OddEyeCam: A Sensing Technique for Body-Centric Peephole Interaction Using WFoV RGB and NFoV Depth Cameras

Daehwa Kim **Keunwoo Park** **Geehyuk Lee**

HCI Lab, School of Computing, KAIST, Daejeon, Republic of Korea
{daehwakim, keunwoo}@kaist.ac.kr, geehyuk@gmail.com

ABSTRACT

The space around the body not only expands the interaction space of a mobile device beyond its small screen, but also enables users to utilize their kinesthetic sense. Therefore, body-centric peephole interaction has gained considerable attention. To support its practical implementation, we propose OddEyeCam, which is a vision-based method that tracks the 3D location of a mobile device in an absolute, wide, and continuous manner with respect to the body of a user in both static and mobile environments. OddEyeCam tracks the body of a user using a wide-view RGB camera and obtains precise depth information using a narrow-view depth camera from a smartphone close to the body. We quantitatively evaluated OddEyeCam through an accuracy test and two user studies. The accuracy test showed the average tracking accuracy of OddEyeCam was 4.17 and 4.47cm in 3D space when a participant is standing and walking, respectively. In the first user study, we implemented various interaction scenarios and observed that OddEyeCam was well received by the participants. In the second user study, we observed that the peephole target acquisition task performed using our system followed Fitts' law. We also analyzed the performance of OddEyeCam using the obtained measurements and observed that the participants completed the tasks with sufficient speed and accuracy.

Author Keywords

Spatially-aware display; Mobile device; Sensing; 3D; Peephole display; Body-centric Interaction;

CCS Concepts

•**Human-centered computing** → **Interaction devices;**

INTRODUCTION

Nowadays, mobile devices have become platforms for increasing number of diverse applications. However, their small screens make it difficult for users to use information-rich applications and switch between the applications [9].

UIST '20, October 20–23, 2020, Virtual Event, USA

ACM ISBN 978-1-4503-7514-6/20/10 ...$15.00.
http://dx.doi.org/10.1145/3379337.3415889

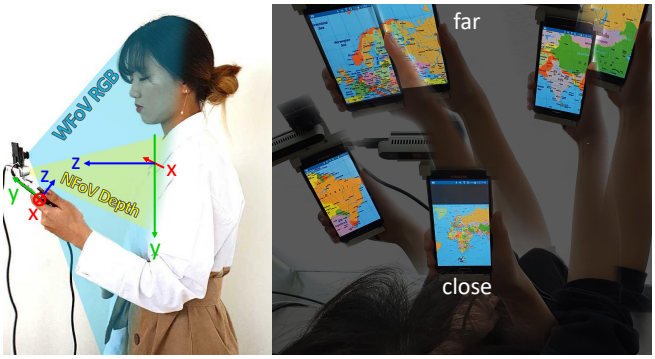

Figure 1. OddEyeCam enables the mobile device tracking of the body of user by combining a wide vision for body tracking and partial depth information (left). Our system supports various body-centric peephole interaction scenarios. An example of world map is shown in Figure 9 (right). Six images were acquired from the same view-point and were blended but not otherwise edited.

Among the many efforts to overcome the limitations due to small screens of mobile devices, the interaction concept of expanding the interaction space to vicinity of the human body [13, 14, 18, 33, 65] has been extensively studied. For instance, this expansion enables users to peep into a large virtual workspace by using the arm-reachable space around their bodies. Additionally, it lets users increase spatial memory recall of virtual information [58], reduce cognitive load when they interact with multiple information sources in a mobile context, and switch applications quickly by utilizing spatial memory and kinesthetic cues [13].

Various approaches have been considered to support such spatial interactions. They range from outside-in tracking methods with an external-vision system or electromagnetic-tracking equipment to inside-out tracking methods with embedded cameras or inertial measurement unit (IMU) sensors. However, the approaches have their own limitations. The outside-in tracking method is difficult to support the portability of mobile devices. The IMU-based approach, which is one of the inside-out tracking methods, tracks the device position relative to the initial position. This approach does not track the user body position; therefore, reliable tracking of the positions in body-centered interfaces is difficult when the user moves. Another approach is face tracking. According to recent neuropsychological studies [3, 52], a human perceives the trunk-centered space as a whole body-centric space rather than the head-centered space. However, face tracking cannot support body-centered interfaces when the face and body are misaligned. Furthermore, another approach is to track external features using a rear

camera. However, this approach cannot support body-centric interfaces because it does not know the body position.

To solve the limitations of the previous approaches, we propose OddEyeCam. Analogous to an odd-eyed cat with two eyes of different colors, we combine the following two types of cameras with different characteristics: an RGB camera with a wide field of view (WFoV) and a depth camera with a narrow field of view (NFoV). Despite the useful function of the depth camera, its field of view is too narrow to capture a peripheral scene [41, 42]. Because an NFoV camera misses a significant portion of a body when it is close to the body, we require a WFoV RGB camera for reliable body tracking. OddEyeCam finds body keypoints from a WFoV RGB image and uses partial depth information from a depth camera to track the body and estimate the device position relative to the body. Recently, many mobile devices have included a ToF depth camera [25, 27, 31, 38, 47, 51] and WFoV RGB camera [4, 5, 32, 46, 48]. OddEyeCam is expected to be a practical method to enable body-centric peephole interaction for mobile devices.

The contribution of this study is a practical inside-out mobile device tracking method to support body-centric peephole interaction. OddEyeCam (1) estimates the absolute device location with respect to the body, (2) provides a wide and continuous interaction space, and (3) enables robust position sensing without imposing restrictions on the user action in (4) both static and mobile environments.

Through three experiments, we verified that our system offers the above-mentioned advantages. We implemented the OddEyeCam prototype and quantitatively evaluated its 3D-tracking accuracy and usability. First, through an accuracy test, we evaluated the extent of accuracy to which OddEyeCam could estimate the x/y/z position of a mobile device, as compared with the OptiTrack tracking system. Second, in our first user study, we evaluated whether our system could sufficiently support various body-centric peephole interfaces and whether it could overcome the limitations of previous studies. Third, we quantitatively characterized the usability of OddEyeCam by performing target-selection tasks through a peephole.

In the rest of this paper, we review the related work and present the requirements and design goals for OddEyeCam derived from the limitations of earlier work. Subsequently, we describe the OddEyeCam prototype and the results of the associated user studies. Finally, we discuss the limitations and future work of OddEyeCam.

RELATED WORK

The target application domains of OddEyeCam are peephole and body-centric interfaces. We review these interfaces and then describe the technical approaches required to realize them.

Peephole and Body-Centric Interfaces

Using spatial information as input has drawn significant attention for the past 20 years [19, 28, 54]. Peephole interfaces enable users to move the handheld device in a 3D physical space and use it as a window to view into a virtual space larger

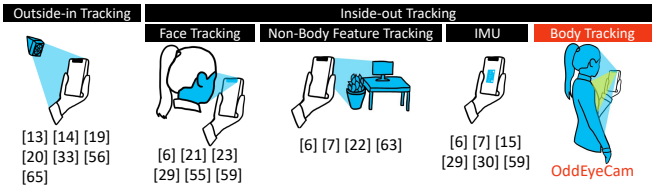

Figure 2. Overview of the technical approaches for mobile device tracking.

than the screen. Spindler et al. showed that spatial manipulation could significantly outperform traditional pinch-drag-flick [56]. Moreover, body-centric interfaces, which map the virtual workspace to the physical location around a user, offer additional benefits. They enable users to utilize their kinesthetic sense to increase in spatial memory recall [58] and can perform mobile context switching tasks quickly and accurately with a low cognitive load [13]. They even enable eyes-free shortcut usage around the body [33].

These above-mentioned studies discussed a common interaction space: peripersonal space, which is the space that surrounds a human body and is easily reachable by human hand [45, 61]. People can perceive their peripersonal spaces using proprioception, as well as vision and hearing. The peripersonal space not only extends the interaction space beyond the small screen but also allows users to utilize proprioception while offering the aforementioned benefits. Therefore, many studies proposed a body-centric interface [13, 14, 15, 33, 53]. A recent neuropsychological study showed that a human could quickly form a peripersonal space for each body part, and that a trunk-centered peripersonal space is the most similar to the whole-body egocentric space representation [52]. Additionally, people judged the egocentric orientation using torso-centered reference frame rather than head-centered reference frame [3]. Notably, HCI studies implicitly used the torso as the reference frame of the user. Peephole displays [65] placed a calendar application and Doodle pad on the left and right sides of the torso, respectively, to locate the applications for the personal reference frame of the user. MultiFi [20] proposed a method of placing an additional sensor in the chest pocket of the user to determine the relative position of the handheld touchscreen to the body.

Technical Approaches for Mobile Device Tracking

Many studies used the outside-in tracking methods by installing equipment including camera [39, 62, 64] and electromagnetic system [16] in the surrounding space of the user, followed by attaching a marker to the mobile device. Inside-out tracking systems, which track the mobile device using sensors inside the mobile device itself, have been considered, and we review them in the following section.

IMU-Based

One of the inside-out tracking methods is to use the motion data obtained using an accelerometer, gyro sensor, and magnetometer. One possible way is to integrate the relative motion of the IMU sensor (e.g., odometery); however, it creates drifts because of sensor biases [29, 43, 50]. Although a reliable position cannot be obtained owing to the drift, there exists a stable reference frame for estimating the orientation of the device. The magnetic field of the earth can be used for the

magnetometer, and the gravitational field can be used for the gyro sensor and accelerometer, and they were utilized for realizing a spatially-aware display [15, 30, 59]. However, the position of the device could not be obtained because these reference frames only provide a rotational reference for the device. Accordingly, Chen et al. [15] remarked that users naturally tilt the device to make view it when they move it. Thus, they estimated position of the the device using its orientation. However, Chen et al. noted that their estimation technique was limited because such tilting did not always perfectly align with the true around-body location of the device in reality [15]. Environmental features can provide a stable reference frame for the position tracking of IMUs and help this approach overcome drift problems (e.g., ARKit 3 by Apple, and ARCore by Google). However, the method still does not track the body position of the user. We will detail the method in the "Non-Body Feature Tracking" section.

Face Tracking

Another approach is to capture the face image using the front camera of the mobile device. For instance, the motion of a facial feature or face bounding box was tracked on a 2D image to estimate the device position relative to the face [23, 29, 55]. Hannuksela et al. estimated the 3D location of a device by modeling the face of the user as a plane in the 3D space [21]. Because the face was used as the reference frame, the user is prohibited from rotating her face to obtain body-centric contents. However, her face naturally rotates when she views the screen by moving the device [15], and the head can thus be misaligned with the body [6]. Because face tracking does not consider the body location, it cannot provide body-centric contents. Additionally, a user naturally tilts the screen to view it while moving the device. In this case, the face image acquired using the front camera is still in the center, even if the device moved. Therefore, no change occurs in the motion of the face features on the 2D image.

Non-Body Feature Tracking

Another vision-based approach involves non-body feature tracking, in which the data of environmental features is acquired using a rear camera [22, 63] and then combined with the IMU data [6, 7] (e.g., SLAM). This approach is not appropriate in mobile environments because the environmental features are changing. Additionally, this approach relatively accumulates the motion data from the initial position, not the body position of the user. Thus, this approach cannot support instantaneous use cases without initial-position calibration step. Babic et al. [6, 7] attempted to automatically perform this calibration by checking whether the device exited the control space. However, a trade-off existed between the interaction-space size and reliable recalibration. The control-space size for Pocket6 [7] was $16\times16\times9$ cm, which is insufficient for body-centric interactions. Additionally, the control space has to be repositioned for every 16 cm of movement while the user is walking.

REQUIREMENTS AND DESIGN GOALS OF ODDEYECAM

In this section, we present the requirements and design goals of OddEyeCam while considering the limitations of earlier works.

The four requirements are:

1. **Absolute location estimation**: OddEyeCam should estimate the absolute location of the mobile device relative to the body while not integrating the relative movements of the device from a initial position.
2. **Wide and continuous interaction space**: To cover various interaction scenarios, OddEyeCam has to continuously track the device location in a wide space around the body.
3. **Robust position tracking without restriction on the action of the user**: The behavior of a user should not be restricted for achieving high tracking performance. Therefore, our system must robustly estimate the location of the mobile device relative to the body.
4. **Usable in a walking situation**: To completely support the portability of mobile devices, OddEyeCam must track the device position relative to the body while the user is moving.

In the evaluation, we confirmed that our method satisfies the above-mentioned requirements. Some requirements, such as "estimating the device position in an absolute manner," are self-evident from the implementation. However, some requirements must be evaluated by the user. We set the design goals as follows to check whether the requirements were satisfied.

1. OddEyeCam enables a user to feel that the content is in a fixed position relative to the body ***even when it was instantaneously used***.
2. OddEyeCam enables a user to feel that the content is in a fixed position relative to the body ***for long usage periods***.
3. OddEyeCam ensures that a user feels that the body-fixed content ***appeared as expected*** even when she moves her mobile device.
4. OddEyeCam makes a user feel ***a continuous movement*** of the content.
5. OddEyeCam makes a user feel ***comfortable with her eyes and wrist*** during the usage period.
6. OddEyeCam enables a user to feel that the content is in a fixed position relative to the body even ***when she is walking***.

We implemented the OddEyeCam system on the basis of these requirements. The design goals were quantitatively verified in the user studies.

ODDEYECAM IMPLEMENTATION

OddEyeCam estimates the 3D location of a smartphone with respect to the body via the process depicted in Figure 3. 1) A WFoV RGB camera provides a whole-body image. 2) An RGB-D camera provides partial body depth and RGB image. 3) An distortion-alleviation module reduces the distortion of the fisheye image through an equirectangular projection. A body-tracking algorithm provides body keypoints from the undistorted image. 4) The body keypoints found in the WFoV image can be projected to the NFoV depth image by combining two cameras. 5) Using keypoints and depth information, as well as additional gravity vector from an accelerometer, we can estimate the body coordinate system. 6) We can obtain the device location with respect to the body by converting the body position (obtained in Step 5) with respect to the camera. The following section detail the implementation of OddEyeCam.

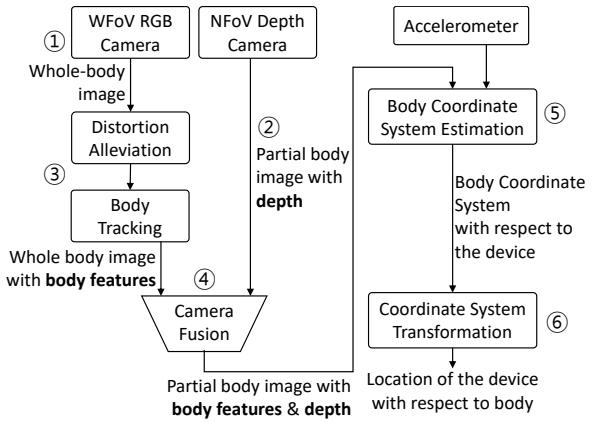

Figure 3. Overall pipeline of OddEyeCam.

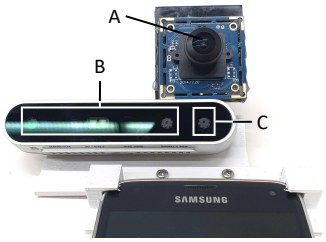

Figure 4. OddEyeCam prototype comprises a WFoV RGB camera (A) and an NFoV RGB-depth camera (C, B).

Hardware Configuration

We used a 180° fisheye-eye lens USB camera as the WFoV RGB camera. Intel RealSense D415 was chosen as the NFoV depth camera in our prototype. The depth camera comprised two IR stereo (65°×40°) cameras and an RGB (69.4°×42.5°) camera. The camera resolutions were 480×270 (depth) and 424×240 (RGB) pixels for Realsense D415 in our settings. The WFoV RGB image was 640×480 pixels in resolution. As shown in Figure 4, the WFoV RGB camera and RGB camera of D415 were vertically aligned with each other. The distance between both lenses was 3.1cm. The cameras were attached to the top-center of the mobile device. To capture the image of the torso, the cameras were tilted at 40° with respect to the mobile device. We used a Samsung Galaxy S5, whose touchscreen had the size of 5.10-inch and the resolution of 1080×1920 pixels. The cameras were connected to a computer via USB cables.

Fusion of the NFoV Depth and WFoV RGB Cameras

The purpose of camera fusion is to find pixels on a depth image that the pixels match with body keypoints on a WFoV RGB image. Perez et al. [41, 42] proposed a similar idea of combining these two cameras, although the implementation and applications were different from ours. A camera model can determine the projected 2D position of 3D points. Therefore, they mapped the depth image with 3D information to a 2D WFoV image. Our system is interested in the depth information of the body keypoints found on a WFoV image, which is the reversed order of what Perez et al. did. Therefore, our aim cannot be simply realized using their method. Thus, We implemented our system to combine cameras, which provides a direct one-to-one conversion map for each pixel.

Figure 5. Fisheye image from WFoV RGB camera (left) and the corresponding undistorted image (right)

Undistortion of Fisheye Image

A fisheye camera uses a different camera model, while a normal RGB camera uses a pinhole camera model, i.e., the perspective projection of 3D objects. We had straighten the distortion of a WFoV RGB image like a normal FoV RGB image. Thus, we calibrated the fisheye camera and transformed the WFoV image via perspective projection. We obtained intrinsic parameters using a fisheye-camera-calibration tool in OpenCV. They enabled us to obtain a perspective projection image shown in Figure 5 (right).

Matching a WFoV Image to a Depth Image

We employed homography, which can transform a 2D image to another 2D image in the 3D space, to obtain the pixel relationship between WFoV and depth cameras. Two cameras were attached 3.1 cm apart from each other, and thus there existed difference between the image locations for an object seen by two cameras (i.e., binocular disparity). We set a range of comfortable arm movements as $20 \sim 60$ cm from the body. Subsequently, we matched two images to minimize the disparity at the center of a comfortable range, not the farthest position. We placed a checkerboard sheet at a distance of 40 cm from the cameras and obtained the homography matrix by using the features on the surface taken two cameras. We used AKAZE [1, 2] as a pattern-matching algorithm, and removed the outliers using RANSAC. The undistorted WFoV image was scaled, translated, and rotated to be matched with the NFoV image by using homography. After the matching, we measured the disparity between two image locations of an "X" mark captured from two cameras at different distances of 20, 40, and 60 cm from the cameras. The disparities of the mark were +34, 0, and -7 pixels for each distance aforementioned, respectively. The physical length of 34 pixels at 20 cm away from the camera was 2 cm, which is a small length. This means when the camera is 20 cm apart from the body, OddEyeCam can find the body keypoints on a depth image 2 cm vertically different position with the WFoV RGB image.

Mapping Body Keypoints Between Two Images

Our pipeline alleviates the distortion of a fisheye image by converting it to an equirectangular-projection image and uses Openpose [11] for estimating body keypoints. Applying perspective projection to a fisheye image stretches its periphery and distorts the shape and size of objects as depicted in Figure 5 (right). Therefore, we used an equirectangular projection image for body tracking, as the distortion was alleviated by projecting an image to the planar rectangular coordinate system. We calibrated the fisheye camera and obtained intrinsic parameters using the OCamCalib Toolbox in Matlab [44]. We calculated the target pixel position of the fisheye image using the intrinsic parameters [37]. We constructed a one-to-one

Figure 6. Result of the fusion of WFoV RGB and NFoV depth cameras. The red dots on the images are body keypoints. The upper three images are from the WFoV camera and the lower two from the depth camera.

conversion map between two cameras, and the body keypoints found in the WFoV image were mapped to the depth image in realtime (see Figure 6). When the cameras so close to the body, that some keypoints were not present in the NFoV depth image. We moved those keypoints to the closest pixel locations in the depth image.

Body and Mobile Phone Coordinate System Estimation

OddEyeCam estimates the body position (origin) and orientation (axes) relative to the phone and then generates the body coordinate system. Figure 1 (left) shows the body coordinate system relative to the phone. The x-axis (red line) is directed from the left (−) to right (+) of a user. The y-axis (green line) is directed along the downward direction. The z-axis (blue line) is directed from the posterior (−) to anterior (+) of a body. The origin of the body coordinate system lies at the center of line joining the two 3D points of shoulders. We estimated the x-axis by fitting the 3D points from the left to right shoulder along a line in the 3D space. To accurately estimate the x-axis using the depth information, the outliers were removed using RANSAC. We initially set the z-axis as a normal vector of the torso that was fitted as a plane. However, the human torso was not perfectly orthogonal to an anterior direction. The torso has a curved shape and slightly leans toward the posterior direction. Therefore, first we estimated the y-axis. To express a human standing-up direction as y-axis, we used the gravity direction measured using the built-in accelerometer. The z-axis was estimated as the cross-product of the x-axis and y-axis after the x-axis was rotated to be orthogonal to the y-axis. Finally, we could reversely obtain the smartphone position with respect to the body from the body coordinate system with respect to the smartphone coordinate system.

OddEyeCam operates at 18-23 fps. Most computing time was taken by OpenPose. The lightweight version of the pose-estimation model can further increase the frame rate.

ACCURACY TEST

Our goal was to evaluate whether OddEyeCam could accurately estimate the 3D position of the device relative to the body. The 3D position of the ground-truth was obtained using the OptiTrack motion-capture system. Optical markers were attached to the camera and the torso of the participant. Because our ground-truth should be human standing-up direction, we rotated the y-axis of the marker on the torso to be perpendicular to the ground. The movement of the marker was sampled

Figure 7. Captured image and shoulder keypoints taken from the front (left) and side (right).

Still / Walk	Clothes		EDE (mm)	Cartesian System (mm)			Polar System(mm, °, °)		
				x	y	z	d	θ	Φ
Still	Casual	Mean	41.72	26.27	20.90	15.52	12.23	3.74	2.59
		Std	26.00	23.59	16.62	14.70	12.75	3.27	1.94
	With Coat	Mean	71.78	55.25	26.54	24.26	18.85	7.96	3.10
		Std	59.40	56.32	21.41	26.54	21.11	7.72	2.39
Walk	Casual	Mean	44.76	28.61	22.96	15.16	13.09	4.44	3.13
		Std	30.25	28.19	17.83	15.15	14.76	4.18	2.37
	With Coat	Mean	71.95	54.83	29.70	19.94	21.29	8.32	3.70
		Std	59.27	57.59	23.15	23.47	24.22	8.35	2.83
Still	With Coat (4)	Mean	52.11	39.70	19.76	16.72	14.24	5.85	2.56
		Std	34.32	15.38	16.27	16.27	14.18	5.05	1.96
Walk	With Coat (4)	Mean	61.15	47.83	23.26	16.12	17.63	7.33	3.17
		Std	48.00	48.15	17.31	18.35	21.28	6.96	2.44

Table 1. Distance errors by axes. EDE means Euclidean distance error. The lower two rows are the error results from four participants who wore a coat that did not cover their necks.

at 120 fps. The smartphone position from OddEyeCam, sampled at 30 FPS, were compared with the ground-truth position, which shared the same-time frame.

Task

The participants were requested to freely and evenly move a smartphone in a 3D volume of ± 60°(θ), ± 60°(ϕ) while varying the distance (d) around the body. A participant would rotate her face following the device. Therefore, this range was set in consideration of the possible range of neck rotation [35]. For the distance between the camera and body, d, we asked every participant to move the device as far and close as possible within a convenient range of the arm. We provided the visual aids of θ and ϕ boundaries on the floor and wall, respectively. The participants were asked to look at an icon at the center of the screen while moving the device to consider the movements for body-centric interfaces.

Procedure

We recruited 10 participants (5 females and 5 males) aged between 19 and 33 years from a recruiting board on a university campus. Two sessions were held: standing and walking. For the walking session, a participant walked back and forth for 1.5m and rotated both clockwise and counterclockwise directions for the body at the ends of the path. The participants changed their clothes twice in each session: a casual apparel and a coat. All the participants completed the task while wearing a coat they had brought. Each participant moved a smartphone around her body for 1 min and repeated it thrice.

Results

We collected a total of 30 fps × 60 s × 3 times × 10 participants = 54,000 datapoints for each combination of (walking/standing)×clothes. Figure 8 depicts the Euclidean distance between the estimated x/y/z locations using OddEyeCam and the ground-truth data. Table 1 presents the distance between the locations estimated using OddEyeCam and the ground-truth data for each axis.

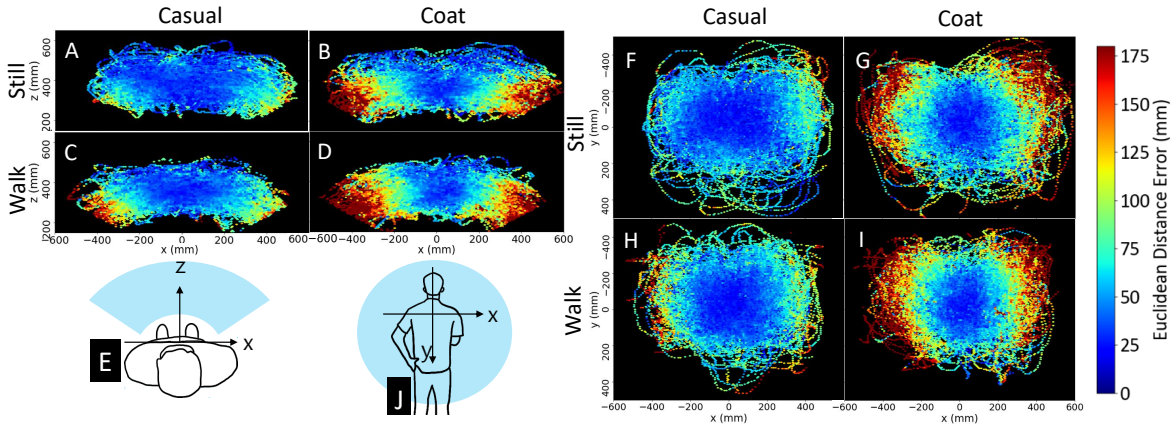

Figure 8. Tracking accuracy (unit: mm) of OddEyeCam. The color bar represents the Euclidean distance between the result of OddEyeCam and the ground-truth data. Figures A, B, C, and D visualize the error in the blue space of Figure E. Figures F, G, H, and I visualize the error in the blue space of Figure J.

The Euclidean distance error of OddEyeCam was 41.72mm (σ=26.00) for the standing session and 44.76mm (σ=30.25) for the walking session. Whether the participants walked or not did not have a significant effect on the results of our system. The tracking accuracy was higher at the center of the body, and lowered as the smartphone moved right/left or up/down. The errors occurred because the pose-estimation model found the body keypoints differently on both the images taken from the front and side, respectively. The estimated keypoints shifted to the right (red dots in Figure 7 right) from the expected location (blue dots in Figure 7 right) as the device moved to the right side of the body. Therefore, the space estimated using our system looked shrunk left/right and up/down.

Our system became more accurate when the participants wore a casual apparels (μ=41.72mm, and σ=26.00) compared with wearing coats (μ=71.78mm, and σ=59.40). The y- and z-position errors did not significantly change whether the participants wore casual apparels (y: μ=20.90mm, σ=16.62 / z: μ=15.52mm, σ=14.70) or coats (y: μ=26.54mm, σ=21.41 and z: μ=24.26mm, σ=26.54). However, a change was observed in the x-position error (casual: μ=26.27mm, σ=23.59 / coat: μ=55.25mm, σ=56.32). The experiment was conducted in winter, and some participants had their necks covered with coats. If the camera was at the top-side around the body, then the thick clothes that covered the neck hid the shoulder, and thus the camera could not capture the hidden part. Accordingly, the shoulder keypoints were located near the neck. Therefore, the system estimated the position closer in the x-axis direction than the actual. For participants who wore coats that did not cover their necks, we achieved higher tracking accuracy (μ=52.11mm, and σ=34.32). The accuracy results are presented at the bottom of Table 1.

EXAMPLE APPLICATIONS

We created several applications to show various design possibilities of OddEyeCam. We divided the design space into six subspaces using the combinations of current mobile device inputs and OddEyeCam: moving mode (standing, walking) $\times$ spatial input $\times$ additional input (touch input, IMU input, none). These example applications are only the instances of the subspaces to emphasize the unique advantages of OddEyeCam that traditional methods cannot achieve. For example, OddEyeCam enables instantaneous use cases by estimating the position without a initial-position calibration step. Additionally, it continuously provides body-centric interaction space while a user is walking. The participants used these applications in the first user study, as explained in the next section. Therefore, in this section, we detail the applications that the participants used in the first user study. Please also see Video Figure on the ACM digital library.

Standing $\times$ Spatial $\times$ Touch: Drag and Drop Between Apps — The interaction concept was proposed to peep into and access a virtual space larger than the screen by physically moving a mobile device [19, 65]. Additionally, users frequently switched from a specific app to a communication app [9]. Inspired by these two aforementioned points, we placed two apps to the personal reference frame of the user, as shown in Figure 9 A. A user can drag a photo in the gallery app and then drop it to the messenger app.

For the convenience of a right-handed user, we located the center of the two apps to be in 5 cm right-side of the chest of the user. Each app was of the same size as the display. The photos in the gallery app were of the size 1.76$\times$1.76 cm. A user could activate the drag state by long-pressing the photo. The photo could be pasted to the messenger app after moving the device left to view the messenger app and then releasing the photo.

Standing $\times$ Spatial $\times$ IMU: Body-Centric Folder — Discrete orientation around the body was useful to retrieve digital objects [14, 33]. We implemented a body-centric folder shown in Figure 9 B. A user can access each folder by placing a smartphone at each location. Moreover, she can select the application by tilting the device without touching it [57].

The first folder was in the front of the left side of torso, as shown in Figure 9 B. The interval between the folders was 8 cm. The deadzones of 2 cm were placed between folders to prevent sudden folder switching. A user could select an app by tilting the device and the app-icon in the folder was highlighted with a yellow background. We detected the tilting using the accelerometer of the smartphone.

Standing × Spatial: Large-Image Viewer — A spatially-aware display can show an image that is larger than the screen, as a user can move the small screen to view the different parts of the image [29]. OddEyeCam can support such interaction scenarios because it can continuously track the location of the mobile device in a large 3D space. As shown in Figure 9 C, we set a curved world map surrounded the body of a user so that she could pan and zoom the map with only spatial input. Moreover, a body-centered world map increased the memory recall of a specific city location.

The map was 38 cm away from the body of the user. If the user brought the device closer to the map, the device peeped into a partial region of the map, which emulating zoom-in. The range of zooming was from 25 to 38 cm. We made the zooming with 2.5 times of gain, thus the user could effectively zoom in and out.

Walking × Spatial × Touch: One-Hand Tagging — The virtual shelves and the body cobwebs anchored around the body were proposed for placing and retrieving information [14, 33]. We designed an application that enables users to tag notifications while paying less visual attention when they are walking. OddEyeCam can support this scenario because it can track the mobile device while its user is walking.

There were the following two layers: a notification bar (close to the user, Figure 9 D right) and tagging zone (far from the user, Figure 9 D left). The user was surrounded by nine tagging zones, which were located 30 cm from her body. The zones were a reminder (on the left side of the user), TODO list (on the center of the user), and fun tag (on the right side of the user). Depending on the height, the levels were different: the duration of the reminder, urgency level, and level of interest. Initially, a user can see a notification bar. After she drags a notification, the bar disappears, and then the tagging zones appear. If the user releases the notification to the desired zone, it is stored in that zone. However, if she takes the device away from the tagging zones, they disappear from the screen. A user can cancel to tag by dropping the notification with the zones disappeared. The zones were divided as 30° horizontally and 15° vertically. The 1° of deadzone was in the zone to prevent a sudden change.

Walking × Spatial × IMU: Getting Directions — Depending on proximity of a user to the device, some systems remove or superimpose visual information [10, 24]. Similarly, a user may require different information depending on the her walking mode. A getting-direction application provides information for searching a destination when the user stands still, and it provides the way to the destination when she walks.

As shown in Figure 9 E left, the application provides four views around the body of the user when she standing still: a zoomed-in map (center, close), zoomed-out map (center, far), and the information of hotels (left) and restaurants (right) near the destination. As shown in Figure 9 E right, the four views for a walking situation are as follows: a map (center, close), simple arrow that directs the user (center, far), next sub-destination (left), and remaining time for arrival of the bus (right). The views were divided by 40°, and there were 6° of

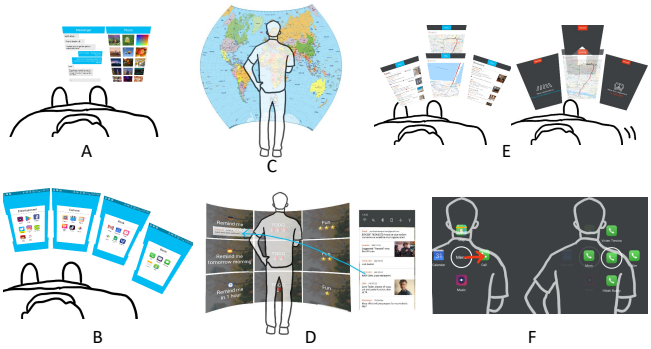

Figure 9. Drag and drop between messenger and gallery app (A), body-centric folder (B), large-image viewer (C), one-hand tagging (D), getting-direction app (E), and marking menu (F).

deadzones to prevent sudden changes of views. A closer view of the map is shown when the device is within 30 cm from the body. Our system detects the walking situations of the user by accumulating the output of a built-in accelerometer.

Walking × Spatial: Marking Menu — We built a two-step marking menu so that users can trigger body-attached shortcuts while paying less visual attention when walking. If a user presses down the screen at a specific location, then a marking menu is activated as shown in Figure 9 F left. Four icons are map (above), calendar (left), call (right), and music (below) app. The distance between the icon and activated location is 10 cm. If a user moved the device to the right, "call" menu would be opened. As shown in Figure 9 F right, four icons (e.g., favorite numbers) newly came up at the moved position. A user could call her "Mom" by moving the device to the left.

USER STUDY 1: DESIGN-GOAL EVALUATION THROUGH SIX APPLICATIONS

The first user study were aimed to check whether OddEyeCam satisfied our design goals and whether it was sufficient for various body-centric peephole applications.

Task

The participants used six example applications that have described above. There existed some simple subtasks for each application. In drag and drop between apps, the participants sent six selected photos from the gallery app to messenger app. In the body-centric folder, the participants moved to a specific folder and selected a specific app following the instruction of the experimenter. The instructions were "Please open the 'Entertainment' folder and then select the 'angry birds' app," etc. In the large-image viewer, the participant enlarged the map and read the text on the world map to search countries and cities following the instruction by the experimenter. The instructions were "You have seen Korea before. Please instantaneously move to the location of Korea and check whether it is still there," etc. In one-hand tagging, the participants classified various types of notifications into their desired tag zones. In the case of the getting-direction application, the participants answered the questions by the experimenter. The questions were "Please open hotel information, and tell me the price of 'Ace Hotel London'," etc. In the marking menu, the participants used shortcuts following the instructions by

the experimenter. The instructions were "Please call mom at one gesture," etc.

Procedure

We recruited 12 right-handed participants (3 women) who were aged between 19 and 27 years from a recruitment board on the university campus. The particpants used each application twice: standing still and walking. For the walking session, the participant walked back and forth for a distance of 2 m and rotated their bodies around clockwise and counterclockwise directions at the ends of the path. Before the experiment, we introduced a spatially-aware display. We used the "Think-aloud protocol" and the participants could give feedback freely without time limitation. After using an application in the standing and walking situations, respectively, each participant scored the statements on the 7-points Likert scale. The statements were as follows: S1) I felt a continuous movement over the body-fixed contents while moving the smartphone. S2) As expected, I felt that the smartphone moved over the body-fixed contents. S3-1) My eyes were comfortable while using the applications. S3-2) My wrist was comfortable while using the applications. S4) The body-fixed contents were at the same location at both the start and end of the sessions. S5) I felt that the contents were fixed to the body as I walked. S6) I could obtain the expected content at the expected location when I accessed it instantaneously. The experiment was performed in one and a half hours. We used a 1€ filter to stabilize the content on the screen [12]. All the participants completed the subtasks for all the applications.

Results

We excluded the data from the first participant because we changed the filter parameters of the "drag and drop between apps" on the basis of his feedback. For the 11 other participants, the same parameter values were used. Figure 10 shows the feedback for each statement in each scenario.

S1, S2, S4, and S6 are related to the tracking performance of OddEyeCam. The results show that the participants felt continuous movement (S1 score=5.9), felt the expected movement over the body-fixed contents (S2 score=5.4), could use the contents in the same location for long usage periods (S4 score=5.6), and could retrieve the expected contents even in the case of instantaneous access (S6 score=5.5). In their feedback, the participants mentioned that they felt as though the contents were fixed to the body (p2, p3, p5, and p7) and that it was possible to naturally use the device without being aware of the need to fix the upper body (p3). Additionally, they replied that the desired content was exactly there when they instantaneously accessed a certain location (p3, and p4).

Using OddEyeCam, the participants could access and interact with the contents fixed to the body while walking (S5 score=4.6). The participants reported that they could easily bookmark notifications or find countries on the world map because they remembered the location of the content relative to the body while they walking (p2, p3, p7, and p8). However, some replied that they could use the content around their body while rotating, whereas some responded that unwanted content briefly appeared while rotating. Notably, one participant stated

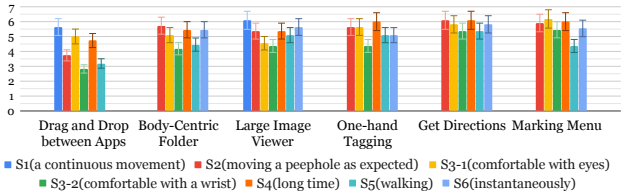

Figure 10. Feedback of participants in user study 1. (1=strongly disagree; 7=strongly agree).

that it would be a rare case to turn by 180° in real walking situations (p2).

All the application examples, except "drag and drop between apps" were well-accepted for all the statements by participants. The usability problems of "drag and drop between apps" were reported from S2, S3-2, and S5. We narrowed down the causes of usability problems in this application on the basis of the feedback from participants. First, they experienced a lag when moving the peephole (p3, p4, p7, and p8). Thereby, the app moved more than expected, and they overshot the target. The lagging was attributed to the 1€ filter. If we further optimize the filter parameters, this lagging problem can be solved, as shown in the second user study. Second, the participants interacted in a narrower space in the case of "drag and drop between apps" than other applications. They had to maintain their arm posture in a two-screen size space, thereby tiring them. Third, they reported that long pressing could not be performed because of body movements while they walked or rotated, thereby making the process of copying photos difficult (p5, p8, and p7). However, despite the negative feedback, all the participants completed the drag and drop task. We can change the "long pressing" to "short tapping" while a user walks.

USER STUDY 2: QUANTITATIVE CHARACTERIZATION OF ODDEYECAM

Although the results from the first user study showed that OddEyeCam satisfied our design goals, we deemed it necessary to characterize our system and understand the capabilities of OddEyeCam. This second user study had two purposes. First, we quantified the human performance when users operated OddEyeCam via a touch input. Second, we diagnosed the "drag and drop between apps", which had a relatively low user rating compared with other scenarios. We used peephole target pointing/acquisition tasks, which are more general but similar to the procedure of "drag and drop between apps".

Task

The target was displayed on a two-screen space (12.7×11.3 cm^2) which was the same space as drag and drop between apps. The participants could peep into the target through 6.35×6.35 cm^2 peephole at the center of the mobile phone, as shown in Figure 11. The target widths were set in the range of 5.9∼32.3 mm with a step of 2.94 mm (50 pixels); thus there were 10 different target widths. For each target width, three distances were specified in the range from 7.05 to 12.7 cm to cover the index of difficulty (ID) value uniformly and widely. Because the target distances were greater than the screen width, the participants had to move the device to

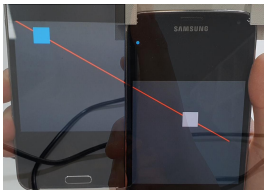

Figure 11. Peephole target acquisition task on OddEyeCam.

view the other target. There were 12 directions each with a step of 30°, thereby covering 360°. We ensured that the participants tapped two opposite directions in series as a round-trip. Therefore, they completed 180 randomized rounds from a total of 360 combinations, i.e., 10 target widths×3 target distances×12 directions (two combinations in one round). If they missed the target, the trial was restarted until succeeds. The two targets were not simultaneously shown through the peephole, and thus it took time to find the first target. We did not want to measure this time. Therefore, we let the participants gain the prior knowledge of the positions of both the targets by connecting the targets with a red line. When the task began, the line disappeared.

Procedure

The participants were asked to tap the target using the thumb of their right hand, which was holding the smartphone. They completed the tasks in the following two situations: standing still, and then walking. To understand the effect of the walking behavior on OddEyeCam, the experiment was conducted on a treadmill. In each session, 180 randomized rounds were divided into 30 rounds, thereby totaling to 6 sets. At the end of each set, the participants took a 1-min break. We encouraged them to take a longer break if they wanted. The participants had spent time practicing before the actual test. They had to perform 30 rounds as the practice task, following which they could practice as many rounds as they wanted. The experiment took 1 h and 40 min for each participant.

Participants

We recruited 12 right-handed participants (4 women) who were aged between 18 and 28 years from the recruitment board of a university campus. The average width of the thumbs of the participants was 19.3mm (σ=2.14mm). We set the walking speed of the treadmill to 3km/h, which is a preferred walking speed when a person is interacting with the smartphone [8]. Because the walking speed preferred by each person can be different, the participants could change the speed if they want, and two of them changed the speeds (p10: 2km/h, p11: 2.7km/h, and others: 3km/h).

Results

Because the target-selection skill of a human follows Fitts' law [17], it is meaningful to ensure that Fitts' law is applied when people select a target using OddEyeCam to verify the usability thereof. The linear regression results between movement time and Fitts' index of difficulty (ID) are depicted in Figure 12. The fitted equations are

$$MT = 0.2966 + 0.3569\log_2(A/W+1) \quad (sec), R^2 = 0.923 \tag{1}$$

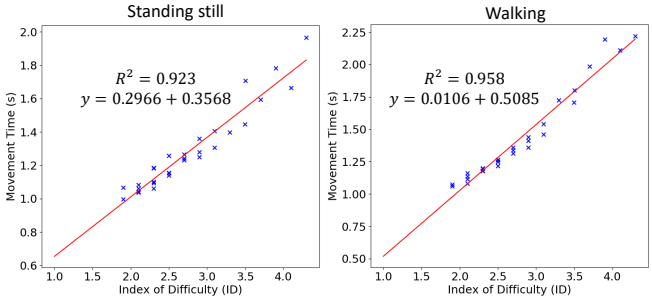

Figure 12. Scatter plot of the movement time vs. the Fitts' Law index of difficulty for peephole target acquisition and linear regression with participants standing still (left) and walking on the treadmill (right).

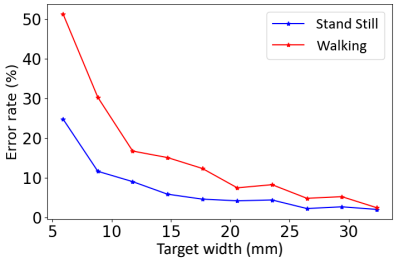

Figure 13. Error rate versus target widths for the peephole target acquisition task. High error rates were concentrated at the target widths less than 8.82 mm.

for the participants standing still and

$$MT = 0.0106 + 0.5085\log_2(A/W+1) \quad (sec), R^2 = 0.958 \tag{2}$$

for the participants walking. In equations 1 and 2, A denotes the target distance, W the target width, and MT the movement time. A peephole target acquisition task using OddEyeCam followed Fitts' law for both standing still and walking situation ($R^2 = 0.923$ for standing still and $R^2 = 0.958$ for walking). In other words, OddEyeCam could support a natural human-arm and hand movement in peephole target pointing tasks. Fitts' law parameters were $a = 0.2966$ and $b = 0.3569$, and throughput was 2.8 bits/s in the standing situation. The information-transmission rate of OddEyeCam was thrice than TinyMotion [63], which is a benchmark of pointing tasks by moving a handheld device. The participants completed OddEyeCam-based peephole pointing task at a throughput of 1.97 bits/s while walking. This throughput value was still 2.2 times higher than the benchmark.

The error rate was defined as the percentage of the number of failures over the total number of trials. The overall error rates were 7.71% for standing situation and 18.81% for the walking situation, and these are relatively high error rates. An ANOVA with Greenhouse–Geisser correction on error rates showed a significant effect of the target width ($F_{4.3,47.298} = 40.789$, $p < 0.001$ for the standing situation and $F_{9,99} = 100.130$, $p < 0.001$ for the walking situation.). As shown in Figure 13, high error rates were concentrated at the small target widths of 5.88 and 8.82 mm. The overall error rate for eight target widths except for two small target widths was 4.45%. The overall error rate was 9.34% for walking, excluding two small target widths. The selection accuracy decreases

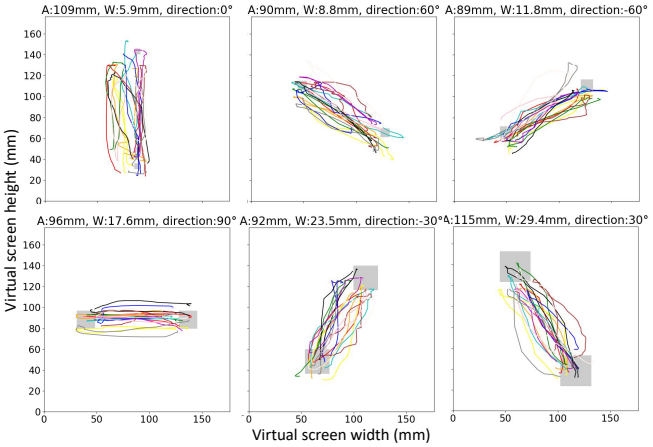

Figure 14. Some round-trip trajectory samples of a smartphone from 12 participants when standing. The same color is the data from one participant. A and W denote the target distance (unit: mm) and target width (unit: mm), respectively.

while walking [8]. We asked the participants to perform the tasks while walking at 3km/h. At this speed, the error rate was approximately 17% for the 9.7 mm of target fixed on the touchscreen from the study of Bergstrom et al. [8] Our task conditions were more difficult than those in Bergstrom et al. [8] Our participants performed a peephole target pointing task rather than fixed target selection on the screen. Additionally, our participants tapped the targets using the thumb of the hand that was holding the device rather than the index finger of the opposite hand. Nonetheless, the error rates were lower than 17% for all the target widths equal to or greater than 11.7 mm.

We analyzed the movements of the device during the peephole selection task to check whether the lag of our system affected the target selection by the participants. Figure 14 shows the round-trip paths of 12 participants. If the lag was significant, overshoot would be observed. We calculated the percentage between the actual path distance and target distance as $Actual\_Path\_Distance/Target\_Distance * 100$. The lagging was not significant because the ratio was 102.05% for the standing situation.

The study results showed the presence of the following two design parameter issues in "drag and drop between apps": 1) 1€ filter parameters and 2) long pressing in the walking situation. First, we adjusted the 1€ filter parameters ($f_{c_{min}}$=0.25, and β=0.0008) to be more responsive in this study. The participants could complete tasks, which is similar to the selecting an photo, quickly (3.1bits/s) and accurately (4.63% error rate of photo-sized targets). Additionally, the moving paths of a smartphone showed that the participants could aim the target with low lagging. Second, participants completed the task of tapping photo-sized targets even while walking with an error rate of 12.37%. A long-press interaction might have affected the usability of the application in the walking condition.

DISCUSSIONS AND FUTURE WORK

Recent mobile devices contain built-in 120° WFoV RGB camera [4, 5, 32, 48, 46] and ToF depth camera [25, 27, 31, 47, 38, 51]. However, we could not use these off-the-shelf mobile devices because the depth camera in OddEyeCam had to be tilted by 40° with respect to the smartphone in order to cover the shoulder area. OddEyeCam may be able to use an off-the-shelf mobile device when a WFoV depth camera, such as the one in Azure Kinect [36] by Microsoft, becomes available in a mobile device in the future.

Azure Kinect provides a depth image of 120° FoV. We considered the possibility of using a WFoV depth camera alone before we started to work on OddEyeCam. We could identify no previous work that presented a human-pose-estimation model using a WFoV depth camera only. The body-tracking system architecture of Azure Kinect also relies on an IR image for estimating body joints [34]. A WFoV depth camera in a mobile device would still need to be used in combination with a WFoV RGB camera for estimating the body coordinate system, but will make OddEyeCam a more feasible option for a mobile device as mentioned above.

We moved the keypoints to the closest point in the FoV of the depth camera when the cameras were so close to the body that the keypoints were outside NFoV. This solution worked satisfactorily in most cases, as confirmed through the three experiments but, when a user tilts the device (e.g., for the function for selecting an application in a body-centric folder), one of the shoulder keypoints could shift excessively. Consequently, the origin of the body coordinate system, which was defined as the center of the two shoulders, also shifted and the position-estimation accuracy of OddEyeCam could be affected. To cope with this problem, we expect that we may be able to use a keypoint on the neck instead, that is, the origin can be set as the neck position, and the x-axis can be obtained using the 3D points from the neck to either shoulder.

The body-tracking module estimates the body keypoints differently in the images taken from the front and side as depicted in Figure 7. Combining the output of the body-tracking module with that of SLAM-based front-facing tracking could help OddEyeCam find keypoints in the same position for all directions and achieve better accuracy.

The current prototype runs a body-tracking model on a desktop. The pose estimation models, such as OpenPose in this study, have many lightweight versions [26, 40, 49, 60]. We plan to utilize one of the body-tracking models for a mobile device for OddEyeCam in our future work.

CONCLUSION

We proposed a practical inside-out mobile-device tracking system, called OddEyeCam, to support body-centric peephole interaction. We evaluated our system through three experiments. OddEyeCam could track the 3D location of the mobile device in an absolute manner in both static and mobile environments with an average error of 4.3 cm. The participants could use various body-centric peephole interaction scenarios on OddEyeCam, and these required continuous tracking in a large 3D space around the body of the user. The task of peephole target selection on OddEyeCam followed Fitts' law, suggesting that OddEyeCam could support a natural human-hand movement. OddEyeCam is a practical method that expands the possibilities of a mobile device to body-centric peephole inter-

action with cameras that are becoming increasingly available in mobile devices.

ACKNOWLEDGMENTS

This research was supported by Next-Generation Information Computing Development Program through the National Research Foundation of Korea(NRF) funded by the Ministry of Science and ICT (2017M3C4A7065963)

REFERENCES

[1] Pablo Fernández Alcantarilla, Adrien Bartoli, and Andrew J Davison. 2012. KAZE features. In *European Conference on Computer Vision*. Springer, 214–227.

[2] Pablo F Alcantarilla and T Solutions. 2011. Fast explicit diffusion for accelerated features in nonlinear scale spaces. *IEEE Trans. Patt. Anal. Mach. Intell* 34, 7 (2011), 1281–1298.

[3] Adrian JT Alsmith, Elisa R Ferrè, and Matthew R Longo. 2017. Dissociating contributions of head and torso to spatial reference frames: The misalignment paradigm. *Consciousness and Cognition* 53 (2017), 105–114.

[4] Apple. 2019. iPhone 11 Pro. (2019). Retrieved July 9, 2020 from `https://www.apple.com/iphone-11-pro/`.

[5] Asus. 2019. ROG Phone II. (2019). Retrieved July 9, 2020 from `https://www.asus.com/Phone/ROG-Phone-II/`.

[6] Teo Babic, Florian Perteneder, Harald Reiterer, and Michael Haller. 2020. Simo: Interactions with distant displays by smartphones with simultaneous face and world tracking. In *Extended Abstracts of the 2020 CHI Conference on Human Factors in Computing Systems*. 1–12.

[7] Teo Babic, Harald Reiterer, and Michael Haller. 2018. Pocket6: A 6dof controller based on a simple smartphone application. In *Proceedings of the Symposium on Spatial User Interaction*. 2–10.

[8] Joanna Bergstrom-Lehtovirta, Antti Oulasvirta, and Stephen Brewster. 2011. The effects of walking speed on target acquisition on a touchscreen interface. In *Proceedings of the 13th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 143–146.

[9] Matthias Böhmer, Brent Hecht, Johannes Schöning, Antonio Krüger, and Gernot Bauer. 2011. Falling asleep with Angry Birds, Facebook and Kindle: A large scale study on mobile application usage. In *Proceedings of the 13th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 47–56.

[10] Michael Brock, Aaron Quigley, and Per Ola Kristensson. 2018. Change blindness in proximity-aware mobile interfaces. In *Proceedings of the 2018 CHI Conference on Human Factors in Computing Systems*. 1–7.

[11] Zhe Cao, Gines Hidalgo, Tomas Simon, Shih-En Wei, and Yaser Sheikh. 2018. OpenPose: Realtime multi-person 2D pose estimation using part affinity fields. *arXiv preprint arXiv:1812.08008* (2018).

[12] Géry Casiez, Nicolas Roussel, and Daniel Vogel. 2012. 1€ filter: A simple speed-based low-pass filter for noisy input in interactive systems. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. 2527–2530.

[13] Jessica Cauchard, Markus Löchtefeld, Mike Fraser, Antonio Krüger, and Sriram Subramanian. 2012. m+ pSpaces: Virtual workspaces in the spatially-aware mobile environment. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 171–180.

[14] Xiang'Anthony' Chen, Nicolai Marquardt, Anthony Tang, Sebastian Boring, and Saul Greenberg. 2012. Extending a mobile device's interaction space through body-centric interaction. In *Proceedings of the 14th International Conference on Human-Computer Interaction with Mobile Devices and Services*. 151–160.

[15] Xiang'Anthony' Chen, Julia Schwarz, Chris Harrison, Jennifer Mankoff, and Scott Hudson. 2014. Around-body interaction: Sensing & interaction techniques for proprioception-enhanced input with mobile devices. In *Proceedings of the 16th International Conference on Human-Computer Interaction with Mobile Devices and Services & services*. 287–290.

[16] NDI Ascension Technology Corporation. 2020. 3D electromagnetic tracking system. (2020). Retrieved July 9, 2020 from `https://www.ndigital.com/about/ascension-technology-corporation/`.

[17] Paul M Fitts. 1954. The information capacity of the human motor system in controlling the amplitude of movement. *Journal of Experimental Psychology* 47, 6 (1954), 381.

[18] George Fitzmaurice and William Buxton. 1994. The chameleon: Spatially aware palmtop computers. In *Conference Companion on Human Factors in Computing Systems*. 451–452.

[19] George W Fitzmaurice. 1993. Situated information spaces and spatially aware palmtop computers. *Commun. ACM* 36, 7 (1993), 39–49.

[20] Jens Grubert, Matthias Heinisch, Aaron Quigley, and Dieter Schmalstieg. 2015. Multifi: Multi fidelity interaction with displays on and around the body. In *Proceedings of the 33rd Annual ACM SIGCHI Conference on Human Factors in Computing Systems*. 3933–3942.

[21] Jari Hannuksela, Pekka Sangi, Markus Turtinen, and Janne Heikkilä. 2008. Face tracking for spatially aware mobile user interfaces. In *International Conference on Image and Signal Processing*. Springer, 405–412.

[22] Thomas Riisgaard Hansen, Eva Eriksson, and Andreas Lykke-Olesen. 2006a. Mixed interaction space—Expanding the interaction space with mobile devices. In *People and Computers XIX—The Bigger Picture*. Springer, 365–380.

[23] Thomas Riisgaard Hansen, Eva Eriksson, and Andreas Lykke-Olesen. 2006b. Use your head: Exploring face tracking for mobile interaction. In *CHI'06 Extended Abstracts on Human Factors in Computing Systems*. 845–850.

[24] Chris Harrison and Anind K Dey. 2008. Lean and zoom: Proximity-aware user interface and content magnification. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. 507–510.

[25] Honor. 2019. View20. (2019). Retrieved July 9, 2020 from https://www.hihonor.com/global/products/smartphone/honorview20/.

[26] Andrew G Howard, Menglong Zhu, Bo Chen, Dmitry Kalenichenko, Weijun Wang, Tobias Weyand, Marco Andreetto, and Hartwig Adam. 2017. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* (2017).

[27] Huawei. 2019. P30 Pro. (2019). Retrieved July 9, 2020 from https://consumer.huawei.com/en/phones/p30-pro/specs/.

[28] Hiroshi Ishii and Brygg Ullmer. 1997. Tangible bits: Towards seamless interfaces between people, bits and atoms. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. 234–241.

[29] Neel Joshi, Abhishek Kar, and Michael Cohen. 2012. Looking at you: Fused gyro and face tracking for viewing large imagery on mobile devices. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. 2211–2220.

[30] Bonifaz Kaufmann and Martin Hitz. 2012. X-large virtual workspaces for projector phones through peephole interaction. In *Proceedings of the 20th ACM international conference on Multimedia*. 1279–1280.

[31] LG. 2019a. G8 ThinQ. (2019). Retrieved July 9, 2020 from https://www.lg.com/us/mobile-phones/g8-thinq.

[32] LG. 2019b. G8X ThinQ. (2019). Retrieved July 9, 2020 from https://www.lg.com/us/mobile-phones/g8x-thinq-dual-screen.

[33] Frank Chun Yat Li, David Dearman, and Khai N Truong. 2009. Virtual shelves: interactions with orientation aware devices. In *Proceedings of the 22nd annual ACM symposium on User interface software and technology*. 125–128.

[34] Zicheng Liu. 2020. 3D Skeletal Tracking on Azure Kinect. (2020). System architecture. p11. Retrieved July 9, 2020 from https://www.microsoft.com/en-us/research/uploads/prod/2020/01/AKBTSDK.pdf.

[35] Tom Mayer, Susan Brady, Elizabeth Bovasso, Priscilla Pope, and Robert J Gatchel. 1993. Noninvasive measurement of cervical tri-planar motion in normal subjects. *Spine* 18, 15 (1993), 2191–2195.

[36] Microsoft. 2020. Azure Kinect DK. (2020). Retrieved July 9, 2020 from https://www.microsoft.com/en-us/p/azure-kinect-dk/8pp5vxmd9nhq?activetab=pivot%3aoverviewtab.

[37] Akira Ohashi, Yuki Tanaka, Gakuto Masuyama, Kazunori Umeda, Daisuke Fukuda, Takehito Ogata, Tatsurou Narita, Shuzo Kaneko, Yoshitaka Uchida, and Kota Irie. 2016. Fisheye stereo camera using equirectangular images. In *2016 11th France-Japan & 9th Europe-Asia Congress on Mechatronics (MECATRONICS)/17th International Conference on Research and Education in Mechatronics (REM)*. IEEE, 284–289.

[38] Oppo. 2018. RX17 Pro. (2018). Retrieved July 9, 2020 from https://oppo-nl.custhelp.com/app/answers/detail/a_id/1365/~/rx17-pro-time-of-flight-%28tof%29-camera-technology.

[39] OptiTrack. 2020. Motion Capture Systems. (2020). Retrieved July 9, 2020 from https://optitrack.com/.

[40] Daniil Osokin. 2018. Real-time 2d multi-person pose estimation on CPU: Lightweight OpenPose. *arXiv preprint arXiv:1811.12004* (2018).

[41] Alejandro Perez-Yus, Gonzalo Lopez-Nicolas, and Josechu J Guerrero. 2016a. A novel hybrid camera system with depth and fisheye cameras. In *2016 23rd International Conference on Pattern Recognition (ICPR)*. IEEE, 2789–2794.

[42] Alejandro Perez-Yus, Gonzalo Lopez-Nicolas, and Jose J Guerrero. 2016b. Peripheral expansion of depth information via layout estimation with fisheye camera. In *European Conference on Computer Vision*. Springer, 396–412.

[43] Milad Ramezani, Debaditya Acharya, Fuqiang Gu, and Kourosh Khoshelham. 2017. Indoor positioning by visual-inertial odometry. *ISPRS Annals of the Photogrammetry, Remote Sensing and Spatial Information Sciences* 4 (2017), 371.

[44] Martin Rufli, Davide Scaramuzza, and Roland Siegwart. 2008. Automatic detection of checkerboards on blurred and distorted images. In *2008 IEEE/RSJ International Conference on Intelligent Robots and Systems*. IEEE, 3121–3126.

[45] Hideo Sakata and Makoto Kusunoki. 1992. Organization of space perception: neural representation of three-dimensional space in the posterior parietal cortex. *Current opinion in neurobiology* 2, 2 (1992), 170–174.

[46] Samsung. 2019a. Galaxy Note10 Series: Note10|Note10+|Note10 5G|Note10+ 5G. (2019). Retrieved July 9, 2020 from https://www.samsung.com/global/galaxy/galaxy-note10/.

[47] Samsung. 2019b. Galaxy S10 5G, camera specification. (2019). Retrieved July 9, 2020 from https://www.samsung.com/us/mobile/galaxy-s10/camera/.

[48] Samsung. 2019c. Galaxy S10 Series: S10e|S10|S10+|S10 5G. (2019). Retrieved July 9, 2020 from https://www.samsung.com/global/galaxy/galaxy-s10/.

[49] Mark Sandler, Andrew Howard, Menglong Zhu, Andrey Zhmoginov, and Liang-Chieh Chen. 2018. Mobilenetv2: Inverted residuals and linear bottlenecks. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 4510–4520.

[50] Davide Scaramuzza and Zichao Zhang. 2019. Visual-Inertial Odometry of Aerial Robots. *arXiv preprint arXiv:1906.03289* (2019).

[51] Michael Schoenberg. 2020. uDepth: Real-time 3D Depth Sensing on the Pixel 4. (2020). Retrieved July 9, 2020 from https://ai.googleblog.com/2020/04/udepth-real-time-3d-depth-sensing-on.html.

[52] Andrea Serino, Jean-Paul Noel, Giulia Galli, Elisa Canzoneri, Patrick Marmaroli, Hervé Lissek, and Olaf Blanke. 2015. Body part-centered and full body-centered peripersonal space representations. *Scientific reports* 5 (2015), 18603.

[53] Garth Shoemaker, Takayuki Tsukitani, Yoshifumi Kitamura, and Kellogg S Booth. 2010. Body-centric interaction techniques for very large wall displays. In *Proceedings of the 6th Nordic Conference on Human-Computer Interaction: Extending Boundaries*. 463–472.

[54] David Small and Hiroshi Ishii. 1997. Design of spatially aware graspable displays. In *CHI'97 Extended Abstracts on Human Factors in Computing Systems*. 367–368.

[55] Misook Sohn and Geehyuk Lee. 2005. ISeeU: camera-based user interface for a handheld computer. In *Proceedings of the 7th international conference on Human computer interaction with mobile devices & services*. 299–302.

[56] Martin Spindler, Martin Schuessler, Marcel Martsch, and Raimund Dachselt. 2014. Pinch-drag-flick vs. spatial input: rethinking zoom & pan on mobile displays. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. 1113–1122.

[57] Ke Sun, Yuntao Wang, Chun Yu, Yukang Yan, Hongyi Wen, and Yuanchun Shi. 2017. Float: one-handed and touch-free target selection on smartwatches. In *Proceedings of the 2017 CHI Conference on Human Factors in Computing Systems*. 692–704.

[58] Desney S Tan, Randy Pausch, Jeanine K Stefanucci, and Dennis R Proffitt. 2002. Kinesthetic cues aid spatial memory. In *CHI'02 extended abstracts on Human factors in computing systems*. 806–807.

[59] Shan-Yuan Teng, Mu-Hsuan Chen, and Yung-Ta Lin. 2017. Way Out: A Multi-Layer Panorama Mobile Game Using Around-Body Interactions. In *Proceedings of the 2017 CHI Conference Extended Abstracts on Human Factors in Computing Systems*. 230–233.

[60] tensor. 2020. Tensorflow lite human pose estimation. (2020). Retrieved July 9, 2020 from https://www.tensorflow.org/lite/models/pose_estimation/overview.

[61] Sandeep Vaishnavi, Jesse Calhoun, and Anjan Chatterjee. 2001. Binding personal and peripersonal space: evidence from tactile extinction. *Journal of Cognitive Neuroscience* 13, 2 (2001), 181–189.

[62] Vicon. 2020. Award Winning Motion Capture Systems. (2020). Retrieved July 9, 2020 from https://www.vicon.com/.

[63] Jingtao Wang, Shumin Zhai, and John Canny. 2006. Camera phone based motion sensing: interaction techniques, applications and performance study. In *Proceedings of the 19th annual ACM symposium on User interface software and technology*. 101–110.

[64] Wii. 2020. Wiimote. (2020). Retrieved July 9, 2020 from http://wii.com//.

[65] Ka-Ping Yee. 2003. Peephole displays: pen interaction on spatially aware handheld computers. In *Proceedings of the ACM SIGCHI Conference on Human Factors in Computing Systems*. 1–8.