ScciencePG
Science Publishing Group

Localization Method Based on Image Processing for Autonomous Driving of Mobile Robot in the Linear Infrastructure

Hyunwoo Song[1, *], Jun Nakahama[2], Yogo Takada[1]

[1]Mechanical and Physical Engineering Graduate School, Osaka City University, Osaka, Japan

[2]Mechanical Engineering, Osaka City University, Osaka, Japan

Email address:
thgudn@naver.com (H. Song), takada@eng.osaka-cu.ac.jp (Y. Takada)
[*]Corresponding author

To cite this article:
Hyunwoo Song, Jun Nakahama, Yogo Takada. Localization Method Based on Image Processing for Autonomous Driving of Mobile Robot in the Linear Infrastructure. *Automation, Control and Intelligent Systems*. Vol. 9, No. 1, 2021, pp. 34-45. doi: 10.11648/j.acis.20210901.15

Abstract: In recent years, the deterioration of infrastructure facilities such as bridges has become a problem. Precautionary measures such as visual inspection and repair by humans are in place as countermeasures for aging; however, there are issues with cost and safety in such inspections. If inspection by robots becomes possible, both these aspects will be improved, which will significantly contribute to the maintenance of infrastructure facilities. In this paper, we propose a complex image processing technique to specify the location of feature points as coordinates through smartphone cameras to obtain the location information of feature points needed for positioning BIREM-IV-P developed to support bridge inspection. The corners located in the bridge inspection environment are used as feature points, and the corners are specified using Harris corner detection, which is a conventional corner detection method, to obtain the position of the feature points. In addition, to compensate for the shortcomings of Harris corner detection, a line segment in the image is detected using the Hough transform, and the intersection points of the line segments are recognized as corners. By combining the results of the two detection methods in this manner, the target feature points can be accurately specified. Then, the position of the feature points of the specified image coordinate system can be changed to the world coordinate system. As a result, it was possible to detect the location of the target feature points in a three-dimensional coordinate system.

Keywords: Corner Detection, Hough Transform, Harris Corner Detection, Image Processing, Coordinate Transformation

1. Introduction

In recent years, the aging of infrastructure facilities, such as bridges and tunnels, has become a problem. Much of the social infrastructure in Japan has been built after the high economic growth period. Although recently constructed bridges are designed to have a lifespan of 100 years, the general lifespan of bridges is known to be around 50 years. In addition, the proportion of bridges that have passed 50 years since construction is approximately 27% as of 2019, which will surge to approximately 52% after 10 years. In particular, the proportion of bridges shorter than 15 m in length will be approximately 59% in 10 years. In addition, there are about 230,000 road bridges in the country with unknown construction years. At present, preventive measures such as visual inspection and repair by people are being undertaken as measures against aging. However, the bridge inspection method is currently performed by inspectors using scaffoldings and special crane vehicles, which require considerable time and inspection cost, and the lack of inspectors and safety issues pose problems. Therefore, if inspection by robots becomes possible, these problems will be alleviated, thereby significantly contributing to the maintenance and management of infrastructure facilities.

In recent times, various robots are being developed as bridge inspection robots [1–9]. There are various types of robots, such as flying, suspension, and adsorption types, but they are mainly divided into flying and adsorption types [10-17]. Among them, the flying robot has a problem in that it is difficult to control because its balance is easily affected

by disturbances such as wind and contact with the bridge. In addition, there is a problem not only when moving, but also when hovering, because power is consumed to maintain posture and position; thus, it is not possible to conduct a long-term inspection. In contrast, adsorption robot can also be moved by adsorbing on the ceiling surface of the lower part of the bridge, and unlike flying robot, they do not use excessive power to control their posture and position even while stationary for bridge inspection. Therefore, in this laboratory, the Bridge Inspection Robot Equipped with Magnets version IV (BIREM-IV) was developed as a robot for bridge inspection [18]. The robot's features include the ability to move freely on steel structures because of the use of magnetic rimless wheels and a high driving capability. However, BIREM-IV and other bridge inspection robots with these characteristics are directly controlled by the inspector; hence, there is a requirement to develop a bridge inspection robot that is fully autonomous.

The most important aspect of autonomous driving to detect of the robot's localization. However, it is difficult to detect the position of the robot as coordinates using the Global Navigation Satellite System (GNSS) because data from satellites cannot be received inside large buildings, such as inside bridge box girders. Therefore, using the characteristics of the light detection and ranging (LiDAR) sensor, we developed the planar LiDAR, which detects the position of the robot as coordinates based on the distance and direction data from the feature points fixed in the experimental environment, and installed it in BIREM-IV [19]. Figure 1, Figure 2, and Table 1 present the appearances, components, and specifications of BIREM-IV-P, respectively.
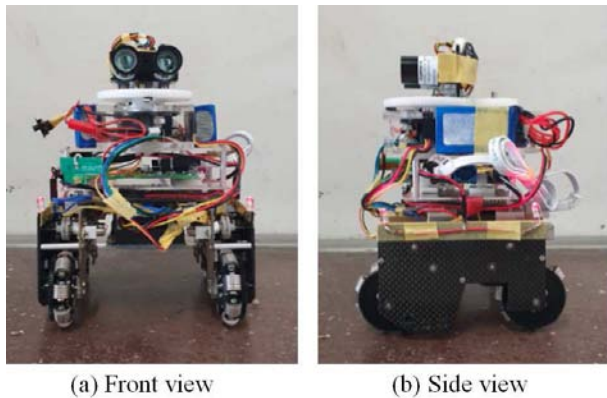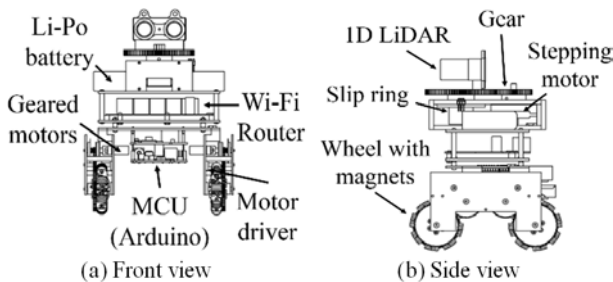

(a) Front view   (b) Side view

Figure 1. *Appearances of BIREM-IV-P.*

(a) Front view   (b) Side view

Figure 2. *Components of BIREM-IV-P.*

Table 1. *Specifications of BIREM-IV-P.*

BIREM-IV-P	
Length	140.0 mm
Width	140.0 mm
Total height	180.0 mm
Mass	1.006 kg
Battery	LiPo, 11.1 V, 600 mAh, 65C
MCU	Arduino Mega 2560 R3

The manufactured planar LiDAR can obtain 360° field of vision information based on the robot by rotating the long measuring distance and a compact one-dimensional LiDAR sensor using a stepping motor. In addition, the feature points of the inspection site can be set in advance using existing information on the bridge structure (design drawings and photos), and the position of the robot can be simply specified based on the feature point information. From these features, BIREM-IV-P is capable of localization using feature point information obtained from planar LiDAR. However, these methods can cause problems where fewer feature points are recognized by planar LiDAR; the greater the distance between the robot and feature points, the greater the error. Therefore, in this study, landmarks present in the advancing direction of the robot are detected using a camera, recognized as feature points, and three-dimensional position coordinates in millimeter units are derived from the obtained two-dimensional position coordinates in pixel units. Then, the reliability of the three-dimensional position coordinates of the obtained feature points is confirmed.

2. Specific Feature Point Detection

In this section, we propose a method for sensing corners in the advancing direction of the robot and detecting target feature points among the detected corners. We aimed to recognize the corners in the experimental environment through image processing of the video captured using a smartphone camera (OS: Android, Model: Galaxy S6 Edge Plus, Manufacturer: Samsung Electronics Co., Ltd). The feature points are fixed in the experimental environment, and their positions are specified as coordinates.
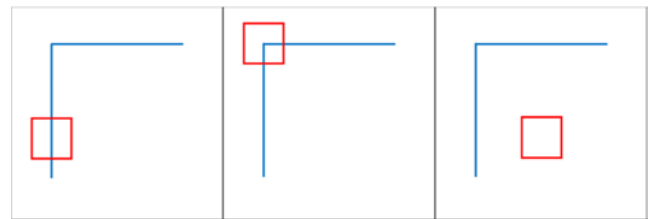
2.1. Traditional Corner Detection

Figure 3. *Moravec corner detection.*

As a pioneering study in corner detection research, Moravec's corner detection method has been studied [20], based on which numerous methods have been proposed [21–24]. As shown in Figure 3, corner detection sets a small area (window) of an image, and when that area moves gradually, it

is recognized as an edge or corner if the rate of change is large. If the rate of change on the x-axis or y-axis is large, it is an edge, and if both are large, it is a corner.

However, this detection method clearly shows shortcomings at the edge of a diagonal line, because the diagonal line does not exhibit a large change in both the x and y directions. Thus, the Harris corner detection method solves this problem [25].

The Harris corner detection is used by converting the camera's two-dimensional color images to grayscale. If the image converted to grayscale is denoted as I, and the amount of change between the point (x, y) on the area and the point $(x + u, y + v)$ moved by u and v respectively in the x and y directions is $E(u, v)$, it can be expressed as

$$E(u, v) = \sum_{x,y} w(x, y)[I(x + u, y + v) - I(x, y)]^2 \quad (1)$$

where $w(x, y)$ is a window function that represents a Gaussian filter. Thus, Equation (1) calculates the amount of change smoothed by the Gaussian filter. Furthermore, if this equation is approximated using the Taylor expansion, it becomes

$$E(u, v) \approx [u, v] M \begin{bmatrix} u \\ v \end{bmatrix} \quad (2)$$

The matrix M in this equation is

$$M = \sum_{x,y} w(x, y) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (3)$$

where I_x and I_y are image derivatives in the x and y directions respectively. When $[I_x, I_y]$ is squared, it becomes the matrix part of M. This matrix part is a matrix describing the variation of M, and it is possible to determine edges or corners by singular value decomposition. However, because it is difficult to calculate complicated eigenvalues, we use the following equation:

$$R = \det(M) - k(\text{trace}(M))^2$$

$$\begin{cases} \det(M) = \lambda_1 \lambda_2 \\ \text{trace}(M) = \lambda_1 + \lambda_2 \end{cases} \quad (4)$$

where $\det(M)$ is the determinant of the matrix, $\text{trace}(M)$ is its trace, k is a constant that is generally 0.04 to 0.06, and λ_1 and λ_2 are the eigenvalues of M. It can be seen that the value of R can be obtained from these. In addition, the eigenvalues of M can be classified as shown in Figure 4.
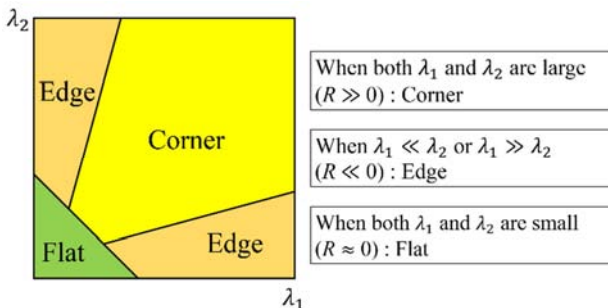

Figure 4. *Classification of image points using eigenvalues of M.*

As a result, we were able to detect the corners using the Harris corner detection method. However, it was not possible to identify which of the detected corners was the target feature point.

2.2. Corner Detection Method Specialized in Feature Detection of Proposed Target

It was possible to capture the target feature point candidates using the Harris corner detection method described in the previous section. However, it was not possible to reliably identify the landmarks as feature points. Therefore, as shown in Figure 5, the proposed method obtains the target feature points by associating the feature points obtained by Harris corner detection with the feature points obtained by Hough transformation.
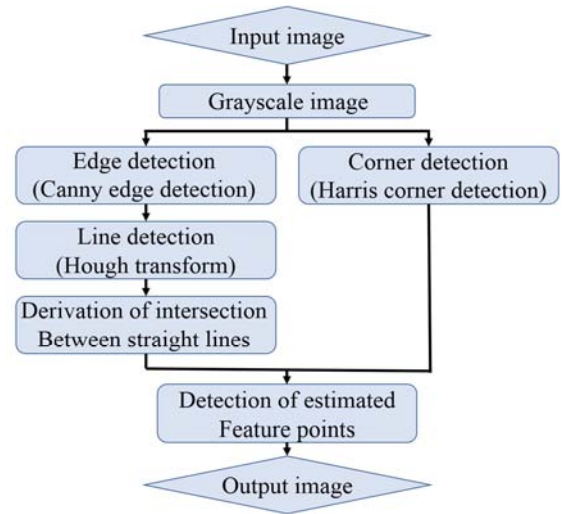

Figure 5. *Flowchart of proposed method.*

2.2.1. Canny Edge Detection

The Canny edge detection method is often used to extract the contour of an image, and it performs noise removal, contour extraction, non-maximum suppression, and hysteresis threshold processing in sequence [26]. As a feature of Canny edge detection, it is possible to reduce the number of edges that cause noise using two thresholds, as well as the number of omissions and false detections of contours.

The input image is smoothed using a Gaussian filter to calculate the differential of the smoothed image. Therefore, the contour is detected by calculating the magnitude and direction of the gradient from the differentiated result, performing non-maximum suppression processing and hysteresis threshold processing.

The Gaussian filter is a filter used to smooth images. Natural smoothing can be performed by using the weighting of the pixel values around the pixel to be processed and the Gaussian distribution.

$$G = I K_g \quad (5)$$

where G is the smoothed image, I is the input image, and K_g

is the kernel of the Gaussian filter. The obtained smoothed image is differentiated using the Sobel filter. The images of the horizontal differential G_x and vertical differential G_y of the Sobel filter are expressed as

$$G_x = GK_x$$

$$G_y = GK_y \qquad (6)$$

where the magnitude and direction of the gradient are calculated from the differential image. From the differential image G, the magnitude of the gradient $|G|$ and direction of the gradient θ are expressed as

$$|G| = \sqrt{G_x^2 + G_y^2}$$

$$\theta = \tan^{-1}\left(\frac{d_y}{d_x}\right) \qquad (7)$$

To simplify the implementation of the thinning process, the contour of the differential image $|G|$ by non-maximum suppression processing is subdivided. Then, the thinning process is performed to supplement the lines. Finally, hysteresis threshold processing is performed, and reliable and unreliable contours are selected from two thresholds (maximum and minimum). The evaluation of contour reliability is presented in Table 2.

Table 2. *Contour evaluation.*

Pixel value	Evaluation
Less than the minimum threshold	Unreliable contour
Minimum threshold to maximum threshold	If there is a highly reliable contour next to it, it is a highly reliable contour. Otherwise unreliable contour
Greater than the maximum threshold	Reliable contour

2.2.2. Hough Transform

The purpose of the Hough transform is to perform groupings of edge points into object candidates by performing an explicit voting procedure over a set of parameterized image objects [27]. The Hough transform is executed using the image adaptation obtained by Canny edge detection to detect the straight lines in the image. As shown in Figure 6, a straight line can be represented by two parameters ρ and y shows as
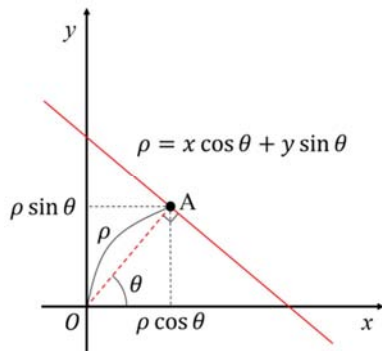

Figure 6. *Straight line expressed in xy coordinates.*

$$\rho = x\cos\theta + y\sin\theta$$

$$= -\frac{\cos\theta}{\sin\theta}x + \frac{\rho}{\sin\theta} \qquad (8)$$

In Figure 6, A is the point before the Hough transform. As shown in Figure 7, the straight lines passing through point A can be represented by ρ and θ.
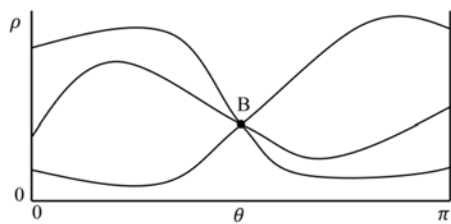

Figure 7. *Straight line expressed in ρ-θ coordinates.*

Because of the symmetry of the figure, θ is considered only as $(0 \leq \theta \leq \pi)$. In addition, ρ is never longer than the diagonal of the input image. There are as many curves in Figure 7 as the number taken up by the Hough transform. In other words, point B where many curves obtained intersect is a straight line where a set of ρ and θ strongly predicted from multiple points of the input image is detected.

2.3. Verification Experiment

In the experiment, feature points were detected based on the proposed method using 640×480 pixel input images obtained from a smartphone camera. As shown in Figure 8, the input image is converted to grayscale.
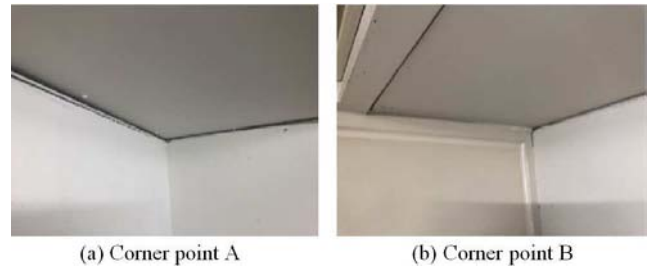

(a) Corner point A   (b) Corner point B

Figure 8. *Input photograph.*

Figure 9 shows the result of converting the input image of the camera to grayscale.

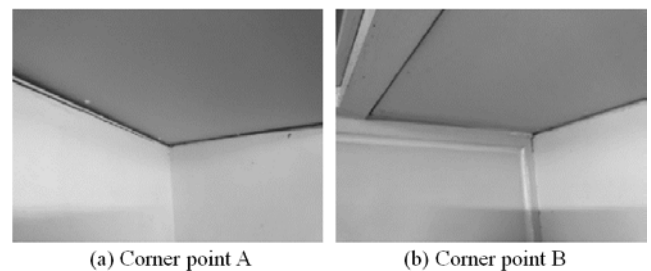(a) Corner point A   (b) Corner point B

Figure 9. *Grayscale image from input photograph.*

Figure 10 shows the result of feature point detection using

the Harris corner detection method, which is the conventional corner detection method described in the previous section, for this grayscale image.
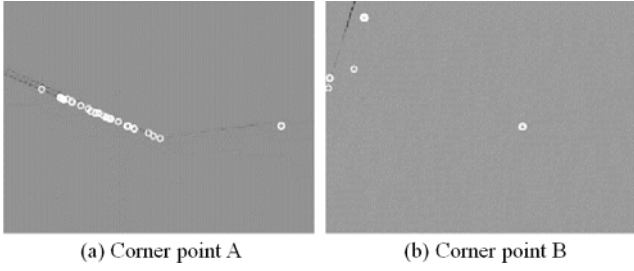

(a) Corner point A   (b) Corner point B

Figure 10. *Result of Harris corner detection.*

In these figures, the positions surrounded by the white circles are detected as corners. Harris corner detection enables the detection of corners, but it can be observed that corners other than the target feature points have also been detected. Therefore, we focused on the geometric features that appear near the corners and narrowed down the target feature points. For this reason, as shown in Figure 11, noise is removed from the grayscale input image by Canny edge detection, and the Hough transform is performed to detect straight lines in the image.
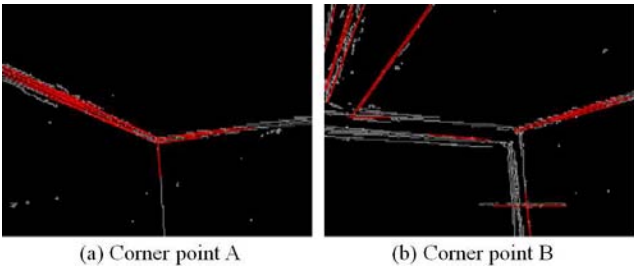

(a) Corner point A   (b) Corner point B

Figure 11. *Result of Canny edge detection with Hough transform.*

Then, as shown in Figure 12, the intersection of the lines obtained from the Hough transform is located, and the target feature points are among these intersections.
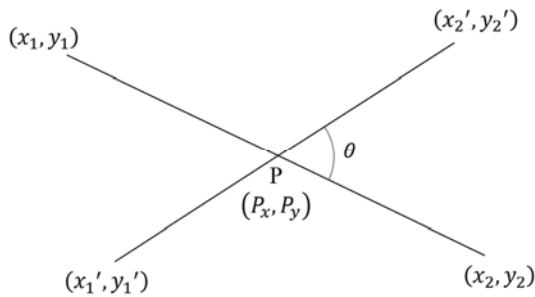

Figure 12. *Intersection of two lines.*

The line segment equations obtained at the start point (x_1, y_1) and end point (x_2, y_2) of the line segment are shown below. However, the x-axis is positive upward, and the y-axis is positive right.

$$y = \frac{y_1-y_2}{x_1-x_2}x + y_2 - \frac{y_1-y_2}{x_1-x_2}x_2 \qquad (9)$$

In addition, the coordinates of the intersection P of the line segment obtained from the other start point (x_1', y_2') and the end point (x_2', y_2') can be obtained. It is expressed as

$$P_x = \frac{ace + bc \times (x_2) - ad \times (x_2')}{bc - ad}$$

$$P_y = \frac{bdf + ad \times (y_2) - bc \times (y_2')}{ad - bc} \qquad (10)$$

where $a = x_1 - x_2$, $b = y_1 - y_2$, $c = x_1' - y_1'$, $d = y_1' - y_2'$, $e = y_2' - y_2$ and $f = x_2' - x_2$. In addition, only intersections with an angle θ ($0° \leq \theta \leq 90°$) between straight lines of 45° or more are considered to prevent the detection of unnecessary intersections as follows:

$$\tan \theta = \frac{m_1 - m_2}{1 + m_1 m_2} \geq 1 \qquad (11)$$

where m_1 and m_2 are the inclinations of the straight lines and are expressed as

$$m_1 = \frac{y_1 - y_2}{x_1 - x_2}, m_2 = \frac{y_1' - y_2'}{x_1' - x_2'} \qquad (12)$$

To the images obtained using the Hough transform, the above is applied to specify the intersection points of the lines. The identified intersection points are represented by green circles as shown in Figure 13.
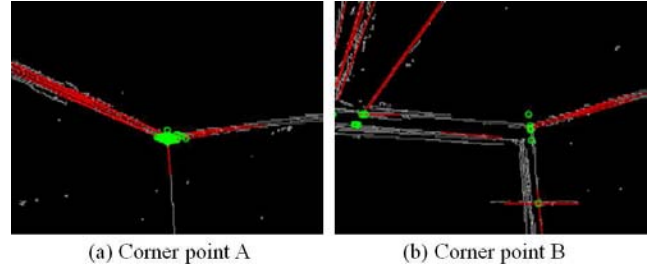

(a) Corner point A   (b) Corner point B

Figure 13. *Derivation of intersections between straight lines.*

As a result, candidates for multiple feature points can be obtained from geometric features. However, it was confirmed that there are many candidates for feature points and points other than corners are also recognized. The above is applied to the results obtained from Harris corner detection and the results are shown in Figure 14. Based on the above results, it was confirmed that the target corner was recognized as the feature point by combining Harris corner detection with data obtained from the Hough transform.
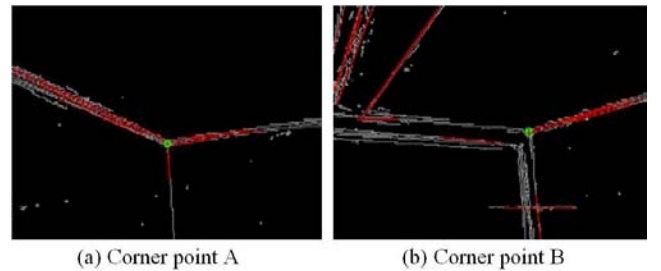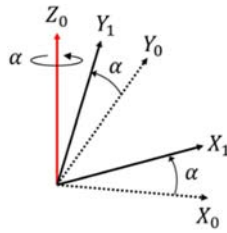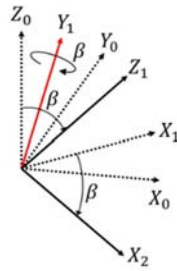

(a) Corner point A   (b) Corner point B

Figure 14. *Corner detection of proposed method.*

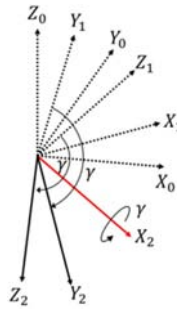# 3. Correspondence Between Two-Dimensional and Three-Dimensional Points

As described in the previous section, image processing was performed using the image data obtained from the camera. The experimental result confirmed that the image coordinates obtained in pixel units of the target feature points in the image were obtained. However, the position in the experimental environment cannot be confirmed by the pixel-based image coordinates on the image. Therefore, this section describes the process of obtaining three-dimensional position coordinates in millimeter units from these image coordinates.

(a) Rotation around Z-axis

(b) Rotation around Y-axis

(c) Rotation around X-axis

Figure 15. *Representation of posture by ZYX Euler angle.*

The camera's posture, considering that it will cover subsequent three-dimensional composite paths, is adopted by the positioning method by Euler angles. Figure 15 presents the posture expression by Euler angles, and the direction of rotation is clockwise with respect to the positive directions of the X-axis, Y-axis, and Z-axis. The Euler angles are a method of expressing the posture of a rigid body as α, β, and γ. In addition, there are various expressions for Euler angles such as ZXZ Euler angles and XYZ Euler angles, but herein we use

ZYX Euler angles. Then, the ZYX Euler angles are rotated once around the Z-axis, as shown in Figure 15 (a), once around the Y-axis, as shown in Figure 15 (b), and once around the X-axis, as shown in Figure 15 (c). This can be associated with roll angles, pitch angles, and yaw angles depending on the camera position.

Then, the ZYX Euler angle is indicated a rotation matrix R. The rotation matrix R rotating by angles α, β, and γ, around each axis of Z, Y, and X, denoted as R_Z, R_Y, and R_X, respectively, can be expressed as

$$R_Z(\alpha) = \begin{pmatrix} \cos\alpha & -\sin\alpha & 0 \\ \sin\alpha & \cos\alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$R_Y(\beta) = \begin{pmatrix} \cos\beta & 0 & \sin\beta \\ 0 & 1 & 0 \\ -\sin\beta & 0 & \cos\beta \end{pmatrix}$$

$$R_X(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos\gamma & -\sin\gamma \\ 0 & \sin\gamma & \cos\gamma \end{pmatrix} \qquad (13)$$

Then, using the above equation, the rotation matrix R of the ZYX Euler angles can be expressed as

$$R = R_Z(\alpha)R_Y(\beta)R_X(\gamma)$$

$$= \begin{pmatrix} C_\alpha C_\beta & C_\alpha S_\beta S_\gamma - S_\alpha C_\gamma & C_\alpha S_\beta C_\gamma + S_\alpha S_\gamma \\ S_\alpha C_\beta & S_\alpha S_\beta S_\gamma + C_\alpha C_\gamma & S_\alpha S_\beta C_\gamma - C_\alpha S_\gamma \\ -S_\beta & C_\beta S_\gamma & C_\beta C_\gamma \end{pmatrix} \qquad (14)$$

where it is represented by $C_\alpha = \cos\alpha$, $S_\alpha = \sin\alpha$, $C_\beta = \cos\beta$, $S_\beta = \sin\beta$, $C_\gamma = \cos\gamma$, and $S_\gamma = \sin\gamma$ for simplification.

3.1. Perspective Projection Transformation

In the perspective projection camera model, the point P_w in the world coordinate system is rendered as the point P_i in the image coordinate system using a smartphone camera, as shown in Figure 16.
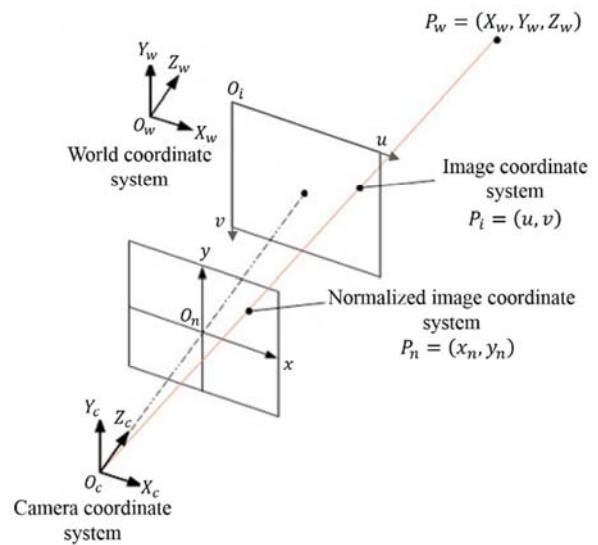

Figure 16. *Relationship of each coordinate system.*

The point P_w in the world coordinate system is placed as an arbitrary three-dimensional point viewed from the camera located at the origin O_c of the camera space, and the point P_i in the image coordinate system is an image of point P_w represented by the image coordinate system. The point P_i represents the pixels of the image captured by the camera, which are formed by intersecting the optical center of the camera with the rays from the point P_w passing through the image plane. When the point P_w in the world coordinate system is represented by the point P_i in the image coordinate system, it is expressed as

$$\begin{bmatrix} X_w \\ Y_w \\ Z_w \end{bmatrix} = R^{-1}\left(sF^{-1}\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} - t \right) \tag{15}$$

where R denotes a 3×3 rotation matrix, t denotes a 3×1 smooth matrix (external parameter), F denotes a camera matrix (internal parameter), and s denotes a constant. The external parameters represent the conversion from the world coordinate system to the camera coordinate system, and the internal parameters represent the conversion from the camera coordinate system to the image coordinate system.

3.1.1. External Parameters

The external parameter is a matrix that converts the world coordinate system into the camera coordinate system. If the world coordinate point viewed from the world coordinate system is $P_w = (X_w \quad Y_w \quad Z_w)^T$ and the camera coordinate point viewed from the camera coordinate system is $P_c = (X_c \quad Y_c \quad Z_c)^T$, it can be expressed as

$$\begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & t_1 \\ r_{21} & r_{22} & r_{23} & t_2 \\ r_{31} & r_{32} & r_{33} & t_3 \end{bmatrix}\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix}$$

$$= [R|t]\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = {}^{O_w}_{O_c}H\begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \tag{16}$$

The simultaneous transformation matrix ${}^{O_w}_{O_c}H$ was used to express the camera's posture and position. In addition, the camera attitude can be expressed using the three degrees of freedom of the Euler angle. Originally, the Euler angle has a singularity, so there are postures that cannot be uniquely expressed by only one Euler angle. However, in this study, feature points are detected on the ceiling surface, and only the yaw angle direction is considered, thus it can be expressed.

3.1.2. Internal Parameters

The internal parameter is a matrix that converts the camera coordinate system into the image coordinate system. To explain this, it is necessary to clarify the conversion from the camera coordinate system to the normalized image coordinate system. The normalized image coordinate system is a coordinate system that considers an image plane placed at a reference position at the center of the image plane.

From Figure 17, the normalized image coordinate point P_n can be expressed as

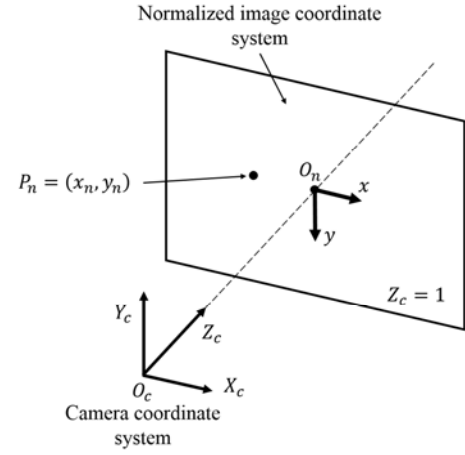$$s\begin{bmatrix} x_n \\ y_n \\ 1 \end{bmatrix} = \begin{bmatrix} X_c \\ Y_c \\ Z_c \end{bmatrix} \tag{17}$$

Figure 17. *Normalized image coordinate system.*

The actual perspective projection point P_n is observed at a shifted point $P_n{'}$ under the influence of lens distortion, as shown in Figure 18. This change is calculated based on the following lens distortion model:
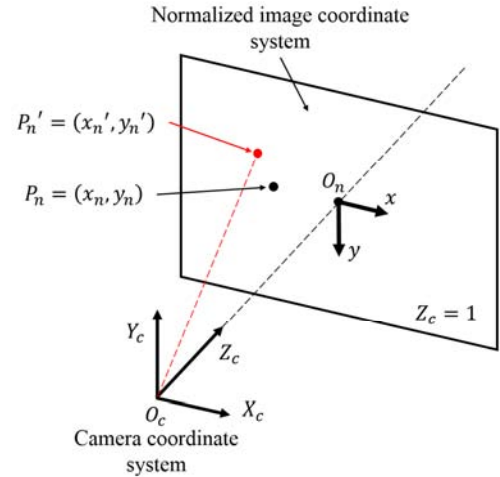

Figure 18. *Lens distortion in image coordinate system.*

$$K = (k_1, k_2, k_3, p_1, p_2) \tag{18}$$

$$x_n' = x_n(1 + k_1r^2 + k_2r^4 + k_3r^6) \\ + \{2p_1x_ny_n + p_2(2x_n^2 + r^2)\}$$

$$y_n' = y_n(1 + k_1r^2 + k_2r^4 + k_3r^6) \\ + \{p_1(2y_n^2 + r^2) + 2p_2x_ny_n\}$$

$$(r^2 = x_n^2 + y_n^2) \tag{19}$$

where K is a parameter that expresses the degree of distortion, k_1, k_2, and k_3 represent the strain in the radial direction, and p_1 and p_2 represent the strain in the circumferential direction. The point $P_n{'}$ in the normalized image coordinate system expresses the position with the center of the image plane as the origin. In contrast, in the image coordinate system, the upper

left side of the image is used as the origin and expressed in pixel units, as shown in Figure 19.
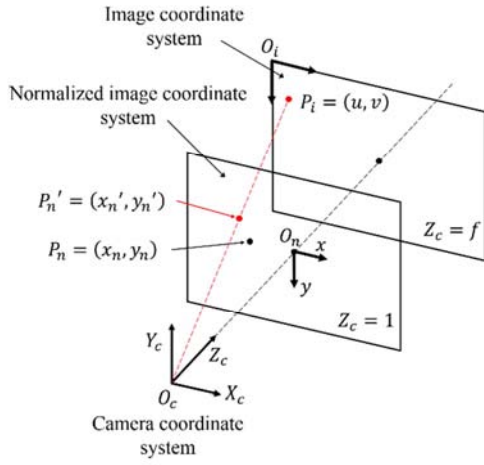

Figure 19. *Relationship between normalized image coordinate system and image coordinate system.*

The point P_i in the image coordinate system can generally be calculated as follows:

$$\begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_n' \\ y_n' \\ 1 \end{bmatrix} = F \begin{bmatrix} x_n' \\ y_n' \\ 1 \end{bmatrix} \quad (20)$$

where F is called the camera matrix as described at the beginning of this section, (f_x, f_y) represents the focal length in pixel units, and (c_x, c_y) represents the optical center (principal point) in pixel units. Based on the above, it can be expressed as follows:

$$s \begin{bmatrix} u \\ v \\ 1 \end{bmatrix} = \begin{bmatrix} fx & 0 & cx \\ 0 & fy & cy \\ 0 & 0 & 1 \end{bmatrix} [R|t] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} = F[R|t] \begin{bmatrix} X_w \\ Y_w \\ Z_w \\ 1 \end{bmatrix} \quad (21)$$

3.2. Estimating of Internal Parameter Using Camera Calibration

Calibration of the camera is used to estimate the focal length of the camera, deviation of the center of the image, and distortion of the lens. As shown in Figure 20, by using a chessboard of a plane with known coordinate information, internal parameters related to the camera are obtained using the correspondence between the two-dimensional points and three-dimensional points as follows.
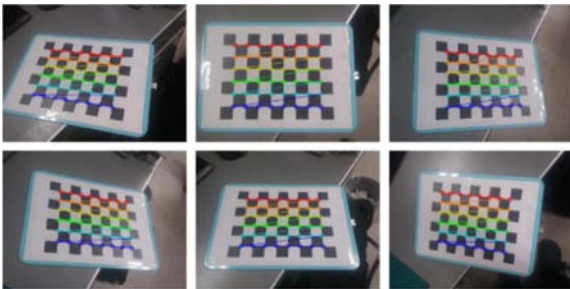

Figure 20. *Camera calibration using chess board.*

$$F = \begin{bmatrix} f_x & 0 & c_x \\ 0 & f_y & c_y \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} 492.15 & 0 & 322.71 \\ 0 & 492.19 & 179.07 \\ 0 & 0 & 1 \end{bmatrix} \quad (22)$$

(unit: pixel)

$$K = (k_1, k_2, k_3, p_1, p_2)$$

$$= (0.320, -2.02, 4.60, -0.001, 0.002) \quad (23)$$

Based on the internal parameters F and K estimated above, the prepared image (640×480) is corrected, and the result is shown in Figure 21. From Figure 21 (b), it can be confirmed that the image is calibrated.

(a) Image before calibration   (b) Image after calibration
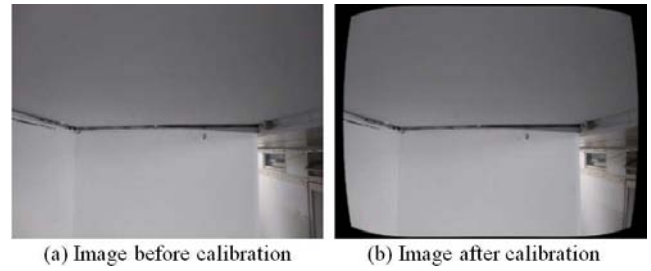
Figure 21. *Result of calibration.*

4. Experiment in Real Environment

In this study, the goal of the bridge inspection robot's location estimation is to recognize the position of the feature point necessary for location estimation as three-dimensional coordinates, and thereby improve the reliability of location estimation. Therefore, by combining the existing Harris corner detection method and the Hough transform to reinforce the problem, the feature points located in the camera shooting direction is detected, and the position is specified as three-dimensional coordinates. To investigate whether the results described in the previous section can be reproduced in the actual environment and how reliably they can be obtained, the experiment was conducted in the environment shown in Figures 22 and 23.
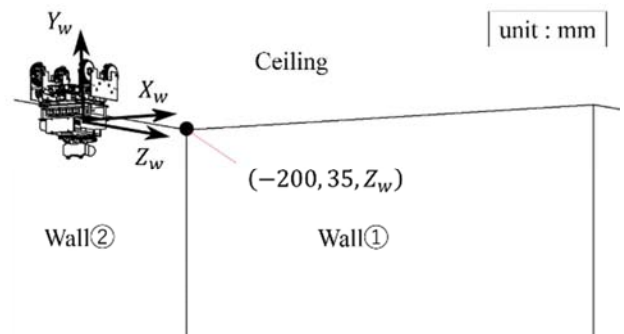

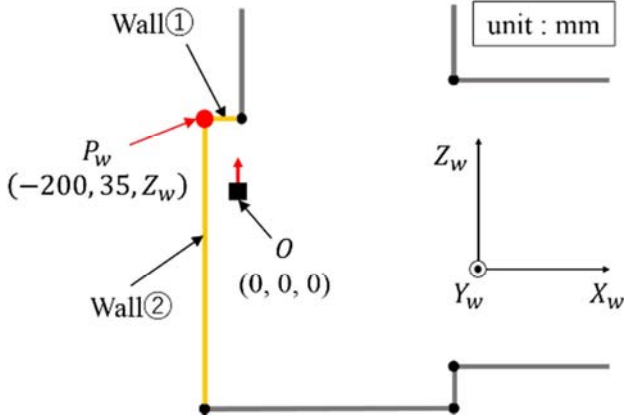**Figure 22.** *Experimental environment.*

Figure 23. *Simplified diagram of experimental environment.*

$$= \begin{pmatrix} C_\beta & 0 & S_\beta \\ 0 & 1 & 0 \\ -S_\beta & 0 & C_\beta \end{pmatrix}. \qquad (24)$$

In this experiment, by providing initial conditions, the position of the target feature point P_w was converted from the image coordinate system to the world coordinate system and compared with the position of the actual feature point. Regarding the conditions of the experimental environment, the calculation was performed with $\beta = 0$ (rad), $X_w = -200$ (mm), and $Y_w = 35$ (mm). In addition, coordinates Z_w on the Z-axis at the feature point P_w were considered at five locations: 500, 600, 700, 800, and 900 mm. By image processing using the method proposed in the previous section, the coordinates of feature points in pixel units in the image coordinate system were detected based on the image at each position.

As described above, experiments were conducted for the detection of each feature point at each position of Z_w, and the results are shown in Table 3.

The pixel coordinates (u', v') of the obtained image coordinate system are compared with the pixel coordinates (u, v) of the feature points of the actual coordinates in the image coordinate system, and the results are shown in Figure 24.

As shown in Figure 22, this experimental environment has been used for a robot running on the ceiling surface, where the rotation matrices around X_w, Y_w, and Z_w are defined as R_X, R_Y, and R_Z and represent the pitch angle, yaw angle, and roll angle in the camera, respectively. In addition, as shown in Figure 23, the angles α and γ in the pitch and roll directions are treated as zero, and thereby, the rotation matrix can be expressed as

$$R = R_Z(0) \times R_Y(\beta) \times R_X(0)$$

Table 3. *Coordinates of feature points in the image coordinate system.*

Z_w (mm)	Real pixel coordinates		Estimated pixel coordinates	
	u (pixel)	v (pixel)	u' (pixel)	v' (pixel)
500	123.61	213.84	112.0	213.5
600	156.55	208.1	151.5	207.0
700	180.34	203.95	189.55	201.36
800	198.28	200.82	206.25	203.76
900	212.25	198.38	214.0	202.0

In the figure, the red circle is the feature point in real coordinates, and the blue circle is the position of the feature point obtained from the image processing. The feature points P_w' in the world coordinate system obtained by image processing can be calculated as shown in Table 4.

Table 4. *Coordinates of feature points in the world coordinate system.*

Z_w (mm)	Estimated three-dimensional coordinates		
	X_w' (mm)	Y_w' (mm)	Z_w' (mm)
500	-200.0	32.675	467.137
600	-200.0	32.621	574.911
700	-200.0	33.475	739.208
800	-200.0	42.39	845.201
900	-200.0	42.177	905.428

The comparison of the coordinates of the feature points (X_w', Y_w', Z_w') obtained by calculation and the coordinates of the actual feature points (X_w, Y_w, Z_w) are shown in Figure 25.

Table 5. *Comparison between real and estimated coordinates in world and image coordinates.*

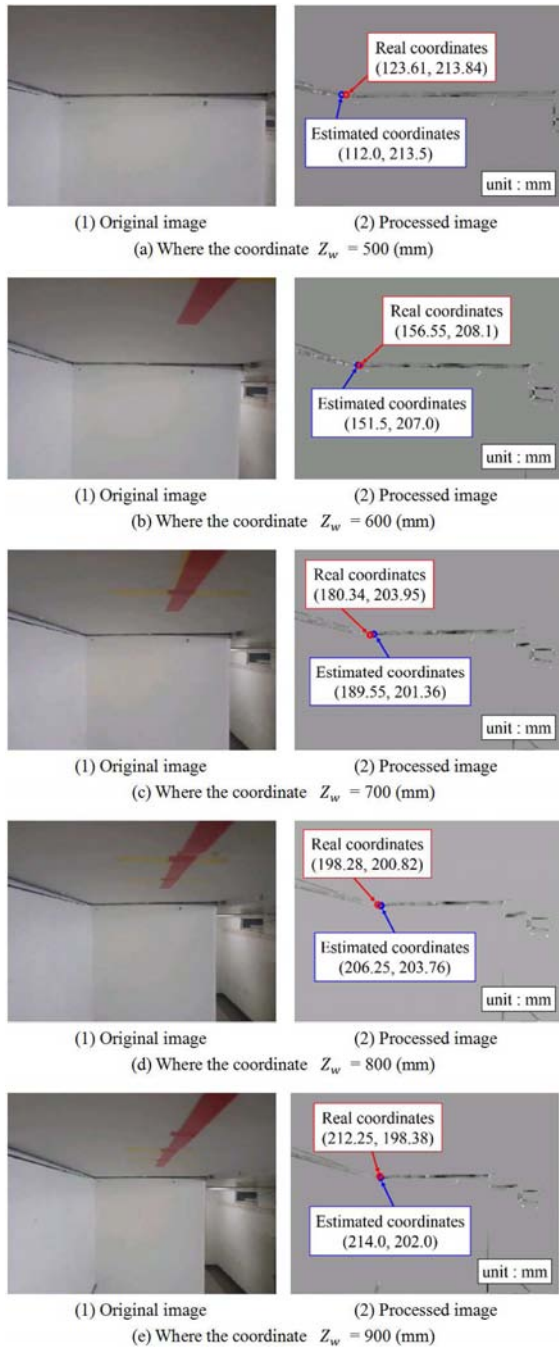| z_w (mm)	Error in the world coordinate system (mm)	Error in the image coordinate system (pixel)
500	39.945	11.615
600	25.202	5.168
700	39.237	9.571
800	45.802	8.497
900	8.999	4.021

Figure 24. *Comparison between real and estimated coordinates in image coordinates.*
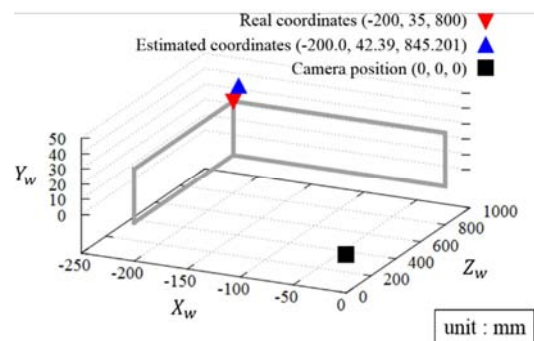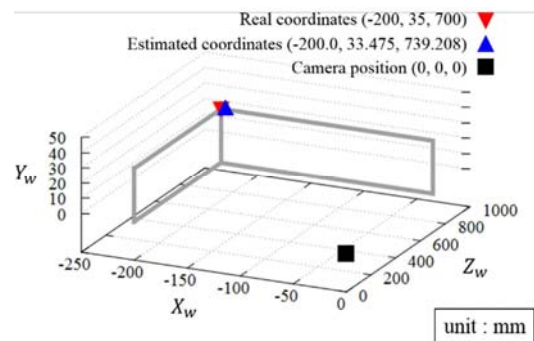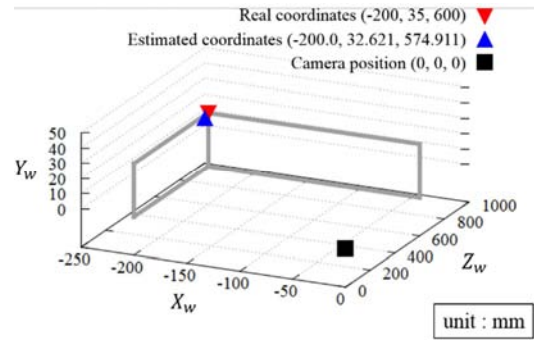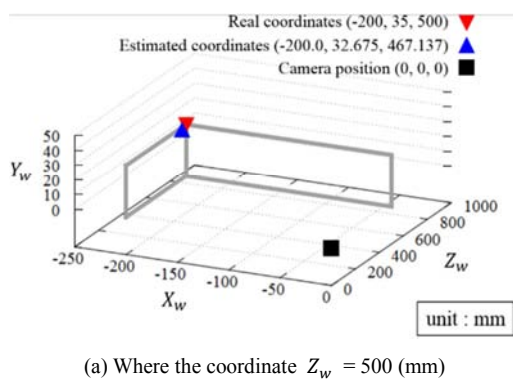
Figure 25. *Comparison between real and estimated coordinates in world coordinates.*

The error distances between the two feature points obtained by comparing the position of the feature point acquired by the image processing with the feature point of the actual

coordinate in the image coordinate system and world coordinate system are shown in Table 5.

The experimental results demonstrate that we were able to identify the corners from the images captured by the smartphone camera and detect the feature points from them. In addition, by converting the pixel coordinates of the feature points in the detected image coordinate system into world coordinates, it was confirmed that the position of the feature points was specified as coordinates. The results of comparing the feature points identified from the respective positions with the feature points of actual coordinates indicated an error of 7.774 pixels on average in the image coordinate system and 31.837 mm on average in the world coordinate system. However, in the lower part of the bridge, which is the actual experimental environment, the distance between the feature points is within 100.0 mm, and if the error is within 21.3 pixels, it can be recognized as the same. Therefore, the results of this experiment were within the acceptable error range. Therefore, it was confirmed that the method proposed in this study could be useful.

5. Conclusion

In this study, image processing was performed using a smartphone camera, and corners were detected as feature points. Based on the data obtained, and the feature points of the actual coordinates, a quantitative evaluation was performed.

The image processing was performed using Harris corner detection, which is a conventional corner detection method. However, Harris corner detection has problems, including erroneous recognition of other feature points with luminosity changes such as scratches and erroneous recognition of oblique lines. Therefore, to overcome these problems, we focused on geometric features such as the corners of the structures and attempted to capture the target feature points. For this purpose, straight lines were detected by acquiring the contours using the Canny edge detection method and performing the Hough transform.

From the perspective projection transformation of the perspective projection camera model, the relational expression between the world coordinate system and the image coordinate system was calculated. By performing perspective projection conversion on the pinhole camera model using a smartphone camera, the relational expression between the world coordinate system and image coordinate system was derived, and the coordinates obtained by image processing and actual coordinates were compared. As a result of comparing and evaluating the estimated and actual coordinates, it was found that the proposed method showed high accuracy with an error within the acceptable range.

In the future, the shooting position of the camera will be estimated using the robot position information obtained from the LiDAR mounted on the bridge inspection robot BIREM-IV-P. In addition, the coordinates of the feature point detected from the position will be combined with the feature point information from the LiDAR sensor. Thus, errors in the position coordinates of the feature points can be reduced, and the accuracy of localization can be improved.

References

[1] Ronny Salim Lim, Hung Manh La, Zeyong Shan and Weihua Sheng, Developing a Crack Inspection Robot for Bridge Maintenance, IEEE International Conference on Robotics and Automation (2011), pp. 6288-6293.

[2] A. Mazumdar and H. H. Asada, Mag-Foot: A steel bridge inspection robot, 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems, St. Louis, MO, 2009, pp. 1691-1696.

[3] W Shen, J Gu, Y Shen Permanent magnetic system design for the wall-climbing robot. Applied Bionics and Biomechanics 3: (2006), pp. 151-159.

[4] S. T. Nguyen and H. M. La, Development of a Steel Bridge Climbing Robot, 2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), Macau, China, 2019, pp. 1912-1917.

[5] P. Ward et al, Climbing robot for steel bridge inspection: design challenges, 9th Austroads Bridge Conference 2014.

[6] Yoshito Okada and Takayuki Okatani, Development of UAV with Passive Rotating Spherical Shell for Bridge Inspection and its Evaluation of Inspection Capability in Real Bridges, Journal of the Robotics Society of Japan, Vol. 34 (2016), No. 2, pp. 119-122.

[7] R. Fujisawa, K. Umemoto, M. Tanaka, N. Sato, N. Nagaya, and M. Katsuyama, Development, Bridge Inspection and Operation of Araneus - Winch Wiring Robot System, J. JSCE. Ser. F4, (Construction and management) special issue, Vol. 73 (2017), No. 1, pp. 26-37 (In Japanese).

[8] K. Nakata, K. Umemoto, K. Kaneko, F. Ryusuke, Development and Operation of Wire Movement Type Bridge Inspection Robot System ARANEUS, Kalpa Publications in Engineering, Vol. 3 (2020), pp. 168-174.

[9] H. Kajiwara, N. Hanajima, K. Kurashige, Y. Fujihira, Development of Hanger-Rope Inspection Robot for Suspension Bridges, Journal of Robotics and Mechatronics, Vol. 31 (2019), No. 6, pp. 855-862.

[10] Y. Okada and T. Okatani, Development of UAV with Passive Rotating Spherical Shell for Bridge Inspection and its Evaluation of Inspection Capability in Real Bridges, Journal of the Robotics Society of Japan, Vol. 34, No. 2 (2016), pp. 119–122.

[11] N. Kanehira, M. Hirai, S. Kashimoto, and S. Echigo, Bridge Mounting Multicopter for Bridge Inspection, Kawada technical report, Vol. 36 (2017), pp. 22–28.

[12] M. Nakao, E. Hasegawa, T. Kudo, N. Sawasaki, Development of a Bridge Inspection Support Robot System Using Two-Wheeled Multicopters, Journal of Robotics and Mechatronics, Vol. 31 (2019), No. 6, pp. 837-844.

[13] K. Hidaka, D. Fujimoto, K. Sato, Autonomous Adaptive Flight Control of a UAV for Practical Bridge Inspection Using Multiple-Camera Image Coupling Method, Journal of Robotics and Mechatronics, Vol. 31 (2019), No. 6, pp. 845-854.

[14] Y. Takada, S. Ito, and N. Imajo, Development of a bridge inspection robot capable of traveling on splicing parts, Inventions 2017, Vol. 2, Issue 3, 22.

[15] W. Lee, and S. Hirose, Contacting Surface-Transfer Control for Reconfigurable Wall-Climbing Robot Gunryu III, Journal of Robotics and Mechatronics, Vol. 25 (2013), No. 3, pp. 439-448.

[16] Rui Wang, Youhei Kawamura, An Automated Sensing System for Steel Bridge Inspection Using GMR Sensor Array and Magnetic Wheels of Climbing Robot, Journal of Sensors, Vol. 2016, pp. 1-15.

[17] H. M. La, T. H. Dinh, N. H. Pham, Q. P. Ha, and A. Q. Pham, Automated Robotic Monitoring and Inspection of Steel Structures and Bridges, Robotica, pp. 1-21 (2016).

[18] Y. Matsumura, K. Kawamoto and Y. Takada, Development of a Compact Wall-Climbing Robot Capable of Transitioning among Floor, Vertical Wall and Ceiling, Journal of the Robotics Society of Japan, Vol. 37 (2019), No. 6, pp. 514-522.

[19] H. Song, J. Nakahama and Y. Takada, Establishing a Localization Method for Mobile Robots to Support Bridge Inspection, The Symposium on Evaluation and Diagnosis, Vol. 18 (2019), pp. 134-139.

[20] H. Moravec, Rover visual obstacle avoidance, Proceedings of the 7th international joint conference on Artificial intelligence, Vol. 2 (1981), pp. 785-790.

[21] S. M. Smith and J. M. Brady, SUSAN -A new approach to low level image processing, International Journal of Computer Vision, Vol. 23, No. 1 (1997), pp. 45-78.

[22] M. Trajkovic, Hedley, M. Fast Corner Detection. Image and Vision Computing. 1998, Vol. 16, pp. 75–87.

[23] E. Rosten, T. Drummond, Fusing Points and Lines for High Performance Tracking, In Proceedings of the 10th IEEE International Conference on Computer Vision, Beijing, China, pp. 17–21 October 2005.

[24] E. Rosten, G. Reitmayr, and T. Drummond, Real-Time Video Annotations for Augmented Reality. Adv. Vis. Comput. 2005, Vol. 3804 (2005), pp. 294–302.

[25] C. Harris, and M. Stephens, A Combined Corner and Edge Detector, In Proceedings of the 4th Alvey Vision Conference (1988), pp. 147-151.

[26] J. Canny, A Computational Approach to Edge Detection, IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 8 (1986), pp. 679-698.

[27] P. V. C. Hough, Machine Analysis of Bubble Chamber Pictures, Proc. Int. Conf. High Energy Accelerators and Instrumentation, 1959.