Spherical Camera Localization by Color Difference Minimization Using 3D Model of the Environment

1st Dongxu Yang[1]
yangdongxu@robot.t.u-tokyo.ac.jp

2nd Hiroshi Higuchi[1]
higuchi@robot.t.u-tokyo.ac.jp

3rd Sarthak Pathak[1]
pathak@robot.t.u-tokyo.ac.jp

4th Alessandro Moro[1]
moro@robot.t.u-tokyo.ac.jp

5th Atsushi Yamashita[1]
yamashita@robot.t.u-tokyo.ac.jp

6th Hajime Asama[1]
asama@robot.t.u-tokyo.ac.jp

***Abstract*—The goal of this study is to propose an occlusion-robust spherical camera localization method using 3D model of the environment. Instead of using specific features, which may cause low estimation accuracy in environment with occlusion, the color information of all pixels in images is used for the estimation. Camera pose is estimated by minimizing the color difference of the taken image and the image generated using former frame and 3D model. To reduce the influence caused by occlusion, a robust evaluation function is used in the estimation process. The effectiveness of the proposed method was confirmed by experimental tests.**

***Index Terms*—localization, spherical camera, 3D model of the environment**

I. Introduction

Recently, drones are more commonly used for various purposes such as the inspection of infrastructures and factories. Instead of people, drones can provide comparatively safer and economical performance. During the inspection, it is necessary to know the pose of the drone correctly. Although localization methods using Global Positioning System (GPS) can work efficiently outdoors [1], when it comes to indoor environments such as factories, GPS has limitations because GPS signals does not work indoors. Many studies are conducted on indoor mobile robots localization using cameras. Cameras are widely used recently because it is flexible and affordable, and is able to provide various kind of information such as color information and feature information.

In this study, 3D model of the environment is used to obtain 3D information in order to complete camera localization. In some related research, depth cameras are used to get 3D information. For example, [2] used an RGB-D camera for real-time 6 DoF (3 DoF translation and 3 DoF rotation) localization by using both depth images and color images. [3] used a depth camera for localization, while using Random Sample Consensus (RANSAC) and 3D point matching. However, RGB-D cameras have a small field of view, so they obtain less information. Cameras with a wider field of view are more effective when used for localization purpose, because the common field of view between images taken before and after camera motion is important. When the rotation of the mobile robot is extremely large, images captured by cameras with narrow vision can be totally different and the localization will fail in this case. Since more information can be obtained by using cameras with a wider field of view, the common field of view between frames is larger. Thus, cameras with a wider range of view are more effective for localization.

In this study, a spherical camera, which has 360 degree field of view, and 3D model of the environment is used. Such 3D model can be easily obtained in advance from the construction CAD models of the environment, or by doing a 3D scan using a Laser range finder. By using the spherical image and depth information from the 3D model, virtual image of a specific view point can be generated. By comparing the virtual image and the image captured after camera motion, the camera pose after the camera motion can be estimated. In indoor environments with occlusion, non-visible region occurs after the camera motion. To reduce the influence brought by occluded regions, a method using the difference of the RGB information of each pixel in the image obtained from the spherical camera is proposed. Instead of using a specific point or feature, using the difference of the RGB information of all pixels in an image is expected to be more robust with occlusion problem. Moreover, the weighting of the difference of the RGB information is adjusted to reduce the influence of occluded regions.

II. Proposed Method

A. Problem Setting

In this paper, a localization method using images obtained from spherical camera and 3D model of the environment is proposed. The initial pose of the camera is assumed to be known. The image taken at the known pose is called key image. Using the known pose, a depth map of the known pose can be generated from the 3D model. Each pixel in the depth map is in correspondence with the pixel in the key image. The image taken by the next moment is set to be the target image whose pose is wanted to be estimated. The motion of the camera is assumed to be small, which means the difference of the key image and the target image should be small.

B. Approach

The overview of the proposed method is shown in Fig. 1. Let the image taken at moment τ be the key image, and the

[1] The University of Tokyo, Tokyo, Japan

978-1-7281-6224-9/20/$31.00

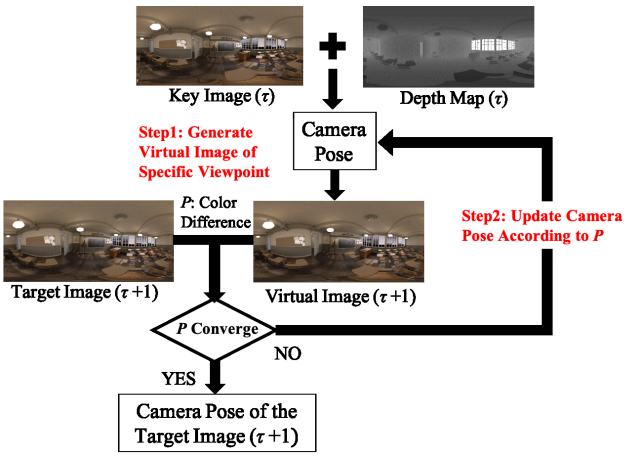


Fig. 1: Overview of proposed method

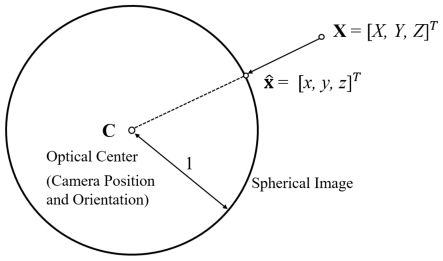


Fig. 2: Spherical camera projection model

image taken at moment $\tau+1$ be the target image. The camera pose of the key image is set to be known because the initial camera pose is known. By using the pose of the key image and the 3D model, a depth map corresponding to the key image can be generated. The purpose of the proposed method is to estimate the camera pose of the target image.

The smaller the color difference of two images, the more similar these two images are, which also means that these two images have high probability to be taken from the same viewpoint. With the known 3D model and the color information from the key image, a virtual image at any camera pose can be generated. Thus, by generating a virtual image of a specific viewpoint and minimizing the color difference between the virtual image and the target image, the pose of the target image can be estimated.

C. Generating Virtual Image of Specific Viewpoint

The projection surface of a spherical camera is a unit sphere whose center is the optical center $\mathbf{c}$ of the camera. A spherical image is an image formed on the spherical camera projection surface, which contains unit vector pixels $\hat{\mathbf{x}} = [x, y, z]^T$. As shown in Fig. 2, the color information of a 3D environment point $\mathbf{X} = [X, Y, Z]^T$ is projected to the unit sphere as a unit vector [4].

The process of generating a virtual image of a specific viewpoint is shown in Fig. 3. First, the key image is changed into a spherical image. For example, the yellow pixel shown

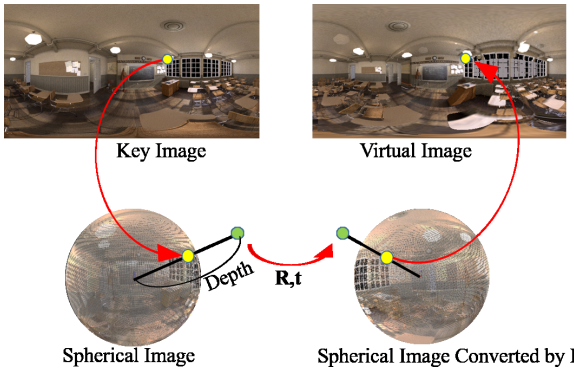


Fig. 3: Generating virtual image of specific viewpoint

in the key image is converted to the corresponding position on the unit sphere. Next, the pixel is converted to the 3D position shown in green by changing the distance between the pixel and the center of the unit sphere to the depth value, which is known from the corresponding depth map. Then, this pixel is rotated and translated by employing the rotation matrix $\mathbf{R}$ and translation vector $\mathbf{t}$, to the wanted position. Then, the pixel is projected to the unit sphere, so that a new spherical image after changing the camera viewpoint is generated. Finally, the spherical image after transformation is changed into an equirectangular image which is the wanted virtual image. A color interpolation (bicubic interpolation) is done to fill the missing pixels after the transformation.

The detail of changing the camera viewpoint of the spherical image is shown in Fig. 4. The purpose is to use the key image of a known viewpoint (camera viewpoint 1) and the corresponding depth map to generate a virtual image of camera viewpoint 2, which is an arbitrary viewpoint. First, by changing the distance between each pixel and the optical center into the depth obtained by using the depth map, a 3D point cloud can be generated. Next, when the camera moves from viewpoint 1 to viewpoint 2, the relative position between all pixels in the 3D point cloud and viewpoint 2 can be calculated. By normalizing the depth between all pixels and viewpoint 2, a spherical image of viewpoint 2 is generated.

In this way, a virtual image of an arbitrary specific viewpoint can be generated with an image and the corresponding depth map.

D. Occlusion Problem

While generating a virtual image, a problem caused by occlusion in the environment may occur.

As shown in Fig. 5, when the camera moves from viewpoint 1 to viewpoint 2, the occluded area that was not visible from viewpoint 1 becomes visible from viewpoint 2. The color information of the occluded area is not included in the key image because it was occluded when the camera was at viewpoint 1. Therefore, the color information of the occluded area can not be correctly estimated when generating a virtual image of viewpoint 2.

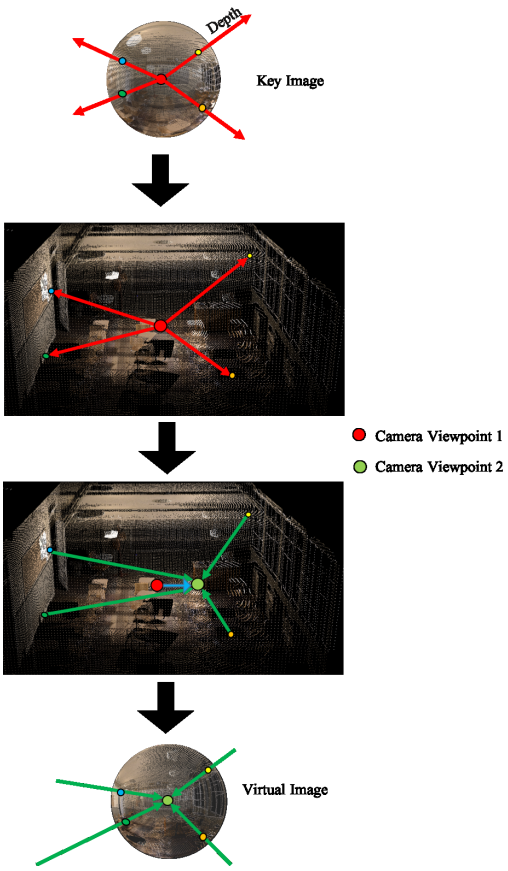


Fig. 4: Generating virtual image of specific viewpoint

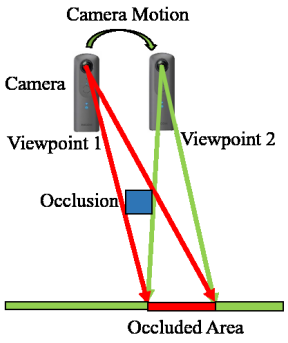


Fig. 5: Occlusion problem

An example of occlusion problem during the virtual image generation is shown in Fig. 6. The color of the area behind the desk can not be correctly estimated because of the occlusion.

The occlusion problem affects the generated virtual image, and also may cause error in the pose estimation process. The weighting of the difference of the RGB information is adjusted to lower the influence brought by occlusion, which is discussed later in section II-F.

Fig. 6: Example of the occlusion problem

E. Camera Pose Estimation Using Color Difference Minimization

The color information of all pixels in an image is used for estimation. Values of R, G, B channels are all used. For the ease of handling, the image matrix was converted to a one-dimensional vector. The evaluation function is the color difference between the virtual image and the target image which is defined as follows:

$$\hat{\mathbf{R}}, \hat{\mathbf{t}} = \arg\min_{\mathbf{R},\mathbf{t}}(P(\mathbf{a})), \tag{1}$$

where $\mathbf{a}$ is the color difference defined as:

$$\mathbf{a} = \mathbf{I}_V(\mathbf{R}, \mathbf{t}) - \mathbf{I}_T. \tag{2}$$

Here, $\mathbf{R}$ is the rotation matrix and $\mathbf{t}$ is the translation vector. $\hat{\mathbf{R}}, \hat{\mathbf{t}}$ are the estimated rotation and translation which makes the color difference the smallest. $\mathbf{I}_V(\mathbf{R}, \mathbf{t})$ is the image vector of the virtual image of a viewpoint which is converted by $\mathbf{R}, \mathbf{t}$. $\mathbf{I}_T$ is the image vector of the target image. P is the evaluation function of the optimization.

With the estimated $\hat{\mathbf{R}}, \hat{\mathbf{t}}$ and the camera pose of the key image, the camera pose of the target image can be calculated.

The evaluation function P used in the optimization is the color difference of the virtual image and the target image which is defined as follows:

$$P(\mathbf{a}) = \sum_{i=1}^{n} \frac{1}{2} a_i^2, \tag{3}$$

where n is the number of components in the image vector. a_i is the i-th component of $\mathbf{a}$.

In this study, the evaluation function is further modified for better performance by changing the weighting of pixels in the image vector. The weighting of each component of the image vector is adjusted to reduce the error caused by the occlusion problem. In this study, the Levenberg-Marquardt method is used for the optimization [5].

F. Weighting of The Evaluation Function

Occlusion problem occurs when generating a virtual image as shown in Figs. 5 and 6. The color information of the occluded area in the image is not known in the key image. Thus, when generating the virtual image, the color of the occluded area can not be correctly estimated. When comparing the virtual image and the target image, the color difference of pixels in the occluded area tends to be higher. To reduce

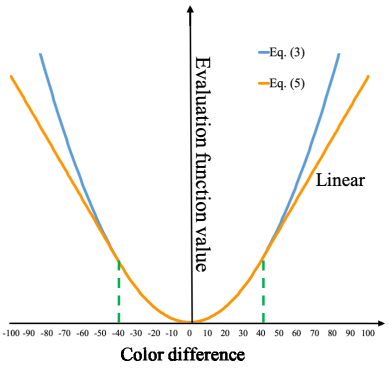


Fig. 7: Evaluation function

Fig. 8: 3D model of a classroom

the influence of the occluded area, the evaluation function P is changed into a Huber loss function [6]. The Huber loss function is defined as follows:

$$P_i(a_i) = \begin{cases} \frac{1}{2}a_i^2 & (|a_i| \leq \delta) \\ \delta(|a_i| - \frac{1}{2}\delta) & (\text{otherwise}) \end{cases}, \tag{4}$$

$$P(\mathbf{a}) = \sum_{i=1}^{n} P_i(a_i), \tag{5}$$

where a_i is the i-th component of $\mathbf{a}$. n is the number of components in the image vector. δ is the threshold. When the error is too high in a region, the probability of occlusion is high. Hence, an error higher than the threshold indicates the presence of occlusion. In this study, the threshold is set to 40 due to the experiment. In different environment, the threshold should be determined according to the amount of occlusion, the lighting conditions, etc.

The evaluation function before and after modification are both shown in Fig. 7. The blue line is the function before changing the weighting (Eq. (3)), and the yellow line is the Huber loss function (Eq. (5)). When the color difference is higher than the threshold, a linear function is used.

III. Experiment and Result

The proposed method was first tested in a simulation environment, and then tested in a real environment.

A. Simulation Experiment

In the simulation experiment, Blender [7], an image rendering software, was used to generate images and depth maps. A 3D model of a classroom is used (Fig. 8). An example of the generated image and depth map is shown in Fig. 9. The

(a) Image (b) Depth map

Fig. 9: Image and depth map generated by Blender

Environment without occlusion Environment with occlusion

Fig. 10: Simulation environment

ground truth of the camera pose is known, thus the result of the proposed method is precisely evaluated.

In order to test the robustness of the proposed method against occlusion, experiments were conducted in two different environments. A classroom without desks and chairs and a classroom with desks and chairs are used because desks and chairs cause occlusion (Fig. 10).

Since a continuous localization method is considered, the camera motion is set to be small. Here, the maximum translation between camera frames is set to ± 0.25 m, and the maximum rotation between camera frames is set to ± 2.5 deg. Images taken from 50 randomly generated camera poses within this range are used in the experiment.

For comparison, proposed method 1 is an experiment using the evaluation function without considering occlusion (Eq. 3) is also conducted. The proposed method 2 uses the evaluation function considering occlusion (Eq. 5). As shown in Table I, four groups of experiments are conducted.

In this experiment, there were some cases when the optimization ended with local minimum and poses were not correctly estimated. Such cases are considered failed cases. Data of failed cases is removed using the Interquartile Range (IQR) method [8]. The average of the data without failed cases and the standard error are shown in Figs. 11 and 12. The success rate in each experiment is shown in Table II.

TABLE I: Groups of simulation experiment

Experiment	Environment	Evaluation function
(1) Proposed method 1 (without occlusion)	Classroom without occlusion	Eq. (3)
(2) Proposed method 2 (without occlusion)	Classroom without occlusion	Eq. (5)
(3) Proposed method 1 (with occlusion)	Classroom with occlusion	Eq. (3)
(4) Proposed method 2 (with occlusion)	Classroom with occlusion	Eq. (5)

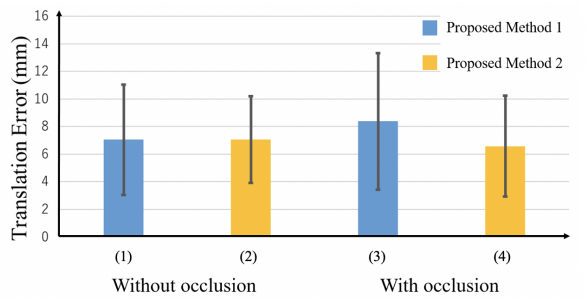


Fig. 11: Translation error

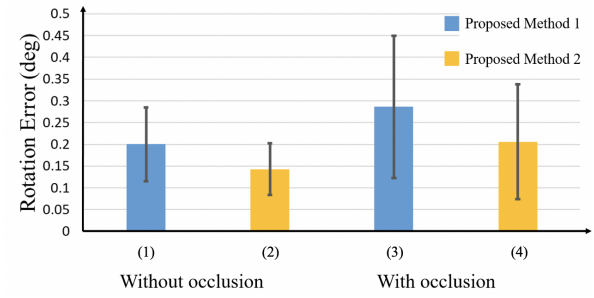


Fig. 12: Rotation error

According to the results of group (1) and group (3) shown in Figs. 11 and 12, it shows that when occlusion occurs in the environment, both translation error and rotation error increased. The success rate of the estimation also decreased (Table II).

From the results of group (3) and group (4), it is clear that the translation error and the rotation error were both reduced by using the proposed method 2 even in an environment with occlusion.

From the results of group (1) and group (2), it can be said that the proposed method 2 also reduced the error and raised the success rate of the estimation in an environment with little occlusion.

TABLE II: Success rate of simulation experiments

Experiment	Success rate [%]
(1) Proposed method 1 (without occlusion)	88.0
(2) Proposed method 2 (without occlusion)	100.0
(3) Proposed method 1 (with occlusion)	82.0
(4) Proposed method 2 (with occlusion)	90.0

TABLE III: Estimation error of real enviroment experiment

	Translation error [mm]	Rotation error [deg]
Average	10.1	0.4
Standard deviation	4.3	0.2

Fig. 13: Experiment environment

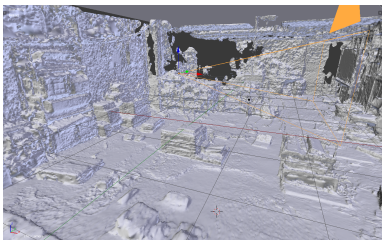


Fig. 14: 3D mesh

B. Real Environment Experiment

This experiment was conducted in a room with a scale of 17 m×9.5 m×3.8 m (Fig. 13).

First, a 3D model of the room was obtained before the experiment. Structure from motion (SfM) method [9] was used to do the 3D reconstruction of the room. An application called PhotoScan was used to complete the 3D model. Nikon D750 was used to take images of the reconstruction. About 1000 images were taken to complete the 3D model of the room. The generated 3D mesh is shown in Fig. 14. By using the 3D mesh, a depth map was generated by Blender (Fig. 15).

A spherical camera (RICOH THETA V) was used to capture images for the estimation. Images taken from 30 camera poses were used. To obtain the ground truth of the camera pose, motion capture (OptiTrack V120) was used. An 'L' shape frame with markers was used.

In this experiment, no failed case occurred. The average and the standard deviation of the estimation error of all data

Fig. 15: Depth map of real environment

TABLE IV: Maximum value of translation and rotation

	Translation [mm]			Rotation [deg]		
Axis	x	y	z	x	y	z
Maximum value	164.4	295.6	14.6	5.0	1.6	3.2

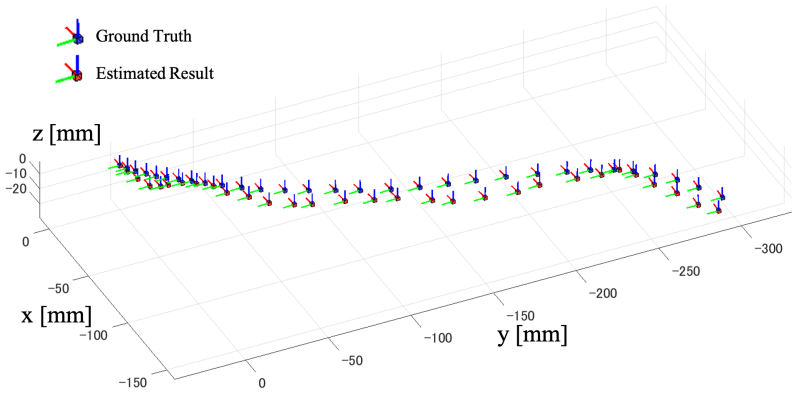
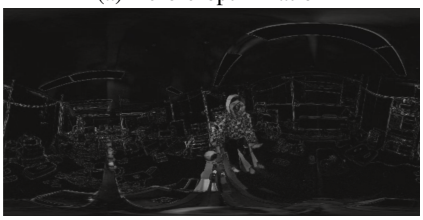


Fig. 16: Pose of the ground truth and the estimated result

(a) Before optimization



(b) After optimization

Fig. 17: Color difference error image of the target image and the generated virtual image

is shown in Table III. In the real environment experiment, the maximum range of translation and rotation was not determined. The maximum value of translation and rotation of the 30 camera poses is shown in Table IV. The pose of the ground truth and the estimated result is shown in Fig. 16. For comparison, the different image of the target image and the virtual image is shown in Fig. 17.

According to Table III, the translation and rotation error is almost at the same level as the result of the simulation experiment.

From the comparison of the target image and the generated virtual image (Fig. 17), it is clear that the target image and the virtual image matches after the optimization. Therefore, it can be said that the estimation is correct.

From the result of this experiment, it is clarified that the proposed method works efficiently in real environments.

IV. Conclusion

In this study, an occlusion-robust spherical camera localization method using 3D model of the environment is proposed. The proposed method was tested in both simulation environment and real environment. The estimation error of the real environment experiment was at the same level as the simulation experiment. Therefore, it is confirmed that camera pose estimation is possible by the proposed method even in an environment with occlusion.

Future work of this study will include: estimation of the initial camera pose, continuous camera pose estimation by regenerating depth map using 3D model, increasing the maximum rotation range. By working on future work listed above, it can be expected that the proposed method will be applicable in the future.

Acknowledgment

The authors would like to thank Pocket Queries Corporation for technical assistance and suggestions based on professional experience.

References

[1] S. H. Kim , C. W. Roh , S. C. Kang and M. Y. Park, “Outdoor Navigation of a Mobile Robot Using Differential GPS and Curb Detection,” Proceedings of the 2007 IEEE International Conference on Robotics and Automation, pp. 3414-3419, 2007.

[2] Z. Fang and S. Scherer, “Real-time Onboard 6DoF Localization of an Indoor MAV in Degraded Visual Environments Using a RGB-D Camera,” Proceedings of the 2015 IEEE International Conference on Robotics and Automation, pp. 5253-5259, 2015.

[3] H. Du, P. Henry, X. Ren, M. Cheng, D. B. Goldman, S. M. Seitz and D. Fox, “Interactive 3D Modeling of Indoor Environments with a Consumer Depth Camera,” Proceedings of the 13th International Conference on Ubiquitous Computing, pp. 75-84, 2011.

[4] J. Courbon, Y. Mezouar, L. Eckt and P. Martinet, “A Generic Fisheye Camera Model for Robotic Applications,” Proceedings of the 2007 IEEE/RSJ International Conference on Intelligent Robots and Systems, pp.1683-1688, 2007.

[5] J. J. More,” The Levenberg-Marquardt Algorithm: Implementation and Theory,” Numerical Analysis, Springer-Verlag, Vol. 630, pp 105-116, 1977.

[6] P. J. Huber, “Robust Estimation of a Location Parameter,” Annals of Mathematical Statistics, Vol. 35, No. 1, pp. 73-101, 1964.

[7] “Blender,” https://www.blender.org/ (access date 2020.01.21).

[8] G. Barbato, E. M. Barini, G. Genta and R. Levi, “Features and Performance of Some Outlier Detection Methods,” Journal of Applied Statistics, Vol. 38, No. 10, pp. 2133-2149, 2011.

[9] M. J. Westoby, J. Brasington, N. F. Glasser, M. J. Hambrey and J. M. Reynolds, “‘Structure-from-Motion ’ ’Photogrammetry: A Low-cost, Effective Tool for Geoscience Applications,” Geomorphology, Vol. 179, pp. 300-314, 2012.