Hand Gesture Recognition System for the Physical Search System

1st Shin Kajihara
Graduate School of Science and Engineering
Saga University
Saga, Japan
email: 20634002@edu.cc.saga-u.ac.jp

2nd Masato Okazaki
Graduate School of Science and Engineering
Saga University
Saga, Japan
email: 22726005@edu.cc.saga-u.ac.jp

3rd Chika Oshima
Faculty of Science and Engineering
Saga University
Saga, Japan
email: sj5872@edu.cc.saga-u.ac.jp

4th Koichi Nakayama
Faculty of Science and Engineering
Saga University
Saga, Japan
email: knakayama@is.saga-u.ac.jp

***Abstract*—In this paper, we proposed a hand gesture recognition system for searching for lost objects using a physical search system (PSS). The PSS detects all displaced objects in a physical space using two cameras and a computer based on image differences-detection technology. When users tell the PSS what the lost object is, using hand gestures to describe it, such as its size and location, may be useful, as may words that describe the object's name, color, and the time it was last seen. The hand gesture recognition system was developed and experiments were conducted to examine how accurately the system can estimate the size indicated by the width between the user's hands. Also, to allow users to register various gestures as the commands they want to use, we investigated the recognition rate of finger gestures. As a result, the system could measure the width between users' hands with almost no errors, based only on the image taken by the camera and a marker. Moreover, the finger gestures could be recognized with high accuracy, unless it was difficult for users to reproduce the gestures that had been pre-registered. In the PSS, the displaced object's images are grouped into clusters that contain the same objects' images and data about their features. When a user tells the PSS the features of what they want to find, using their hand gestures, the PSS can present to the user images of the object in an appropriate folder (cluster) that matches the request. Finally, once the user identifies the lost object's image, the PSS displays where and when the object was last seen/lost.**

***Index Terms*—Hand gesture; physical search system; MediaPipe.**

I. Introduction

Keyword and image searches are often used to search for data online. By contrast, the physical search system (PSS) [1], [2] looks for objects in physical space without requiring any sensors, other than a camera or data for pre-learning, and enables the retrieval of any object that has moved within a given physical space.

We sometimes use hand gestures when telling someone about a lost object; "a board about this size" with our hands outstretched, or "the remote control was over there," while pointing with the index finger. Even when using the PSS, it is desirable to be able to input information about the object's size and an approximate location in physical space using hand gestures.

Hand gesture recognition can be broadly divided into wearable and non-wearable types. In wearable types, there are an acceleration sensor [3], [4] and optical markers, such as color and reflective markers [5], [6]. However, for daily use, it is inconvenient to wear devices and markers on the hands and fingertips.

In non-wearable types, Leap Motion [7] can recognize hands and fingers by irradiating infrared rays with a small device, but they can only be detected up to a distance of about 0.5m from the device. OpenPose [8] can acquire the position and posture of fingers only based on camera images. With a high-resolution zoom camera, even far-distant hand gestures can be recognized, but the positions of photographed fingers can only be acquired two-dimensionally. Users do not always make gestures toward the camera because they tend to point in the direction where they think the lost object is, or they express the shape of the object in three dimensions. Therefore, a hand gesture recognition system needs to work not only in a non-wearable format, but also to acquire three-dimensional (3D) positions in physical space (hereinafter "the world coordinate system").

MediaPipe Hands [9], [10], a non-wearable type, acquires the estimated Z coordinate, in addition to 2D coordinates (X, Y) from a camera image. In this paper, we propose a system that can indicate features of a lost object based on hand gestures. Users can command the system with the hand gestures they determine to search for the lost object with the PSS, such as indicating a size of the lost object, pointing to the approximate location of the object before it was lost. No sensors, other than a camera in a real space and MediaPipe, are required.

In the next section, the PSS is introduced; the experiment's results are explained in Section II. Then, in Section III, a hand gesture recognition system is proposed. Two experiments are

conducted to examine how accurately the system can estimate the length between a user's hands and interpret their finger gestures. The paper concludes in Section IV.

II. Physical Search System (PSS)

A. Overview

This section explains the overall structure of the proposed PSS [1] [2]. Figure 1 shows the PSS' hardware configuration. The PSS consists of two cameras that constantly capture the target area and a computer that processes the images photographed by the cameras. The PSS' software configuration consists of a displaced object detection unit that extracts the displaced objects from images photographed by each camera, a displaced object image-clustering unit that creates clusters, and a search results display unit that retrieves and displays the displaced objects.

In the displaced object detection unit [2], the photographed images are processed in the order in which they are photographed. The image at a certain time is then compared at the pixel level with the image photographed at a previous time. When a pixel with a difference of a certain standard or more is detected, it is determined that something has been displaced. The PSS can also detect people, and the photographs can define areas in which no one or nothing is present. In other words, the PSS does not yet detect objects that are moving/rotating around of the center of gravity, but it can detect displaced objects by comparing sets of images.

When objects overlap, we can obtain expected results, if they are displaced in order. For example, Object A is placed at a certain place. After the system has photographed an image near Object A, Object B is placed on top of Object A. If the PSS photographs the place again before each object is moved, both Objects A and B will be detected correctly. When two overlapping objects move together, Objects A and B are detected as a single object, so if Object A is pulled out from under Object B, Object A will not be detected. However, if Object A is placed elsewhere, it will be detected as a displaced object.

Figure 2 shows the differences between the two images in white. The target area is cropped as a rectangle [11]. The cropped image is called a "displaced object image." As described in Section II-B, in the displaced object images clustering unit, the displaced objects' images are grouped into clusters that contain the images of the same objects in each location and stored in the PSS. In the search result display unit, as shown in Figure 3, when a PSS user searches for a lost object (a displaced object), the search results are displayed in an application that displays augmented reality (AR) using an AR marker and an AR display terminal [2].

B. Two-step Feature Clustering Algorithm

This section describes a two-step feature clustering algorithm (TFA [2]). At first, the displaced object images are processed with the x-means clustering algorithm [18]. Then, the PSS user manually deletes a few folders (clusters) in

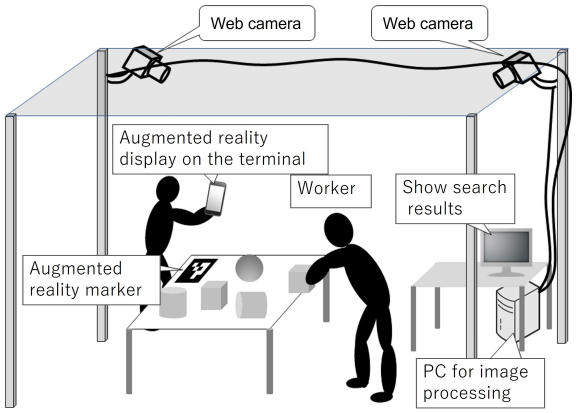

Fig. 1. Construction of the PSS hardware [2].

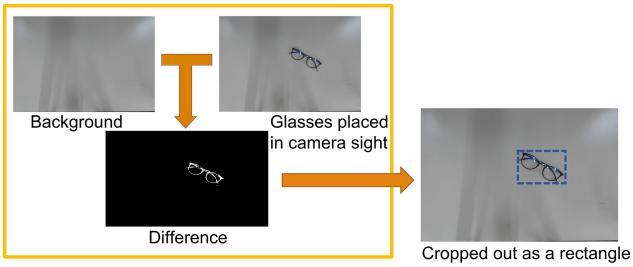

Fig. 2. How to crop out a displaced object [11].

which only noisy images are included. The reason why x-means was employed is that x-means is a method of clustering while automatically estimating the number of clusters k of k-means [19]. Therefore, the x-means clustering is a type of unsupervised learning like the k-means, wherein the data points (the features of the displaced object images) are grouped into different clusters based on their degree of similarity.

Next, a method of generating displaced object images with the feature are explained. ResNet50 [12], [13] is applied to the displaced object images to quantify their features. ResNet50 is a convolutional neural network that is pre-trained on ImageNet [14], an image database. Therefore, the user does not need to prepare any image-learning data.

ResNet is a residual network designed to alleviate the vanishing/exploding gradient problem caused by stacking residual

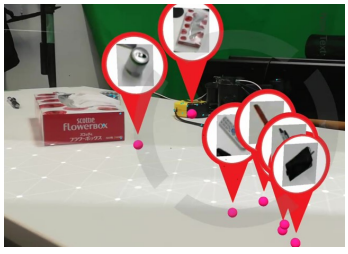

Fig. 3. Icons displays where and when the object was last seen/lost [2].

blocks. In ResNet-50, one residual block consists of three convolution layers. The size of the convolution kernel, which is the element of convolution operation in the convolutional layer, should be smaller than the size of the input image [15]. The stacked layers in the residual blocks have $1 \times 1, 3 \times 3$, and 1×1 convolution layers. The 1×1 convolution first reduces the dimensions. In the next layer, the bottleneck 3×3 layer, the features of the images are calculated. Then, the dimension of depth is again added in the next 1×1 layer (bottleneck) [16]. The final convolutional layer outputs 2048 feature maps of size 7×7.

In the PSS, the images of the displaced object are resized to 224 $\times$ 224 pixels and entered into ResNet50. Then, each resized image is flattened into the 100352-dimentional vector ($1 \times 7 \times 7 \times 2048$, which is a tensor: $depth(none) \times width \times height \times channel$) [17]. We call these "image features" in this paper.

Then, the displaced object images with the feature are processed with the x-means clustering algorithm [18]. A cluster number is assigned to each cluster, as determined by the x-means method. All displaced object images are stored in folders according to their cluster number.

There are a few folders (clusters) in which only noisy images are included. The PSS user manually deletes these. Then, the same processing protocol as used in the first stage is performed again for all images in the remaining clusters (the second step).

However, some noisy images will remain in a few folders [2], so there is room for improvement in accuracy. Therefore, a linking method (LM) was proposed to improve the accuracy of the TFA clustering [1].

C. Linking Method

This section describes the LM [1]. LM eliminates noisy images by creating pairs of images of highly similar displaced objects based on photographs taken simultaneously by two cameras [1].

In the PSS [2], two cameras (Cameras A and B) usually take pictures of the same displaced object at the same time from different angles. However, noisy images are photographed by only one of the two cameras, because noisy images are a result of misrecognition due to light rays or mistakes in cropping out the object parts of the images. Therefore, as shown in Figure 4, in the LM, pairs of the displaced object images with high degrees of similarity are created from the displaced object images derived from the photographs taken simultaneously by Cameras A and B. This process is called "linking" in this paper. In other words, a pair combination is created with a displaced object image derived from Camera A's photograph and another displaced object image derived from Camera B's photograph. Displaced object images obtained from only one of the cameras cannot be paired.

Next, the method for calculating the similarity between the displaced object images is explained. "imgsim [20]" is a library for computing perceptual hashes of images. The "distance" between images can be calculated using the imgsim library. The distances between the displaced object image, "a1," derived from Camera A's photograph and the displaced object images, "b1 to bx," derived from Camera B's photograph, are calculated. The higher the degree of image similarity, the smaller the distance between them. The distance between identical images is 0.

As shown in Figure 5, pairs are created in order, starting from those with the smallest distance value (the highest degree of similarity) between two images. For example, when two displaced object images are obtained from Cameras A and B's photographs taken at a certain time, there are four possible pair combinations. The image that is paired with another image is excluded from the candidate images for the other pairs. In addition, combinations with distance values exceeding 23 are not considered pairs. A gathering of the pairs is called a "pair group."

D. Brush up TFA Clustering Results with Pair Groups Created Using LM

This section explains a method for combining the TFA and the LM. Figure 6 shows that the displaced object images are updated by comparing the TFA results with that of LM; then, the clusters (folders) are reorganized. The displaced object images in the clusters that do not overlap with the displaced object images of the pair group are deleted. In this process, the noisy images and the images for which one camera has failed to detect a displaced object can be deleted from the folders.

Finally, the clusters created by TFA are reorganized. If two displaced object images that are paired belong to different clusters, they are processed as follows: the similarity (distance) between each of two images and other images that belong to the same cluster of each of two images is calculated using the imgsim library. Next, the averages of the distances in each cluster are calculated. The one with the larger average value moves to the cluster that includes the other displaced object image with the smaller average value. For example, Image_p, which belongs to Cluster_P, is paired with Image_q, which belongs to Cluster_Q. The distance values are calculated between Image_p and each of the other images in Cluster_P, and between Image_q and each of the other images in Cluster_Q. Then, the averages of the distance values are calculated for both Cluster_P and Cluster_Q. If the average of distances between Image_p and the other images in Cluster_P is larger than that of Cluster_Q, Image_p is moved to Cluster_Q.

E. Experiments for the usefulness of LM

1) Aim: In this section, we detail an experiment conducted to compare the accuracy of clustering between the combination of LM with TFA and TFA alone [1].

2) Method: Figure 7 shows ten objects on a table. The objects were a red pen, a green pen, a smartphone tripod, a box of tissues, a cup of coffee, a black smartphone, a box of darts, a dart, a plastic bag of replacement dart feathers, and gum tape. Two cameras were located so that the entire table could be photographed from two different directions. Even if

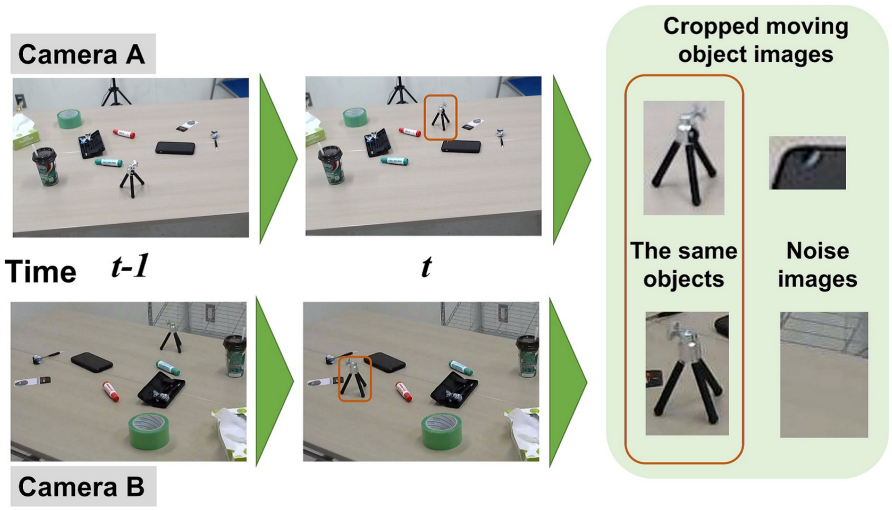

Fig. 4. Noisy images cannot be paired with another image.

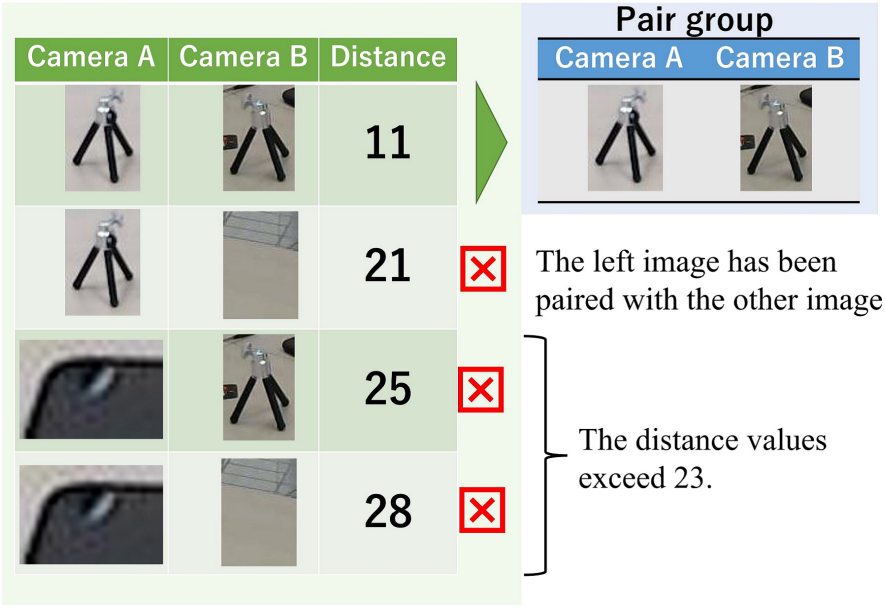

Fig. 5. Create a pair based on the similarity between two images.

the PSS is running, when there are people present, nothing will not be photographed.

During the experiment, one of the authors moved one of the objects on the table and then moved beyond the ranges of the cameras. After confirming that the PSS recognized the displaced object, they moved the next object on the table. This method was applied to the ten objects. They moved each object 10 times in two conditions, “LM with TFA” and “TFA.”

This process was repeated twice on two different tables in two different rooms, Rooms C and D. In Room C, the ten objects were on a desk, as shown in Figure 8. This is a dimly lit space because three displays are lined up, and the fluorescent lamp is not directly overhead. In Room D, a large table was placed in the center of the room, with fluorescent lights directly above it. There was nothing around it to block the light, as shown in Figure 9.

The PSS created clusters in both conditions. The accuracy of the clustering is indicated in precision values, recall values, and F-measures. All displaced object images showing one of the ten objects are regarded as an “actual positive.” The cluster in which the most images is included is considered to be a correct cluster, and the displaced object images of the correct cluster are regarded as a “predicted positive.” In the predicted positive images, the actual positive images are considered to be true positives (TP), and the others are considered false positives (FP). In the actual positive images, the images that

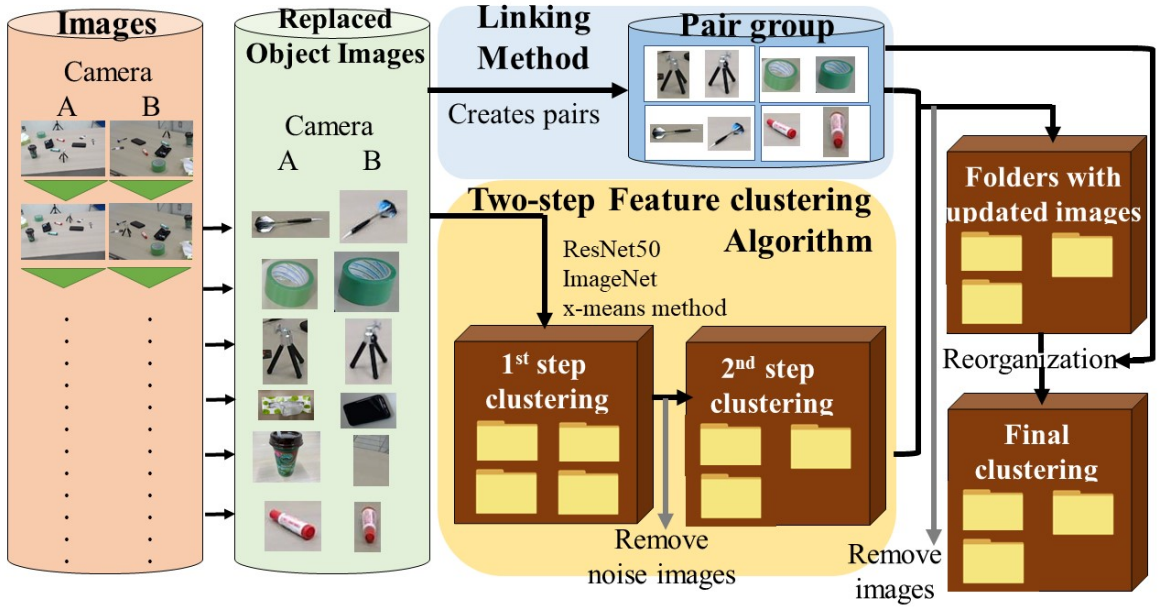

Fig. 6. Combine the results of the linking method (LM) with the two-step feature clustering algorithm (TFA) to update the images; then, reorganize the clustering.

are not in the correct cluster are considered false negatives (FN).

The recall is calculated using the following formula.

$$Recall = \frac{TP}{TP + FN} \quad (1)$$

The precision is calculated using the following formula.

$$Precision = \frac{TP}{TP + FP} \quad (2)$$

The F-measure is calculated using the following formula. The F-measure represents the harmonic mean of precision and recall.

$$F - measure = \frac{2Precision * Recall}{Precision + Recall} \quad (3)$$

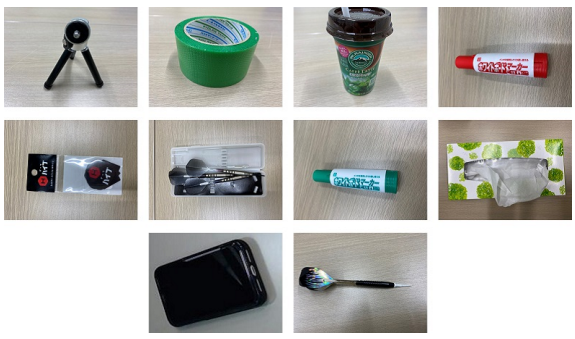

Fig. 7. Ten kinds of objects for the experiment.

Fig. 8. Desk in Room C.

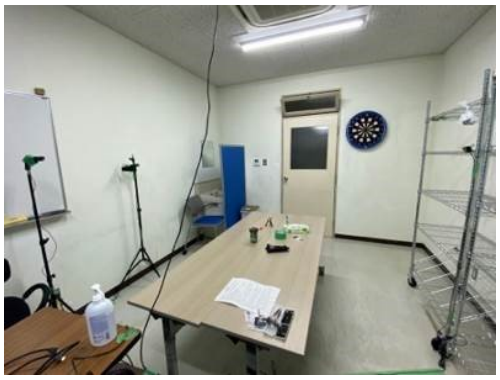

Fig. 9. Table in Room D.

Fig. 10. Images that missed a part of the gum tape.

3) Results:

Comparison of the Results of TFA and LM with TFA

The PSS created 11 clusters in both conditions. Each displaced object image was grouped into one to five clusters, depending on the type of displaced object. For example, all images of the smartphone tripod were grouped into Cluster_3 in the LM with TFA condition. The images of the green pen were grouped into five kinds of clusters in the TFA condition.

Table I shows the recall values, precision values, and F-measures to compare the results of the two conditions in both Rooms C and D [1]. For eight out of ten objects, the recall values were higher in the LM with TFA condition than in the TFA alone condition. In particular, the recall value of the gum tape under the LM with TFA condition was improved by 17%, compared to the TFA alone condition, and it became even closer to 100%.

For all objects, the F-measures were higher in the LM with TFA condition than in the TFA alone condition. However, the precision values of the LM with TFA condition were almost the same as that of the TFA alone condition. The displaced object images, the smartphone, the box of darts, and the dart remained around 30%.

Comparison of the Results of Rooms C and D

Table II shows the precision values, recall values, and F-measures to compare the results of Rooms C and D [1]. The number of images is less than Table I because of the results for each room. Therefore, the number of clusters was different, and these evaluated values between Tables I and II are different. The F-measures of all displaced object images in Room D were higher than that of Room C.

F. Discussion

The results showed that LM with TFA improved the clustering over TFA alone. As an example of improvement, the images of the gum tape were grouped into four clusters in the TFA alone condition because there were some images that missed a part of the gum tape, as shown in Figure 10. However, in the LM with TFA condition, most images of gum tape could be grouped into one cluster.

The F-measures for the red and green pens were not high, even in the LM with TFA condition. Since the shapes of these pens are similar, it was difficult to group them into one cluster using these algorithms. An algorithm using color features should be applied to such objects [2].

For the smartphone tripod and the cup of coffee, the precision values in the LM with TFA condition were lower than in the TFA alone condition. In the LM process, contrary to our expectations, the images of the smartphone tripod and the cup of coffee were paired with noisy images. Something similar to these objects was photographed as noisy images.

There were differences in accuracy depending on the room. The table in Room D was brighter than the desk in Room C. It can be suggested that the brighter space led to higher accuracy in detecting the displaced objects.

III. Hand Gesture Recognition System

A. Overview

In this section, a hand gesture recognition system is proposed. It consists of a registration mode, which corresponds to the initial settings when starting to use the system, and a recognition mode, which recognizes hand gestures by actually using the system. Section III-B describes the camera registration phase and its coordinate system, while Section III-C describes the gesture registration phase for defining the gestures to be discriminated. Section III-D describes the position acquisition phase, which obtains finger positions, and Section III-E describes the gesture recognition phase, which determines to which gesture the shape created by fingers corresponds.

B. Registration Mode: Camera Registration Phase

An augmented reality marker (AR marker) printed in an arbitrary size is placed at an arbitrary location that can be seen by all cameras. The experiment detailed in this paper used an ArUco marker [21] printed in a 570 mm $\times$ 570 mm square, as shown in Figure 11. The center of the marker was set as the origin of the world coordinate system. Each camera reads the marker and then calculates and stores its own position and orientation (camera parameters) in the world coordinate system.

C. Registration Mode: Gesture Registration Phase

In this phase, gestures to be discriminated are registered with the proposed system. The position and orientation of each finger are estimated by MediaPipe Hands [9], [10]. Figure 12 shows the numbered coordinates of 21 parts of each finger that MediaPipe Hands can acquire from each camera based on its images [10]. The coordinates are two-dimensional (X, Y) in the camera image, and the depth coordinates (Z) are estimated by MediaPipe Hands. Since the video is processed as time-series data at 10 frames per second, the coordinates of the hand gesture video for 1 second are registered in the system as real values of 21 points $\times$ 3 axes (X, Y, Z) $\times$ 10 frames.

TABLE I
ACCURACY OF CLUSTERING IN THE CONDITIONS TFA ALONE AND LM WITH TFA.

	Recall		Precision		F-measure	
Displaced objects	*TFA*	*LM with TFA*	*TFA*	*LM with TFA*	*TFA*	*LM with TFA*
Red pen	0.50	0.53	0.63	0.64	0.56	0.58
Green pen	0.38	0.43	0.47	0.43	0.42	0.43
Tripod	0.93	1.00	1.00	0.95	0.96	0.98
Box of tissues	0.49	0.63	1.00	1.00	0.65	0.77
Cup of coffee	0.91	1.00	1.00	0.94	0.95	0.97
Smartphone	0.42	0.58	0.23	0.29	0.30	0.39
Box of darts	0.44	0.69	0.30	0.30	0.36	0.42
Dart	0.75	0.75	0.24	0.27	0.36	0.40
Plastic bag	0.66	0.63	0.48	0.57	0.56	0.60
Gum tape	0.76	0.93	1.00	1.00	0.86	0.96
Average	0.62	0.72	0.64	0.64	0.60	0.65

TABLE II
ACCURACY OF CLUSTERING IN THE CONDITIONS OF ROOMS C AND D.

	Recall				Precision				F-measure			
	TFA		*LM with TFA*		*TFA*		*LM with TFA*		*TFA*		*LM with TFA*	
Displaced objects	C	D	C	D	C	D	C	D	C	D	C	D
Red pen	1.00	0.88	1.00	0.75	0.43	0.45	0.43	0.55	0.60	0.60	0.60	0.63
Green pen	0.87	0.81	0.95	0.63	0.48	0.81	0.50	0.45	0.62	0.55	0.66	0.53
Tripod	0.76	1.00	0.90	1.00	0.90	0.90	1.00	0.90	0.83	0.95	0.95	0.95
Box of tissues	0.50	1.00	0.50	1.00	0.50	0.80	1.00	1.00	0.50	0.89	0.67	1.00
Cup of coffee	0.93	1.00	0.71	1.00	0.87	0.86	1.00	1.00	0.90	0.92	0.83	1.00
Smartphone	0.64	0.81	0.76	0.63	0.20	0.39	1.00	0.50	0.31	0.53	0.86	0.56
Box of darts	0.53	0.81	0.47	0.63	0.22	0.39	0.57	0.48	0.31	0.53	0.52	0.63
Dart	0.69	0.56	1.00	1.00	0.09	0.90	0.12	0.95	0.17	0.69	0.22	0.97
Plastic bag	1.00	0.80	0.95	0.93	0.82	0.92	1.00	0.93	0.90	0.86	0.98	0.93
Gum tape	0.78	1.00	0.78	1.00	1.00	0.95	1.00	1.00	0.88	0.97	0.88	1.00
Average	0.77	0.87	0.80	0.86	0.55	0.74	0.76	0.78	0.60	0.75	0.72	0.82

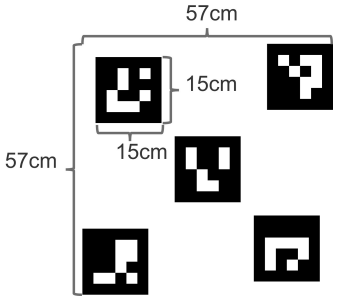

Fig. 11. A marker for determining the orientation and position of the camera on the world coordinate system.

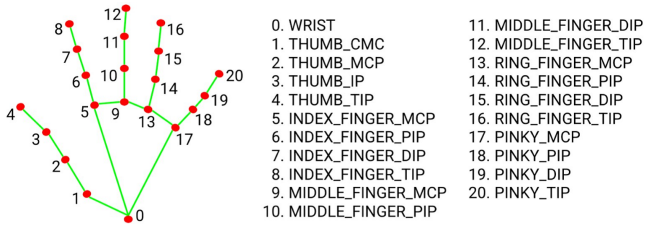

Fig. 12. 21 hand landmarks (quoted from [10]).

However, this coordinate is the value in the coordinate system for each camera. As shown in Section III-D, the coordinate values are converted to values in the world coordinate system based on the camera's unique parameters (see Section III-B).

D. Recognition Mode: Position Acquisition Phase

This section shows how to convert the coordinates of the hand in the camera's coordinate system to the world coordinate system. First, 0: WRIST (Figure 12) (hereinafter, "wrist coordinate") is used as the representative value of the hand position. When only one camera is used, there is a large error in the depth coordinate (Z-axis) in the camera coordinate system; therefore, the 3D position (coordinate) of the hand in the world coordinate system cannot be determined uniquely. With this system, however, since the camera parameters are obtained using AR markers, the position and orientation of each camera in the world coordinate system can be calculated. Then, since the positions of the hands are detected simultaneously with two cameras, their 3D positions in the world coordinate system can be calculated.

To explain in detail, the coordinates in the world coordinate system are obtained from the perspective projection transformation matrix, which indicates the position and orientation of the two cameras and the wrist coordinate in the screen obtained by each camera. The wrist coordinate is estimated to lie on a straight line passing through the camera in 3D space, calculated from the position and orientation of one camera and

the wrist coordinate estimated from the images taken by that camera. Similarly, based on the image of the other camera, the wrist coordinate is estimated to lie on another straight line passing through the camera in 3D space. The midpoint of the line segment representing the shortest distance from these two straight lines is the wrist coordinate in 3D space.

E. Recognition Mode: Gesture Detection Phase

The time-series data of each recognized finger position (hand gesture) is coordinate-transformed so that the positions and orientations match the registered hand gestures (see Section III-C). As shown Figure 13, the position and orientation of the hand gesture is transformed to overlap with the positions and orientations of the registered gestures at Positions 0, 5, and 17 (see Figure 12). Then, the similarity between the hand gesture captured by the camera and the registered hand gesture is determined. If the registered hand gesture has 10 frames, the similarity is continuously determined for the latest 10 frames of the obtained data. Three similarities are used: cosine similarity for position, Euclidean distance of position of each part, and velocity of wrist position. When each similarity is higher than the threshold value, it is determined that the corresponding hand gesture was performed.

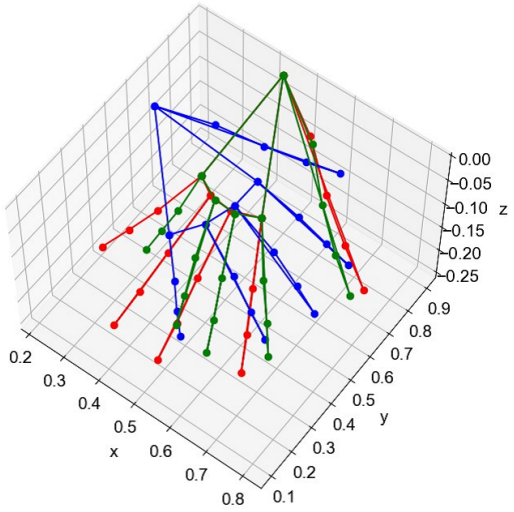

Fig. 13. The hand gesture is transformed to overlap with the registered gestures.

F. Evaluation of Size Expressions by Hand Gestures

1) Aim: This section verifies how accurately the system can estimate the size indicated by the user's hand gesture in real space.

2) Setting: As shown in Figure 14, four cameras [22] with a height of 2600 mm facing the direction of a square table were placed at the four corners of a square with a side of 3200 mm. The height of the table was 660 mm, and each side was 800 mm long. The ArUco marker [21] was placed in the center of the table as the origin of the world coordinate system.

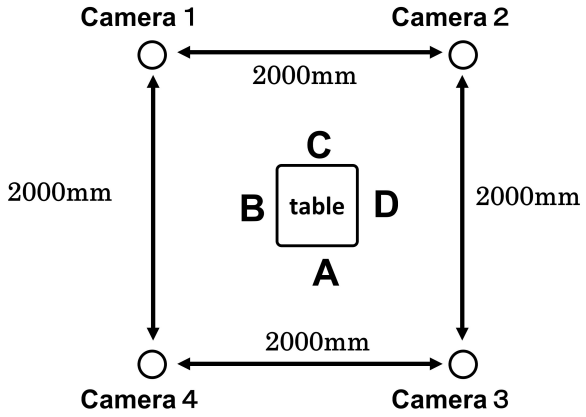

Fig. 14. A Table and Four Cameras.

3) Method: Participants were five males in their twenties. The space between their left and right hands was fixed using a wristband with a string attached. As shown in Figure 15, the wristbands were made of Velcro with 300-, 600-, or 900 mm-long strings. First, the participants separated their arms to the full length of the wristband string while they clenched their fists. Then, the participants turned their palms upwards. In this experiment, the system regarded the palm-up action as a gesture indicating length and measured the width between the hands.

The first author preregistered in the system these palm-up gestures with lengths of 300, 600, and 900 mm. Each participant stood facing the center of the table in an assigned position, A–D (see Figure 14). Then, the participant was instructed to perform the palm-up gesture with the 300 mm wrist band. After the length between their hands was recognized, they moved clockwise and performed the palm-up gesture with the same wrist band again. After they completed it for all four positions, they performed it with the remaining two length wristbands, again moving to each of the four positions. A total of 60 measured values (5 participants, 4 positions, 3 length wristbands) was acquired.

The absolute and relative errors for each measurement value were calculated as follows:

$$AE = |d_i - d| \tag{4}$$

$$RE = \frac{|d_i - d|}{d} \times 100 \tag{5}$$

where AE means the absolute error and RE means the relative error. d_i means the measurement value and d means the length of each wrist band (300, 600, or 900 mm).

Then, means, standard deviations (SD), and coefficients of variation (CV) were calculated for each condition (300, 600,

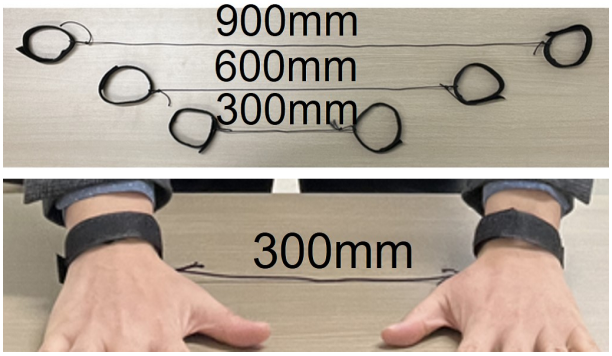

Fig. 15. Wristbands

distance of 300 mm, while the other had a horizontal distance of 600 mm from the desk.

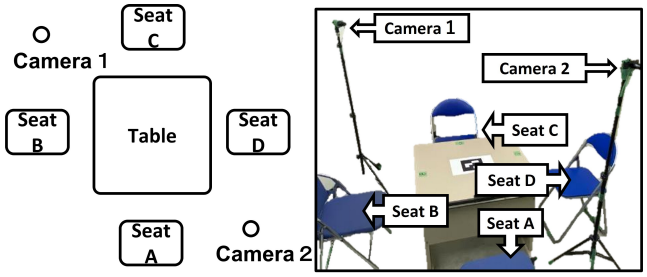

Fig. 16. Four Seats and Two Cameras Around the Table.

and 900 mm). Because the means were drastically different from one another, the CV was calculated as follows:

$$CV = \frac{s}{\bar{d}} \tag{6}$$

s and $\bar{d}$ indicate the standard deviation and means of the width measurements, respectively. CV is the ratio of the standard deviation to the mean.

4) Results: Table III shows the width measurements between the hands using the system, AE, and RE. The mean, SD, and CV are shown at the ends of the columns.

The means of AE were 14.5, 40.2, and 48.6 mm in each condition, respectively. The average error was less than 5 cm. The means of RE show that the error rate was smallest under the 300 mm condition.

By contrast, the CV results show that the measurement values while wearing the 300 mm wristband had the greatest variation among the three conditions (see Table III).

5) Discussion: In this section, the participants wore wristbands connected by 300, 600, and 900 mm strings on both hands. They were asked to make a gesture of fully spreading their hands. The error between the actual spreading length and the length measured by the system was investigated. Because the average errors were small, the results showed that it is possible to convey the length of a search object to the system using this hand gesture.

G. Recognition of Finger Gestures' Accuracy

1) Aim: In addition to hand gestures that indicate the size of objects, various gestures can be registered as commands that users want to use. In this section, 10 kinds of finger letters were used to examine the recognition rate of detailed finger gestures.

2) Setting: Figure 16 shows the setup for the experiment. Four seats (A–D) with different angles of 90 degrees were prepared around a square table. The height of each seat was 440 mm, the size of the seat was about 400 mm square, and the height of the highest point of the backrest was 800 mm from the floor. Two web cameras [22] were set on the diagonal extension of the desk. One of the cameras had a horizontal

3) Method: Participants were 11 males in their twenties who had never used finger gestures. Each participant sat in Seats A–D in turn. The participants were presented with a list of finger letters, as shown in Figure 17. The "a, i, u, e, o" are the Japanese vowels. The "ka, ki, ku, ke, ko" show five consonant and vowel combined pronunciations. The participants were asked to spell the following five words with their fingers: "a-ka," "i-ku," "u-ki," "o-ke," and "ko-e." These finger letters were preregistered in the system by the second author. However, these were not intended to be recognized as "characters." These finger letters were used to examine how recognizable the finger gestures were.

Each participant was seated facing the center of the table in an assigned seat, from A–D. Then, they was instructed to perform the first finger letter, a-ka, with his right hand until the camera recognized it. The orientation of the hands and fingers was arbitrary. If a gesture was not recognized after repeating it 10 times, it terminated as a recognition failure. After a-ka was recognized, they moved clockwise and performed the same finger gesture again. After completing this in all four seats, they performed the remaining four words' finger gestures, again, sitting in each of the four seats. A total of 220 types of finger gesture data (11 participants, 4 seats, 5 types of hand gestures) were acquired.

When the first author piloted the recognition of the finger letters using this method, the recognition rate was almost 100% for any gesture and any seat. The participants were asked to examine the list of finger letters at the beginning of the experiment and then performed the finger letters without prior practice.

4) Results: A total of 199 of the 220 finger letter trials (90.5%) were recognized in fewer than 10 trials. The average number of trials for 199 was 1.98 times.

Table IV shows the recognition rate and average number of attempts per finger gesture. The gestures for o-ke and i-ku had both a high recognition rate and a low average number of trials. O-ke moves from "o," with all fingers curled, to "e," with four fingers extended. I-ku moves from "i," with only the little finger upright and the others curled, to "ku," with four fingers extended horizontally.

TABLE III
WIDTH INDICATED BY HAND GESTURE AS MEASURED BY THE SYSTEM.

	String length								
	300mm			600mm			900mm		
Participant	Measured value(mm)	AE (mm)	RE (%)	Measured value(mm)	AE (mm)	RE (%)	Measured value(mm)	AE (mm)	RE (%)
1	275.4	24.6	8.2	617.1	17.1	2.8	895.3	4.7	0.5
1	286.9	13.1	4.4	603.1	3.1	0.51	889.6	10.4	1.2
1	282.6	17.4	5.8	618.8	18.8	3.14	960.4	60.4	6.7
1	286.8	13.2	4.4	650.7	50.7	8.45	957.7	57.7	6.4
2	316.7	16.7	5.6	659.4	59.4	9.9	990.5	90.5	10.1
2	329.3	29.3	9.8	647.4	47.4	7.9	994.9	94.9	10.6
2	317.2	17.2	5.8	629.7	29.7	5.0	998.5	98.5	11.0
2	317.1	17.1	5.7	614.8	14.8	2.5	982.7	82.7	9.2
3	307.2	7.2	2.4	605.0	5.0	0.8	938.7	38.7	4.3
3	298.4	1.6	0.5	649.8	49.8	8.4	967.3	67.3	7.5
3	301.5	1.5	0.5	626.2	26.2	4.4	966.0	66.0	7.3
3	298.6	1.4	0.5	650.2	50.2	8.4	956.7	56.7	6.3
4	276.3	23.7	7.9	656.3	56.3	9.4	880.2	19.8	2.2
4	322.4	22.4	7.5	654.7	54.7	9.1	910.7	10.7	1.2
4	274.3	25.7	8.6	668.1	68.1	11.4	888.1	11.9	1.3
4	288.0	12.0	4.0	661.6	61.6	10.3	897.5	2.5	0.1
5	312.8	12.8	4.3	631.7	31.7	5.3	899.8	0.2	0.0
5	296.8	3.2	1.1	641.9	41.9	7.0	965.0	65.0	7.2
5	316.2	16.2	5.4	652.5	52.5	8.8	971.1	71.1	7.9
5	312.6	12.6	4.2	664.6	64.6	10.8	962.5	62.5	7.0
Mean	300.9	14.5	4.8	640.2	40.2	6.7	943.7	48.6	5.4
SD	17.0	–	–	20.3	–	–	39.8	–	–
CV	0.06	–	–	0.03	–	–	0.04	–	–

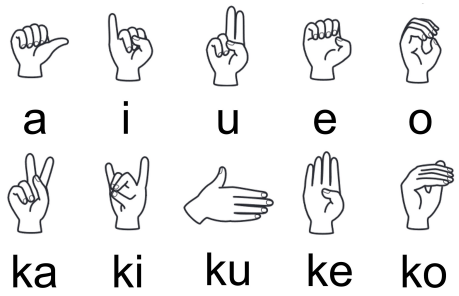

Fig. 17. Finger letters for Japanese phonetics.

Ko-e moves from "ko," with all fingers extended, to "e," with all fingers curled. The overall images of the hands are similar to one another, since the fingers are not extended upward; however, the recognition rate was not so low.

The recognition rate for a-ka was the lowest, even though the recognition rate seemed to be high, because a-ka moves from "a," with the thumb extended to "ka," with the index and middle fingers extended upward.

Table V shows recognition rate and average number of attempts per seat. The recognition rates for Seats A and C, in which the participants' right hands were photographed from behind, were slightly lower, but all recognition rates exceeded 80%, and the average number of trials was less than 2.27 times, regardless of the direction of the hand gesture.

Table VI shows recognition rate and average number of attempts per participant. Although there were differences in the recognition rates among participants, all achieved at least 80%. Participants looked at pictures of the finger letters and imitated them, but there were differences in their accuracy, which is thought to have led to the differences in their recognition rates.

TABLE IV
RECOGNITION RATE AND AVERAGE NUMBER OF ATTEMPTS PER FINGER GESTURE.

Finger letter	Recognition rate (%)	Average number of attempts
a-ka	72.7	1.84
i-ku	97.7	1.58
u-ki	93.2	2.20
o-ke	100.0	1.59
ko-e	88.6	2.74

TABLE V
RECOGNITION RATE AND AVERAGE NUMBER OF ATTEMPTS PER SEAT.

Seat	Recognition rate (%)	Average number of attempts
A	89.1	2.20
B	94.5	1.71
C	80.0	2.27
D	98.2	1.80

5) Discussion: Based on the experiment's results, the finger gestures performed within the range captured by the two cameras can be generally recognized from any direction and by all participants. However, if the gesture itself includes an action that is difficult for the participant, the recognition rate declined. Therefore, it is important to set finger gestures that are easy for system users to operate.

TABLE VI
RECOGNITION RATE AND AVERAGE NUMBER OF ATTEMPTS PER PARTICIPANT.

Participant	Recognition rate (%)	Average number of attempts
P1	100.0	1.60
P2	100.0	1.45
P3	80.0	2.50
P4	95.0	1.47
P5	80.0	1.63
P6	100.0	1.90
P7	90.0	2.56
P8	80.0	2.75
P9	100.0	1.65
P10	85.0	2.29
P11	85.0	2.29

IV. CONCLUSION

The physical search system (PSS) was developed to search for lost objects in physical space. The PSS detects all objects displaced in a physical space using two cameras and a computer. Besides voice and text input, it would be useful to use hand gestures to tell the PSS what to look for. In this paper, we investigated the accuracy rate of hand gestures to indicate the size of an object. The participants wore wristbands connected by 300, 600, and 900 mm strings on their hands. They spread their hands to the full length of the wristband strings, 300, 600, and 900 mm, while they clenched their fists, and the system measured the width between them. The results showed average absolute errors of 14.5, 40.2, and 48.6 mm in the conditions of 300, 600, and 900 mm, respectively. The system could measure the widths between users' hands with little error. Then, the recognition rates of five kinds of finger gesture series were examined. The recognition rates were from 72.7–100.0%. The finger gestures could be recognized with high accuracy if the participants could easily imitate the gestures the system had learned in advance.

In the future, a series of flow: a user inputs the features of a lost object into the PSS using hand gestures, then, the PSS finds images of the object with these features in the database, and presents where it is, will be evaluated from the aspects of the system's function and interface.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number 20H04470 and 22K00211.

REFERENCES

[1] S. Kajihara, M. Okazaki, C. Oshima, and K. Nakayama, "Improving a Physical Search System that Detects Even Unknown Displaced Objects Using Image Differences," EMERGING 2022, The Fourteenth International Conference on Emerging Networks and Systems Intelligence, IARIA, pp. 13-18, 2022.

[2] S. Kajihara, M. Okazaki, K. Kawabata, H. Furukawa, C. Oshima, et al., "Proposal and verification of a physical search system that does not require pre-learning data and sensors other than cameras," IPSJ Transactions on digital practices, vol. 3, no. 2, pp. 76–92, 2022. (in Japanese)

[3] K. Muraoka and T. Terada: A Motion Recognition Method by Constancy Decision, IPSJ Journal, vol. 52, no. 6, pp. 1968-1979, 2011.

[4] D. Kim, O. Hilliges, S. Izadi, A. D. Butler, J. Chen, et al., "Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor," Proceedings of the 25th annual ACM symposium on user interface software and technology, pp. 167–176, 2012.

[5] Vicon Motion Systems Ltd., VICON. Available: https://www. vicon.com/ (accessed May 20, 2023)

[6] NaturalPoint Inc., OptiTrack. Available: https://optitrack. com/ (accessed May 20, 2023)

[7] Leap Motion, Inc., Leap Motion Controller. Available: https://www.ultraleap.com/ (accessed May 20, 2023)

[8] Z. Cao, G. Hidalgo, T. Simon, S. Wei, and Y. Sheikh, "OpenPose: realtime multi-person 2D pose estimation using Part Affinity Fields," In arXiv preprint arXiv:1812.08008, 2018.

[9] F. Zhang, V. Bazarevsky, A. Vakunov, A. Tkachenka, G. Sung, C. Chang, and M. Grundmann, "MediaPipe Hands: On-device Real-time Hand Tracking," Fourth CVPR Workshop on Computer Vision for Augmented and Virtual Reality, rXiv:2006.10214, 2020.

[10] Google, "MediaPipe." Available: https://google.github.io/mediapipe/ (accessed May 20, 2023)

[11] R. Hamasaki and K. Nakayama, "A deep learning system that learns a discriminative model autonomously using difference images," Proceedings of the Genetic and Evolutionary Computation Conference Companion, ACM, pp. 1683–1685, 2019.

[12] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," Proceedings of the IEEE conference on computer vision and pattern recognition, pp. 770-778, 2016.

[13] ResNet50. Available: https://jp.mathworks.com/help/deeplearning/ref/resnet50.html;jsessionid=c2d1fcfb1eb58ff18ab9a8beff0c (accessed May 20, 2023)

[14] ImageNet. Available: https://www.image-net.org/ (accessed May 20, 2023)

[15] B. Li and D. Lima, "Facial expression recognition via ResNet-50," International Journal of Cognitive Computing in Engineering, vol. 2, pp. 57-64, 2021.

[16] S. Bhattacharyya, "Understand and Implement ResNet-50 with TensorFlow 2.0," Towards Data Science. Available: https://towardsdatascience.com/understand-and-implement-resnet-50-with-tensorflow-2-0-1190b9b52691 (accessed May 20, 2023)

[17] C. Chen, W. Zhu, J. Steibel, J. Siegford, J. Han, et al., "Classification of drinking and drinker-playing in pigs by a video-based deep learning method," Biosystems Engineering, vol. 196, pp. 1-14, 2020.

[18] D. Pelleg and A. W. Moore, "X-means: Extending k-means with efficient estimation of the number of clusters," Proceedings of ICML 2000, vol. 1, pp. 727-734, 2000.

[19] P. S. Bradley and U. M. Fayyad, "Refining initial points for k-means clustering," Proceedings of ICML 1998, vol. 98, pp. 91–99, 1998.

[20] imgsim. Available: https://github.com/Nr90/imgsim (accessed May 20, 2023)

[21] S. Garrido-Jurado, R. Muñoz-Salinas, F. J. Madrid-Cuevas, and M. J. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," Pattern Recognition, vol. 47, no. 6, pp. 2280-2292, 2014.

[22] Logicool, "C270 web camera." Available: https://www.logitech.com/en-eu/products/webcams/c270-hd-webcam.960-001063.html (accessed May 20, 2023)