7), the following equation that converts the coordinate system from the depth sensor coordinate system to the fish-eye camera image coordinate system is obtained. In Section II-B, the one-to-one relationship between each point in the fish-eye camera image coordinate system and each 3D vector in the fish-eye camera coordinate system is found. Therefore, by using Eqs. (1) and (8), the information of textures can be given to the 3D coordinates' point cloud of the obstacle from the fish-eye camera images.

$$\tilde{\mathbf{P}}_f = \mathbf{K} \mathbf{M}^{-1} \tilde{\mathbf{P}}_d. \quad (8)$$

F. Projection of obstacles to the floor

The extracted 3D point cloud with colors is projected orthogonally to the floor. Specifically, an extracted 3D point of an obstacle $\mathbf{A} [x_w, y_w, z_w]^T$ in the robot coordinate system is

converted to a point $\mathbf{B} [x_w, y_w, 0]^T$ on the floor. This processing is executed for all of the extracted 3D points of obstacles. Then, when obstacles exist in the surrounding environment, the robot cannot enter the back of the obstacles in general, and let it be called the non-enterable area in this paper. The explanation below is the method of the generation of the non-enterable area. First, the same processing as described above is executed for the 3D point that represents the depth sensor's position in the robot coordinate system, and let the projected coordinate point be represented as \mathbf{C} . Second, straight lines are drawn from \mathbf{C} to all of the projected points of obstacles. Finally, the non-enterable areas are generated as the region that consists of line segments drawn on the opposite side of the robot with respect to obstacles. In the integration process shown in Fig. 4, the non-enterable area is superimposed onto the conventional bird's-eye view image first, and the projected points of obstacles are superimposed onto it second. Finally, the image that shows correctly the positional appearance with respect to the robot can be obtained.

III. EXPERIMENT

A. Experimental settings

Figure 1 shows the arrangement of the robot, four fish-eye cameras, and a depth sensor used in this experiment. The fish-eye cameras and the depth sensor are encircled by the broken line and the full line, respectively. The camera is Grasshopper3 GS3-U3-41C6C-C of Point Grey Research Inc., the fish-eye lens is FE185C086HA-1 of Fujinon Inc., and the depth sensor is Kinect v2 of Microsoft Corporation. The four fish-eye cameras are mounted on the corners on the upper surface of the robot. The depth sensor is mounted on the front upper surface of the robot. Also, three orange strips of papers are hung on the front surface of the robot in order to find the robot's closest point to the obstacle on the image that is captured actually from above. In this experiment, the validity of the proposed method in one direction is confirmed, because it is simple to expand it in every direction.

Experiments were done in an indoor environment where an obstacle was arranged in front of the robot as shown in Fig. 1. In this experiment, a blue tube was used as the obstacle because such an obstacle often exists in indoor disaster sites, and is shown inaccurately in the bird's-eye view image as shown in Fig. 2. Therefore, assessing the validity of the proposed method is easy in comparison with the conventional method. In this experimental situation, the bird's-eye view image was generated by the proposed method. In the same experimental situation, the bird's-eye view image was generated by the conventional method. By comparing the two bird's-eye view images with the image captured from above, the assessment of the proposed method was done.

B. Experimental results

Figure 6 shows the experimental results. Figure 6 (a) shows the image captured actually from above in the situation as shown in Fig. 1. In the same situation, the bird's-eye view image generated by the conventional method and the bird's-eye view image generated by the proposed method are shown in Fig. 6 (b) and (c), respectively. The red dotted line in Fig. 6 (a), (b), and (c) is the boundary line of the area where the obstacle is observed actually from above, and the robot cannot get beyond the boundary due to the collision with the obstacle. The blue dotted line in Fig. 6 (a), (b), and (c) is the line

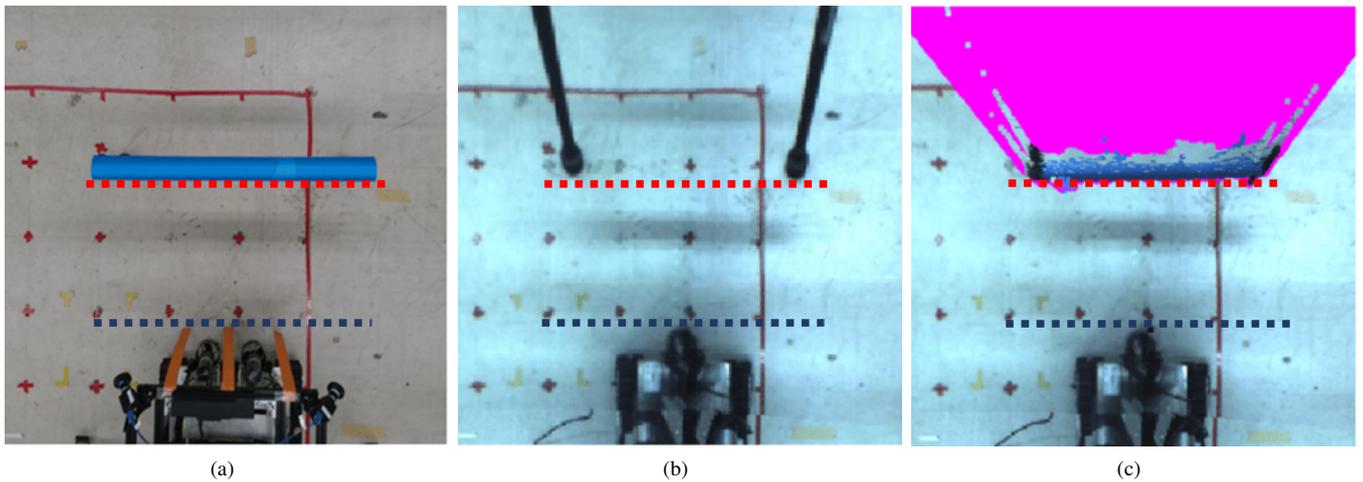


Fig. 5. Experimental results. (a) The image captured actually from above. (b) The bird's-eye view image generated by the conventional method. (c) The bird's-eye view image generated by the proposed method.

segment that passes through the robot's closest point to the obstacle, and is parallel to the boundary line. In Fig. 6 (a), the upper side of the obstacle, namely the blue pipe is not shown correctly in the same position in Fig. 6 (b). When operators control the robot remotely by using the bird's-eye view image of Fig. 6 (b), there is a danger that the operators misidentify the surrounding environment, and the robot collides with the blue pipe. On the other hand, in the bird's-eye view image generated by the proposed method as shown in Fig. 6 (c), the blue pipe is shown correctly on the boundary line, which is the same that shown in Fig. 6 (a). In Fig. 6 (c), the non-enterable area, which is the area beyond the obstacle is shown by vivid pink color. This can prevent the robot from colliding with the obstacle, even when it is difficult to identify the presence of the obstacle due to its thin appearance seen from just above.

As a quantitative verification, the positional accuracy of the obstacle shown on the bird's-eye view was evaluated. The distance from the blue line to the red line is 114 pixel on the image of Fig. 6 (a), and the distance on the image of Fig. 6 (c) is 117 pixel. The difference between the two distances on the images 3 pixel corresponds to 12.5 mm, which is calculated as real world scale. Actually, the blue pipe in Fig. 6 (c) is shown at almost the same position of Fig. 6 (a). Therefore, by using the bird's-eye view image generated by the proposed method, the operators can remote-control the robot safely without the misidentification of the surrounding environment, and without the robot's collision with obstacles.

IV. CONCLUSION

In this study, we proposed a method regarding the bird's-eye view system that improves the visibility of the surrounding environment, and prevent the robot from colliding with obstacles. By this proposed method, their positional relationship with respect to the robot could be rendered correctly on the bird's-eye view image. Therefore, it can be helpful for the operators to remote-control the robots safely, when surveying and recovery work are conducted in indoor disaster sites. In the future, it is necessary to expand the proposed method in every direction in order to improve the visibility of surrounding environments.

ACKNOWLEDGEMENTS

This work was in part funded by Tough Robotics Challenge, ImpACT Program of the Council for Science, Technology and Innovation (Cabinet Office, Government of Japan), and a part of this study is the result of "HRD for Fukushima Daiichi Decommissioning based on Robotics and Nuclide Analysis" carried out under the Center of World Intelligence Project for Nuclear S&T and Human Resource Development by the Ministry of Education, Culture, Sports, Science and Technology of Japan.

REFERENCES

- [1] F. Matsuno and S. Tadokoro: "Rescue Robots and Systems in Japan," *Proceedings of the 2004 IEEE International Conference on Robotics and Biomimetics (ROBIO2004)*, pp. 12–20, 2004.
- [2] M. W. Kadous, R. K. Sheh and C. Sammut: "Effective User Interface Design for Rescue Robotics," *Proceedings of the 1st ACM SIGCHI/SIGART Conference on Human-Robot Interaction (HRI2006)*, pp. 250–257, 2006.
- [3] C. C. Lin and M. S. Wang: "Topview Transform Model for The Vehicle Parking Assistance System," *Proceedings of the 2010 International Computer Symposium (ICS2010)*, pp. 306–311, 2010.
- [4] J. Mingyue, Y. Sun and J. Wang: "Obstacle Detection in Stereo Bird's Eye View Images," *Proceedings of the 2014 Information Technology and Artificial Intelligence Conference (ITAIC2014)*, pp. 254–257, 2014.
- [5] T. Sato, A. Moro, A. Sugahara, T. Tasaki, A. Yamashita and H. Asama: "Spatio-Temporal Bird's-Eye View Images Using Multiple Fish-eye Cameras," *Proceedings of the 2013 IEEE/SICE International Symposium on System Integration (SII2013)*, pp. 753–758, 2013.
- [6] Y. C. Liu, K. Y. Lin and Y. S. Chen: "Bird's-eye View Vision System for Vehicle Surrounding Monitoring," *Proceedings of the 2008 International Workshop on Robot Vision (RobVis2008)*, pp. 207–218, 2008.
- [7] S. Shimizu, J. Kawai and H. Yamada: "Wraparound View System for Motor Vehicles," *Fujitsu Scientific & Technical Journal (FSTJ2010)*, Vol. 1, No. 46, pp. 95–102, 2010.
- [8] D. Scaramuzza, A. Martinelli and R. Siegwart: "A Flexible Technique for Accurate Omnidirectional Camera Calibration and Structure from Motion," *Proceedings of the 4th IEEE International Conference of Computer Vision Systems (ICVS2006)*, pp. 45–52, 2006.
- [9] D. Scaramuzza, A. Martinelli and R. Siegwart: "A Toolbox for Easily Calibrating Omnidirectional Cameras," *Proceedings of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2006)*, pp. 5695–5701, 2006.