Proceedings of the 2013 IEEE/SICE International
Symposium on System Integration, Kobe International
Conference Center, Kobe, Japan, December 15-17,

Spatio-Temporal Bird's-eye View Images Using Multiple Fish-eye Cameras

Takaaki Sato, Alessandro Moro, Atsushi Sugahara, Tsuyoshi Tasaki, Atsushi Yamashita, and Hajime Asama

***Abstract*— In camera images for urban search and rescue (USAR) robots teleoperation, it is important to reduce blind spots and get surroundings as much as possible because of safety requirements. We propose a method to create synthesized bird's-eye view images from multiple fish-eye cameras as spatio-temporal data which can reduce blind spots. In practical use, it is very important to get images robustly even when some troubles such as camera broken and network disturbances occur. This method develops showing bird's-eye view images robustly even if some of images are not acquired by compensating past stored spatio-temporal data to these images. Effectiveness of the proposed method is verified through experiments.**

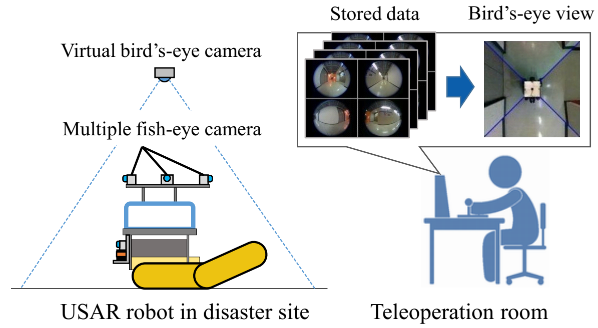


Fig. 1: Proposed method.

I. Introduction

In this research, we propose a method to create a robust bird's-eye view images using stored data of multiple fish-eye cameras (Fig. 1). More specifically, we describe a principle of a spatio-temporal bird's-eye view which compensates missing parts of bird's-eye view images from stored data.

On March 11, 2011, the great eastern japan earthquake and tsunami caused the accident of Fukushima Daiichi Nuclear Plant. From that time, investigation and decommissioning of the nuclear reactor has been a serious problem. The main cause is that no one can enter the center area because of the high radiation exposure.

Teleoperating an urban search and rescue (USAR) robot is an effective way to solve this serious problem, and various robots have been designed and manufactured [1]. However, little attention has been given to be used in real site. For USAR robot studies, what seems to be lacking is considering many cases of disaster and developing functions aimed for practical use. It is a very important issue because if USAR robots work in a real disaster sites, we can reduce workers and risks of secondary disaster [2].

Teleoperation remains an important part of our interaction with USAR robot [3]. We have to consider requirement specification for USAR robots such as types of camera, controller, graphic user interface, and some additional sensors like a laser range finder. In a teleoperation, usually a few cameras are mounted on a robot to understand environments, and Images are sent to displays by network communication.

T. Sato, A. Yamashita, and H. Asama are with Department of Precision Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan {satoh, yamashita, asama}@robot.t.u-tokyo.ac.jp

A. Moro is with Department of Precision Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan, and Ritecs, 11-5-3 44 Shibasaki, Tachikawa-shi, Tokyo, Japan alessandromoro.italy@ritecs.co.jp

A. Sugahara and T. Tasaki are with Toshiba R&D Center, Toshiba Corporation, I Komukai Toshiba-cho, Saiwa-ku, Kawasaki 210, Kanagawa, Japan {atsushi.sugahara, tsuyoshi.tasaki}@toshiba.co.jp

II. Related works

Over the last few decades, a lot of studies have been made on reducing blind spots for teleoperation [4]. A Normal approach is a simple way which uses multiple cameras and showing these images individually. In this method, there is a problem that operators cannot understand surroundings easily because it needs some estimations of relations between images from each camera image.

Showing bird's-eye view images are one of the effective methods to solve the estimation problem. Nagatani *et al.* [5] mounted looking down cameras to a poll on a robot. With this solution which reduces the field of view, operators see surroundings, but also a robot. It helps operators understand relations between surroundings and a robot with no estimation, however this system can work only in high ceiling environment because of a tall pole, so that this method cannot adapt in low ceiling disaster site.

Several studies have been made on creating virtual bird's-eye view images from multiple cameras to solve ceiling problem. These studies mounted multiple wide view cameras on a car, and created a top view from these cameras by image processing using geometric transformation [6], [7], [8]. These methods have already been practically used as park assist systems in some cars [9], [10]. Virtual bird's-eye view images can show the surroundings of a robot as bird's-eye-view images so that operators can understand relations between surroundings and a robot more easily. In addition, Virtual bird's-eye view enables to work in low ceiling environment. Based on these advantages, we also mount multiple fish-eye cameras on a robot and create bird's-eye view images. The clear difference between our proposal

978-1-4799-2625-1/13/$31.00 ©2013 IEEE

and bird's-eye view park assist system is use. Park assist system is used for only parking. On the other hand, our proposal system is used for various purposes needed in teleoperation. It means that creating bird's-eye view is one of the functions and purposes in our method. Our system also provides other images by using multiple fish-eye cameras.

Several studies used spatio-temporal data of single camera and create a view from rear above [11], [12]. These studies mounted a GPS and a camera on the front of a robot, stored images as spatio-temporal data by synthesizing localization result. Bird's eye-view images from rear above was created by showing a past image and pasting robot CG on a current position in a past image. Spatio-temporal captured data can be reused so that it can extend and compensate the area which is not captured at current time. For the reasons above, we also create bird's-eye view as spatio-temporal data.

These studies reduce blind spots, however the problem which we have to also consider is practical use. When USAR robots are used in disaster sites, many troubles have to be considered such as camera broken and network disturbance. These troubles often cause images not available so that recovering these missing images is very important. Using multiple cameras can reduce the risk of no view, so that we create bird's-eye view from multiple cameras.

In summary, considering missing images is very important even though reducing blind spots and showing a relation between surroundings and a robot is also important. Past information can be reused to recover missing images so that we propose creating bird's-eye view images from multiple fish-eye cameras as spatio-temporal data. It develops teleoperation more robustly.

III. Proposed method

We propose a method to construct bird's-eye view images using spatio-temporal data from multiple fish-eye cameras. To create a bird's-eye view images, first we capture images from multiple fish-eye cameras. Fish-eye images are rectified from fish-eye images to perspective images by spherical mapping, then transformed to bird's-eye view image by perspective transform. We combine each bird's-eye view image to a bird's-eye view image.

After creating bird's-eye view image, it is stored as spatio-temporal data by synthesizing localization data and time data. If some parts of images are not acquired, this method compensates for parts by reusing past stored spatio-temporal data. Finally, compensated bird's-eye view images are displayed to an operator.

A. Rectification

Fish-eye camera has wide angle that produces strong visual distortion. Because of the wide view, we can capture under region images even if lens orientation and ground are parallel. An example of captured images are shown in Fig. 2(a). To recognize images easily, spherical mapping is one of the methods to rectify fish-eye images to pinhole images [13]. It can change the area of rectification. Front image is shown in Fig. 2(b), left image is shown in Fig. 2(c), right image is shown in Fig. 2(d).

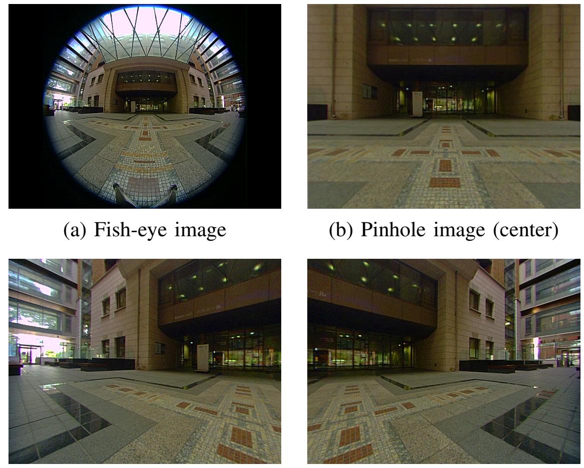

(a) Fish-eye image (b) Pinhole image (center)
(c) Pinhole image (left) (d) Pinhole image (right)

Fig. 2: Example of capture image and rectification.

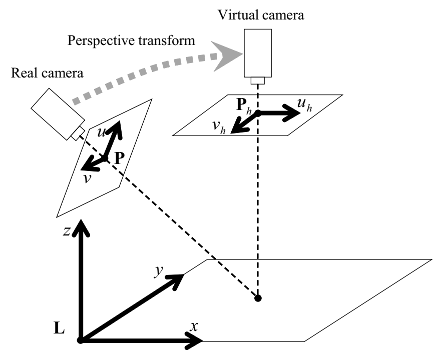


Fig. 3: Perspective transform.

B. Perspective transform

To create bird's-eye view images from each camera, we use a perspective transform. Geometric transformation between real camera and virtual camera is shown in Fig. 3. First we consider the relation between local coordinate $\mathbf{L} = [x, y, z]^T$ and real camera coordinate $\mathbf{P} = [u, v]^T$. This relational expression is described by using homography matrix $\mathbf{H}$ (3×4), that is to say, there is a relational expression between a matrix which includes local coordinate $\mathbf{P}_L = [x, y, z, 1]^T$ and another matrix which include real camera coordinate $\mathbf{P}_r = [u, v, 1]^T$ as follows:

$$\mathbf{P}_r = \mathbf{H}\mathbf{P}_L. \tag{1}$$

Then if we postulate projection plane is $z = 0$, the expression is more simpler as follows:

$$\begin{aligned} \mathbf{P}_r &= \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} h_{11} & h_{12} & h_{13} & h_{14} \\ h_{21} & h_{22} & h_{23} & h_{24} \\ h_{31} & h_{32} & h_{33} & h_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ 0 \\ 1 \end{bmatrix} \\ &= \begin{bmatrix} h_{11} & h_{12} & h_{14} \\ h_{21} & h_{22} & h_{24} \\ h_{31} & h_{32} & h_{34} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \equiv \mathbf{H}_r\tilde{\mathbf{P}}_L. \end{aligned} \tag{2}$$

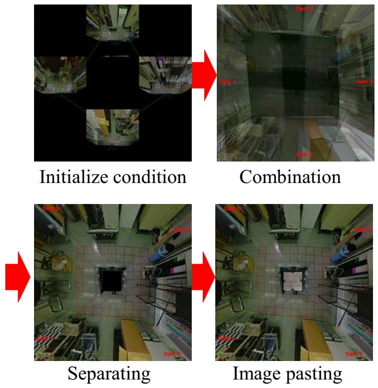


Fig. 4: Combination Procedure.

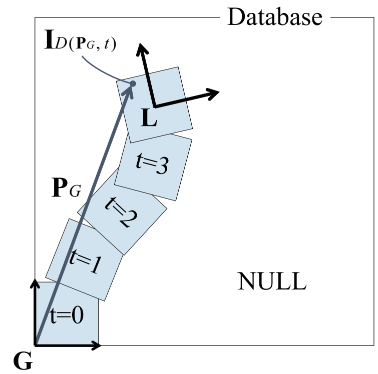


Fig. 5: Database of spatio-temporal data.

In the same way, the relationship between matrix which include real camera coordinate and a matrix including virtual camera coordinate $\mathbf{P}_h$ is described as follows:

$$\mathbf{P}_h = \mathbf{H}_h \tilde{\mathbf{P}}_L, \tag{3}$$

where $\mathbf{H}_h$ is also homography matrix when projection plane is $z = 0$. Then we obtain a relation between real camera coordinate and virtual camera coordinate as follows:

$$\mathbf{P}_r = \mathbf{H}_r \mathbf{H}_h^{-1} \mathbf{P}_h, \tag{4}$$

When a bird's-eye view image is created, it must be noted that we must consider all of the virtual camera positions and orientations are similar.

C. Combination

Figure 2 shows an example of combination part. After creating bird's-eye view images, we combine these images together and create a bird's-eye view. To calibrate each virtual camera, we put squares such as checker pattern on the ground. Each camera has overlapping regions so that each image has shared some squares. These squares should be in the same position in combined bird's-eye view. We calibrate bird's-eye view by using this characteristic. After calibration, overlapped regions of each image are separated. Finally a robot image which is taken at top is pasted in the combined bird's eye-view image.

D. Spatio-temporal data

In order to store bird's-eye view image as spatio-temporal data, getting time and localization are needed. Spatio-temporal bird's-eye view stores past images to a database, however local coordinate moves with a robot moves. Due to the reason, global coordinate system $\mathbf{G}$ is defined when a robot starts moving. We consider global coordinate $\mathbf{P}_G = [x', y']^T$ and a matrix which includes global coordinate $\tilde{\mathbf{P}}_G = [x', y', 1]^T$. To estimate localization, dead reckoning is a simple method, however errors are caused by slipping. To consider this error, we also use a laser range finder which is one of the distance-measurement sensor. It gives translation vector $\mathbf{t}$ and rotation matrix $\mathbf{R}$ by applying current range scan data and previous data to ICP (Iterative Closest Point) algorithm [14], [15]. Then local coordinate can be transferred to a global coordinate as follows:

$$\tilde{\mathbf{P}}_G = \begin{bmatrix} \mathbf{R} & \mathbf{t} \\ \mathbf{0} & 0 \end{bmatrix} \tilde{\mathbf{P}}_L. \tag{5}$$

By using this expression, current bird's-eye view images $\mathbf{I}_{(x',y',t)}$ which have color data such as RGB is stored as spatio-temporal data $\mathbf{I}_{D(x',y')}$.

E. Compensation

Spatio-temporal bird's-eye view image is created from a database. The database has a size d_s, and data is updated as a robot moves. Outline of the database is shown in Fig. 5. It compensates current bird's-eye view images if images have missing areas.

Note that when missing areas are detected, operators should understand troubles before compensation. Because spatio-temporal data is created at the other robot position and time so that compensated bird's-eye view images sometimes confuses operators. To avoid this confusion, our method displays a message before compensation.

If there is past spatio-temporal data in a database, it is overwritten by latest data. Because a latest data has minimum error of localization. In addition, the data saves memory resource. Finally compensated current image $\mathbf{I}_{(x',y',t)}$ is displayed to an operator.

The algorithm for storing and compensation is as follows:

Algorithm 1 Storing and Compensation

```
for x̃ = −d_s/2 to d_s/2 do
  for ỹ = −d_s/2 to d_s/2 do
    if I_(x',y',t) =NULL && I_D(x',y') ≠ NULL then
      I_(x',y',t) ⇐ I_D(x',y')
    else
      I_D(x',y') ⇐ I_(x',y',t)
    end if
  end for
end for
```

IV. Demonstration of Bird's-eye view

Toward a more practical use of USAR robot, we integrated our bird's-eye view system with a USAR robot teleoperation system in Anti-disaster Unmanned System R&D Project of the New Energy and Industrial Technology Development Organization (NEDO) . In a demonstration of the project, we showed the effectiveness of bird's-eye view.

A. Environment

A USAR robot which was developed in Anti-disaster Unmanned System R&D Project is shown in Fig. 6(a), and an example of teleoperation is shown in Fig. 6(b). The left image is frontal view image with laser range finder data. Right image is our bird's-eye view image, and bottom images are past captured images.

Four multiple fish-eye cameras (NM33, made of OPT Corporation) which have 180° view and 15 fps maximum frame rate look each direction, and a LRF (UTM-30LX, made of HOKUYO Corporation) are mounted on the robot. The height of the robot is 70.0cm.

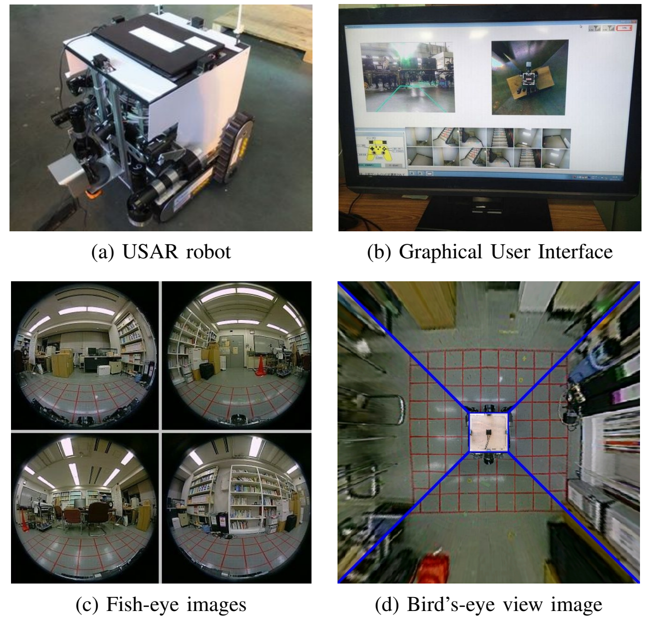
An original resolution of fish-eye image (Fig. 7(c)) is 640×480 pixel, and we created bird's-eye view image (Fig. 7(d)) which has 500×500 pixel. Graphical user interface obtains the bird's-eye view images at 9 fps by wireless network.

Environment in the demonstration is shown in Fig. 9(a). The robot which has a width of 65cm moves narrow path. The width of path is 80cm so that moving with no collisions is difficult. Teleoperation using only front images can hardly move this narrow path, and collisions sometimes make troubles. Due to the reason, demonstration without bird's-eye view was never conducted.

B. Result

Figure 7 shows an example scene of the demonstration. The robot had gone through the path 2 times with no collisions. Before the demonstration, we tested same experiment 4 times, which was also no collisions. This is because bird's-eye view images can show a relationship between the robot and environment. It means bird's-eye view improves collision safety of a rescue robot in narrow disaster site.

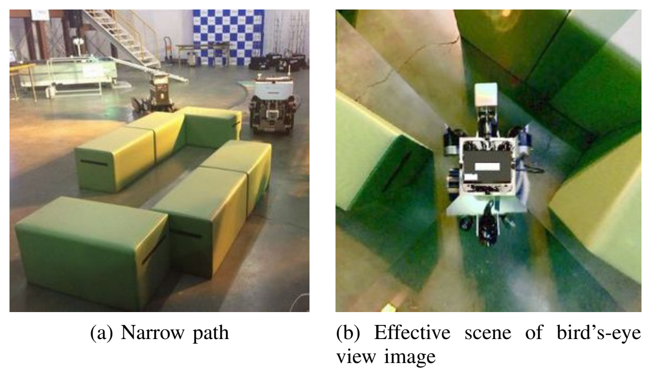
In addition, we conducted another experiment which evaluates the accuracy of positioning. Six examinees teleoperated the robot one time. The robot moved straight at stop line 4 meters along ahead and stop at the line. First an examinee teleoperates with bird's-eye view and front view. Second the examinee teleoperates with only front view. The result is shown in Fig. 8. It clearly shows that bird's-eye view improves the accuracy of positioning of the robot. Distance error is less than 10cm when examinees used bird's-eye view. The result indicates two types advantage of bird's-eye view. One is the same advantage mentioned above that bird's-eye view images can show a relationship. The other is bird's-eye view can show surroundings with no blind spots. Frontal image cannot show under regions because the angle of view is low so that stop line is disappearing when robots come as near the line.



(a) USAR robot (b) Graphical User Interface

(c) Fish-eye images (d) Bird's-eye view image

Fig. 6: Teleoperation system of USAR robot and an example of original fish-eye images and bird's-eye view image.



(a) Narrow path (b) Effective scene of bird's-eye view image

Fig. 7: Demonstration result.

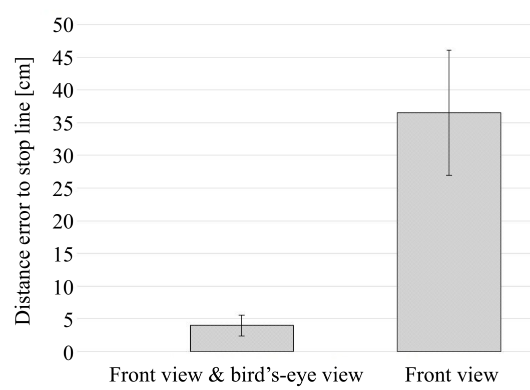


Fig. 8: Evaluation of distance error to a stop line. Error bars indicate standard deviation.

V. Experiment

In an experiment, we make certain that spatio-temporal bird's-eye view images are created robustly even when some of images are missing. We consider the case that rear camera images are not acquired because of some troubles. USAR robot is the same one which we performed in the demonstration.

A. Environment

The USAR robot is shown in Fig. 9(a). Difference in the demonstration is a position of LRF. We mounted a LRF on top of the robot.We teleoperated robot in the room. An example of teleoperation scene is shown in Fig. 9(b). The robot moved down straight hallway shown in Fig. 9(c).

An environment map and captured positions is shown in Fig. 9(d). In this experiment, the robot goes 4m, and capture the bird's-eye view image every robot goes 1m.

Figure 9(e) shows our graphical user interface. It can change the view to bird's-eye view image, each fish-eye image, and each rectified image by touching button. We use touch panel type PC so that it's very easy to change a view. In case camera positions and orientations are changed due to some troubles such as collision, the system can support to recalibrate the bird's-eye view parameters instantly.

In this experiment, bird's-eye view images are created from high resolution fish-eye images (1536 $\times$ 1536 pixel). It makes high quality images, however computation time is increased. Graphic user interface obtains images at 4 fps.

B. Result

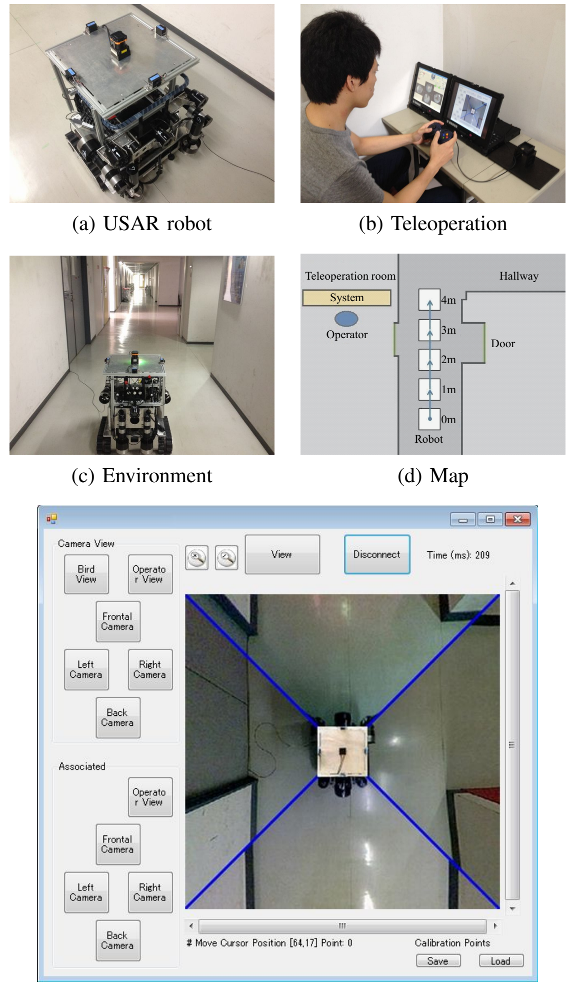
The main result is shown in Fig. 10. Vertical axis are arranged in each item. First from the top is the front view which is made by rectifying a fish image. This is a one of the advantages of a fish-eye camera. That is to say, fish-eye camera image is not only used to create the bird's-eye view images, but also various images needed for teleoperation.

Second from the top is the case which bird's eye-view images are created from all of cameras. There is no troubles, however if some of images are not acquired, bird's eye-view has blind spots as third from the top. We have to consider many troubles in practical use like this case. Forth from the top is a spatio-temporal bird's-eye view which we propose. Horizontal axis are arranged in a position of the robot.

Now we discuss the spatio-temporal bird's-eye view. In the 0m position, there is no spatio-temporal data to compensate missing parts of rear image so that there is no compensation. However, this is not a serious problem because this situation happens only when the robot starts to move.

In the 1m position, some of missing parts are compensated even though still there are missing parts. This is just because still there are no stored data, however it is not a big problem because the robot just started moving.

On the after 2m position, all of missing parts are compensated. The result shows that spatio-temporal bird's-eye view can show images robustly even when some parts of images are not acquired. Effective scenes of spatio-temporal bird's-eye view occur when rear camera is broken at narrow path.



(a) USAR robot (b) Teleoperation

(c) Environment (d) Map

(e) Graphical user interface

Fig. 9: Teleoperation system and experimental environment.

Crawler type of USAR robot does not distinguish between front and rear so that USAR robot can go back safely by using spatio-temporal bird's-eye view images which compensate rear view images.

Spatio-temporal bird's-eye view shows surroundings robustly. However, there is a problem which we should solve as future works. Spatio-temporal bird's-eye view shows an incorrect image on the 3m position. Thw view shows the black edge of a door, but normal bird's-eye view does not show the edge. This problem is caused by the error of localization. In featureless environment such as this hallway, ICP algorithm does not work well. We should have to consider more types of localization method, or constructing new methods.

VI. Conclusion

In this research, we propose a method to construct spatio-temporal bird's-eye view images using multiple fish-eye cameras. We stored bird's-eye view images with localization and getting time as spatio-temporal data. If bird's-eye view images have blind spots, spatio-temporal data compensates these spots. It develops teleoperation more robustly.

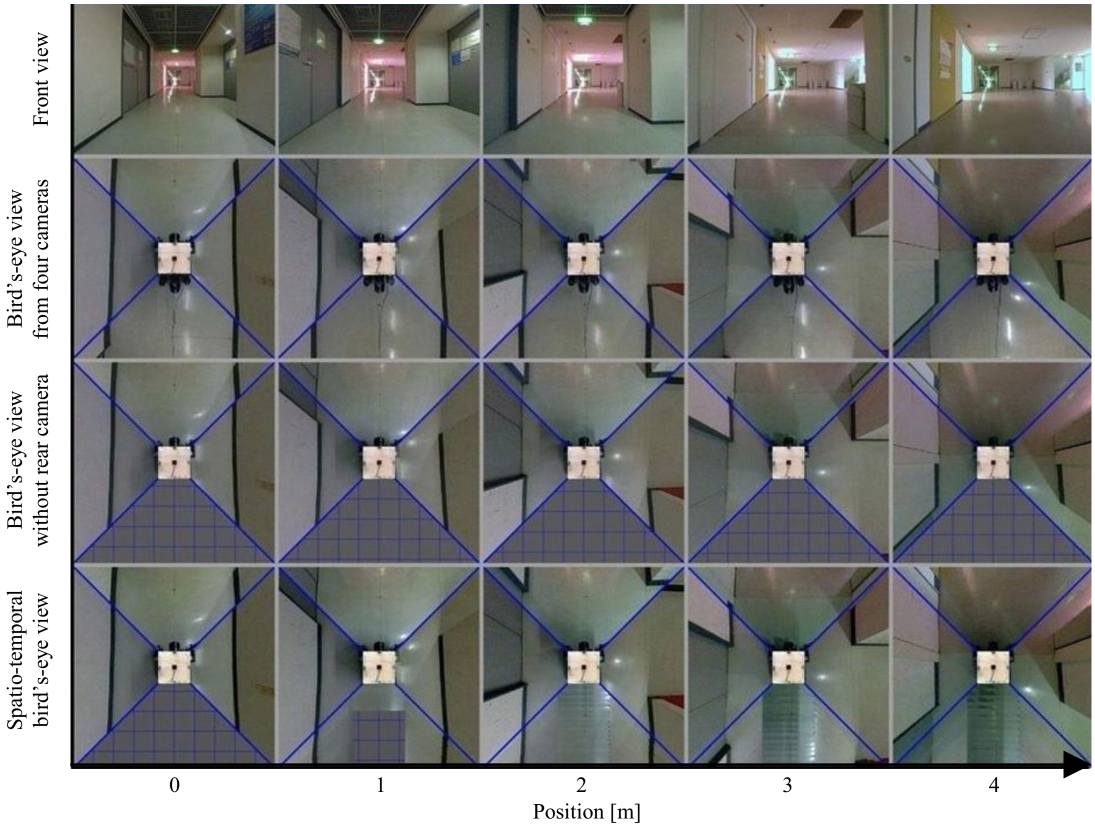


Fig. 10: Result of images. Bird's-eye view images are created from four cameras so that if images are not available, there are blind spots. Spatio-temporal bird's-eye view compensates these blind spots by using past stored images.

ACKNOWLEDGMENT

This work was in part supported by of the Anti-disaster Unmanned System R&D Project of the New Energy and Industrial Technology Development Organization (NEDO), Japan, MEXT KAKENHI, Grant-in-Aid for Young Scientists (A) 22680017, and the Asahi-Glass Foundation.

REFERENCES

[1] S. Ali A. Moosavian, Hesam Semsarilar, Tetsushi Kamegawa, and Arash Kalantari, "Design and Manufacturing of a Mobile Rescue Robot", *Proceedings of the 2006 IEEE / RSJ International Conference on Intelligent Robots and Systems*, (2006), pp.3982–3987.

[2] Satoshi Tadokoro, "Rescue Robotics Challenge", *Proceedings of the 2010 IEEE Workshop on Advanced Robotics and its Social Impacts*, (2010), pp.92–98.

[3] Mohammed Waleed Kadous, Raymond Ka-Man Sheh, and Claude Sammut, "Effective User Interface Design for Rescue Robotics", *Proceedings of the 2006 ACM Conference on Human Robot Interaction*, (2006), pp.250–257.

[4] Thomas B. Sheridan, "Teleoperation, Telerobotics and Telepresence: A Progress Report", *Proceedings of the IFAC Control Engineering Practice*, (1995), pp.205–214.

[5] Keiji Nagatani, Seiga Kiribayashi, Yoshito Okada, Satoshi Tadokoro, Takeshi Nishimura, Tomoaki Yoshida, Eiji Koyanagi, and Yasushi Hada, "Redesign of Rescue Mobile Robot Quince - Toward Emergency Response to the Nuclear Accident at Fukushima Daiichi Nuclear Power Station on March 2011 - ", *Proceedings of 2011 IEEE International Workshop on the Safety, Security and Rescue Robotics*, (2011), pp.13–18.

[6] Yohei Ishii, Keisuke Asari, Hitoshi Hongo, and Hiroshi Kano, "A Practical Calibration Method for Top View Image Generation", *Proceedings of the 2008 IEEE International Conference on Consumer Electronics*, (2008), pp.1–2.

[7] Yu-chih Liu, Kai-ying Lin, and Yong-sheng Chen, "Bird's-Eye View Vision System for Vehicle Surrounding Monitoring", *Robot Vision - RobVis*, (2008), pp.207–218.

[8] Tobias Ehlgen, and Tomas Pajdla, "Monitoring Surrounding Areas of Truck-Trailer Combinations", *Proceedings of the 5th International Conference on Computer Vision Systems*, (2007).

[9] Akihiro Kanaoka, Teruhisa Takano, Daisuke Sugawara, Sotaro Otani, Masayasu Suzuki, Satoshi Chinomi, and Ken Oizumi, "Development of the Around View Monitor (Special Feature: The Development of Safety Technology)", *Nissan Technical Review*, (2008), pp.37–41 (in Japanese).

[10] Seiya Shimizu, Jun Kawai, and Hiroshi Yamada, "Wraparound View System for Motor Vehicles", *Fujitsu Scientific & Technical Journal*, (2010), pp.95–102.

[11] Masataka Ito, Noritaka Sato, Maki Sugimoto, Naoji Shiroma, Masahiko Inami, and Fumitoshi Matsuno, "A Teleoperation Interface using Past Images for Outdoor Environment", *Proceedings of the SICE Annual Conference*, (2008), pp.3372–3375.

[12] Shiroma Naoi, Kagotani Georges, Sugimoto Maki, Inami Masahiko, and Matsuno Fumitoshi, "A Novel Teleoperation Method for a Mobile Robot Using Real Image Data Records", *Proceedings of the 2004 IEEE International Conference on Robotics and Biomimetics*, (2004), pp.233–238.

[13] Ciaran Hughes, Patrick Denny, Martin Glavin, and Edward Jones, "Equidistant Fish-eye Calibration and Rectification by Vanishing Point Extraction", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.32, No.12, (2010), pp.2289–2296.

[14] Paul J. Besl and Neil D. McKay, "A Method for Registration of 3-D Shapes", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol.14, No.2, (1992), pp.239–256.

[15] Feng Lu, and Evangelos Milios, "Robot Pose Estimation in Unknown Environments by Mathching 2D Range Scans", *Proceedings of the 1994 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (1994), pp.935–938.