

Camera Orientation Estimation in Leaking Indoor Environment via Vanishing Point of Water Drops

Hao Xu, Ren Komatsu, Hanwool Woo, Angela Faragasso, Atsushi Yamashita and Hajime Asama

Abstract—In this paper, we propose a novel camera orientation estimation method based on the computation of the vanishing point of water drops in leaking indoor environment. Camera orientation estimation is an important component of robots as it allows them to perform complex tasks such as three-dimensional (3D) reconstruction of different environments. Camera estimation usually involves sensors, such as cameras or encoders and sophisticated processing algorithms. In recent years, computer vision techniques have been widely used to estimate the camera orientation in robotics-related research as visual sensing can improve the autonomy of the systems. Although most of these methods perform well in outdoor environments, they are problematic in the environments of indoor disasters, where common visual features may be missing due to collapse and erosion. To solve these problems, we developed a novel technique that employs particular characteristics of leaking indoor environment. Our method uses the vanishing point generated from the trajectories of water drops, to estimate the rotation of the camera. The proposed technique can potentially be applied for inspecting nuclear power plants. Computer-simulated and real data experiments have been performed to evaluate the accuracy of the proposed method. The results of these experiments demonstrate that our method can detect the vanishing point of water drops and estimate the rotation angle accurately.

I. INTRODUCTION

Robots are currently used in a wide variety of industries, from daily living assistance to space exploration. Robotic provide safety benefits since they can perform tasks in environments that would be too dangerous or small or narrow for humans to access. For instance, robots can work in the presence of radiation or in extreme temperatures and pressure. Robots have been used in nuclear plant decommissioning, reactor stabilization, inspection, and clean up activities [1]. They can be equipped with sensors and operated remotely to perform inspection tasks. Figure 1 shows a multi-sensing inspection robot that is equipped with a radiation detector, camera, and thermometer that can be used for exploration of nuclear power plants. In [2] a similar remote-controlled mobile robot that carried an inspection system that was able to rotate around the pan and tilt axes was used to investigate the primary containment vessel (PCV) of the Fukushima Nuclear Power Plant.

*A part of this work 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.

H. Xu, R. Komatsu, H. Woo, A. Faragasso, A. Yamashita and H. Asama are with the Department of Precision Engineering, School of Engineering, The University of Tokyo, 7-3-1 Hongo, Bunkyo-ku, Tokyo, Japan. {xuhao, komatsu, woo, faragasso, yamashita, asama}@robot.t.u-tokyo.ac.jp.

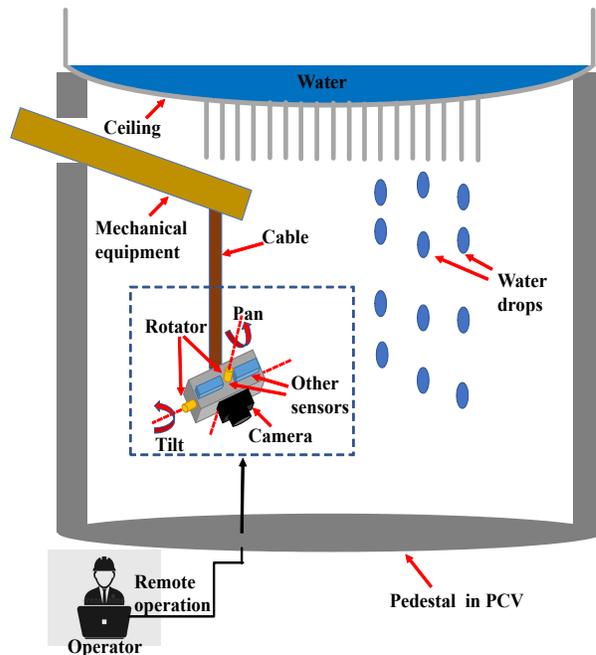


Fig. 1. The inspection system composed of a camera and multiple sensors is hung with an electric cable to perform inspection tasks in the PCV where water drops are dripping consistently from the ceiling.

Accurate camera orientation estimation is necessary for computer vision techniques, such as image stitching and 3D reconstruction. Additionally, camera orientation estimation can help to track not only the movement of the robot, but also the movement of mounted sensors. In [3], a 3D reconstruction scheme of radiation source distribution was used for nuclear power plants inspection. Camera orientation information was required for the robot to be able to create the 3D radiation imaging scheme.

The estimation of camera orientation in indoor environments has been well researched. Camera orientation estimation, which is an important part of simultaneous localization and mapping (SLAM) system, was realized by matching image feature points of multi-view images in [4]. However, this camera orientation estimation method is sensitive to illumination and texture information and therefore cannot work in the environments where the camera would have difficulty detecting and matching the feature points, such as the Fukushima Daiichi Nuclear Power Plant disaster. Figure 2 [5] shows a picture taken in the PCV of unit 2. As shown in Fig. 2, the environment of the PCV of unit 2 lacks illumination and texture.



Fig. 2. A picture taken in the PCV of unit 2 of Fukushima Daiichi Nuclear Power Plant [5].

In addition to the feature points, many other image features are also used to estimate the camera orientation in indoor environments. For example, the line features in man-made indoor environments were employed to solve the Perspective-n-Line (PnL) problem for the estimation of the camera orientation estimation in [6]. The vanishing points of natural geometric lines were detected to estimate the camera orientation in [7]. However, these camera orientation estimation methods cannot be applied in the environments of disaster sites where the natural feature lines have been interrupted by collapse or corrosion. In addition to the methods based on computer vision, an inertial measurement unit (IMU) and cable encoder were used for rotation estimation [8]. However, IMU measurements have drift errors. Additionally, encoder data may not always be available. For example, the encoders can malfunction in an environment where high radiation is present [9].

This study proposes a novel method for estimating the camera orientation by employing a visual feature, i.e, the vanishing point, detected by processing water drops. This method can be used in the leaking indoor environments and can successfully be used in situations where a camera would otherwise have difficulty extracting the feature points or the feature lines of the environment. This research focuses on camera orientation estimation in indoor environments with water drops dripping consistently from a ceiling, like the Fukushima Daiichi Nuclear Power Plant disaster where water that was normally used for injection cooling was consistently dripping from the PCV ceiling. Other studies have taken into account the phenomenon of dripping water. A micro aerial vehicle (MAV) was operated remotely to inspect the containment vessel for the nuclear site decommissioning and the experiments were performed to test the performance of the MAV under the environment with the dripping water in [10].

The remainder of this paper is organized as follows. Section II describes in detail the processes of detecting the water drop trajectories, calculating the vanishing point, and estimating the camera rotation. The effectiveness of our proposed method is evaluated by experiments in Section III.

Section IV presents our conclusions and plans for future work.

II. ROTATION ESTIMATION FRAMEWORK

In the indoor environment of the PCV, water drops dripped vertically from the ground from fixed positions due to the rugged structure of the PCV ceiling. Therefore, the trajectories of the water drops can be considered a set of parallel lines in a 3D space. A set of parallel lines in 3D space will converge into one vanishing point on the camera image by projective transformation. The coordinate of the vanishing point in the pixel coordinate system changes when the camera orientation changes. This is used to estimate the camera rotation. In the proposed setting, the remote-operated, robot-mounted camera can rotate around the tilt and pan axis. Additionally, we assume that the operator would stop the camera for a while to collect enough video data before changing the camera orientation again. Figure 3 gives an overview of our rotation estimation framework. Inputs are two videos recorded before and after the camera orientation changes. Output is the rotation angle we want to estimate. The framework can be divided into three steps: water drops detection, vanishing point estimation, and rotation estimation. We describe the details of each step in the following subsections.

A. Water Drops Detection

A background subtractor algorithm [11] was applied to two image sequences that were recorded by the same camera. This method allows the moving water drops to be detected against the stationary background. Then, each frame that was outputted by the background subtractor was processed by the image preprocessing module, which was composed of image binarization [12], morphology opening operation, and morphology centroid extraction. Image binarization was performed for the subsequent image morphology process. The morphology opening operation was performed to remove salt and pepper noise and enhance the images of the water drop masses. Morphology centroid extraction was performed to estimate the centers of the water drops, which is represented as the two dimensional (2D) point $\mathbf{p}(u, v)$ in the pixel coordinate system whose origin was the top left vertex of video frames. Finally, the temporal integration module was performed as:

$$\mathbf{I}_i = \{\mathbf{p}_i^j | j = 1, 2, 3, \dots, m\}, \quad (1)$$

$$\mathbf{P} = \bigcup_{i=0}^n \mathbf{I}_i, \quad (2)$$

where, \mathbf{p}_i^j is the j -th point in the i -th frame, and \mathbf{I}_i is the set of all the 2D points in i -th frame. \mathbf{P} represents the set composed of 2D points from all the frames. Figure 4 shows the results of that operation and the detected water drops.

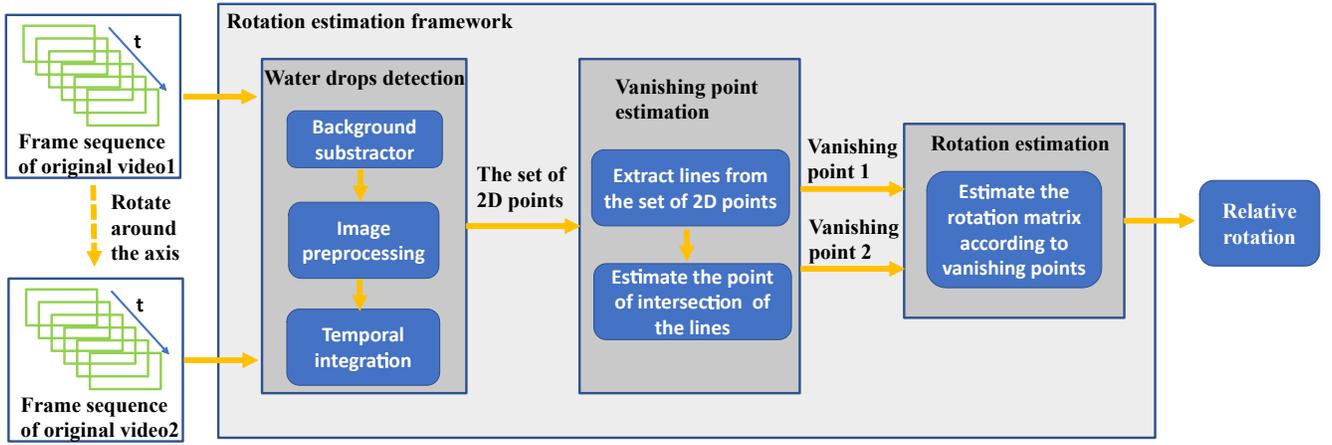


Fig. 3. Overview of the rotation estimation framework based on the vanishing point of water drops.

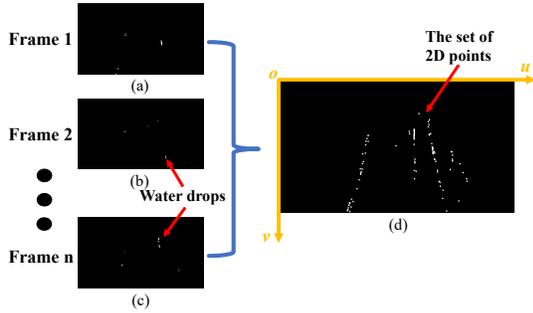


Fig. 4. Water drop detection: (a), (b), (c) are images of detected water drops (i.e., white masses in the images) after each frame is processed by background subtractor and image preprocessing modules. They are binary images including the detected water drop masses. (d) is the illustration of the set of 2D points of the detected water drops.

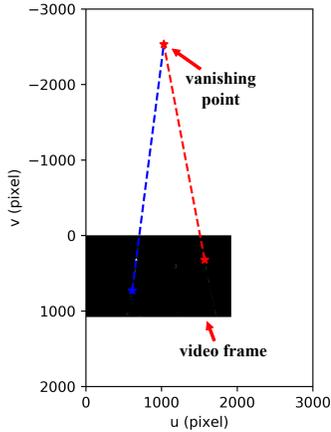


Fig. 5. Vanishing point computation: The blue and red dotted lines represent two trajectory lines of water drops. Their intersection point is the vanishing point.

B. Vanishing Point Estimation

Vanishing point estimation calculated the pixel coordinates of two vanishing points \mathbf{p}_1 and \mathbf{p}_2 based on the set of 2D points \mathbf{P} that were computed in the previous subsection. As mentioned in Section II, the trajectories of water drops

in 3D space can be represented as a bundle of parallel lines that converge into the vanishing point on the image through projective transformation. The proposed algorithm estimated the lines from the set of 2D points of water drops on the images and then calculated the intersection point (i.e., the vanishing point) of the lines, as shown in Fig. 5. Algorithm 1 describes the process used for detecting the lines of water drops and estimating the vanishing point. Here, $\tilde{\mathbf{p}} = (u, v, 1)^T$ represents the homogeneous pixel coordinate of the vanishing point. The trajectory lines of the water drops are estimated with the Random Sample Consensus (RANSAC) [13] and Least Squares algorithms. $|\mathbf{P}|$ is the size of the set of the 2D points and $\mathbf{P}_{\text{inlier}}$ represents those points that are used to fit a line by the RANSAC algorithm. λ is a threshold parameter set preliminary for triggering the end of line detection. The smaller λ is set, the more lines that can be detected. If λ is set too small though, it will lead to the greater probability of detecting wrong lines due to the influence of noise. In the experiments described in this paper, λ was set as the ratio between the size of the set of 2D points and the expected amount of lines. $\mathbf{q}_i = (k_i, -1, b_i)^T$ is the parameter vector of the line and \mathbf{A} is the parameter matrix composed of parameter vectors, which is represented in

Algorithm 1 Vanishing point estimation

Input: the set of 2D points \mathbf{P}

Output: homogeneous coordinate of the vanishing point $\tilde{\mathbf{p}}^*$

```

i = 0
while |P| > λ do
    P_inlier ← RANSAC(P)
    i ← i + 1
    q_i ← LeastSquares(P_inlier)
    P ← P - P_inlier
    |P| ← |P| - |P_inlier|
end while
A ← [q_1; q_2; ...]_{i × 3}^T
p̃* ← arg min_p ||A p̃||

```

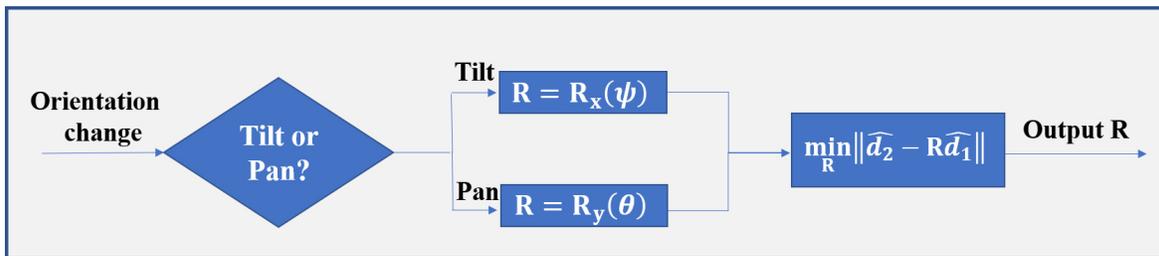


Fig. 6. Decision tree of the rotation estimation step.

Eq. (3). To estimate the only intersection point from multiple lines, we performed the minimization of $\|\mathbf{A}\tilde{\mathbf{p}}\|$, which is solved by singular value decomposition (SVD) method. Here, \mathbf{A} is formulated as:

$$\mathbf{A} = \begin{pmatrix} k_1 & k_2 & k_3 \\ -1 & -1 & -1 & \dots \\ b_1 & b_2 & b_3 \end{pmatrix}^T, \quad (3)$$

where k_i and b_i ($i = 1, 2, 3, \dots$) are the parameters of the line equation $k_i x - y + b_i = 0$.

C. Rotation Estimation

The details of the rotation estimation are shown in Fig. 6. The camera is rotated by remote operations and the rotations are distinguished as pan or tilt. As shown in Fig. 7, the vanishing point is the intersection point of the image plane and the line, which passes through the optical center of the camera and parallel to the dripping direction of the water drops. Additionally, the position of the vanishing point in the image is correlated to the orientation of the camera [14]. Here, the homogeneous pixel coordinate of the vanishing point is represented as $\tilde{\mathbf{p}}$ and the unit direction vector of the vanishing point in the camera coordinate is represented as $\hat{\mathbf{d}} \in \mathbb{R}^3$. The relation between them can be written as:

$$\hat{\mathbf{d}} = \mathbf{K}^{-1}\tilde{\mathbf{p}}/\|\mathbf{K}^{-1}\tilde{\mathbf{p}}\|, \quad (4)$$

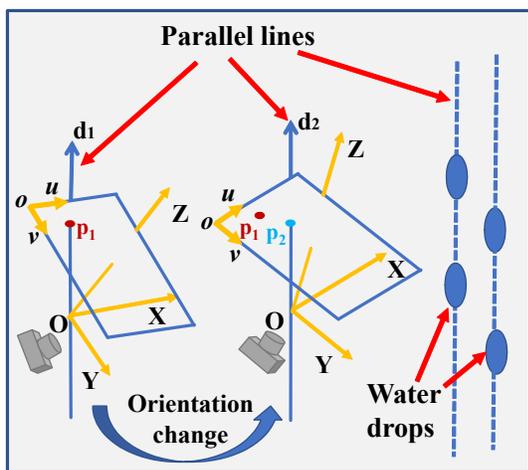


Fig. 7. $\mathbf{p}_1, \mathbf{p}_2$ are the vanishing points and $\mathbf{d}_1, \mathbf{d}_2$ are direction vectors of the vanishing points in the camera coordinate system. When the camera orientation changes, the direction vector of the vanishing point also changes.

where \mathbf{K} is the camera intrinsic parameter matrix. The relation between two unit direction vectors of the vanishing points (i.e. $\hat{\mathbf{d}}_1$ and $\hat{\mathbf{d}}_2$) can be represented as:

$$\hat{\mathbf{d}}_2 = \mathbf{R}\hat{\mathbf{d}}_1, \quad (5)$$

where \mathbf{R} is the rotation matrix of the camera. The computation of the tilt (i.e, rotation around x axis) and pan (i.e, rotation around y axis) angles were performed independently, as we assumed that the camera mounted on the robot could be rotated remotely around the tilt and pan axis. The tilt angle is represented as ψ and the pan angle is represented as θ . The rotation matrix of the tilt $\mathbf{R}_x(\psi)$ and pan angle $\mathbf{R}_y(\theta)$ is represented as:

$$\mathbf{R}_x(\psi) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \psi & -\sin \psi \\ 0 & \sin \psi & \cos \psi \end{pmatrix}, \quad (6)$$

$$\mathbf{R}_y(\theta) = \begin{pmatrix} \cos \theta & 0 & \sin \theta \\ 0 & 1 & 0 \\ -\sin \theta & 0 & \cos \theta \end{pmatrix}. \quad (7)$$

Finally, the rotation angle tilt or pan can be solved by using the optimization method Levenberg-Marquardt (LM) [15] algorithm to solve (5):

$$\mathbf{R}^* = \arg \min_{\mathbf{R}} \|\hat{\mathbf{d}}_2 - \mathbf{R}\hat{\mathbf{d}}_1\|. \quad (8)$$

III. EXPERIMENTS

Simulated experiments were performed by rendering virtual images with the 3D simulation software Blender [16] to prove the feasibility of the proposed rotation estimation method based on the vanishing point of water drops. We also simulated a indoor environment with dripping water and used an actual camera to collect video data and verify the proposed rotation estimation framework. We evaluated the effectiveness of our method with quantitative error analysis.

A. Simulated

The simulation software Blender can render images recorded by cameras and output the true value of the tilt and pan rotation angles of the cameras, as shown in Fig. 8. In the Figure, the ellipses represent the water drops and a camera was used to render the images of water drops. Additionally, there were three positions where water drops were dripping vertically to the ground and their drip trajectories were

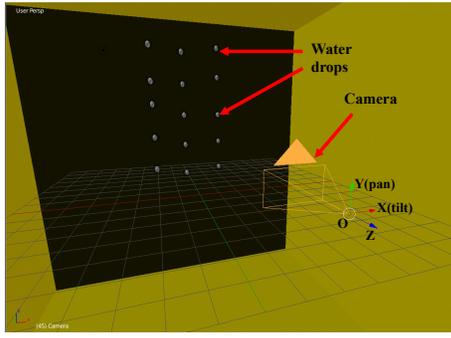


Fig. 8. A computer-simulated experimental environment created in Blender. The gray ellipses represent water drops. A camera was used to render the images of water drops.

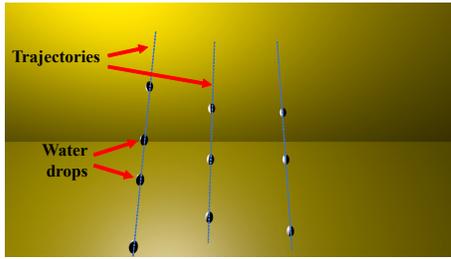


Fig. 9. An example of an image rendered by the camera in Blender. This image was rendered when the tilt angle was 40° and pan angle was 0° . The blue dotted lines are reference lines, which were not rendered by Blender and added manually.

parallel in the 3D space. The computer-simulated experiment was performed to verify whether the camera rotation could be estimated based on the calculation of the vanishing point of the water drops. Therefore, we focused on the second and third steps of our proposed framework for this experiment and simplified the model of the water drops as static objects. Figure 9 is an example of an image rendered by the camera. The figure shows that the three trajectories of water drops tend to converge into one point (i.e., the vanishing point). The pixel coordinates of the water drop centers in the images were manually marked in the image. Four center points were used to fit two different lines and then the intersection point of the two lines was estimated as the vanishing point. Finally, the rotation angle was calculated using the vanishing points of the two orientations. We performed the experiments for three situations (i.e., only rotation around the pan angle of the camera, only rotation around the tilt angle of the camera, and rotation around the tilt or pan angle of the camera alternately). We rotated the camera by 5° each time. We changed the camera orientation for all three model situations. Figure 10 shows error analysis of these experiments. The mean errors were all less than 0.5° , which proves the effectiveness of the proposed rotation estimation method. The errors were generated from the inaccurate estimation of water drop centers, lines, and the vanishing point. The errors were analyzed in the worse case. The errors could be improved by automated algorithms, such as estimating the water drop centers with image processing methods, estimating the lines

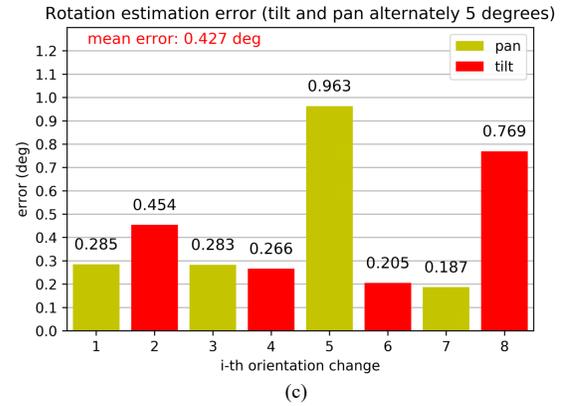
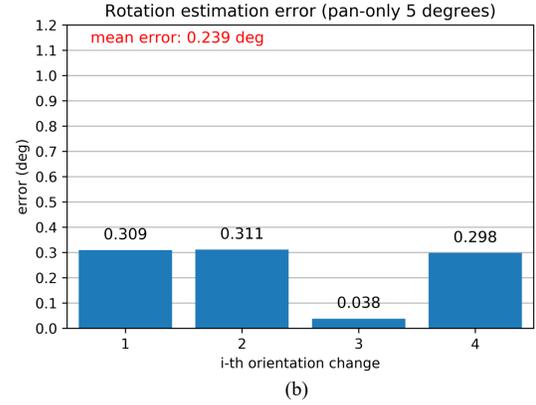
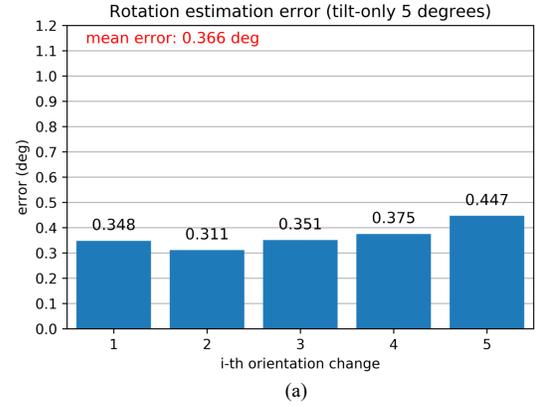


Fig. 10. Rotation estimation errors: (a), (b), (c) correspond to pan, tilt and both rotation respectively.

with the Least Squares algorithm, integrating the temporal information to obtain more water drops, or estimating the intersection point with SVD optimization applied to multiple lines. These algorithms were used in the real-world experiment described in the following subsection.

B. Real-world Environment

Figure 11 shows the set-up of the real-world experiment. The set-up included a camera, a rotator, water drop generators and a cardboard box that had little texture. The camera used was GoPro Hero7 Black [17] in linear mode recording 30 frame per second (fps) and in

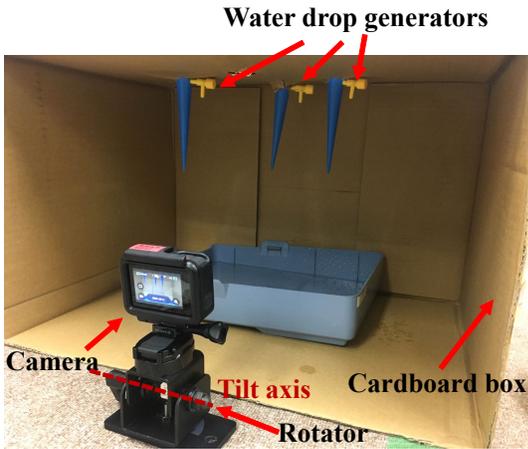


Fig. 11. Real-world experiments.

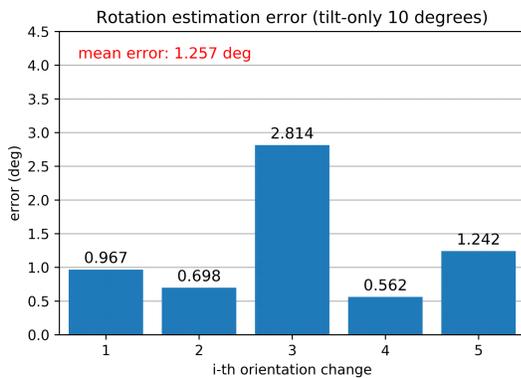


Fig. 12. Rotation estimation errors. The camera was rotated around the tilt axis 10° each time.

1920 × 1080 pixels format. A rotator was used to rotate the camera around the tilt axis. Water drops fell from the yellow spigots shown in Fig. 11. Cardboard box was used to simulate an indoor environment and to affix the water drop generators. The intrinsic parameter was calibrated using the method shown in [18]. The experiment process complies with the framework we described in Section II. We rotated the camera five times and we changed the tilt angle by 10° . In this experiment, the camera was able to successfully detect the water drops and estimate the correct rotation angle. The result of error analysis of the rotation estimation is shown in Fig. 12, which proves the feasibility of the proposed framework in a real environment. The error of the third orientation change was larger than others, which was the result of the loud noise in the processing of detecting the water drops. Additionally, the accuracy of the result is also influenced by the measurement of the rotator and the relative location between the rotator and the camera, which could be improved in future experiments.

IV. CONCLUSIONS

In this paper, a novel method that employed the vanishing point of water drops was proposed and used to estimate the camera rotation in indoor environments with dripping

water. The camera orientation framework was composed of water drop detection, vanishing point estimation, and rotation estimation. Computer-simulated and real-world experiments data were conducted to confirm the effectiveness of our method. The proposed solution can be used for remote inspection by robots via mounted cameras and to compute rotation estimation when a camera would have difficulty detecting feature points or lines in an environment.

In future work, we will attempt to overcome the limitations of the proposed method related to the singularity that happens when the rotation axis is parallel to the direction in which the water drops are dripping. This makes it impossible for the robot to estimate the rotation angle. Additionally, in future work we hope to be able to allow the camera to rotate arbitrary instead of just around one axis each time.

REFERENCES

- [1] S. Kawatsuma, "Contribution by incorporation of laser technology and robotics technology for Fukushima Daiichi Nuclear Power Plants decommissioning," *Reza Kenkyu*, vol. 45, no. 7, pp. 413–417, 2017.
- [2] Investigation of PCV of unit 2, https://irid.or.jp/wp-content/uploads/2018/04/20180426_1.pdf, accessed 27 Aug. 2020.
- [3] D. Kim, H. Woo, Y. Ji, et al., "3D radiation imaging using mobile robot equipped with radiation detector," *Proceedings of the IEEE/SICE International Symposium on System Integration*, pp. 444–449, 2017.
- [4] R. Mur-Artal, J. D. Tardos, "ORB-SLAM2: an open-source slam system for monocular, stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, pp. 1255–1262, 2017.
- [5] The picture taken in PCV, https://photo.tepco.co.jp/library/180119_01/180119_05.jpg, accessed 27 Aug. 2020.
- [6] C. Xu, L. Zhang, L. Cheng and R. Koch, "Pose estimation from line correspondences: A Complete Analysis and a Series of Solutions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 6, pp. 1209–1222, 2017.
- [7] P. V. Gakne, K. O'Keefe, "Monocular-based pose estimation using vanishing points for indoor image correction," *Proceedings of the IEEE International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–7, 2017.
- [8] A. C. Murtra, J. M. M. Tur, "IMU and cable encoder data fusion for in-pipe mobile robot localization," *IEEE Conference on Technologies for Practical Robot Applications*, pp. 1–6, 2013.
- [9] D. Hamonic, J. D. Saussine, E. Feuilleley, "Radiation hardening methodology applied on absolute optical encoder of a total dose beyond 100 kGy," *IEEE Fifth European Conference on Radiation and Its Effects on Components and Systems*, no. 99TH8471, pp. 382–386, 1999.
- [10] D. Thakur, G. Loianno, L. Jarin-Lipschitz, et al., "Autonomous inspection of a containment vessel using a micro aerial vehicle," *Proceedings of the IEEE International Symposium on Safety, Security, and Rescue Robotics*, pp. 1–7, 2019.
- [11] Z. Zivkovic, F. Van Der Heijden, "Efficient adaptive density estimation per image pixel for the task of background subtraction," *Pattern recognition letters*, vol. 27, no. 7, pp. 773–780, 2006.
- [12] N. Otsu, "A threshold selection method from gray-level histograms," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 9, no. 1, pp. 62–66, 1979.
- [13] M. A. Fischler, R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.
- [14] R. Hartley, A. Zisserman, *Multiple view geometry in computer vision*, Cambridge University Press, 2003.
- [15] J. J. Moré, "The Levenberg-Marquardt algorithm: implementation and theory," *Numerical analysis*, Springer, Berlin, Heidelberg, pp. 105–116, 1978.
- [16] Blender, <https://www.blender.org>, accessed 27 Aug. 2020.
- [17] GoPro, <https://gopro.com/en/cn/>, accessed 27 Aug. 2020.
- [18] Z. Zhang, "A flexible new technique for camera calibration," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1330–1334, 2000.