Komatsu *et al. Robomech J* (2020) 7:15
https://doi.org/10.1186/s40648-020-00163-4

ROBOMECH Journal

RESEARCH ARTICLE Open Access

Free viewpoint image generation system using fisheye cameras and a laser rangefinder for indoor robot teleoperation

Ren Komatsu[1*], Hiromitsu Fujii[2], Yusuke Tamura[1], Atsushi Yamashita[1] and Hajime Asama[1]

Abstract

In robot teleoperation, a lack of depth information often results in collisions between the robots and obstacles in its path or surroundings. To address this issue, free viewpoint images can greatly benefit the operators in terms of collision avoidance as the operators are able to view the robot's surrounding from the images at arbitrary points, giving them a better depth information. In this paper, a novel free viewpoint image generation system is proposed. One approach to generate free viewpoint images is to use multiple cameras and Light Detection and Ranging (LiDAR). Instead of using the expensive LiDAR, this study utilizes a cost-effective laser rangefinder (LRF) and a characteristic of man-made environments. In other words, we install multiple fisheye cameras and an LRF on a robot. Free viewpoint images are generated under the assumption that walls are perpendicular to the floor. Furthermore, an easy calibration for estimating the poses of the multiple fisheye cameras, the LRF, and the robot model is proposed. Experimental results show that the proposed method can generate free viewpoint images using cameras and an LRF. Finally, the proposed method is primarily implemented using OpenGL Shading Language to utilize a graphics processing unit computation to achieve a real-time processing of the multiple high-resolution images. Supplementary videos and our source code are available at our project page (https://matsuren.github.io/fvp).

Keywords: Robot teleoperation, Free viewpoint images, Human interface, Real-time visualization

Background

Visualizing the surrounding environment of a robot is important for an efficient robot teleoperation. Therefore, visualization methods have been comprehensively studied over a long period [1–7]. One of the difficulties in robot teleoperation is the lack of depth perception. During robot teleoperation, the operators view images captured by the cameras mounted on the robot, predict the situation of the robot, and decide the next move for the robot. However, the images do not provide much depth information, which sometimes leads to collisions between the robots and obstacles.

To address the issue of obstacle collision, Keyes et al. investigated the relationship between camera positions and collisions of teleoperated robot [4]. They compared a forward-facing camera, which provides first-person view images, to an overhead camera, which provides third-person view images. From the comparison, they concluded that the third-person view images were more beneficial for obstacle avoidance as operators could see both the robot body itself and the obstacles.

Some studies have been conducted to provide third-person view images without the overhead cameras to deal with some spatial constraints, e.g., environments with a low ceiling. Sato et al. demonstrated a system that generates bird's-eye view images from multiple first-person view images captured by the cameras on the robot [5]. They warped the first-person view images by homography transformation and combined them to generate bird's-eye view images. They concluded that

*Correspondence: komatsu@robot.t.u-tokyo.ac.jp
[1] Graduate School of Engineering, The University of Tokyo, Tokyo 113-8656, Japan
Full list of author information is available at the end of the article

the bird's-eye view images helped operators to avoid collisions as it is easy for them to understand the relationship between the robot and the obstacles in the bird's-eye view images. However, if obstacles were higher than the floor, they were not visualized correctly in the bird's-eye view images, leading to collisions between the robots and obstacles. Thus, Awashima et al. superimposed 3D points of obstacles obtained using depth sensors onto the bird's-eye view image to visualize the locations of the obstacles correctly [6].

In robot teleoperation, enabling the operators to view images from arbitrary viewpoints is beneficial because these images give them a depth perception by motion parallax. Showing multiple images taken by different positions of cameras might give the operators some depth perceptions as well, however, displaying a lot of sensor information to operators is not recommended because a lot of sensor information tends to confuse operators [1]. Sun et al. proposed a system that provides third-person view images from arbitrary viewpoints for the teleoperation of construction machines [7]. They approximated the surrounding environment as a semi-spherical mesh model, and the images captured by the cameras on the construction machine were projected onto the mesh model to generate third-person view images. This approach works well in construction sites; however, it generates distorted third-person view images in some situations such as indoor where the surrounding environment is hardly approximated as a semi-spherical mesh model. As a result, the distorted third-person view images make it more difficult for operators to perceive the distance between the robot and the obstacles.

Ferland et al. proposed a method to generate third-person view images using a camera and a laser rangefinder (LRF) for an indoor robot teleoperation [2]. However, the third-person view images were only available in front of the robot. Therefore, in order to investigate the surrounding environment, turning the robot in various directions is necessary. Moving the robots without sufficient information of the surrounding environment might lead to collisions with obstacles. Hence, it is desirable to view the surrounding environment without moving the robot.

In this study, we propose a novel third-person view generation system for an indoor robot teleoperation. This system is called a free viewpoint image generation system as it provides third-person view images from arbitrary viewpoints, not only from the front of the robot as in [2], but also from the surrounding of the robot. As opposed to the previous method [7], the generated free viewpoint images are not distorted, hence, the operators can perceive the distance between the robot and the obstacles easily. Furthermore, the operators can view the guide maps and room number signs in the images, which helps them understand the indoor scene structures. An example of the indoor scene with guide maps is shown in Fig. 1.

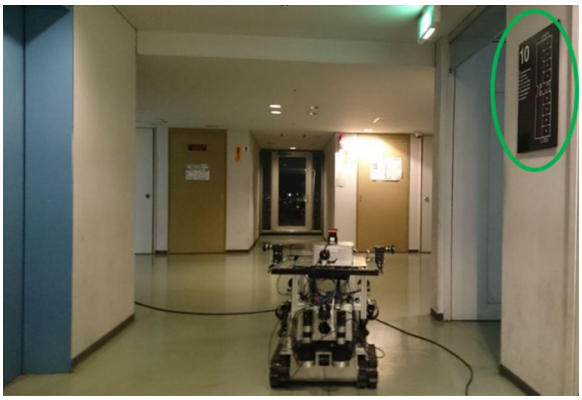


Fig. 1 Example of an indoor scene. The guide map is marked with a green circle on the upper right of the image. The guide map is beneficial for operators to decide directions for robots to reach their destination

The three-dimensional (3D) model of the surrounding environment and its texture are required to generate the free viewpoint images. One method to measure the 3D model is to use the 3D Light Detection and Ranging (LiDAR). Instead of using LiDARs, this study utilizes a two-dimensional (2D) LRF and a characteristic of man-made environments because the minimum range of the LiDARs is bigger for indoor environments (e.g., 1 m for a Velodyne HDL-32E) compared to that of the LRFs (e.g., 0.1 m for a Hokuyo UTM-30LX), and LRFs are much more cost-effective. We also use multiple high-resolution fisheye cameras to get the texture of the surrounding environment. Besides that, an easy calibration for estimating the poses of the multiple fisheye cameras, an LRF, and the 3D models is proposed. The proposed method is mostly implemented with OpenGL Shading Language (GLSL) to utilize a graphics processing unit (GPU) computation to achieve a real-time processing of multiple high-resolution images. Consequently, the operators can change the viewpoints smoothly, which helps them obtain a depth perception by motion parallax.

Overview of the proposed method

To generate free viewpoint images, 3D models of the surrounding environment and its texture are required. In this study, we assumed that the indoor environment consists of only three elements: one robot, one floor, and walls (Fig. 2). Figure 2 is an illustration of the assumed indoor environment. This assumption is widely valid for indoor environments, especially for corridors as shown in Fig. 1. Therefore, only 3D models, relative poses, and

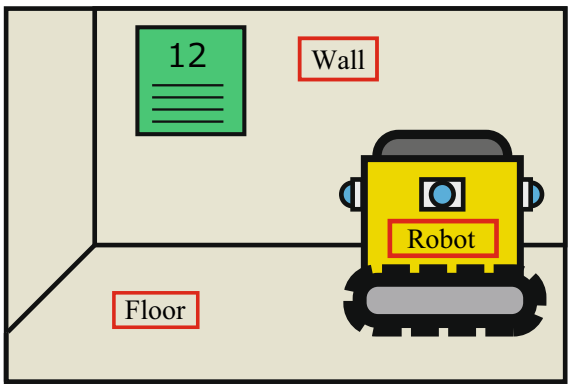


Fig. 2 Assumption of an indoor environment. It consists of three elements: robot, floor, and wall

textures of the robot, the floor, and the walls are needed to generate the free viewpoint images when this assumption is employed.

The overview of our proposed method is illustrated in Fig. 3. First, the 3D models of the robot, the floor, and the walls are obtained. Among these three 3D models, only the robot model has a texture. Second, the relative poses of the robot, the floor, and the walls are estimated and combined so that they have the same world coordinate. Finally, the textures of the floor and the walls are given by the fisheye cameras with known pose on the robot, and free viewpoint images are generated from the 3D models of the surrounding environment with textures.

System configurations

In this study, an unmanned ground vehicle (UGV) is used as an indoor teleoperated robot. Multiple fisheye cameras facing various directions are installed on the robot so that the surrounding environments are captured. Moreover, each camera has some overlapped coverage regions between adjacent cameras for the proposed calibration, which is explained in "Estimate relative poses" section.

An LRF is installed on the top of the robot so that the scan lines are parallel to the floor to estimate the 3D model of the walls, which is explained in "Estimate 3D models" section.

Estimate 3D models

3D model of the floor

Only a single floor exists in the assumed indoor environment, and thus, the 3D model of the floor can be easily approximated as a plane.

3D model of the robot

The 3D model of the robot does not change during teleoperation. Therefore, it is possible to prepare its 3D model in advance. For example, 3D CAD models provided by manufacturing companies are available in some cases.

In this study, open source structure-from-motion software [8–11] is used to create a dense 3D model with the texture from multiple images. As such, the 3D model of a robot can be obtained easily even without the provision of the 3D models by the manufacturing companies.

3D models of the walls

In order to estimate the 3D models of the walls, this study utilizes a cost-effective LRF and the characteristic of man-made environments. The said characteristic is that the walls are perpendicular to the floor. This assumption is similar to the Manhattan world assumption [12] and the Atlanta world assumption [13], which is commonly observed in indoor environments. By using this

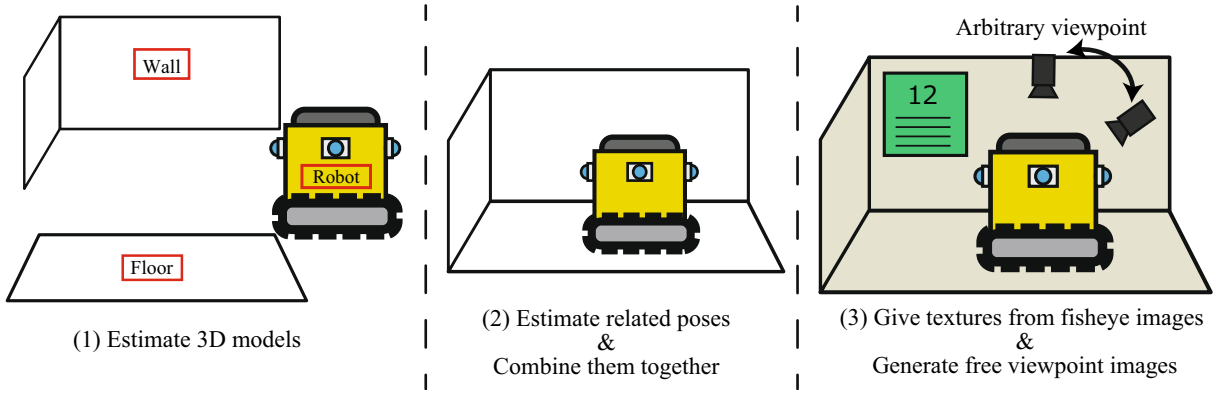


Fig. 3 Overview of the proposed method. From left to right: (1) 3D models of the robot, the floor, and the walls are estimated. Only the robot model has a texture. (2) The relative poses of the robot, the floor, and the walls are estimated and combined. (3) Their textures are obtained from fisheye images, and free viewpoint images are generated

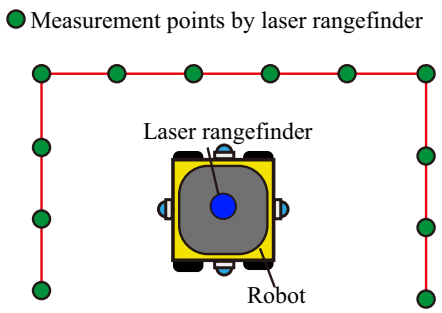


Fig. 4 Estimating the 3D models of the walls (Top-view). The yellow rectangle, the blue circle, and the green circles represent the robot, the LRF, and the measurement points by the LRF, respectively. The red lines connecting the green circles represent the 3D models of the wall

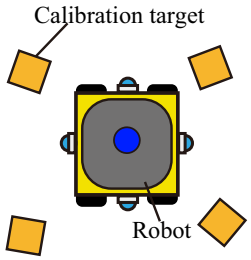


Fig. 5 Calibration targets placed on the floor (Top view). If four cameras are mounted on the robot, four square boards should be placed as the calibration targets on the floor so that at least two cameras can capture them

characteristic, the 3D models of the walls are estimated (Fig. 4). Figure 4 illustrates the estimation of the 3D models of the walls from the measurement points by the LRF. The yellow rectangle, the blue circle, and the green circles represent the robot, the LRF, and the measurement points by the LRF, respectively. The red lines connecting the green circles represent the 3D models of the walls. To be more specific, one of the rectangle meshes of the walls is defined by the following set of the four vertices:

$$\left\{ \begin{pmatrix} p_x^i \\ p_y^i \\ 0 \end{pmatrix}, \begin{pmatrix} p_x^i \\ p_y^i \\ h_{\text{wall}} \end{pmatrix}, \begin{pmatrix} p_x^{i+1} \\ p_y^{i+1} \\ 0 \end{pmatrix}, \begin{pmatrix} p_x^{i+1} \\ p_y^{i+1} \\ h_{\text{wall}} \end{pmatrix} \right\}, \tag{1}$$

where $(p_x^i, p_y^i)^\mathsf{T} \in \mathbb{R}^2$ is the location of the i-th measurement point of the LRF, and h_{wall} is the height of the walls which is a parameter given by the users based on the scene. The local coordinate of the LRF is defined, so that the scan lines are perpendicular to z-axis.

Estimate the relative poses

In this section, the relative poses of the 3D models of the robot, the floor, the walls, and the multiple fisheye cameras on the robot are estimated.

First of all, we define the world coordinate so that the plane of the floor is in an xy-plane, and the origin of the world coordinate $\mathbf{O} = (0, 0, 0)^\mathsf{T}$ is set to a point on the floor. The z-axis is directed upward. In other words, $(p_x, p_y, 0)^\mathsf{T}$ and $(p_x, p_y, h)^\mathsf{T}$ indicate a point on the floor and a point at the height h from the floor, respectively.

As the 3D model of the walls is defined by the LRF in Eq. (1), the relative pose of the LRF should be estimated instead of that of the walls.

Therefore, the parameters that need to be estimated are as follows:

- Relative poses of the floor and the fisheye cameras
- Relative poses of the robot model
- Relative pose of the LRF

Relative poses of the floor and fisheye cameras

The intrinsic parameters of the fisheye cameras are needed to be estimated in advance [14, 15]. We use OCamCalib as it supports a wide field of view of the fisheye cameras (up to 195 degrees) and the closed-form expressions for the distortion and the undistortion.

The relative poses of the floor and the fisheye cameras are estimated in the similar manner as done in [5] except that instead of estimating the homography matrix of each image, we estimate the actual camera poses. In the previous method, square boards as calibration targets are placed on the floor, so that two adjacent cameras are able to capture the same target as shown in Fig. 5. For example, assuming that the poses of four cameras are to be estimated, then four targets should be placed on the floor. Next, four corners of each target are selected manually in each image and the corresponding points between two images are used for estimating the parameters.

In this study, instead of selecting the corners of the targets manually, AprilTag [16, 17] is employed as the calibration targets so that the corners of the targets are detected automatically. AprilTag uses a 2D bar code to distinguish the targets, which make it possible to find the corresponding points between two images automatically.

The relative poses are calculated by minimizing the reprojection error between the corresponding points on the images, which is formulated as follows:

$$E_{\text{reproj}} = \sum_{c_k \in \mathcal{C}} \sum_{i \in \mathcal{V}_{c_k}} \sum_{j=1}^{4} ||\omega_{c_k}\left(\mathbf{q}_{ij}^w\right) - \mathbf{u}_{ij}^{c_k}||^2, \tag{2}$$

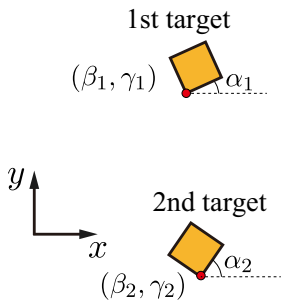


Fig. 6 Parameterization of the calibration targets (*xy*-plane). The parameters β_i and γ_i represent the location of the *i*-th target, and α_i represents the rotation of the *i*-th target

where $\mathcal{C}$ is the set of all the cameras, $\mathcal{V}_{c_k}$ is the set of all the indices of the targets which can be seen from the camera c_k, and $\mathbf{u}_{ij}^{c_k} \in \Omega$ is the point corresponding to $\mathbf{q}_{ij}^{w}$ captured by the camera c_k in the image domain Ω. $\mathbf{q}_{ij}^{w} \in \mathbb{R}^3$ is the *j*-th corner location of the *i*-th target in the world coordinate, which is formulated as follows:

$$\left(\mathbf{q}_1^p \ \mathbf{q}_2^p \ \mathbf{q}_3^p \ \mathbf{q}_4^p \right) = \begin{pmatrix} 0 & L & L & 0 \\ 0 & 0 & L & L \\ 0 & 0 & 0 & 0 \end{pmatrix}, \tag{3}$$

$$\mathbf{q}_{ij}^{w} = \begin{pmatrix} \cos\alpha_i & -\sin\alpha_i & 0 \\ \sin\alpha_i & \cos\alpha_i & 0 \\ 0 & 0 & 1 \end{pmatrix} \mathbf{q}_j^p + \begin{pmatrix} \beta_i \\ \gamma_i \\ 0 \end{pmatrix}, \tag{4}$$

where L is a side length of the targets. α_i, β_i, and γ_i are the parameters to be optimized, which determine the pose of the *i*-th target as shown in Fig. 6. The parameterization of the calibration targets in *xy*-plane is illustrated in Fig. 6.

$\omega_{c_k} : \mathbb{R}^3 \to \Omega$ is a function that projects a point $\mathbf{q}_{ij}^{w}$ in the world coordinate onto an image plane of a camera c_k, which is formulated as follows:

$$\omega_{c_k}\left(\mathbf{q}_{ij}^{w}\right) = \pi_{c_k}\left(\mathbf{R}_{c_k w}\mathbf{q}_{ij}^{w} + \mathbf{t}_{c_k w}\right), \tag{5}$$

where $\mathbf{R}_{c_k w} \in \mathrm{SO}(3)$ and $\mathbf{t}_{c_k w} \in \mathbb{R}^3$ are the rotation matrix and the translation vector, respectively, to determine the pose of the camera c_k. $\pi_{c_k} : \mathbb{R}^3 \to \Omega$ is a fisheye projection function [14, 15].

The cost function in Eq. (2) is minimized by using the Levenberg Marquardt (LM) method provided by [18] to estimate the camera poses, $\mathbf{R}_{c_k w}$ and $\mathbf{t}_{c_k w}$. The Rodrigues' rotation formula is employed to represent the rotation $\mathbf{R}_{c_k w}$ to avoid redundant parameterization. The initial values for the LM method are calculated by IPPE [19] which is suitable for estimating a camera pose using a plane-based calibration target.

Relative poses of the robot model

The 3D model of the robot is created using open source structure-from-motion software [8–11] where the scale of the 3D model cannot be obtained. Therefore, in addition to the relative pose between the robot model and the world coordinate, the relative scale is also needed to be estimated.

The relative scale and pose of the robot model are estimated by selecting some reference points $\mathbf{q}_i^m$ in the robot model coordinate and their corresponding $\mathbf{q}_i^w$ in the world coordinate. As the camera poses are already estimated, the camera locations are used as the reference points for the estimation. After selecting the points corresponding to the locations of the cameras in the model coordinate, the relative scale and pose, s, $\mathbf{R}_{wm}$, and $\mathbf{t}_{wm}$, are estimated by minimizing the following cost function:

$$E(s, \mathbf{R}_{wm}, \mathbf{t}_{wm}) = \sum_{c_k \in \mathcal{C}} ||\left(s\mathbf{R}_{wm}\mathbf{q}_{c_k}^m + \mathbf{t}_{wm}\right) - \mathbf{q}_{c_k}^w||^2, \tag{6}$$

where $\mathbf{q}_{c_k}^w$ is the location of the camera c_k in the world coordinate, which is calculated as follows:

$$\mathbf{q}_{c_k}^w = -\mathbf{R}_{c_k w}\mathbf{t}_{c_k w}, \tag{7}$$

and $\mathbf{q}_{c_k}^m$ is the location of the camera c_k in the robot model coordinate. The cost function in Eq. (6) can be solved by the closed-form solution [20].

Relative pose of the LRF

The LRF is installed on the top of the robot, so that the scan lines are parallel to the floor, and the height of the LRF does not affect the 3D model of the walls. Therefore, only the translation $t_{wl}^{xy} \in \mathbb{R}^2$ in the *xy* plane and the rotation $\alpha_{wl} \in \mathbb{R}$ around the *z*-axis should be estimated.

The procedure for estimating the relative pose of LRF is almost similar to the method of estimating the relative pose of the robot model, except that only 2D rigid body transformation is required for the LRF. Some measurement points of the LRF and the corresponding points in the world coordinate are chosen, and the relative pose of the LRF is estimated by [20]. In this study, the corners are selected as they are easily distinguished from the measurement points as can be seen in Fig. 4. The corresponding points in the world coordinate are selected in the bird's-eye view image, which is generated by projecting the fisheye images onto the 3D model of the floor by using the method described in "Give textures from fisheye images" section.

Give textures from fisheye images

The fisheye camera images are projected onto the 3D model of the surrounding environment geometrically to give textures by using the intrinsic parameters of the cameras and the relative poses. At first, all the 3D models

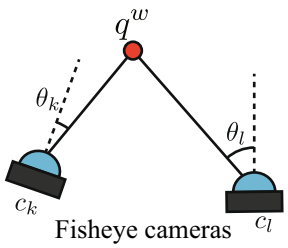


Fig. 7 Illustration of the incident angle. Camera c_k has the smallest incident angle θ_k and c_l has the second smallest incident angle θ_l. $\mathbf{q}^w$ is a point of a model in the world coordinate

are combined using the relative poses so that these 3D models have the same world coordinate. A texture of a point of a 3D model, $\mathbf{q}^w$, given by a camera c_k is formulated as follows:

$$I_{c_k}^{\text{tex}}(\mathbf{q}^w) = I_{c_k}\left(\omega_{c_k}\left(\mathbf{q}^w\right)\right), \tag{8}$$

where $I_{c_k}(u)$ is the color value of a pixel located at $u \in \Omega$ in an image of a camera c_k.

Blending in boundary

In the projection of the multiple fisheye camera images, textures may be obtained from multiple fisheye cameras for the same 3D position. In this study, the pixel values obtained from the center of the fisheye camera images are preferentially used as the spatial resolutions of the fisheye cameras can be decreased and distortion might exist when the incident angles of the optical rays are large. However, if the textures are obtained from a single camera image, the boundary between an area whose texture is given by a camera and another area whose texture is given by another camera is obviously visible. Therefore, a blending of the textures from two fisheye camera images is used near the boundary.

In the blending, two fisheye cameras, c_k with the smallest incident angle θ_k and c_l with the second smallest incident angle θ_l, are selected (Fig. 7). The weight for the blending is determined based on the incident angle, which is formulated as follows:

$$I^{\text{tex}}(\mathbf{q}^w) = \alpha I_{c_k}^{\text{tex}}(\mathbf{q}^w) + (1-\alpha) I_{c_l}^{\text{tex}}(\mathbf{q}^w), \tag{9}$$

$$\alpha = \begin{cases} 1 & (\theta_l - \theta_k \geq \theta_{\text{th}}) \\ \left(1 + \sin\left(\frac{\theta_l - \theta_k}{2\theta_{\text{th}}}\pi\right)\right)/2 & (\text{otherwise}) \end{cases}, \tag{10}$$

where $I^{\text{tex}}(\mathbf{q}^w)$ is the final texture at point $\mathbf{q}^w$ after the blending, and θ_{th} is a parameter that determines the blending range of textures for two fisheye cameras.

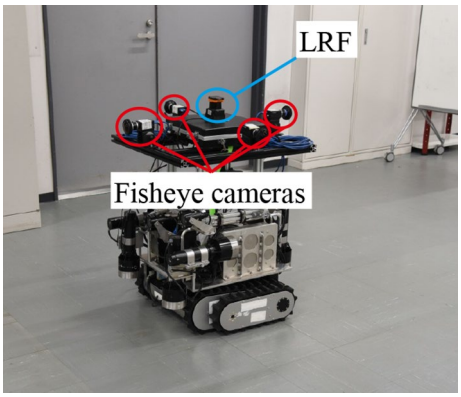


Fig. 8 Experimental setting. Four fisheye cameras and one LRF are mounted on the robot

Real-time visualization

Acquiring and processing the data in real-time are required for real-time visualization. Therefore, one thread per sensor is launched for real-time data acquisition.

Processing multiple high-resolution images in real-time requires high computation resources. Therefore, the whole processes including the undistortion of the fisheye images, the projection, and the blending are implemented by GLSL so that the GPU parallel processing is utilized. All the relative poses, the 3D models, and the intrinsic parameters of the fisheye cameras are loaded into the GPU at the beginning, and only the fisheye camera images and the 3D model of the walls are updated during operations.

Experiments

Experiments were conducted in the basement floor of Engineering Building No. 14 in the University of Tokyo.

Settings

Four fisheye cameras and one LRF were mounted on a robot as shown in Fig. 8. Four fisheye cameras were used because the robot was approximated as a cuboid. Assumed that the field of view (FoV) of fisheye cameras is more than 180 degrees, attaching a fisheye camera on each face of the cuboid except the top and bottom faces is enough to capture the surrounding environments. We used Grasshopper3 GS3-U3-41C6C-C color cameras, Fujinon FE185C086HA-1 fisheye lenses with the FoV of 185 degrees, and a Hokuyo UTM-30LX LRF. All the computations were performed using a laptop (Vaio Z), with a 2-core CPU @3.3GHz (Intel Core i7-6567U), a 16GB RAM, and an integrated GPU (Intel Iris Graphics 550). A wireless HDMI adapter was plugged into the laptop for sending the generated free viewpoint images to the operator, and a wireless mouse was used to change the

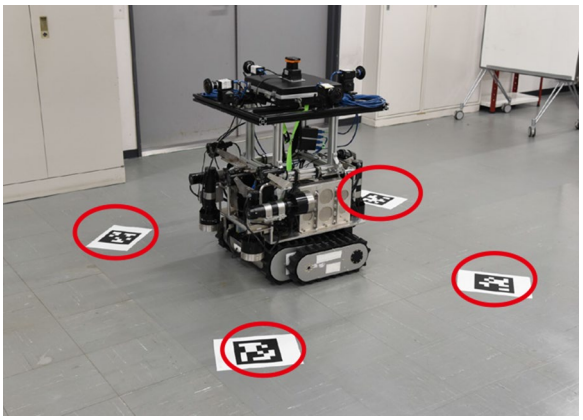


Fig. 9 Four AprilTags [16, 17] circled in red are placed on the floor for the calibration

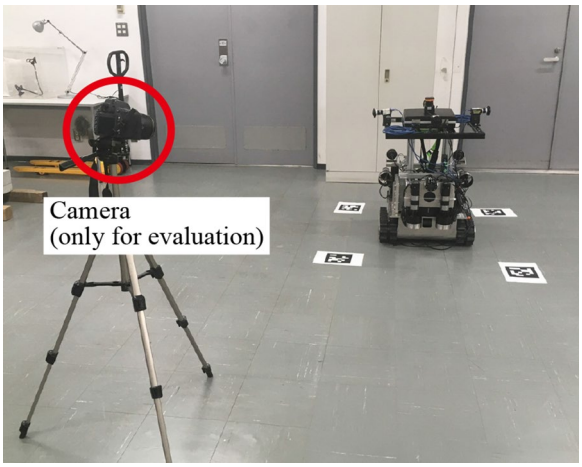


Fig. 10 Example of scenes when images of the robot are taken from outside by using a reference camera (only for evaluation)

viewpoint of the free viewpoint images. The height of the walls, h_{wall}, was set to 3.0 m according to the scene, while the blending range of textures, θ_{th}, was set to 15 degrees.

The image resolution was set at 1600×1600 after cropping the field of view regions. As the resolution of the images is high, the cropping was applied by the hardware setting of the camera to make the most out of the limited bandwidth of the USB 3.0 hub in the laptop.

Calibration

After installing the cameras and an LRF, the relative poses of the floor, the cameras, the robot model, and the LRF were estimated using the proposed method. We placed four AprilTags on the floor as shown in Fig. 9 and took images using the cameras. All of the calibration procedures were automatic except for selecting the four points corresponding to the locations of the cameras in the robot model, two or three corner points in the measurement points of the LRF, and their corresponding points in the bird's-eye view image. Details are presented in our supplemental video at our project page.[1]

Evaluation for free viewpoint images

To evaluate the generated free viewpoint images, we employed an actual camera (reference camera) to take images of the robot from outside, as shown in Fig. 10. We took ten images of the robot from various directions, and Fig. 10 depicts an example of the scenes where the reference camera was placed at the back right of the robot. Every time an image of the robot was taken using the reference camera, we estimated the pose of the reference camera. Then, given the pose and the intrinsic parameters of the reference camera, the proposed method generated a virtual image from the same viewpoint using only the mounted cameras and an LRF.

We used the structural similarity (SSIM) [21] index and peak signal-to-noise ratio (PSNR) as evaluation metrics to compare the free viewpoint images generated by the proposed method to the images taken by the reference camera. SSIM handles appearance similarity in terms of human visual perception, whereas PSNR handles the pixel value difference directly. Therefore, it is worth using both SSIM index and PSNR as evaluation metrics. We masked out the robot-body region in the images before evaluations as we found out that the robot-body region has worse evaluation metrics, and the quality of the robot model is beyond the scope of this study.

Results

Free viewpoint images generation

The generated free viewpoint images from various viewpoints are shown in Fig. 11. In this figure, the images generated by the previous method [7], images taken from outside of the robot, and images generated by the proposed method are presented from the first row to the last row of the figure, respectively. The values written below the first and last rows in Fig. 11 are the SSIM index and the PSNR calculated using the images of the second row as reference images.

As observed in the first row of Fig. 11, free viewpoint images generated by the previous method [7] are distorted. This is because the previous method approximated the surrounding environment as a semi-spherical mesh model, but the actual environment did not verify the assumptions. On the other hand, the proposed method (the last row of Fig. 11) is capable of generating

[1] https://matsuren.github.io/fvp.

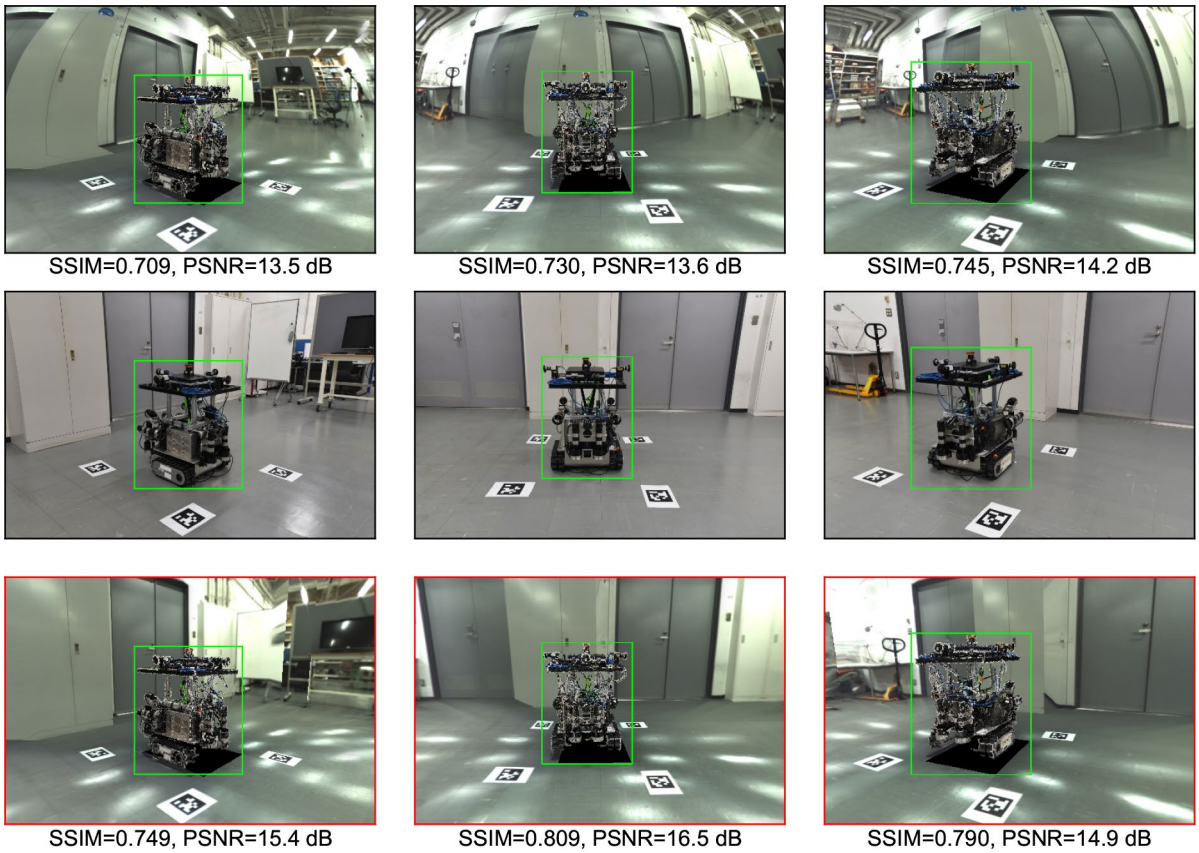


Fig. 11 Free viewpoint images from various viewpoints. From the first row to the last row: images generated by the previous method [7], images taken from outside of the robot, and images generated by the proposed method. The values written below the first and last rows are the SSIM index and the PSNR, which were calculated using images of the second row as reference images. The green rectangle in each image represents the robot-body region, which was masked out before evaluation

free viewpoint images as if they were captured from outside of the robot.

A quantitative evaluation of the appearance of the generated free viewpoint images was also performed (Table 1). Ten images were taken by the reference camera at various locations, and the mean values of the evaluation metrics of the proposed method and the previous method [7] are listed in Table 1. This table indicates that the proposed method achieved higher values than the previous method.

Table 1 Quantitative evaluation of the appearance of the generated free viewpoint images

Method	SSIM	PSNR (dB)
Previous [7]	0.722	13.7
Proposed	*0.769*	*15.4*

N = 10

The mean value of SSIM and PSNR are used as evaluation metrics (higher is better)

Real-time visualization

The frame rate of the fisheye camera was set to 25 Hz due to the limited bandwidth of the USB 3.0 hub in the laptop. Therefore, the whole procedure of data loading into GPU, processing, and visualizing should be performed in less than 40 ms for real-time visualization. Table 2 shows the time consumed when the proposed method generates a free viewpoint image with a resolution of 1920 × 1080 (full-screen mode). In Table 2,

Table 2 Time consumed by the proposed method to generate a free viewpoint image with a resolution of 1920 × 1080 in a laptop (Vaio Z)

Phase	Time
Load data into GPU	6.0 ms
Process and visualize	4.2 ms
Total	10.2 ms

Fig. 12 Free viewpoint image generation in a corridor. **a** Generated by the previous method [7]. **b** Generated by the proposed method. The previous method generated distorted free viewpoint images, which makes it more difficult for operators to perceive the distance between the robot and the walls. On the other hand, the distance between the robot and the walls is obvious in images generated by the proposed method

"Load data into GPU" indicates the phase for loading four fisheye camera images (1600 × 1600) and one scan data of the LRF, and "Process and visualize" indicates the process done by GLSL. As can be observed in Table 2, the proposed method achieved real-time visualization even with an integrated GPU of a normal laptop.

Exploration of a corridor

Another experiment for exploring an environment using a teleoperated robot was conducted in a corridor. In this experiment, we controlled the robot remotely by viewing the generated free viewpoint images. Figure 12 a, b show the free viewpoint images generated by the previous method [7] and the proposed method, respectively. It should be noted that the same robot location and viewpoint were used to generate both free viewpoint images in Fig. 12 a, b. As observed in Fig. 12 a, it is difficult to perceive the distance between the robot and the walls when using the generated free viewpoint images because of distortion, which may cause collisions. On the other hand, the proposed method generates less distorted free viewpoint images, which help operators perceive the distance between the robot and the walls easily. The exploration of the corridor is provided in our supplemental video at our project page.[2]

Conclusion and future works

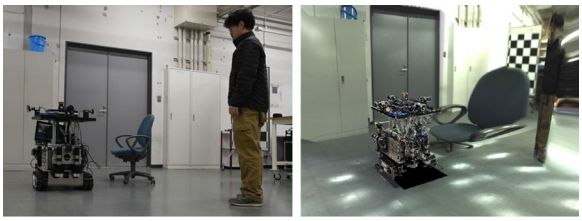
In this paper, we proposed a novel free viewpoint image generation system for indoor robot teleoperation. Multiple fisheye cameras and an LRF were installed on a robot, and free viewpoint images were generated under the assumption that the indoor environment consisted of three elements: one robot, one floor, and walls that are perpendicular to the floor. Furthermore, an easy calibration for estimating the poses of the multiple fisheye cameras, the LRF, and the robot model was proposed.

The experimental results showed that the proposed method is capable of generating free viewpoint images as if they were captured from outside of the robot, and the system visualizes the surrounding environment of the robot in real-time owing to the implementation by GLSL even with an integrated GPU of a normal laptop.

We focused specifically on building a system that generates free viewpoint images in this study, however, evaluating the effectiveness of our proposed method in terms of usability would be also interesting to know. Therefore, we leave the usability evaluation for future works.

Our proposed method only works under the assumption that the indoor environment consisted of three elements: one robot, one floor, and walls that are perpendicular to the floor. There are two different things occur when other objects exist in the environment as can be observed in Fig. 13 a, b. In one case their heights are shorter than the LRF scan lines, the objects are projected onto the floor. In another case their heights are taller than the LRF scan lines, the objects are estimated as walls. Thus, the proposed method generates wrong free viewpoint images. Therefore, as future works, we intend to employ a semantic segmentation to the images to distinguish between the objects under assumption (robots, floors, and walls) and other objects. We intend to use a depth reconstruction by multi-view stereo to estimate the 3D models for other objects.

[2] https://matsuren.github.io/fvp.



a b

Fig. 13 Example of scenes where the assumption of this study is not valid. A chair and a human exist other than a robot, a floor, and walls in this environment. The height of the LRF scan lines is taller than that of the chair but shorter than that of the human. **a** Taken from an outside camera. **b** Generated free viewpoint image by the proposed method. The chair is projected onto the floor, and the human is estimated as walls

Acknowledgements
The authors would like to thank the members of the Intelligent Construction Systems Laboratory, The University of Tokyo for their useful suggestions, especially Mr. Shingo Yamamoto and Mr. Takumi Chiba from Fujita Corporation and Dr. Kazuhiro Chayama from KOKANKYO Engineering Corporation.

Authors' contributions
RK conceived the presented idea and carried out the experiment with help from HF. RK wrote the manuscript in consultation with HF, YT, AY, and HA. All the authors read and approved the final manuscript.

Funding
A part of this study is financially supported by the Nuclear Energy Science & Technology and Human Resource Development Project (through concentrating wisdom) from the Japan Atomic Energy Agency / Collaborative Laboratories for Advanced Decommissioning Science.

Availability of data and materials
Our source code is available at our project page (https://matsuren.github.io/fvp). The data that we used to calculate the SSIM index is available from the corresponding author, RK, upon request.

Competing interests
The authors declare that they have no competing interests.

Author details
[1] Graduate School of Engineering, The University of Tokyo, Tokyo 113-8656, Japan. [2] Faculty of Advanced Engineering, Chiba Institute of Technology, Chiba 275-0016, Japan.

Received: 24 January 2020 Accepted: 28 February 2020
Published online: 12 March 2020

References

1. Yanco HA, Drury JL, Scholtz J (2004) Beyond usability evaluation: analysis of human-robot interaction at a major robotics competition. Hum Comput Interact 19(1):117–149
2. Ferland F, Pomerleau F, Le Dinh C, Michaud F (2009) Egocentric and exocentric teleoperation interface using real-time, 3D video projection. In: Proceedings of the 4th ACM/IEEE international conference on human-robot interaction: 11–13 March 2009. California, USA. IEEE, New York, pp 37–44
3. Nielsen CW, Goodrich MA, Ricks RW (2007) Ecological interfaces for improving mobile robot teleoperation. IEEE Trans Robot 23(5):927–941
4. Keyes B, Casey R, Yanco H, Maxwell B, Georglev Y (2006) Camera placement and multi-camera fusion for remote robot operation. In: Proceedings of the IEEE International Workshop on safety, security and rescue robotics. Gaithersburg, USA, pp 22–24
5. Sato T, Moro A, Sugahara A, Tasaki T, Yamashita A, Asama H (2013) Spatio-temporal bird's-eye view images using multiple fish-eye cameras. In: Proceedings of the 2013 IEEE/SICE international symposium on system integration: 15–17 Dec 2013, Kobe, Japan. IEEE, New York, pp 753–758
6. Awashima Y, Komatsu R, Fujii H, Tamura Y, Yamashita A, Asama H (2017) Visualization of obstacles on bird's-eye view using depth sensor for remote controlled robot. In: Proceedings of the 2017iInternational workshop on advanced image technology: 6–8 Jan, 2017. Penang, Malaysia
7. Sun W, Iwataki S, Komatsu R, Fujii H, Yamashita A, Asama H (2016) Simultaneous tele-visualization of construction machine and environment using body mounted cameras. In: Proceedings of the 2016 IEEE international conference on robotics and biomimetics: 3–7 Dec 2016. Qingdao, China. IEEE, New York, pp 382–387
8. Wu C (2011) VisualSFM: a visual structure from motion system. http://ccwu.me/vsfm
9. Furukawa Y, Ponce J (2010) Accurate, dense, and robust multi-view stereopsis. IEEE Trans Pattern Ana Mach Intell 32(8):1362–1376
10. Schonberger J L, Frahm J M (2016) Structure-from-motion revisited. In: Proceedings of the IEEE Conference on computer vision and pattern recognition: 26 Jun–1 Jul 2016. Las Vegas, USA. IEEE, New York, pp 4104–4113
11. Schonberger J L, Zheng E, Frahm J M, Pollefeys M (2016) Pixelwise view selection for unstructured multi-view stereo. In: Proceedings of the 14th European conference on computer vision: 8–16 Oct 2016, Amsterdam, The Netherlands, pp 501–518
12. Coughlan JM, Yuille AL (2001) The Manhattan world assumption: regularities in scene statistics which enable Bayesian inference. Advances in neural information processing systems. MIT press, Cambridge, pp 845–851
13. Schindler G, Dellaert F (2004) Atlanta world: an expectation maximization framework for simultaneous low-level edge grouping and camera calibration in complex man-made environments. In: Proceedings of the 2004 IEEE computer society conference on computer vision and pattern recognition: 27 June–2 July 2004. IEEE, Washington, DC, USA., New York, pp 203–209
14. Scaramuzza D, Martinelli A, Siegwart R (2006) A flexible technique for accurate omnidirectional camera calibration and structure from motion. In: Proceedings of IEEE international conference of computer vision systems: 4–7 Jan 2006, IEEE, New York, USA, pp 45–52
15. Scaramuzza D, Martinelli A, Siegwart R (2006) A toolbox for easily calibrating omnidirectional cameras. In: Proceedings of the 2006 IEEE/RSJ international conference on intelligent robots and systems: 9–15 Oct 2006. Beijing, China. IEEE, New York, pp 5695–5701
16. Olson E (2011) Apriltag: a robust and flexible visual fiducial system. In: Proceedings of the 2011 IEEE International conference on robotics and automation: 9–13 May 2011, Shanghai, China. IEEE, New York, pp 3400–3407
17. Wang J, Olson E (2016) Apriltag 2: efficient and robust fiducial detection. In: Proceedings of the 2016 IEEE/RSJ international conference on intelligent robots and systems: 9–14 Oct 2016, Daejeon, Korea. IEEE, New York, pp 4193–4198
18. Agarwal S, Mierle K (2012) Ceres solver. http://ceres-solver.org
19. Collins T, Bartoli A (2014) Infinitesimal plane-based pose estimation. Int J Comput Vis 109(3):252–286
20. Umeyama S (1991) Least-squares estimation of transformation parameters between two point patterns. IEEE Trans Pattern Anal Mach Intell 13(4):376–380
21. Wang Z, Bovik AC, Sheikh HR, Simoncelli EP (2004) Image quality assessment: from error visibility to structural similarity. IEEE Trans Image Process 13(4):600–612

Publisher's Note