3D Hand Pointing Recognition over a Wide Area using Two Fisheye Cameras

Azusa Kuramochi and Takashi Komuro(✉)

Saitama University, Saitama, Japan
komuro@mail.saitama-u.ac.jp

Abstract. In this paper, we propose a user interface for large displays that allows pointing operations from a wide area. Two fisheye cameras installed on both sides of the display are used to capture images of a wide area in front of the display, and the system recognizes the user's skeleton and allows the user to perform pointing operation. Due to the wide viewing angle of the fisheye cameras, the baseline length between the cameras can be long. The experimental result showed that the pointing accuracy was higher with the fisheye cameras than with the standard cameras when operating at the same distance.

Keywords: Gesture recognition · Panoramic images · Pose estimation

1 Introduction

In recent years, large touch panel displays are becoming widespread, making it possible to provide interactive information in public places. However, there is a problem that a touch panel can only be used if when the user is near the display. Moreover, in the case of large displays, there are problems such as inaccessible places and the need to move large distances for operation.

On the other hand, systems that allows remotely selecting objects on a display by hand pointing have been proposed. However, some of them use markers [1–3] or devices [4, 5] for hand posture recognition, and a user need to wear them on his/her hand. Some other systems recognize hand pointing using cameras installed in the environment, such as on a wall or a ceiling [6, 7]. This makes it difficult to move the system, and also camera calibration is required after installation. There are some systems that use only a camera/cameras [8, 9] or a Kinect sensor [10] installed around the display, but the recognizable range is limited to the camera's field of view, and users have to perform operation within the range.

In this paper, we propose a user interface for large displays that allows pointing operations from a wide area. Two fisheye cameras installed on both sides of the display are used to capture images of a wide area in front of the display, and the system recognizes the user's skeleton and allows the user to perform pointing operation based on the information.

C. Stephanidis et al. (Eds.): HCII 2021, LNCS 13095, pp. 58–67, 2021.
https://doi.org/10.1007/978-3-030-90963-5_5

2 3D Pointing Recognition from Fisheye Camera Images

2.1 System

Figure 1 shows the configuration of the system and a user operating the system. The system utilizes two fisheye cameras to recognize a user's 3D hand pointing over a wide area. A cursor is displayed at the point where the user extended her arm toward the display.

Since the user's pointing operation is recognized using fisheye cameras, the user can operate the system from a wider range than using normal cameras. Figure 2 shows the field of view using standard cameras and that using fisheye cameras. Due to the wide viewing angle of the fisheye cameras, the baseline length between the cameras can be long, which would increase the positional accuracy in the depth direction. In this system, the two fisheye cameras were put on both sides of the display.

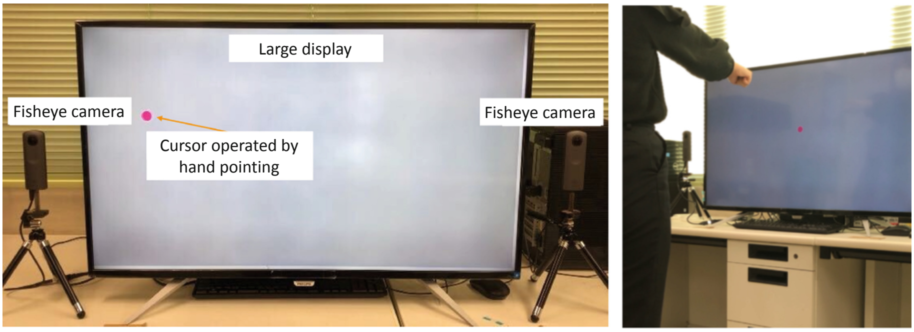


Fig. 1. Configuration of the system (left) and a user operating the system (right).

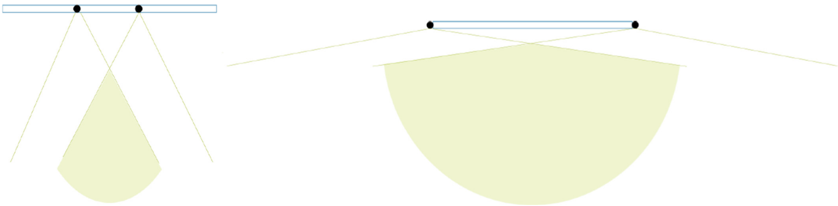


Fig. 2. Field of view using standard cameras (left) and that using fisheye cameras (right).

2.2 Hand Pointing Recognition

The system recognizes the user in front of the display from the omnidirectional image captured by the fisheye camera. First, the images acquired by the fisheye cameras are converted to those in the equirectangular format, which are represented by (θ, ϕ) coordinates. In this study, we used RICOH THETA V for the fisheye cameras, which has two fisheye cameras, one on the front and one on the back, and can output

360-degree panoramic images in the equirectangular format. However, we used only 180-degree part of the panoramic images that is captured by the front fisheye camera.

Figure 3 shows an example of an image captured by the fisheye camera, and the image converted to that in equirectangular format. The images captured by the fisheye cameras cover a large indoor area and can recognize a user's operations from a wide range.

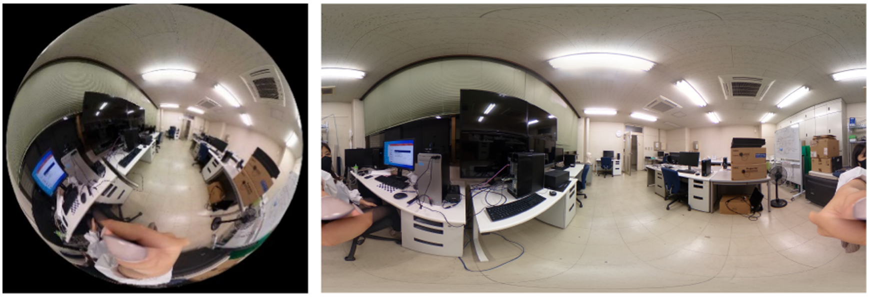


Fig. 3. An example of an image captured by the fisheye camera (left), and that in equirectangular format (right).

Next, the user's skeleton is extracted from the equirectangular images from the left and right fisheye cameras, respectively, using OpenPose [11]. OpenPose is an implementation of deep-learning-based multi-person pose estimation. An example of extracted skeleton from left and right equirectangular images is shown in Fig. 4.

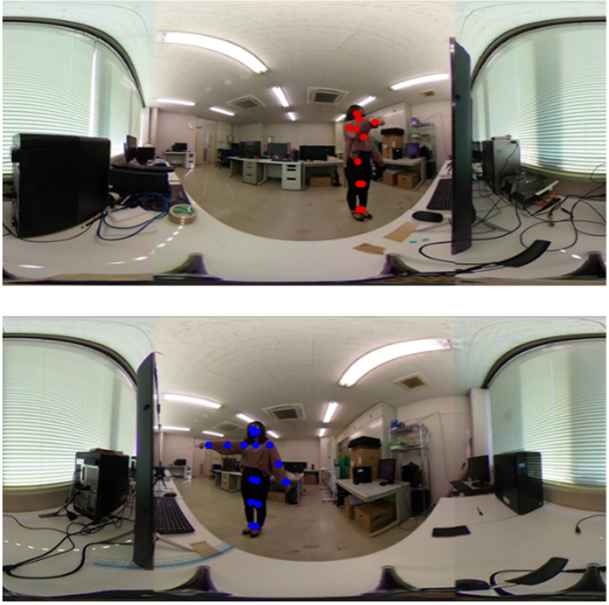


Fig. 4. An example of extracted skeleton from left and right equirectangular images.

The three-dimensional positions of the elbow and wrist of the user's right arm are used to calculate the pointing position. First, the coordinates (θ, ϕ) of each joint point in an equirectangular image is converted into plane coordinates (x, y) using the following equations. f is the focal length of the virtual camera and can be set to any value.

$$x = f\frac{\sin\,\theta\cos\phi}{\cos\,\theta}, \quad y = f\frac{\sin\,\theta\sin\,\phi}{\cos\,\theta}$$

Next, the 3D coordinates (X, Y, Z) of each joint are calculated from the planar coordinates of the corresponding joint points (x_l, y_l) and (x_r, y_r) in the left and right camera images using the following equations of parallel stereo. b is the baseline length between the left and right cameras.

$$X = \frac{b(x_l + x_r)}{2(x_l - x_r)}, \quad Y = \frac{b(y_l + y_r)}{2(x_l - x_r)}, \quad Z = \frac{bf}{x_l - x_r}$$

As shown in Fig. 5, a cursor is displayed at the point where the straight line passing through the 3D coordinates of the elbow (X_1, Y_1, Z_1) and wrist (X_2, Y_2, Z_2) intersects the display. By doing so, the cursor is displayed at the point where the user extends his/her arm.

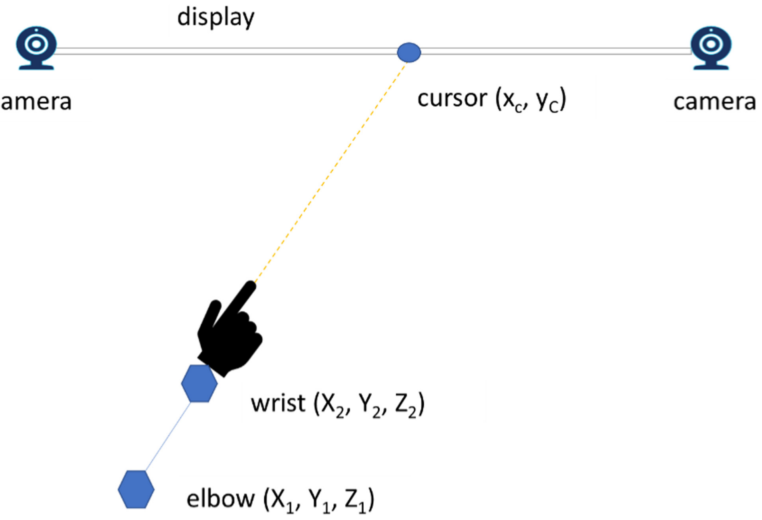


Fig. 5. Field of view using standard cameras (left) and that using fisheye cameras (right).

The coordinates of the cursor (x_c, y_c) are calculated by the following equations.

$$x_c = X_1 + k(X_2 - X_1), \quad y_c = Y_1 + k(Y_2 - Y_1), \quad k = -\frac{Z_1}{Z_2 - Z_1}$$

Since the cursor coordinates are calculated in a coordinate system in real space, they are converted to that in the display coordinate system. Figure 6 shows the 3D positions of the wrist and elbow, and the 2D cursor position calculated from the left and right equirectangular images in Fig. 6.

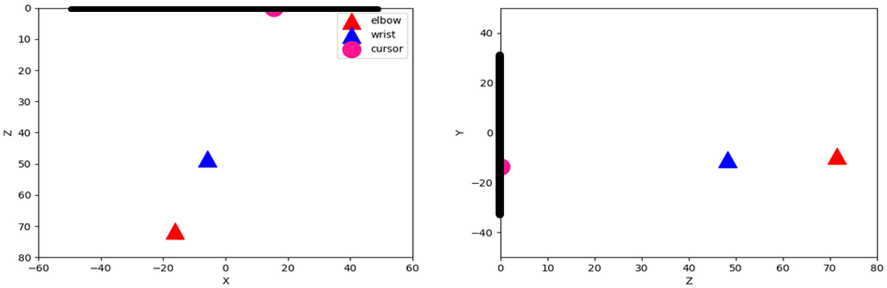


Fig. 6. 3D positions of the wrist and elbow, and the 2D cursor position.

3 Performance Evaluation

We conducted an experiment to evaluate the pointing accuracy of the proposed system.

3.1 Procedure

Fifteen circular objects with a radius of 5 cm were presented on a 43-inch display, and we asked six participants to select each object by hand pointing with their right hand. The object that the participant had to point at was changed every 10 min and was indicated in blue. The participants performed operation at the center and 60 cm left/right positions, 80 cm away from the display, and at the center, 140 cm away from the display, respectively. We set the baseline length between the two cameras to 120 cm. For comparison, we also performed recognition using two standard USB cameras a baseline length of 30 cm. The participants performed operation only 140 cm away from the display due to the narrow field of view. Figure 7 shows a participant performing operation and the participants' standing positions.

The fisheye cameras had a resolution of 1920 × 960 pixels. The standard cameras had a resolution of 1920 × 1080 pixels. The frame rate of both the fisheye cameras and standard cameras was around 10 fps.

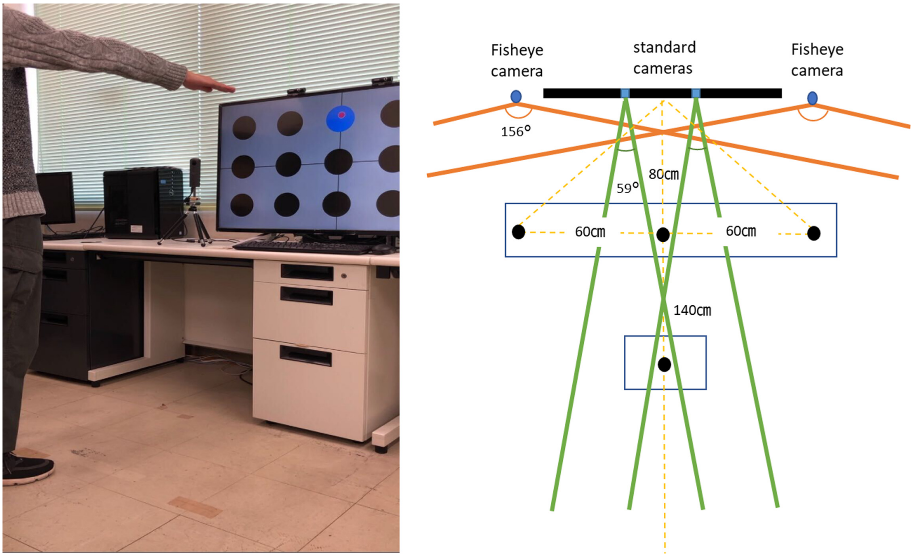


Fig. 7. A participant performing operation (left) and the participants' standing positions (right).

The following exponential smoothing filter was applied to the cursor positions to reduce jittering.

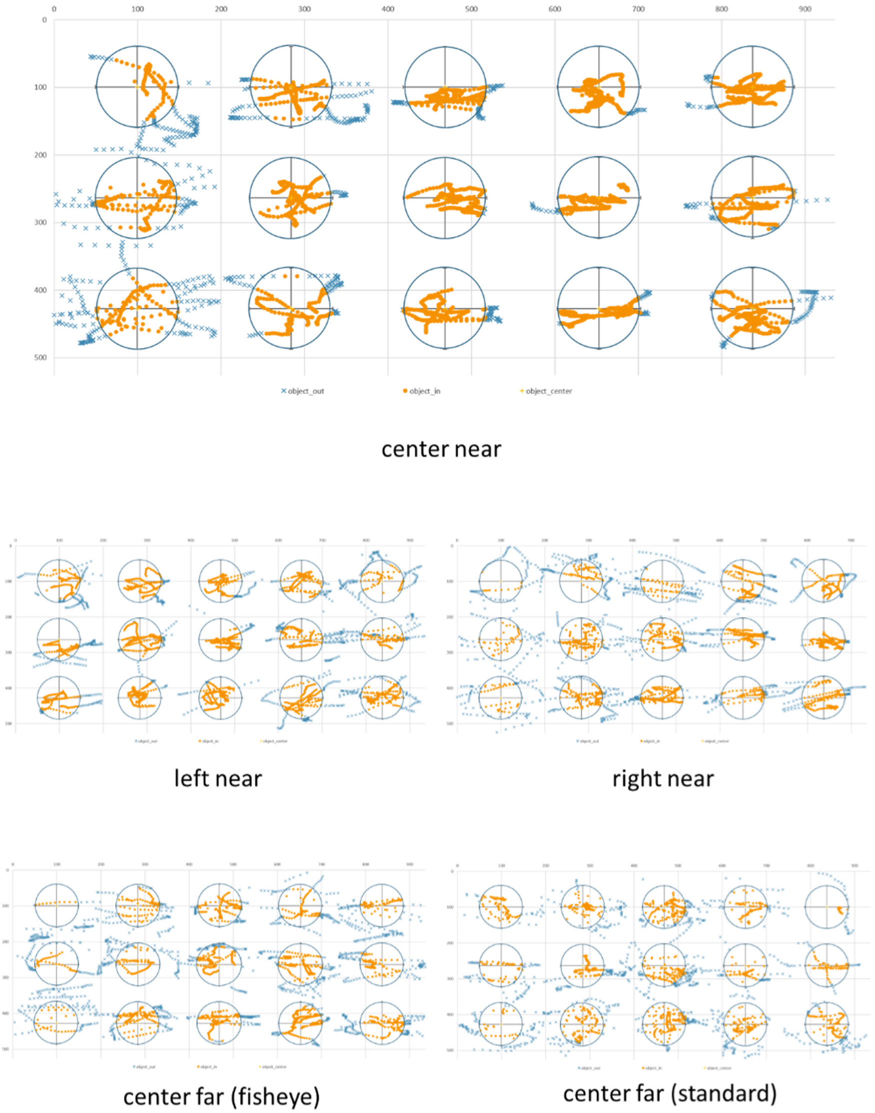
$$y_k = \alpha y_{k-1} + (1 - \alpha)x_k$$

We set the filter parameter α to 0.95. If the skeleton extraction by OpenPose fails, the value of the previous output y_{k-1} was used as the output y_k.

3.2 Results

Figure 8 shows the scatter plot of pointer coordinates when the participants were at the center, 80 cm away from the display. The last 50 pairs of coordinates in the 10 s of selecting one object were taken. The total of 300 pairs of pointer coordinates of six participants were taken for one object. The pointers that were inside the target object were plotted in orange, and those outside the target object were plotted in light blue. The pointers whose distances from the center of the target were more than 10 cm were not plotted.

Figure 9 shows the success rate for each operating position and object position, and Fig. 10 shows the number of failures in skeleton extraction. The success rate was over 90% for objects close to the operating position, but tended to drop as the object for objects farther away from the operating position. Also, operations near the display had a higher success rate than those away from the display.



center near

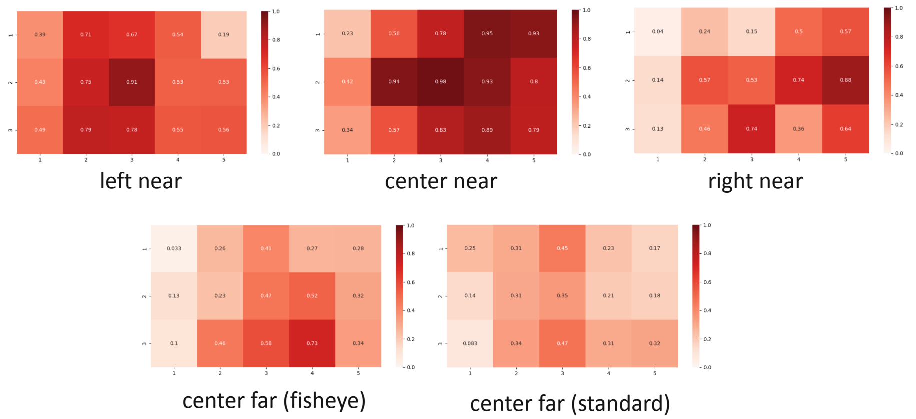
left near

right near

center far (fisheye)

center far (standard)

Fig. 8. Scatter plot of pointer coordinates for each operating position and object position.



left near

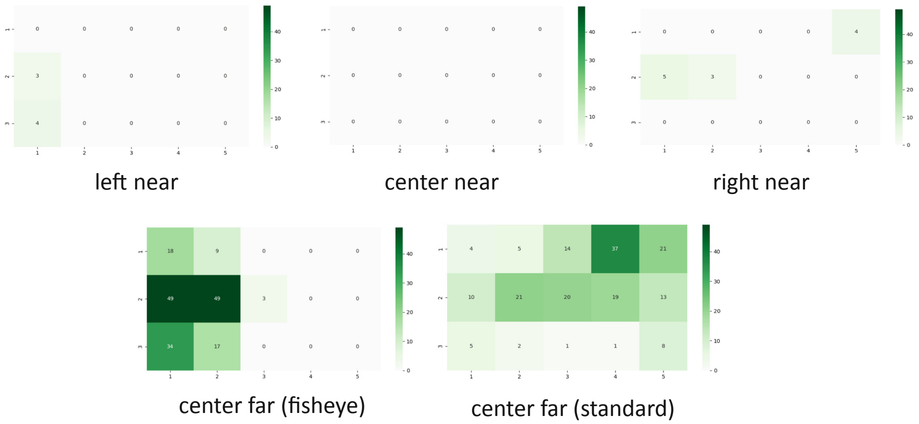
center near

right near

center far (fisheye)

center far (standard)

Fig. 9. Success rate for each operating position and object position.



left near

center near

right near

center far (fisheye)

center far (standard)

Fig. 10. Number of failures in skeleton extraction for each operating position and object position.

3.3 Discussion

The experimental results showed that the pointing accuracy of operations away from the display was lower than that of operations near the display. This is because the size of the user in the camera image becomes smaller as the user moves away from the camera, and the accuracy of skeleton extraction by OpenPose decreases.

However, even when operating at the same distance, the results were slightly more accurate with the fisheye cameras than with the standard cameras. This may be because the fisheye cameras have a longer baseline length, which results in a higher depth accuracy. Fisheye cameras have a wider field of view than standard cameras and the baseline length can be set longer, which is one of the advantages of using fisheye cameras.

The failure in skeleton extraction often occurs when the elbow and wrist appear to overlap in the camera image. As shown in Fig. 10, the number of failures increased when users are pointing at the target near the camera. Since the fisheye camera were installed on left and right sides outside the display, the number of failures was larger in the peripheral area of the display. On the other hand, the standard cameras were installed near the center top of the display and the number of failures was larger around that area. It would be effective to apply a method to compensate for the failure of skeleton extraction in one camera image with the extraction results of the other camera image.

The overall pointing accuracy in this experiment was not very good. This is due to the low positional accuracy of OpenPose’s skeleton extraction, which resulted in shifting between frames. The accuracy would be improved by a combined method such as using OpenPose to detect rough joint positions and then aligning the positions by object tracking.

4 Conclusion

In this paper, we proposed a user interface that allows a user to perform operations by hand pointing from a wide area. By using two fisheye cameras, the user's operating space is extended compared to that using standard cameras, and by extracting the user's skeleton points and used them for pointing recognition, the user can perform remote operation without using a device.

In the proposed method, the skeleton points are extracted from the equirectangular images captured by two fisheye cameras and the 3D coordinates of each point are calculated. A cursor is displayed at the point where the straight line passing through the 3D coordinates of the elbow and wrist intersects the display. We implemented the method above and showed that a user can perform pointing operation.

There are three issues to be addressed in the future. The first is the simultaneous operation by multiple people. In this study, we assumed that there was only one person in the camera images, and the persons in the images acquired by the two cameras could be regarded as the same person. However, when there are multiple people in the images at the same time, it is necessary to recognize the same person in the left and right camera images.

The second is to improve the pointing accuracy. Possible solutions include compensating for false detection in one camera image with the other camera image, or a combination of skeleton extraction and object tracking.

The last is the variety of operations. In our method, the cursor is displayed and the user can only move the cursor to the object. Since OpenPose can recognize not only skeleton points of the body but also those of fingers, it would also be possible to make the system recognize operations using hand gestures.

References

1. Jota, R., Nacenta, M., Jorge, J., Carpendale, S., Greenberg, S.: A comparision of ray pointing techniques for very large displays. In: Proceedings of Graphics Interface, pp. 269–276 (2010)
2. Rateau, H., Rekik, Y., Grisoni, L., Jorge, J.: Talaria: Continuous drag & drop on a wall display. In: Proceedings of the ACM International Conference on Interactive Surfaces and Spaces, pp. 199–204 (2016)
3. Matulic, F., Vogel, D.: Multiray: multi-finger raycasting for large displays. In: Proceedings of the CHI Conference on Human Factors in Computing Systems, Article No. 245 (2018)
4. Pietroszek, K., Tahai, L., Wallace, J., Lank, E.: Watchcasting: freehand 3D interaction with off-the-shelf smartwatch. In: Proceedings of IEEE Symposium on 3D User Interface, pp. 172–175 (2017)
5. Haque, F., Nancel, M., Vogel, D.: Myopoint: pointing and clicking using forearm mounted electromyography and inertial motion sensors. In: Proceedings of the Annual ACM Conference on Human Factors in Computing Systems, pp. 3653–3656 (2015)
6. Schick, A., van de Camp, F., Ijsselmuiden, J., Stiefelhagen, R.: Extending touch: towards interaction with large-scale surfaces. In: Proceedings of ACM International Conference on Interactive Tabletops and Surfaces, pp. 117–124 (2009)
7. Hu, K., Canavan, S., Yin, L.: Hand pointing estimation for human computer interaction based on two orthogonal-views, In: Proceedings of International Conference on Pattern Recognition, pp. 3760–3763 (2010)
8. Matsuda, Y., Komuro, T.: Dynamic layout optimization for multi-user interaction with a large display. In: Proceedings of the International Conference on Intelligent User Interfaces, pp. 401–409 (2020)
9. Endo, Y., Fujita, D., Komuro, T.: Distant pointing user interfaces based on 3D hand pointing recognition. In: Proceedings of ACM International Conference on Interactive Surfaces and Spaces, pp. 413–416 (2017)
10. Makela, V., James, J., Keskinen, T., Hakulinen, J., Turunen, M.: It's natural to grab and pull: retrieving content from large displays using mid-air gestures. IEEE Pervasive Comput. **16** (3), 70–77 (2017)
11. Cao, Z., Hidalgo, G., Simon, T., Wei, S., Sheikh, Y.: OpenPose: realtime multi-person 2D pose estimation using part affinity fields. IEEE Trans. Pattern Anal. Mach. Intell. **43**(1), 172–186 (2021)