SICE JOURNAL OF CONTROL, MEASUREMENT, AND SYSTEM INTEGRATION
2022, VOL. 15, NO. 2, 145–161
https://doi.org/10.1080/18824889.2022.2103631

RESEARCH ARTICLE

OPEN ACCESS

UAV manipulation by hand gesture recognition

Shoichiro Togo and Hiroyuki Ukida

Graduate School of Advanced Technology and Science, Tokushima University, Tokushima, Japan

ABSTRACT

In this study, we discuss a unmanned aerial vehicle operation system by recognizing human gestures. Here, we focus on both dynamic and static gestures, such as moving the right hand repeatedly or holding it in a certain position. And, we propose two methods, one is a feature-based (FB) method to detect the position of the right hand in an image and identify the gesture form features estimated by FFT, and the other is a machine learning (ML) method to detect the position of the right hand in an image and identify the gesture by the framework of the ML. In experiments, we compare the results of gesture recognition by each method. As a result, the recognition rate of the FB method is higher than that of the ML method under the conditions assumed in the FB method. But, in other cases, the ML method is higher than that of the FB method. The ML method is also effective in terms of extensibility, such as adding more types of gestures.

ARTICLE HISTORY
Received 25 October 2021
Revised 20 May 2022
Accepted 14 July 2022

KEYWORDS
Gesture recognition; UAV manipulation; feature extraction; hand region estimation; fast Fourier transform; machine learning; OpenPose; long short-term memory

1. Introduction

In recent years, unmanned aerial vehicle (UAV) has become increasingly popular for the purpose of aerial photography and inspection. In addition to the above purposes, UAV is also used in many other fields such as transportation, rescue, surveying, and so on. It is expected to permeate a part of our daily lives in the future. For this reason, a man machine interface that allows anyone to easily operate UAV or robot is necessary.

One of the most common ways to operate an unmanned robot is to use a controller. However, the operator must be in possession of the controller and must be proficient in its use. On the other hand, if a robot can be controlled using gestures, which is a “natural interface” for humans to operate with natural and intuitive motions, anyone can operate it easily. From this point of view, we have discussed a rover robot manipulation system using human gestures [1].

We can consider two types of gestures. One is static gestures, and the other is dynamic gestures. Static gestures are gestures of the shapes of human hands or the postures of the human bodies. It can be obtained from a one-shot image of a camera. On the other hand, dynamic gestures are gestures that use the movements of the human hands or bodies as time-series data. In general, static gestures enable fast recognition because the time required to acquire the necessary data is short. Dynamic gestures require more time to be recognized because they obtain data of movements during a certain period of time, but it is possible to use a large number of gestures by combining movements. In this study, we deal with dynamic gestures by treating the case of stationary human motion as one of them.

Many methods have been investigated for gesture recognition. Conventional methods often use self-designed unique features to detect body parts for gesture recognition from camera images or to recognize gestures from their shapes and movements. This method is referred to as the “feature-based method (FB method)” in this paper. On the other hand, as will be shown later in Section 2, many methods for detecting body parts and for gesture recognition using machine learning (ML) have been studied recently. We call this method “ML method.”

In this study, we investigate the method of a UAV operation by recognizing dynamic gestures. Previous studies have discussed the FB method [2]. In this method, the position of the human hand is extracted from the camera image using colour features, and gesture recognition is performed using motion features based on the Fourier transform. In this paper, we also investigate a gesture recognition method using ML method with OpenPose [3] and long short-term memory (LSTM) [4]. Here, we compare the gesture recognition of FB method and ML method, and discuss the features of each method.

In the following, we show related work and aim of this study in Section 2. Section 3 shows the definition of gestures used in this study. In Section 4, we propose gesture recognition methods based on FB method and ML method. In Section 5, we show the method and results

CONTACT Hiroyuki Ukida ukida@tokushima-u.ac.jp

of preliminary experiments to implement the proposed method. In Section 6, we show the comparison results and effectiveness of the two proposed methods. Finally, in Section 7, we summarize our research and show future works.

2. Related work and aim of this study

In the problem of manipulating a robot by gesture recognition, many methods of detecting and recognizing the shape of a human hand have been studied in order to indicate the direction of movement of the robot or the direction of an object. Although Refs. [5,6] are not robot operations, in Ref. [5], a method for indicating the position of an object using a pointing gesture is discussed, and in Ref. [6], a method for identifying complex hand shapes is discussed. In Ref. [7], for a mobile robot equipped with a fisheye camera, a method is proposed to indicate the direction of movement to the robot by the direction of operator's arm. In this method, an operator wears a marker of a characteristic colour to facilitate detection of the hand position. In Ref. [8], authors propose a method to move a hand robot in the same way as a human hand by detecting the shape of the hand from camera images in two directions. Although these methods may include continuous motion, they are essentially static gesture recognition.

On the other hand, a method for recognizing dynamic gestures of moving hands and arms has been proposed. In Ref. [9], gestures are recognized using dynamic programming (DP) matching, which features temporal changes in the hand's centre of gravity position, orientation, and area. In Ref. [10], gesture recognition is performed using a support vector machine (SVM) by calculating the features of periodic motion using fast Fourier transform (FFT). In Ref. [11], the authors propose a gesture recognition method that uses pre-made templates for both arm movements.

In recent years, many methods for recognizing dynamic gestures caused by hand and body movements have been investigated using ML frameworks. In addition, UAVs are used as operation targets. In Ref. [12], a method for identifying hand movements and controlling UAV flight using deep neural network (NN) is proposed. This study uses the leap motion controller, which is a device specialized for hand motion detection. Reference [13] recognizes a dynamic gesture of moving both hands by the ML method using partial colour images and optical flow motion from camera images mounted on a UAV as input data. The recognition is done by pose-based convolutional neural network. Reference [14] is also a gesture recognition method using camera images mounted on a UAV. This study proposes a method to identify static gestures using convolutional neural network (CNN) and dynamic gestures using Tiny-YOLOv2 (YOLO: You Only Look Once). In Ref. [15], human detection using OpenPose and gesture recognition by hand shape using CNN are studied.

In this study, we initially investigated a method to control a rover robot with dynamic gestures [1]. In particular, we devised features specific to the fact that the gesture is a repetitive motion of the right hand and proposed a recognition method for it. Microsoft Kinect [16] was also used for the detection of human motion. After this study, we changed the object of operation from a rover robot to a UAV. Since it is difficult to mount a device such as Kinect on a UAV, a method using images from a single camera is considered.

Currently, UAVs are often operated using controllers. In order to operate the UAV according to the operator's intention, the operator must be proficient in the operation of the controller. The purpose of this paper is to make it possible for anyone to operate a UAV easily and intuitively. As the operation of UAV, we consider the following two types:

A. To move the UAV (forward and backward, left and right, up and down, etc.).
B. To give a command to the UAV to perform some tasks (e.g. taking aerial photos, loading/unloading cargo).

In this study, we aim to be able to perform both A and B operations in the future by increasing the number of gestures. However, in this paper, since we focus on the investigation of gesture recognition methods, we will only deal with A as the operation of the UAV.

On the other hand, one of the problems of such gesture recognition is safety, since the UAV's camera needs to capture the human, and both of them are in close proximity. In this study, we use a small UAV as shown in Section 5, which can fly safely at a distance of about 1 m from a human. On the other hand, larger UAVs need to be kept farther away for safety. In this case, gesture recognition becomes difficult because the human in the image is captured in a small size. However, using a camera with a zoom function will make gesture recognition possible.

Since it is difficult for a UAV to capture a human while moving, it is assumed that the UAV will be hovering when it recognizes the gesture. This allows us to assume that the camera is almost stationary. In order to make it easy for anyone to operate the UAV, we assume that the human gesture is performed for a few seconds, during which time the UAV recognizes the gesture from images captured by the hovering camera.

The gesture recognition considered in this study is intended for UAV operation. It is desirable to give commands to the UAV continuously through gesture recognition. For this purpose, it is necessary to search for the operator so that the operator can be captured in the camera image for gesture recognition again after the UAV has performed a certain action. Therefore, we propose an operator detection method, and a practical UAV operation system based on gesture recognition is constructed and evaluated in this study. In other studies, gesture recognition methods using images captured by UAVs have been evaluated, but practical operation systems based on gestures have not been proposed.

From the above, the flow of UAV operation by gesture recognition in this study can be summarized as follows:

1. The UAV recognizes gestures and receives commands.
2. It performs actions corresponding to commands.
3. It searches for the operator to recognize the next gesture.

This study adopts repeated up–down or left–right motions of the right hand as natural human gestures. As the gesture recognition method, we consider the FB method, which is a hand detection based on colour information and recognition based on features using the Fourier transform [2]. This is an effective method for obtaining features of repetitive motion. In addition, this paper discusses the ML method, too. By using the ML method, it is possible to relax the restrictions of the conditions and environment for hand detection. Here, we also consider human pose detection using OpenPose [3] and gesture recognition methods using LSTM [4]. LSTM is an effective method for identifying time-series data such as gesture motions. In this study, we construct our own ML framework that is suitable for the gesture movements to be handled. Although methods similar to the FB method and ML method have been used in other studies, this study makes its own modifications to improve the recognition rate of gestures. In this study, the results of gesture recognition by each method are compared, and the effectiveness of each method is discussed.

3. Definition of gestures

In this section, we describe the gestures used in this study. First, in Figure 1, the point near the chest is called "the reference point," and the area within a certain range centred on the reference point is called "the reference region." It is also divided into four regions, each 90° upper, under, left, and right, centre on the reference point. "The detection region" consists of four regions: left, right, under, and reference. The reason for not using the upper area is that the face and neck are located in the upper region, in which the hand has similar colour information.

Next, we describe the motion of the gestures. The gesture is initiated when the hand enters the reference region. The gesture is a combination of the detection region and the hand motion. Figure 2 shows examples of the hand motions. The hand motion is one of the static motion or dynamic motion. Static motion is the motion of moving the hand from the reference region to a detection region and stopping the hand at a certain point, as shown in Figure 2(a). On the other hand, the dynamic motion repeats the motion of moving the hand from the reference region to the detection region and back to the reference region again, as shown in Figure 2(b). The gestures to be used in this study are seven in total: three static motion gestures in left, right, and under detection regions and three dynamic motion gestures to the left, right, and under of detection regions, and one static motion gesture in the reference region. However, the operator should wear long sleeves and hide the non-gesturing left hand behind the body when gesturing.

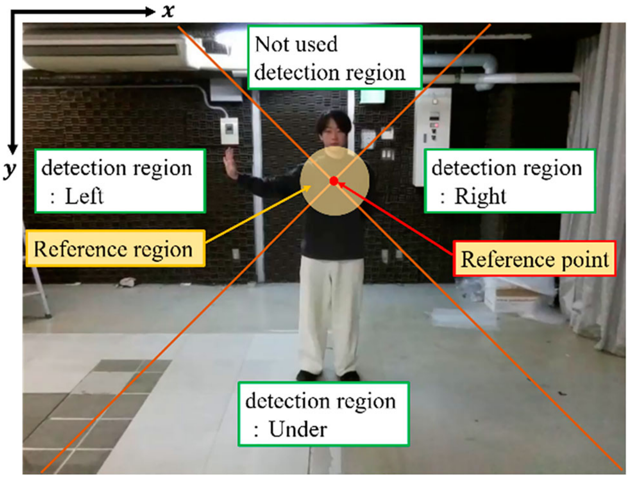


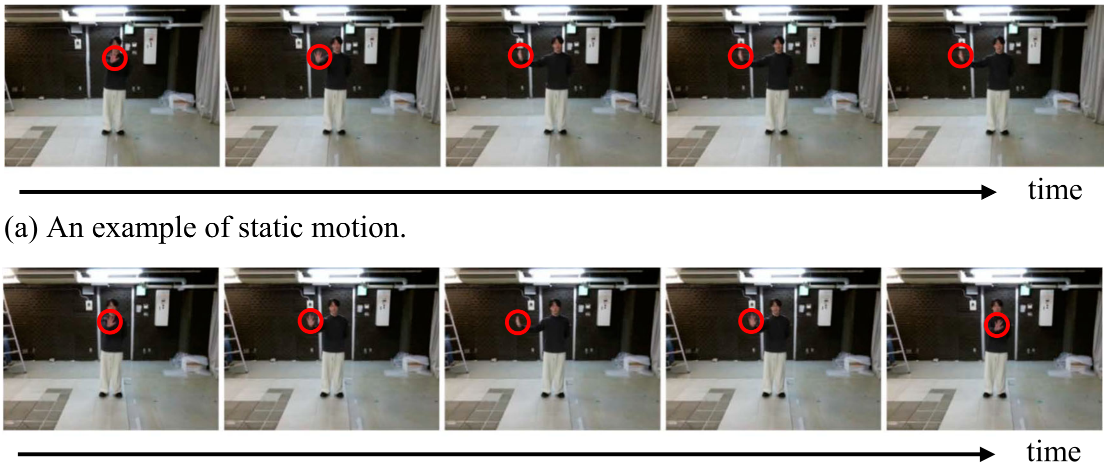
Figure 1. Reference point, reference region, and detection regions.

After the gesture is recognized, the UAV executes the motion command corresponding to the gesture (execution of UAV motion commands). Motion commands of the UAV corresponded to each gesture are predetermined as shown in Table 1. In the following, when the detection region is on the right and the detection motion is static motion, it is abbreviated as "right static."

4. Gesture recognition methods

We show the gesture recognition method using the FB method in Section 4.1. And, we show the gesture recognition method using the ML method in Section 4.2. In addition, the purpose of this study is to manipulate a UAV using gestures. As described in the previous section, after the UAV recognizes a gesture, it performs a motion associated with the gesture. Then, the gesture recognition process and the UAV operation process are repeated again until the stop gesture is performed.

sHowever, the UAV does not always return to the front of the operator after performing a motion related to a gesture. There is a possibility of losing the operator. In Section 4.3, we describe the process of re-detection



(a) An example of static motion.

(b) An example of dynamic motion.

Figure 2. Types of gestures.

Table 1. Gestures and related UAV motion commands.

Gesture	Right	Left	Under	Reference
Static	Climb	Descent	Forward	Landing
Dynamic	Move right	Move left	Backward	(Not assigned)

method for the operator. Therefore, our proposed method repeats "gesture recognition process," "UAV motion process," and "operator detection process."

4.1. Gesture recognition by FB method

In the gesture recognition process, a NN is used to estimate the hand region, and the frequency component of the hand position time series is used to determine which gesture is being performed from the time-series data of the estimated hand region coordinates. This section is divided into two parts: the hand region estimation method and the gesture judgment method. First, the hand region estimation method is shown in Step 1 through Step 3 below.

Step 1. The image is acquired from a camera, and the human region is detected as a rectangle using histograms of oriented gradients (HOG) features and SVM. Here, the two-dimensional coordinates of the starting point of the detected human region (the upper left side of the rectangle) are P_{xi} and P_{yi}, respectively, the height and width of the detected size are H_i and W_i, respectively, and i represents the ith image (Figure 3). To detect human regions, the trained detector of the image processing library OpenCV [17] is used.

Step 2. Determine the reference point near the chest. A human being is said to be 7 heads tall at 12 years old and elderly, and 8 heads tall at adult. From this reason, the operator is assumed to be 7.5 heads tall, and near the chest of the person is assumed to be 1.7 heads from the top of head. From this assumption, the two-dimensional coordinates B_{xi} and B_{yi} of the reference point are calculated using Equations (1) and (2). From these equations, the position of the reference point (B_{xi}, B_{yi}) is the lower end of 1.7 heads from the top of the 7.5 heads.

$$B_{xi} = P_{xi} + \frac{W_i}{2}, \tag{1}$$

$$B_{yi} = P_{yi} + H_i\left(\frac{1.7}{7.5}\right). \tag{2}$$

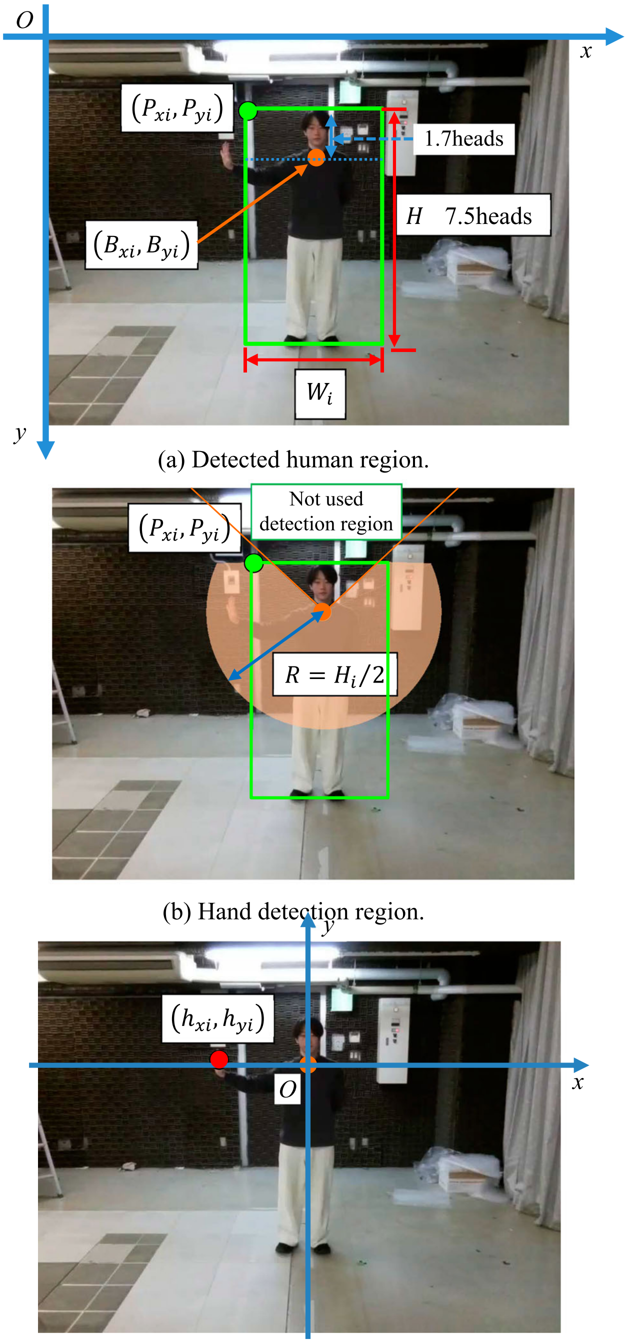
Step 3. Estimate the hand region using NN by following the steps from Steps 3.1 to 3.4. The NN is consisted by three layers. The input layer has six nodes and inputs colour information for each pixel, the intermediate layer has 20 nodes, and the output layer has two nodes and outputs either the hand region or other region. The colour information consists of six values: H, S, V, Y, Cb, and Cr.

Step 3.1. The range of motion of the hand can be fixed. Therefore, we set "the hand detection region," and the right hand position is searched with in this region. We assume that the hand detection region satisfies following conditions. The hand detection region is shown in Figure 3(b):

- Inner area of a circle whose centre is (B_{xi}, B_{yi}) and its radius $R = H_i/2$.
- Eliminate the area where y-coordinate is upper than P_{yi}.
- Eliminate the area of "Not used detection region" where it includes human's head in Figure 1.

Step 3.2. Convolve the input image with a 5 by 5 pixels size Gaussian filter and create 1/4 times image by down sampling. By using this small image, we can reduce the search area of the hand region and speed up the process.

Step 3.3. Distinguish the hand region in the hand detection region of a 1/4 times image by NN. After that, we perform a four-connected labelling of the region distinguished as the hand region, and the centre



(a) Detected human region.

(b) Hand detection region.

(c) Transformed right hand coordinates.

Figure 3. Hand position detection.

of gravity of the region above a certain threshold is used as the hand region coordinate.

Step 3.4. In the original image, the coordinates transformed into the reference point centred coordinate system with the reference point as the origin and the right and upward direction as the positive direction of x and y, respectively, are the final hand region estimated 2D coordinates (h_{xi}, h_{yi}).

Next, when the estimated hand region coordinates (h_{xi}, h_{yi}) enter the reference region, the gesture judgment is started. Repeat Steps 1–3 above until the gesture judgment is started. The gesture judgment method is shown in Steps 4–8 below.

Step 4. Repeat Steps 1–3 to save eight estimated hand region coordinates (h_{xi}, h_{yi}) in chronological order. Note that the reason for setting the amount of data to 8 is based on the results of preliminary experiments of gesture recognition considering the speed of hand movement in dynamic gestures (about 2.5 s per round-trip) and the process time of the right hand coordinate (h_{xi}, h_{yi}) detection (167 ms per one image).

Step 5. Perform FFT on each coordinate of the eight saved data. Calculate the values of the real part $\mathrm{Re}(n)$ in Equation (3) and the imaginary part $\mathrm{Im}(n)$ in Equation (4), and then calculate the amplitude spectrum $\mathrm{aps}(n)$ from Equation (5). Due to the symmetry of the FFT, the computational complexity is half of the total number of samples:

$$\mathrm{Re}(n) = \frac{1}{N}\sum_{k=0}^{N}\left\{x(k)\cos\left(\frac{2\pi nk}{N}\right)\right\} \quad (0 \le n \le 3), \tag{3}$$

$$\mathrm{Im}(n) = \frac{1}{N}\sum_{k=0}^{N}\left\{-x(k)\sin\left(\frac{2\pi nk}{N}\right)\right\} \quad (0 \le n \le 3), \tag{4}$$

$$\mathrm{aps}(n) = \sqrt{\mathrm{Re}(n)^2 + \mathrm{Im}(n)^2} \; (0 \le n \le 3), \tag{5}$$

where N is the total number of data, n is the frequency, and $x(k)$ is the kth sampled data.

Step 6. Obtain the evaluation value E. As shown in Equation (6), the evaluation value E is the maximum value among the amplitude spectra $\mathrm{aps}_x(n)$ and $\mathrm{aps}_y(n)$ of each coordinate added together for each frequency n.

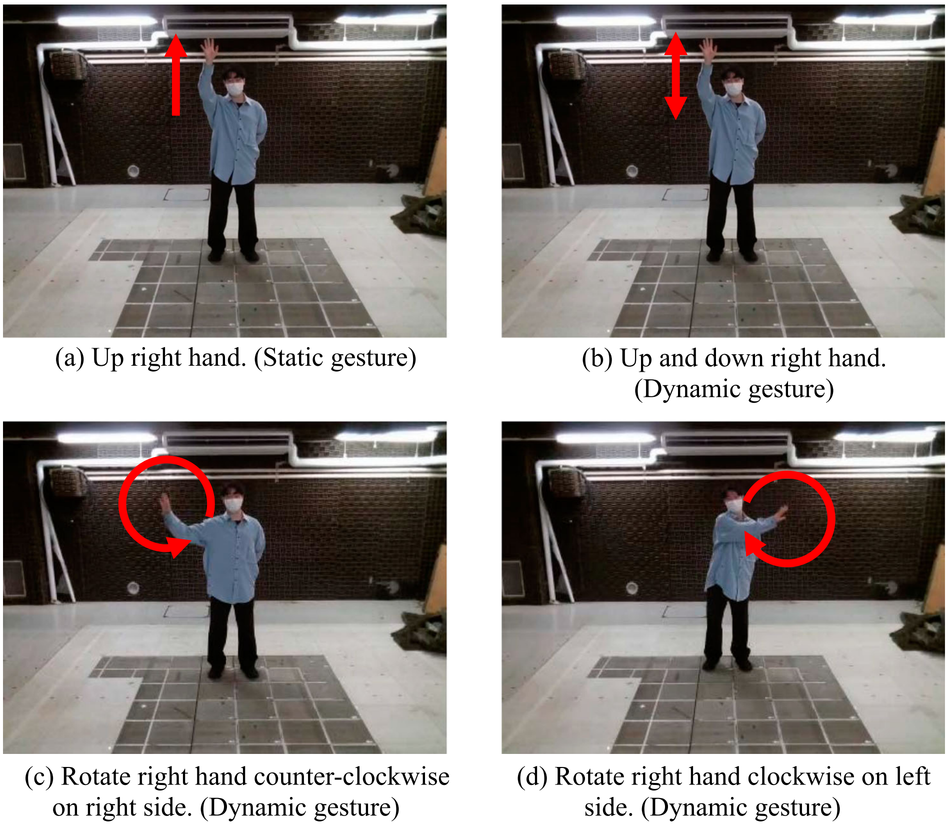
$$E = \max_n\{\mathrm{aps}_x(n) + \mathrm{aps}_y(n)\}. \tag{6}$$

Step 7. From the evaluation value E, the threshold value is used to judge which motion corresponds to "static motion," "dynamic motion," or "not a gesture." The threshold value is set through preliminary experiment in Section 5.1.

Step 8. Make a final judgment on the gesture. If the evaluation value is dynamic motion or static motion, if all eight data are in the reference region or in the same detection region, it is the dynamic gesture of that detection region. However, in the case of the same detection region, the reference region is not included in the count; this case is the static gesture of that detection region. If there is even one region that is different, it is not a gesture. If it is judged that it is not a gesture, delete the oldest one of the eight estimated hand region coordinate data, return to Step 1 and add one new hand region coordinate data for the gesture judgment.

4.2. Gesture recognition by the ML method

In the FB method shown in Section 4.1, when the operator's clothing changes, the right hand may not



(a) Up right hand. (Static gesture)

(b) Up and down right hand. (Dynamic gesture)

(c) Rotate right hand counter-clockwise on right side. (Dynamic gesture)

(d) Rotate right hand clockwise on left side. (Dynamic gesture)

Figure 4. Additional gestures.

be detected correctly if there are multiple skin colour regions in the image that are the same as the right hand. It also fails to detect the right hand correctly when the illumination changes or under outdoor sunlight. Moreover, if we want to add gestures with different movements, the value of E in Equation (6) may not correctly identify them.

Therefore, in this section, we propose a gesture recognition method based on ML. For the gestures, we use 11 types, including the 7 types shown in Section 3 and the 4 types used in the experiments described in Section 6.3. The four additional gestures are shown below.

- Up the right hand above the reference region and hold it still. There is a face with the same skin colour near the right hand (static gestures; Figure 4(a)).
- Iterate the movement of the right hand up from the reference region and down into the reference region (dynamic gesture; Figure 4(b)).
- Rotate the right hand counter-clockwise on the right side of the body (dynamic gesture; Figure 4(c)).
- Rotate the right hand clockwise on the left side of the body (dynamic gesture; Figure 4(d)).

The position of the right hand in the image, which is the input data for gesture recognition, is detected using OpenPose. OpenPose is a deep learning-based method for determining the pose (skeletal data) of a person in an image. This method estimates the two-dimensional coordinates of the 15 points in the image (Figure 5). Even if there are multiple people in the image, the skeletal data can be detected for each of them. The programme of OpenPose is available at Ref. [18]. In this study, the programme is created by combining it with the deep NN module of OpenCV, referring to the contents of Ref. [19].

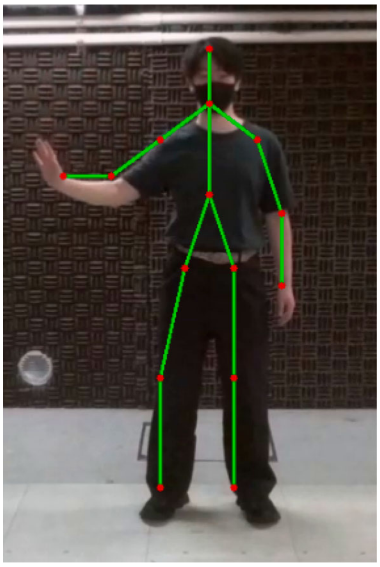


Figure 5. Skeletal data of a person.

As a method to recognize gestures from the time-series coordinate data of the right wrist obtained by OpenPose, we use ML with the LSTM framework in

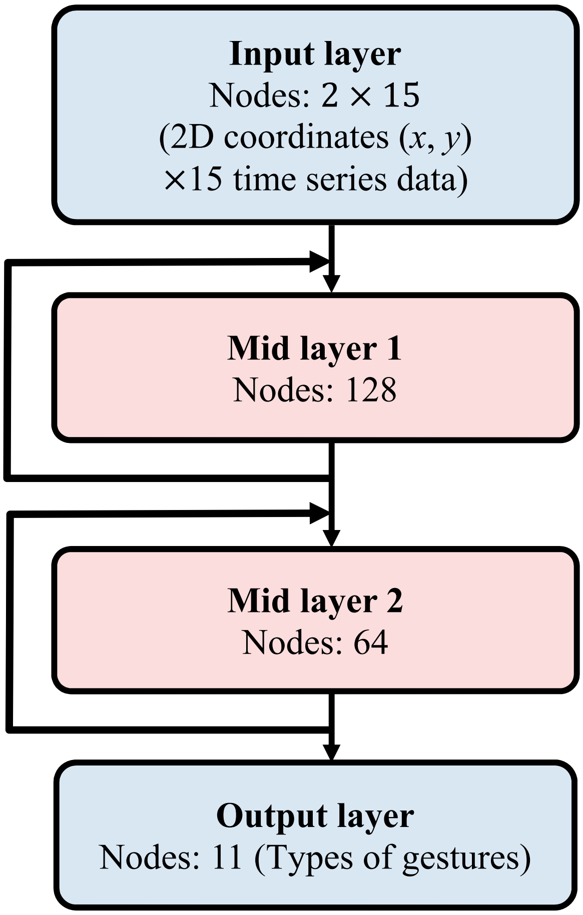


Figure 6. Structure of the proposed LSTM framework.

this study. LSTM is an improved model of recurrent neural network (RNN). RNN is an effective method for time-series and continuous data such as speech recognition and language translation. It differs from general CNNs in that the results of the output or intermediate layers are returned to the previous intermediate or input layers. The LSTM is an RNN with a management mechanism for the information of the storage period.

In this study, we use a framework with the following original structure for LSTM (Figure 6):

- Input layer: time-series 2D coordinates of the right wrist.
- Mid layer 1: 128 nodes.
- Mid layer 2: 64 nodes.
- Output layer: types of gestures.

Next, we show the recognition process. First, we obtain the coordinate data of the right wrist.

Step 1. Obtain human skeletal data from the images taken by the UAV by OpenPose. Here, we assume that there is only one person in the image. When detecting skeletal data with OpenPose, we ignore the confidence level of skeletal data detection in OpenPose library so that all skeletal data could be detected.

Step 2. Estimate the reference region from the skeletal data. Because the coordinates to be obtained are different, the range is slightly different from that of Figure 1. Here, the following range is defined as the reference region (Figure 7(a)) where (x_n, y_n) are the coordinates of the neck, (x_b, y_b) are the coordinates of the body, and d_b is the distance between (x_n, y_n) and (x_b, y_b):

- Horizontal range: from $(x_n - 0.3 \cdot d_b)$ to $(x_n + 0.3 \cdot d_b)$.
- Vertical range: from y_n to y_b.

Step 3. The coordinates of the detected right hand (x_r, y_r) in Figure 7(a) are transformed to (x'_r, y'_r) in Figure 7(b) as shown in Equation (7) so that they are in a coordinate system with (x_b, y_b) as the origin and the length of d_p as a constant value D (here, $D = 100$ pixels).

$$\left.\begin{aligned} x'_r &= (x_r - x_b) \cdot D/d_p \\ y'_r &= (y_r - y_b) \cdot D/d_p \end{aligned}\right\} \tag{7}$$

Next, we perform the ML process using LSTM. In this process, we use the necessary number of right wrist coordinate data for training (refer to Section 5.2 for details).

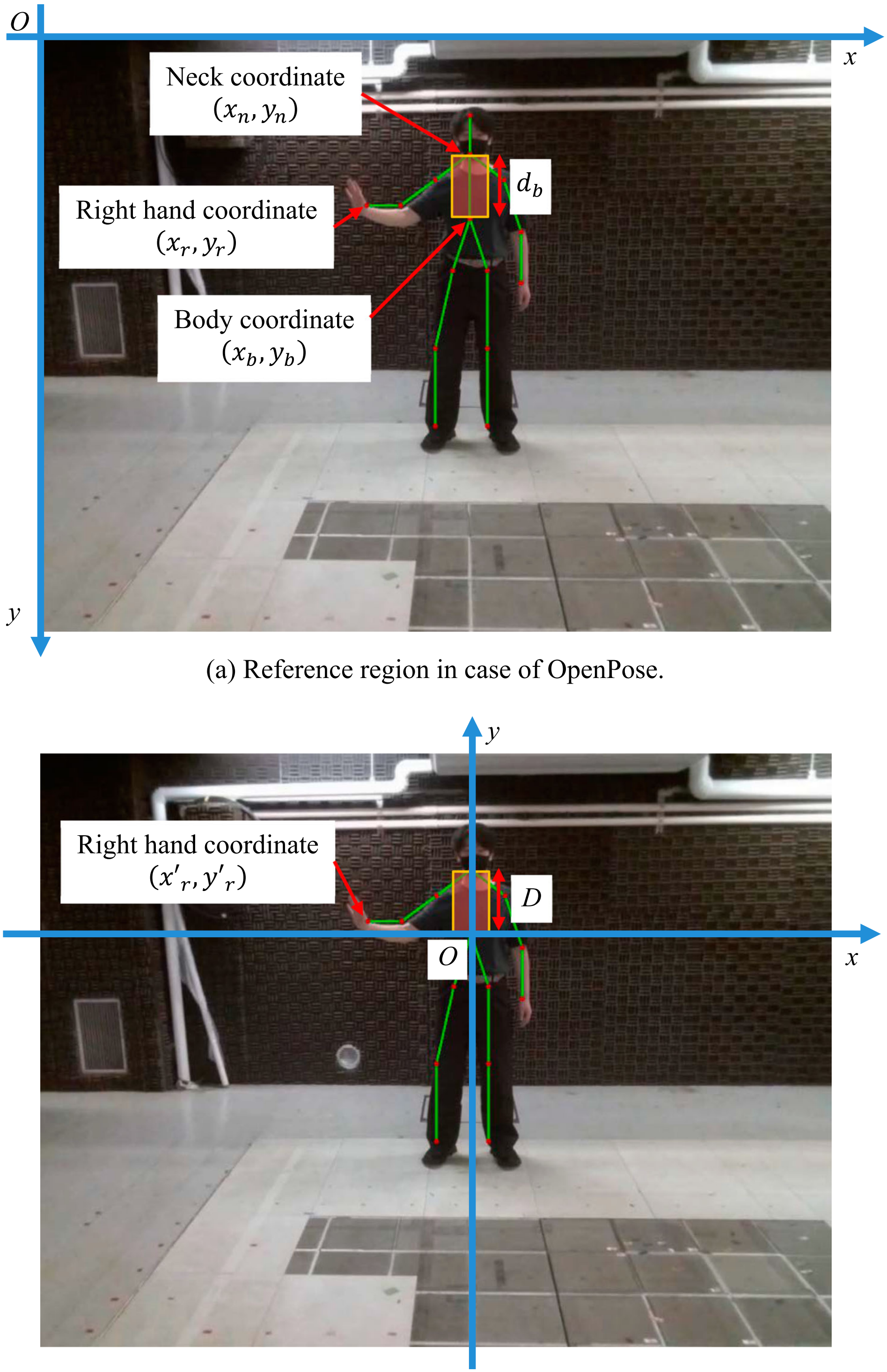
After that, in the gesture recognition process, once the coordinate data for n is obtained using the above method, the recognition process is performed using LSTM, and the gesture with the highest probability in the output layer is used as the recognition result. Note that our method does not include the class "unrecognizable" in the output layer, so the result is always one of the 11 gestures.

4.3. Operator detection method

If any of the following three conditions (a)–(c) are met during the gesture recognition process, the "operator detection process" is performed.

(a) When the human (operator) region cannot be detected.
(b) When the y-coordinate of the centre of gravity of the human region is extremely located on the upper or lower side of the image.
(c) When the x-coordinate of the centre of gravity of the human region is extremely located on the left or right side of the image.

The operator assumes to follow the motion commands of the UAV to some extent and perform the gestures. However, accurate following is difficult, especially in the vertical direction. As the gesture recognition proceeds, it is expected that the operator will gradually be obscured from the angle of view. Therefore, by setting the above conditions, the UAV automatically adjusts its distance, height, etc. and moves to a position where gesture recognition is possible again. Hence, we propose a



(a) Reference region in case of OpenPose.

(b) Transformed right hand coordinates.

Figure 7. Hand position detection by OpenPose.

method which moves the UAV automatically to a position where gesture recognition is possible again. The method of operator re-detection is shown in Steps 1–4 below.

Step 1. Acquire image from the current location and detect human region and face region. If neither can be detected, the UAV rotates around to look for the operator. Note that the detection size of the face region is a square. For face region detection, we use a trained Haar-like feature classifier from the image processing library OpenCV.

Step 2. The direction and height of the UAV adjust to operator by moving it based on the centre of gravity coordinates of the detected size (human region or face region).

Step 3. Repeat the face detection five times to obtain the average width of one side of the detection size (square).

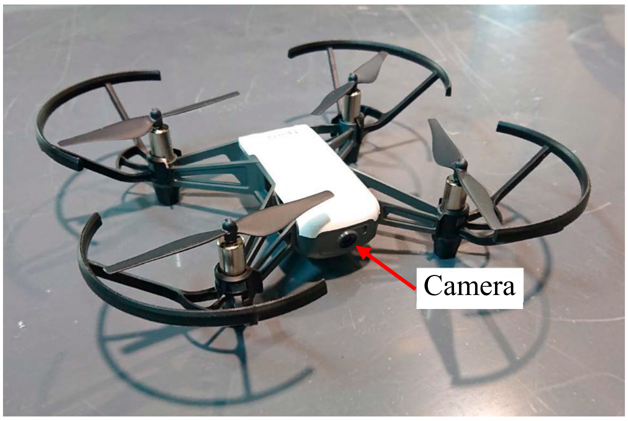


Figure 8. UAV (Tello).

Step 4. This average value is substituted into Equation (8) to obtain the estimated distance between the operator and the UAV:

$$y = 231.2x^{-1} - 0.9152, \tag{8}$$

where x is the average value of the face region detection size (one side) (pixel) and y is the estimated distance between the UAV and the operator (m). Equation (8) is calculated from preliminary experiments in Section 5.3. The UAV is moved to adjust the distance between the UAV and the operator based on y. (This distance is set to 3.0 m in experiments.)

Step 5. Repeat Steps 1–4 again. If no adjustment is required, return to the gesture recognition process.

5. Preliminary experiments

Section 5.1 shows the details of the preliminary experiment to set the threshold value of Step 7 in Section 4.1. Section 5.2 shows the learning data and results of ML of LSTM. Section 5.3 shows the details of the preliminary experiment to find Equation (8) to estimate the distance between the operator and the UAV in Step 4 in Section 4.3.

The UAV used in the experiments in this section and Section 6 is the Tello (Figure 8) by Ryze Technology [20]. The Tello is equipped with a single camera with a resolution of 960 by 720 pixels. In addition, Tello connect to a PC using a wireless local area network (LAN). Tello captures operator's image and send it to PC via Wi-Fi. PC processes the gesture recognition method and sends motion commands to Tello.

5.1. Threshold determination

In this experiment, we examine the distribution of the evaluation value E and estimate thresholds to distinguish gesture motions.

5.1.1. Experimental environment

This experiment is conducted indoors, where there are no major disturbances such as wind or lighting changes. The two subjects (operators) of the experiment are A, who have performed the gesture many times, and B, who have performed the gesture several times. The subject is the only one around the UAV, and the UAV is always in a hovering state. The distance between the subject and the UAV is about 3.2 m, and the height of the UAV from the floor is about 1.5 m.

5.1.2. Experimental method

The subjects perform static motion and dynamic motion gestures in the direction of each detection region, and the evaluation values are calculated and save according to Steps 1–6 in Section 4.1. In addition, 20 times the gestures of each motion are performed for each detection region. Then, the normal distribution curves are calculated from the saved evaluation values E for each motion using Equation (9):

$$f(x) = \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(x-\mu)^2}{2\sigma^2}\right), \tag{9}$$

where x is the evaluated value, μ is the mean of the evaluated values, and σ^2 is the variance of the evaluated values.

5.1.3. Experimental results

The obtained normal distributions are shown in Figure 9. The horizontal axis represents the evaluation value and the vertical axis represents the distribution of the evaluation value. The blue colour (the left side distribution) is the normal distribution curve for static motion, and the orange colour (the right side distribution) is the normal distribution curve for dynamic motion. Figure 9 shows that the peaks of the distribution of evaluation values for each motion are far apart. Therefore, we set the threshold for determining the gesture motion based on the range of $\mu \pm 3\sigma$ (σ: standard deviation of evaluation values), which contains 99.73% of the data. We set the range of static motion from 1.0 to 15.0, and the range of dynamic motion is from 25.0 to 63.0. A gesture that does not satisfy both thresholds is considered to be "not a gesture."

5.2. ML and learning data

In this section, we describe the ML process of the LSTM shown in Section 4.2. Here, we prepared 108 time-series data for each of 11 types of gestures (7 types in Section 3 and 4 types in Section 4.2). One time-series data consists of 15 pairs of right wrist coordinates. For each gesture, we randomly selected the time-series data and classified them as follows.

- For training: 88 time-series data multiplied by 11 types.
- For validation: 11 time-series data multiplied by 11 types.
- For test: 9 time-series data multiplied by 11 types.

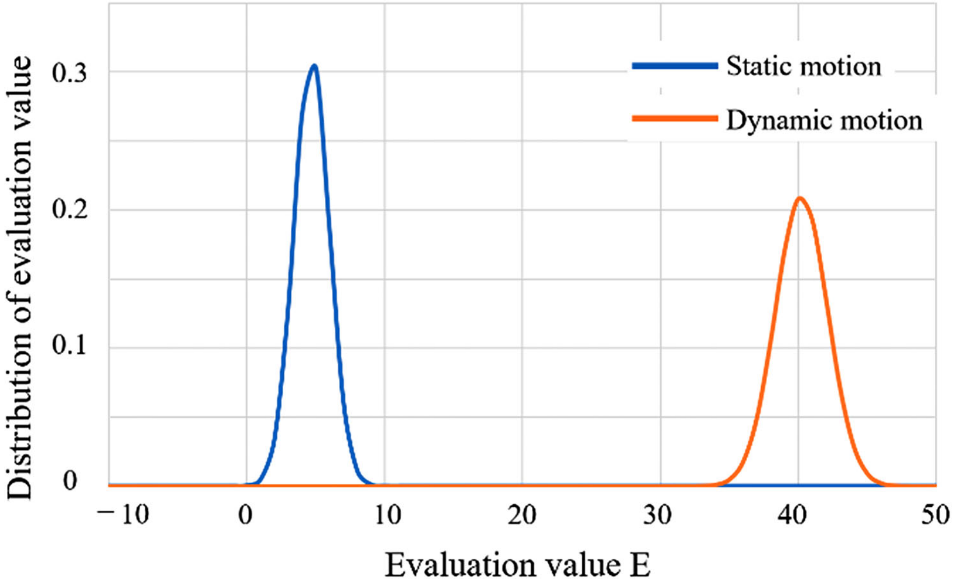


Figure 9. Normal distribution of evaluation values for each motion.

The LSTM is trained with the following training parameters using the training and validation data. The values of these parameters are obtained experimentally.

- Activation function: tanh
- Optimization algorithm: Adam
- Learning rate: 0.01
- Loss function: mean squared error
- Batch size: 16
- Number of training epochs: 1000 (finish if there is no change in the loss of validation data between 20 epochs)

After the training, the loss is 0.0127, and the accuracy is 0.999. Using the test data for the trained LSTM, the recognition accuracy is 94.9% (94 correct answers out of 9 by 11 time-series data).

5.3. Distance estimation

In this experiment, to estimate the distance between the operator and the UAV, we examine the relationship between the detected face size and the distance between the operator and the UAV.

5.3.1. Experimental environment

The experimental location and surrounding environment are the same as in Section 5.1. Three subjects are selected for the experiment.

5.3.2. Experimental method

The UAV places at a distance of 1.5 m from the operator and is gradually raised manually. Next, we detect the face region by hovering the UAV at a height where the face is within the angle of view. The face detection size is saved. The subject is kept stationary at the position. The height is changed within the range where the face is within the angle of view and the face detection size is saved. It saves 30 face detection sizes for each distance. Repeat this process in increments of 0.5 m up to 5.0 m.

5.3.3. Experimental results

The results of plotting the relationship of the face detection size (one side) against the distance between the operator and the UAV are shown in Figure 10. The horizontal axis is the average value of the face region detection size (one side) (pixel), and the vertical axis is the distance between the operator (subject) and UAV (Tello). Based on the relationship between the data, an inverse proportional approximation formula using the least-squares method is used as the formula to calculate the estimated distance between the operator and the UAV (Equation (8)).

6. Experimental results

6.1. Competition of gesture recognition

In this section, we show the comparison results of gesture recognition by FB method and ML method.

6.1.1. Experimental environment

In this experiment, four subjects (operators) perform the seven types of gestures shown in Section 3, and the recognition rate for each method is evaluated. The experience level of the gestures of the four subjects (A, B, C, and D) is as follows:

- A: performed the gestures many times.
- B: performed the gestures several times.
- C and D: perform the first gesture.

In this experiment, the UAV is fixed at a position 3.5 m in front of the subject and at a height of about 1.5 m from the floor, and the gestures are captured as

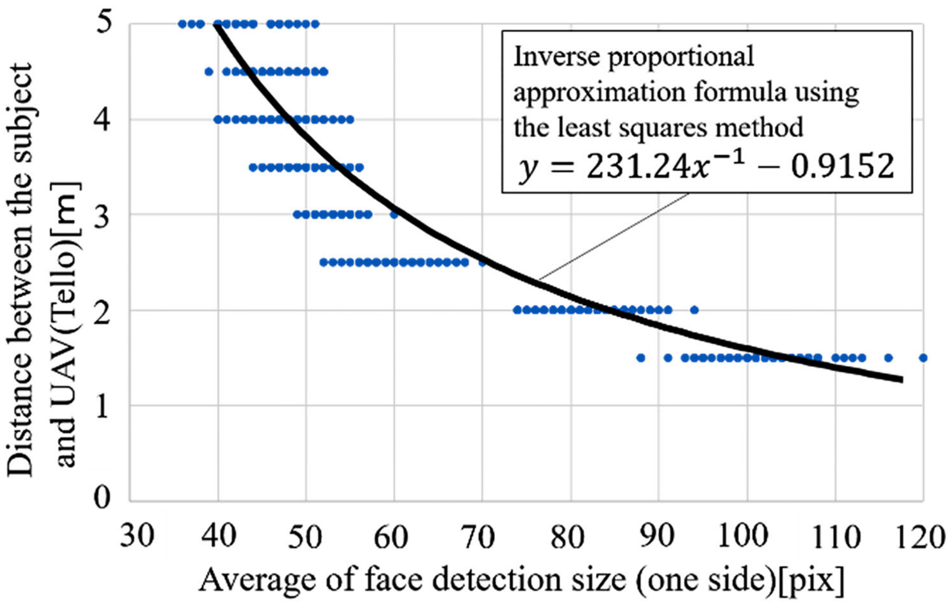


Figure 10. Relation between face detection size and distance.

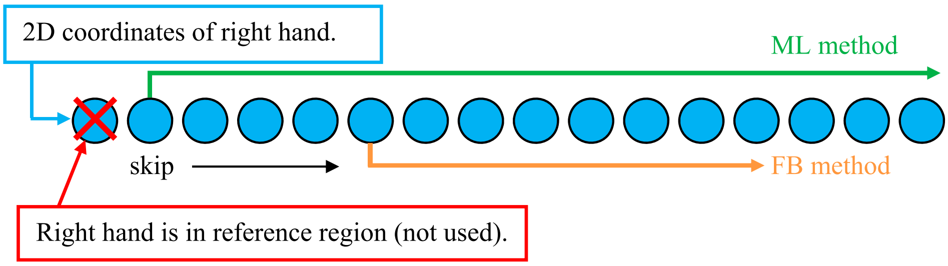

2D coordinates of right hand.
ML method
skip
FB method
Right hand is in reference region (not used).

Figure 11. Time-series data of each method.

a video by the UAV's camera. The subject performs the seven gestures in one sequence. Ten sequences are recorded as a video. Therefore, each subject performs 70 gestures. We will evaluate the results for four subjects.

The FB method and the ML method use the same video image. Each method detects the position of the right hand about every 167 ms. One gesture is performed for about 2.5 s. In the ML method, after the right hand is in the reference region, 15 time-series data (2D coordinates) of the right wrist are captured (first data of the right hand position in the reference region is not included). In the FB method, after the right hand is in the reference region, four position data are skipped because the gesture can be distinguished easily static or dynamic. And then eight time-series data are captured (Figure 11).

6.1.2. Experimental results

Table 2 shows the recognition results by the FB method, and Table 3 shows the recognition results by the ML method. As for "p/n" in the table, n is the number of gestures performed by the subject, and p is the number of gestures correctly recognized among them. And red cells mean ($p < n$), that is, there is a case of fail gesture recognition. This notation is also used in the tables in the following sections.

The total recognition rate of the FB method is 98.9% and that of the ML method is 97.9%. The reason for the low recognition rate of the ML method is that the detected position of the right wrist by OpenPose is sometimes different from the actual one, as shown in Figure 12. This error has nothing to do with the skill of the operator. Therefore, it is clear that the FB method can recognize gestures with high accuracy as long as the environment and other conditions are within the assumed range. On the other hand, the ML method also has a high recognition rate of gestures, but in order to achieve higher accuracy, it is necessary to investigate outlier detection and correction methods for the time-series data of skeletal coordinates obtained by OpenPose.

6.2. UAV manipulation

6.2.1. Experimental environment

Next, we show the results of gesture recognition while actually flying the UAV. In this section, we use the FB method, which has a high recognition rate as described in the previous section. In addition, four subjects are performed gestures. Their experience of gestures is as

Table 2. Gesture recognition rate (FB method).

Subject	Left dynamic	Right dynamic	Under dynamic	Under static	Right static	Left static	Reference static	Rate (%)
A	10/10	10/10	10/10	10/10	10/10	10/10	9/10	98.6
B	10/10	10/10	10/10	10/10	10/10	10/10	10/10	100.0
C	10/10	10/10	10/10	10/10	10/10	10/10	10/10	100.0
D	10/10	10/10	9/10	10/10	10/10	10/10	9/10	97.1
							Total	98.9

Table 3. Gesture recognition rate (ML method).

Subject	Left dynamic	Right dynamic	Under dynamic	Under static	Right static	Left static	Reference static	Rate (%)
A	10/10	10/10	9/10	10/10	10/10	10/10	7/10	94.3
B	10/10	10/10	10/10	10/10	10/10	10/10	10/10	100.0
C	10/10	10/10	10/10	10/10	10/10	10/10	8/10	97.1
D	10/10	10/10	10/10	10/10	10/10	10/10	10/10	100.0
							Total	97.9

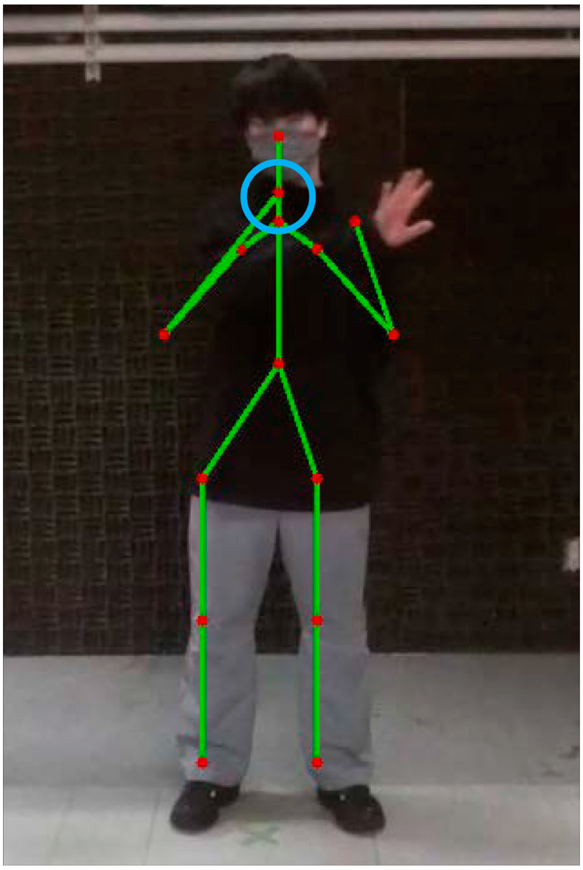


Figure 12. Incorrected right wrist position (circle) by OpenPose.

same as in the previous section. The process of the experiment is shown below.

Step 1. The Tello is placed in a certain position and the operator stands about 3.2 m away from the Tello.

Step 2. Take off the Tello. Manually adjust the Tello to a position where the operator's entire body is within the angle of view (about 1.5 m height). After adjustment, switch to automatic gesture recognition process.

Step 3. The operator grasps the signal from the Tello according to the number of sounds produced by the PC. This sound has three patterns as shown below. It also sounds automatically when each condition is entered.

- Sound once: Start the gesture recognition process.
- Sound twice: Gesture recognition is complete and motion command is executed.
- Sound three times: Start operator re-detection process.

Step 4. When the gesture recognition starts, the operator performs the gesture. Once the motion command is executed, the operator follows the UAV. The operator repeats this process. The gestures are performed in the following order: (left dynamic, right dynamic, under dynamic, under static, right static, left static, and reference static). (Note that this order is as same as that of Section 6.1.) The seventh "reference-static" gesture is recognized, the landing of the motion command is confirmed, and the process is ended. The moving distance and speed of Tello are set to be the same for all motions.

This process (from Step 1 to Step 4) is considered as one experiment. In addition, each operator performs the experiment five times. If a different gesture is recognized, the next gesture is performed without performing that gesture again.

6.2.2. Experimental results

Figure 13 shows the scene of this experiment. Table 4 shows the experimental results. Subject A shows no misrecognition from the first to the fifth experiment, and the gesture recognition rate is 100%. However, false recognitions are observed in the experiments of Subjects B, C, and D.

Figure 14(a) shows the time chart of the first experiment for Subject A, who is recognized correctly, and Figure 14(b) shows the time chart of the fifth experiment for Subject D, who show false recognition. In Figure 14, the horizontal axis is the elapsed time since the switch to automatic gesture recognition process, and the vertical axis is labels representing types of movements of Tello (it includes the motions of rotations of the operator detection method). In addition, the yellow part represents the gesture recognition until the hand enters the reference area, the red part represents

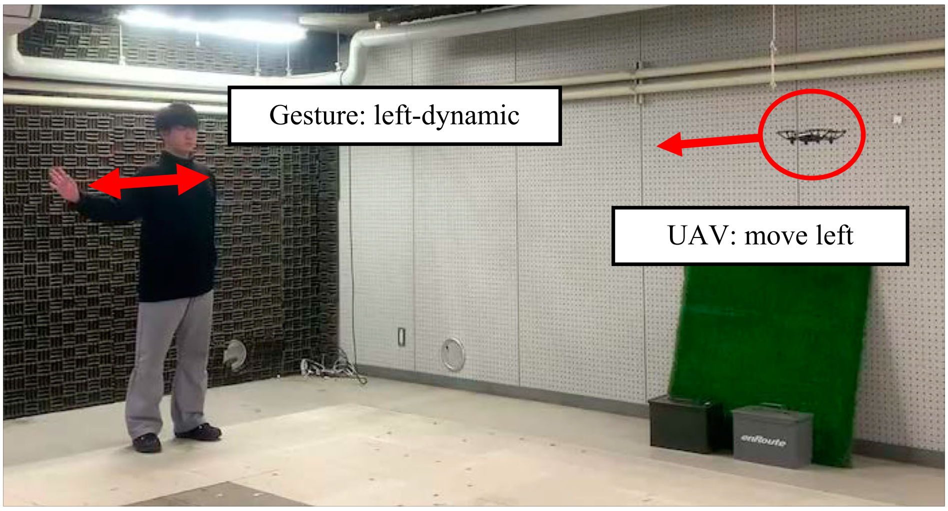


Figure 13. Scene of experiments.

Table 4. Gesture recognition rate under flying UAV.

Subject	Left dynamic	Right dynamic	Under dynamic	Under static	Right static	Left static	Reference static	Rate (%)
A	5/5	5/5	5/5	5/5	5/5	5/5	5/5	100.0
B	5/5	5/5	4/5	5/5	5/5	5/5	5/5	97.1
C	5/5	5/5	4/5	5/5	5/5	5/5	5/5	97.1
D	5/5	5/5	4/5	5/5	5/5	4/5	5/5	94.3
							Total	97.1

the gesture recognition after the hand enters the reference area and the blue part represents the execution of the motion command. From Figure 14(a), all UAV motion commands are correctly executed. Therefore, we can see that the gestures are also recognized correctly. In Figure 14(b), the third gesture is not recognized. The reason is that the position of the right hand could not be detected. In the hand detection region shown in Figure 3(b), the area above the reference point, where the face and neck exist, is excluded from the detection region. Therefore, if the right hand is located above the reference point, the right hand may not be detected. We think that the gesture is not recognized because the right hand is often above the reference point in the gesture motion.

Throughout the entire experiment, the gesture recognition rate is 100% in four out of five experiments for both Subjects B and C, three out of five experiments for Subject D who are found to be falsely recognized few times. The gesture recognition rate for the entire experiment is 97.1%. The average processing time from the start of gesture judgment to the end of recognition is about 4.0 s. It is assumed that this is a time that does not place a burden on the operator. Also, the average sampling rate for this experiment is 2.6 fps.

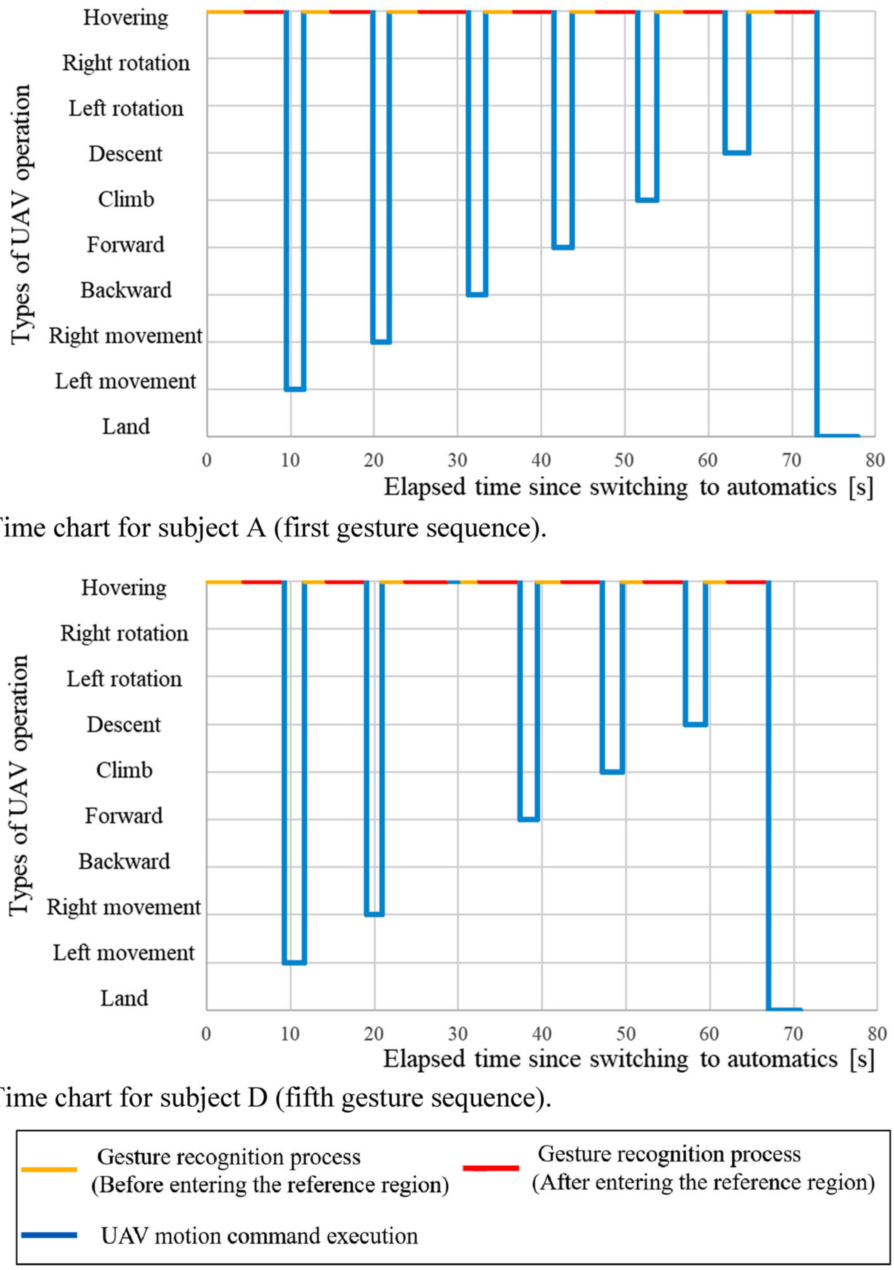
In this experiment, the operator re-detection shown in Section 4.3 was not performed. To demonstrate the effectiveness of this method, we also conduct gesture recognition experiments when the operator intentionally moves the location where the gesture is performed. In this section, we assume that the operator intentionally moves one step in the opposite direction of the UAV when the UAV moves to the left by the first gesture. In the case of other gestures, the operator follows the movement of the UAV as shown in Section 6.2.1, Step 4.

The time chart of this experiment is shown in Figure 15. In this figure, the green line from 15 sec. to 30 sec. represents the movement of the UAV for the operator search. After the first gesture, the operator moves in the opposite direction to the UAV, then the UAV captures the operator on the right edge of the camera image. In order to keep the operator in the centre of the image, the UAV is rotated to the right three times. Next, to adjust the distance to the operator, the UAV moves forward and backward. Subsequent gesture recognitions are successful. We conducted these experiments several times, and in all cases, gesture recognition could be continued by performing operator re-detection. Therefore, in order to operate the UAV with repeated gesture recognition, it is effective to search again when the operator is lost.

6.3. Robustness of gesture recognition

6.3.1. Experimental environment

Finally, in order to evaluate the robustness of the gesture recognition method, we show the results of gesture recognition with different clothes than in Section 6.1.



(a) Time chart for subject A (first gesture sequence).

(b) Time chart for subject D (fifth gesture sequence).

Figure 14. Time charts of experiments.

And, for the ML method, we also show the recognition results of the added gestures.

Gesture experiment 1. For the two subjects (A and D), the UAV is fixed to capture a video of them wearing different coloured or short-sleeved clothes from Section 6.1 and performing one sequence of the seven gestures. Using this video as input, we compare the recognition rate of gestures between the FB method and the ML method. Subject A has performed the gesture many times before. Subject D performs the gesture for the first time.

Gesture experiment 2. The same two subjects (A and D) perform one sequence of the four gestures described in Section 4.2 for each. The UAV's camera captures these sequences as videos. Using these videos as inputs, we estimate the gesture recognition rate using the ML method. In this experiment, the recognition rate by the FB method is not estimated because this method cannot calculate the features.

6.3.2. Experimental results

Table 5 shows the recognition results of the FB method and Table 6 shows the recognition results of the ML method for gesture experiment 1. The overall recognition rate of the FB method is 60.0% and that of the ML method is 98.6%. The reason for the low recognition rate of the FB method is that the left hand position is sometimes mistakenly detected as the right wrist position. Figure 16(a) shows the image when the right wrist is falsely detected. Figure 16(b) shows the result of the same person's image, which is correctly detected by OpenPose. Incorrect detections also occur when some of the clothes contain colours similar to skin.

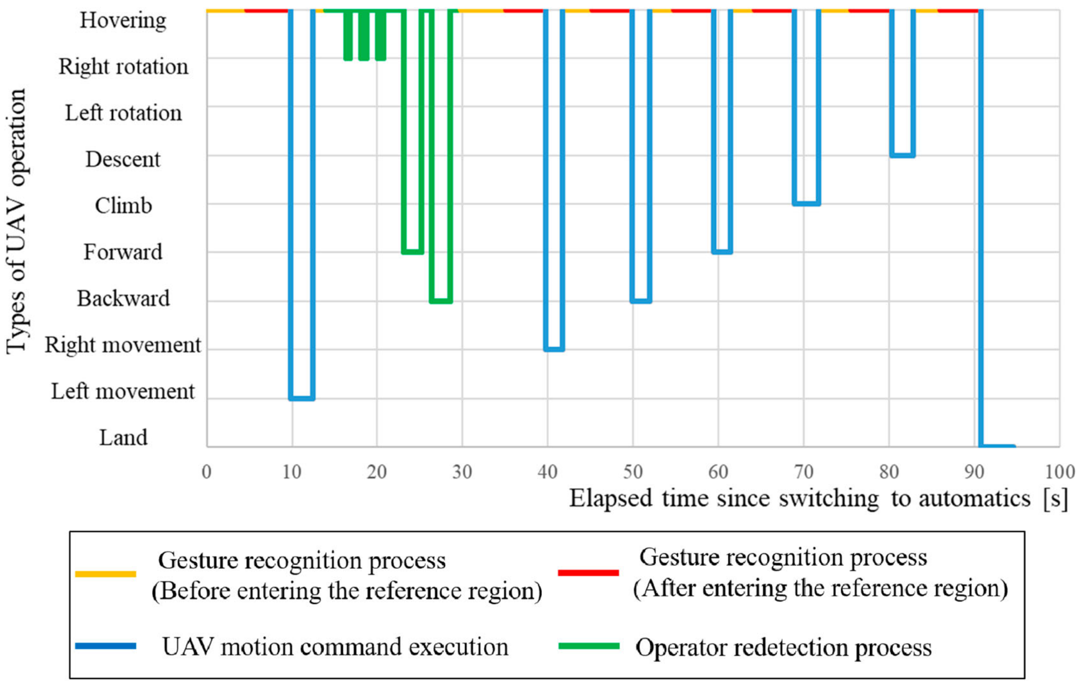
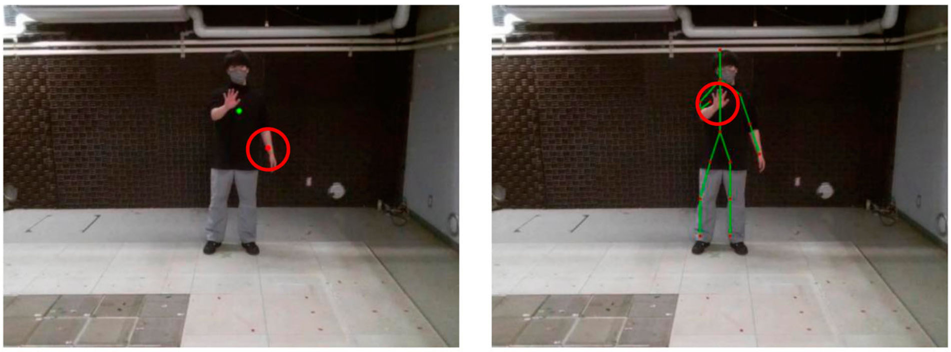


Figure 15. Time chart in case of operator re-detection process.



(a) Falsely detected right hand position by FB method.

(b) Correctly detected right hand position by ML method (OpenPose).

Figure 16. Competition of the right hand detection.

Table 5. Gesture recognition rate (gesture experiment 1, FB method).

Subject	Left dynamic	Right dynamic	Under dynamic	Under static	Right static	Left static	Reference static	Rate (%)
A	1/5	0/5	4/5	4/5	5/5	5/5	0/5	54.3
D	5/5	0/5	5/5	5/5	0/5	4/5	4/5	65.7
							Total	60.0

Table 6. Gesture recognition rate (gesture experiment 1, ML method).

Subject	Left dynamic	Right dynamic	Under dynamic	Under static	Right static	Left static	Reference static	Rate (%)
A	5/5	5/5	4/5	5/5	5/5	5/5	5/5	97.1
D	5/5	5/5	5/5	5/5	5/5	5/5	5/5	100.0
							Total	**98.6**

Next, Table 7 shows the recognition results of the ML method for gesture experiment 2. The overall recognition rate by the ML method is 92.5%. Although the number of gestures used in the evaluation is rather small, the recognition rate is almost the same as the previous results of the ML method. From these results, it can be said that the ML method is more robust to changes in clothing and environment than the FB method and can be easily extended to recognize gestures with different movements by simply preparing training data. However, in the ML method, it is necessary to correct the time-series coordinate data for

Table 7. Gesture recognition rate (gesture experiment 2).

Subject	Raise static	Up and down dynamic	Rotate right dynamic	Rotate left dynamic	Rate (%)
A	5/5	5/5	5/5	5/5	100.0
D	5/5	5/5	3/5	4/5	85.0
				Total	92.5

more accurate gesture recognition, as described in Section 6.1. In addition, we need more discussion about the movements and types of gestures that can be recognized by the ML method.

7. Conclusion

In this paper, we investigate a system that can recognize human gestures and operate a UAV. Here, we examine the FB method, which detects the position of the right hand from an image using features and identifies gestures, and the ML method, which detects the position of the right hand from an image and identifies gestures. We evaluate and compare each method.

In this study, we focus on both dynamic and static gestures, in which the right hand is repeatedly moved or held in a certain position. In the FB method, the coordinates of the right hand are obtained from the image using the colour information. Eight time-series data are Fourier transformed, and the resulting features are used to identify the gestures. On the other hand, the ML method estimates human skeletal data using the OpenPose library and obtains the coordinates of the right wrist from it. Then, we use 15 time-series data to identify the gestures in the LSTM framework. After the gesture is recognized, the UAV will move according to the gesture, but in order to be able to recognize the gesture continuously, we also propose a method to detect the operator.

In the experiment, we compare the gesture recognition results of the FB method and the ML method. As a result, under the conditions assumed in the FB method, the recognition rate of this method is higher than that of the ML method. In the ML method, the right wrist position obtained by OpenPose may contain errors, which may affect the recognition results. However, the recognition rate by the FB method decreases under conditions that are not expected for the FB method, but there are no significant change in the recognition rate for the ML method. Therefore, the ML method is more effective in various environments. In addition, we confirm that the UAV can search for the operator after performing a movement corresponding to the recognized gesture. We are able to show that the UAV can be continuously controlled by gestures.

We also show that the ML method can easily discriminate other gestures with different movements by learning them. However, the FB method is necessary to design new features and study their calculation methods, and it is not easy to increase the number of gestures. Therefore, when the environment in which the UAV flies, the operator's clothing and the types of gestures are limited, the FB method can provide highly accurate gesture recognition. But, when gesture recognition in various environments or when more types of gestures is needed, recognition by the ML method will be effective.

In the future, we should discuss how to correct the coordinates of the human skeletal data using OpenPose, taking into account the fact that the data is a time series of human movements. We will also devise new gestures for UAV operation, including movements other than those of the right hand, and evaluate their recognition accuracy. In order to verify the practicality of this study, we need to conduct outdoor gesture recognition experiments. We plan to investigate the gesture recognition rate under various conditions such as wind, rain, and different lighting conditions. And we plan to improve the system for gesture recognition by the on-board computer of the UAV to realize autonomous flight control.

Disclosure statement

No potential conflict of interest was reported by the author(s).

Notes on contributors

Shoichiro Togo He received his B.E. from Tokushima University in 2020. He also received his M.E. from Graduate School of Advanced Technology and Science, Tokushima University in Mar. 2022. His main research interests are robot vision and ML.

Hiroyuki Ukida He received his B.E. and M.E. from Okayama University in 1992 and 1994, and received the Ph.D. degree in informatics from Kyoto University in 2003. He is currently an associate professor of Tokushima University. His main research interests are image processing, robot vision, and ML. He is a member of IEEE.

References

[1] Tanaka K, Ukida H. Mobile robot operation by gesture recognition using iterative motion. IEEJ Trans Electron Inf Sys. 2015;135(8):944–953 [In Japanese].

[2] Togo S, Ukida H. Gesture recognition using hand region estimation in robot manipulation. In: Proceedings of the SICE annual conference 2021; 2021 Sept 8–10; Tokyo, Japan. p. 1119–1124.

[3] Cao Z, Hidalgo G, Simon T, et al. Openpose: realtime multi-person 2D pose estimation using part affinity fields. IEEE Trans Pat Anal Mach Intell. 2021;43(1):172–186.

[4] Gers F, Schmidhuber J, Cummins F. Learning to forget: continual prediction with LSTM. Neural Comput. 2000;12:2451–2471.

[5] Tamura Y, Sugi M, Arai T, et al. Target identification through human pointing gesture based on human-adaptive approach. J Robot Mechatron. 2008;20(4): 515–525.

[6] Dung L, Mizukawa M. Fast hand feature extraction based on connected component labeling, distance transform and hough transform. J Robot Mechatron. 2009;21(6):726–738.

[7] Takahashi Y, Toshida K, Hibino F, et al. Human pointing navigation interface for mobile robot with spherical vision system. J Adv Comput Intell Intell Inf. 2011;15(7):869–877.

[8] Hoshino K, Kasahara T, Tomida M, et al. Gesture-world environment technology for mobile manipulation – remote control system of a robot with hand pose estimation. J Robot Mechatron. 2021;24(1): 180–190.

[9] Tanaka S, Umeda K. Operating a mobile robot by gesture recognition. Trans IEE Jpn. 2001;121-C(9): 1457–1463 [in Japanese].

[10] Takahashi M, Irie K, Terabayashi K, et al. Gesture recognition based on detection of periodic motion. J Robot Soc Jpn. 2010;28(6):756–765 [in Japanese].

[11] Wan K, Sawada H. Gesture recognition based on the probability distribution of arm trajectories. SICE J Control Meas Syst Integr. 2009;2(5):263–270.

[12] Hu B, Wang J. Deep learning based hand gesture recognition and UAV flight controls. Int J Autom Comput. 2020;17(1):17–29.

[13] Perera AG, Law YW, Chahl J. UAV-GESTURE: a dataset for UAV control and gesture recognition. In: Leal-Taixe L, Roth S, editors. Computer vision – ECCV 2018 workshops (ECCV 2018). Lecture notes in computer science, vol. 11130. Cham: Springer; 2019.

[14] Kassab MA, Ahmed M, Maher A, et al. Real-time human-UAV interaction: new dataset and two novel gesture-based interacting systems. IEEE Access. 2020;8: 195030–195045. doi:10.1109/ACCESS.2020.3033157

[15] Liu C, Sziranyi T. Real-time human detection and gesture recognition for on-board UAV rescue. Sensors. 2021;21:2180. doi:10.3390/s21062180

[16] Kinet for Windows [Internet]. Microsoft; [cited 2021 Oct 25]. Available from: https://developer.microsoft.com/ja-jp/windows/kinect/

[17] OpenCV [Internet]. [Cited 2021 Oct 25]. Available from: https://opencv.org/

[18] CMU-Perceptual-Computing-Lab/openpose [Internet]. [Cited 2021 Oct 25]. Available from: https://github.com/CMU-Perceptual-Computing-Lab/openpose/

[19] Deep Learning Based Human Pose Estimation Using OpenCV [Internet]. [Cited 2021 Oct 25]. Available from: https://learnopencv.com/deep-learning-based-human-pose-estimation-using-opencv-cpp-python/

[20] Tello [Internet]. Ryze Technology; [cited 2021 Oct 25]. Available from: https://www.ryzerobotics.com/jp/tello/