SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

January 2020, Vol. 63 112101:1–112101:20
https://doi.org/10.1007/s11432-019-1517-3

A flexible technique to select objects via convolutional neural network in VR space

Huiyu LI[1]* & Linwei FAN[2,3]

[1] *School of Computer Science and Technology, Shandong University, Jinan* 250101*, China;*
[2] *Shandong Province Key Lab of Digital Media Technology, Shandong University of Finance and Economics, Jinan* 250061*, China;*
[3] *Shandong Co-Innovation Center of Future Intelligent Computing, Shandong Technology and Business University, Yantai* 264005*, China*

Received 8 April 2019/Revised 17 June 2019/Accepted 1 August 2019/Published online 23 December 2019

Abstract Most studies on the selection techniques of projection-based VR systems are dependent on users wearing complex or expensive input devices, however there are lack of more convenient selection techniques. In this paper, we propose a flexible 3D selection technique in a large display projection-based virtual environment. Herein, we present a body tracking method using convolutional neural network (CNN) to estimate 3D skeletons of multi-users, and propose a region-based selection method to effectively select virtual objects using only the tracked fingertips of multi-users. Additionally, a multi-user merge method is introduced to enable users' actions and perception to realign when multiple users observe a single stereoscopic display. By comparing with state-of-the-art CNN-based pose estimation methods, the proposed CNN-based body tracking method enables considerable estimation accuracy with the guarantee of real-time performance. In addition, we evaluate our selection technique against three prevalent selection techniques and test the performance of our selection technique in a multi-user scenario. The results show that our selection technique significantly increases the efficiency and effectiveness, and is of comparable stability to support multi-user interaction.

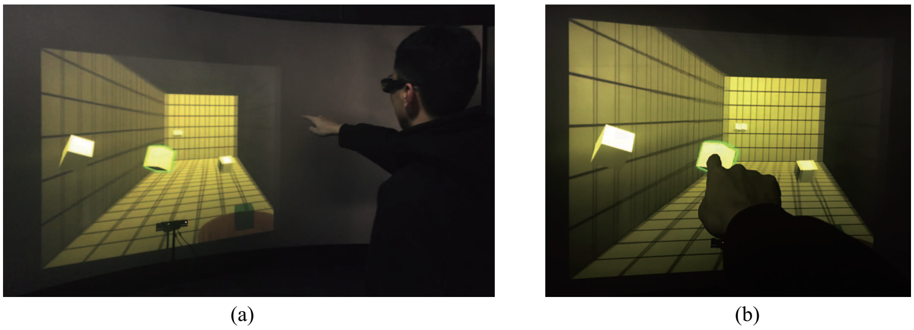
Keywords convolutional neural network, interaction techniques, pose estimation, virtual reality, 3D selection

Citation Li H Y, Fan L W. A flexible technique to select objects via convolutional neural network in VR space. Sci China Inf Sci, 2020, 63(1): 112101, https://doi.org/10.1007/s11432-019-1517-3

1 Introduction

Large display projection-based virtual reality (VR) systems [1–3], which present stereoscopic virtual environments, have emerged as the prevalent display paradigms. Furthermore, with all users head-tracked, large display projection-based VR systems have the ability to generate virtual environments that align with the real space and the viewed positions of virtual objects inside the virtual space. Because the alignment of the real and virtual environments has the potential to allow the user to intuitively perceive his body as part of the virtual space, the efficient support of selecting stereoscopically displayed objects is necessary in a head-tracked projection-based virtual environment (VE). To enable users to intuitively select virtual objects on a large display from a distance, the ray-casting selection technique [4] has been commonly used in projection-based VR systems. In this technique, the target object is determined by computing the intersections of rays emanating from spatial interaction devices, such as laser pointers [5,6] or 3D tracking equipment [7–9], through the display plane. Without interaction devices, the most intuitive

* Corresponding author (email: huiyl91@163.com)


(a)

(b)

Figure 1 (Color online) The illustration of our technique (once the object is selected, the contour is marked): (a) the user points at the desired object with his fingertip; (b) the object is selected when the fingertip occludes part of the object.

instrument is the human hand. In recent studies [10, 11], the use of human hand ray-casting has been proposed and evaluated.

When adopting the users' hands as the interaction devices, the first priority is to precisely track the user's body. Having been widely used in various interaction techniques, an RGB-D camera such as Microsoft's Kinect, based on vision techniques, can effectively track the bodies of multiply users. However, occlusion is a significant problem in the real deployment of single front-view camera systems. In recent years, many methods have been proposed to solve this problem. Kim et al. [12] proposed a depth-based tracking method for hand tracking and occlusion handling. By using depth cues for occlusion detection, Zohra et al. [13] proposed an effective face recognition method. However, these proposed solutions are suitable only for processing the occlusion of a single user. To resolve mutual occlusion when tracking multiple users, multi-camera based tracking systems sense the tracking area from different angles by using multiple RGB-D cameras, such as Out of Sight [14]. However, adopting more cameras leads to additional issues, such as high cost, difficulty in calibration, and inconvenient deployment setup for users.

Recently, the use of a deep convolutional neutral network (CNN) has enabled significant progress in multi-person pose estimation [15, 16]. By using a single, affordable and unintrusive RGB-D camera, the existing state-of-the-art CNN-based multi-person pose estimation methods have shown the potential to precisely track multiple users that encounter heavy occlusion or are in close proximity. These methods can accurately track each joint position and even fingertip, and avoid shading due to the use of Kinect equipment, unveiling new possibilities for selection techniques. Typically, CNN-based multi-person pose estimation methods include bottom-up and top-down methods. Bottom-up approaches, such as Cao et al. [16], detect body joints and assign them to people instances; therefore they achieve faster run time than top-down approaches. In contrast, top-down methods [17–19] first detect people and then adopt a single person pose estimation (SPPN) method to predict final pose for each person. In this manner, top-down methods enable higher accuracy than bottom-up methods. Because the SPPN method is performed for each person instance, top-down methods are extremely slow.

In traditional large display projection-based VR systems, misalignment between real and virtual object positions occurs when multiple users share a single-view stereoscopic display. To provide a separate, correct image for each eye of each user, two general approaches [20, 21] have been proposed. The first approach [20] is to correctly generate an individual image for each user. However, this approach is limited by costly extensions and image separation techniques. The second approach [21] is to adapt the stereoscopic images and depth cues to minimize the perceived distortion.

Inspired by the previous studies, we propose a flexible selection technique that enables multiple users to intuitively select virtual objects via CNN in a single-view stereoscopic VR environment, as shown in Figure 1. First, to support multi-user selection, our technique adopts a real-time CNN-based body tracking method that requires only a single RGB-D camera to accurately track multiple users. Because real-time performance is the primary factor of multi-user interaction, we adopt the bottom-up architecture for our CNN-based body tracking method. Second, after obtaining the tracked skeletons of the users, our technique adopts a region-based selection method that enables users to select virtual objects by directly pointing at the targets using their fingertips. To this end, the proposed selection technique is comparatively flexible in contrast with classical selection techniques which adopt expensive tracking equipment such as motion capture systems [11, 22]. Finally, to allow multi-user selection in a single-view virtual environment, our technique applies a multi-user merge method to effectively reduce the misalignment between the visual perception and actions of each user.

Within the scope of this paper, the virtual environment is rendered by projection-based display system. To this end, our selection technique requires the users to face the screen when selecting in the virtual environment. For our CNN-based body tracking method, a comparative experiment is performed to evaluate the the precision and the real-time performance by comparing with state-of-the-art CNN-based pose estimating methods. To evaluate the performance of our selection technique, we conduct two formative user studies to compare with prevalent selection techniques and test the effectiveness of our selection technique in multi-user interaction scenario. In summary, the main contributions of this paper are as follows:

(1) A flexible 3D selection technique that allows multiply users to rapidly and precisely select virtual objects using their fingertips in a single-view stereoscopic VR environment.

(2) To support multi-user interaction, a real-time CNN-based body tracking method is proposed to generate stable 3D skeletons of multi-users. The proposed body tracking method is able to achieve considerable estimation accuracy on the premise of ensuring real-time performance.

(3) A region-based selection method is proposed to adopt the tracked skeleton of the user to select virtual objects in a single-view stereoscopic VR environment. By using the extended region of the tracked fingertip, our region-based selection method allows the user to accurately select target objects in the virtual environment.

(4) To allow multiple users selecting virtual objects in the same virtual environment, we propose a multi-user merge method by appropriately adjusting users' viewpoints and interactions to make their perceptions and actions consistent.

The remainder of this paper is organized as follows. Section 2 reviews previous studies on pose estimation and ray-casting selection technique. Section 3 describes in detail the process and calculation of our technique, while Section 4 presents the evaluations that analyze the performance of our technique and discusses the results. Finally, we conclude with future work in Section 5.

2 Related work

2.1 Pose estimation

Human pose estimation has been extensively studied for many years. Because many studies focus on detecting the body parts of individuals [15,23], it is significant to localize anatomical key-points. However, challenges occur when inferring multi-user poses. The so-called top-down approach [24] is a classic method of directly estimating single-user positions by leveraging existing techniques. However, the top-down approach suffers from computational cost when detecting multiple users. In contrast, bottom-up approaches have the potential to decouple runtime complexity from the number of people in the image. Yet, previous studies using the bottom-up method [17,25] fail to retain the gains in efficiency owing to the costly global inference required by the final parse. For instance, Pishchulin et al. [25] proposed seminal work of bottom-up approach to jointly label part detection candidates and associate them with individual users. However, an NP-hard problem occurs in solving the integer linear programming problem over a fully connected graph, and the average processing time is on the order of hours. Based on ResNet [26] and image-dependent pairwise scores, Insafutdinov et al. [17] used stronger part detectors to vastly improve the runtime. However the method still takes several minutes per image, with a limitation on the number of part proposals.

2.2 Ray-casting selection technique

As the commonly used selection technique, the ray-casting technique [27,28] adopts a virtual ray emitting from a user's hand or input device and detects the intersections of these rays to make a selection. Although this technique is very simple, it suffers from a number of problems. These issues are mostly caused by natural hand tremor and tracker jitter, making it difficult to control the ray using this technique. When objects are in small size [29], the ray cast may become even more difficult to use because selecting such objects by pointing requires a high level of precision.

To address these issues, a number of improvements have been proposed. By adding a cone-shaped volume to the ray, the cone-casting [27] extends the ray cast to make it easier to select virtual objects at a distance. Haan et al. [28] presented the snapping technique which uses a selection volume to calculate and accumulate scores over time for each object to estimate the target. In addition, the bubble-cursor [30] is a 2D technique that dynamically resizes a circular cursor to make this cursor contain only one object. Vanacken et al. [31] extended the bubble-cursor to a 3D version by utilizing a virtual sphere instead of a circle. In a cluttered virtual scene, these two techniques actually perform worse, because the ray or the cursor constantly snaps or resizes to select new targets when even small movements occur. By changing the control-display ratio, some other techniques improve ray cast accuracy, either automatically (e.g., PRISM [32], when moving slowly) or manually (e.g., ARM [33], by pressing a button), or by providing the ability to zoom (e.g., zoom-and-pick [34]). Although very high levels of precision can be achieved by these techniques, they all have limitations. For PRISM and ARM, a significant mismatch of the physical pointing direction and the perceived pointing position happens during the selection process, and the mapping is nonlinear. Zoom-based techniques suffer from a potential loss of detail. In addition, users are required to interact very carefully and with full attention through these techniques. In contrast, SQUAD [35] uses progressive refinement to narrow the choice of objects from which to select. This technique presents the objects that are contained in a sphere-cast and displays these objects onto a Quad-menu in the screen. The user selects the part of the Quad-menu that contains the desired object, and the objects that are in the same part are then used to fill in the Quad-menu, in the same manner as the original objects. The technique iteratively processes until the desired object is individually contained in part of the Quad-menu.

3 The proposed selection technique

We aim to develop a selection technique to effectively achieve agreement between users' actions and their perceptions when multiple users view a single-view stereoscopic display. For instance, if a user attempts to select a virtual object, the object will be selected when the user's fingertip occludes part of the object in his vision. Therefore, the precision and consistency of perception are the key points of the selection technique. Regarding precision, we propose a real-time body tracking method to precisely track the 3D joint data of the users. For consistency, we introduce the region-based selection method and multi-user merge method to achieve correct agreement between the users' interactions and perceptions.

3.1 Real-time body tracking

The first priority of our technique is to precisely track the heads and hands of multiple users, which is the basis of subsequent processing. In general, a feasible approach is to adopt available real-time body estimation techniques using an RGB-D camera, such as the Microsoft Kinect. However, Kinect works inefficiently when the body parts of single user are heavily occluded, or multiple users are in close proximity. In contrast, CNNs have been proved to effectively solve this issue in body estimation and maintain good real-time performance. Thus, we implement a body tracking method with multi-stage CNNs to accurately estimate the 3D body poses of multiple users. The method takes the color and depth images of the users as the inputs. Inspired by Zhe et al. [16], the method uses a two-branch CNN to predict the 2D joints of each user in the color images. Finally, all body joints are assembled into full 3D

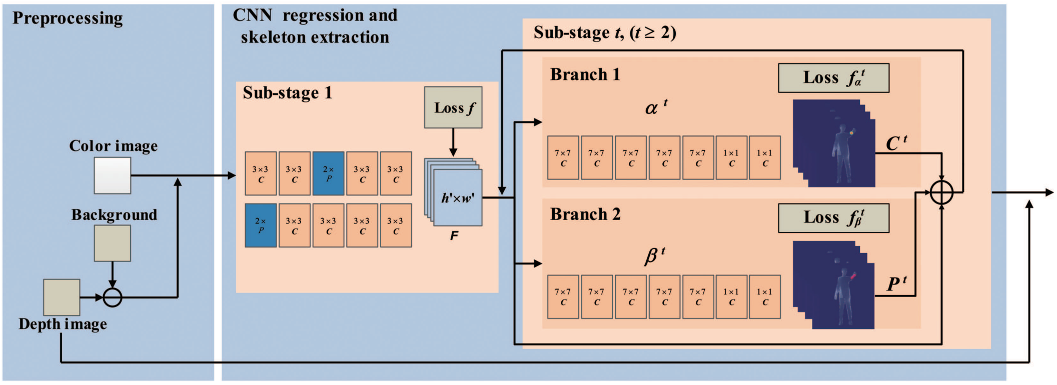


Figure 2 (Color online) Architecture of the two-stage body tracking method. Using the multiple sub-stage CNN with two branches, the method eventually estimates correct full body poses for each user in the depth image.

body poses for all users in the depth image. As shown in Figure 2, we divide the body tracking method into two stages: (1) preprocessing and (2) CNN pose regression and skeleton extraction. The details are as follows.

3.1.1 *Preprocessing*

The preprocessing stage mainly eliminates the background of the original color image while preserving the main bodies of all users. A simple version is to initially capture a background image, which contains only the stationary scene of the detection area, and directly eliminate the background from the subsequent color images. However, even slight changes in illumination are detrimental to this approach. Because many researchers [36, 37] have proved that human body can be effectively extracted using depth image, we adopt a depth-image-based method to detect the main bodies of all users in the depth image. The method initially captures a depth image that contains the experimental environment, except the users, as the background. Using the background depth image, the method eliminates the stationary scene from the depth image with a background subtraction process to produce a non-background depth image. All pixels that indicate users are retained in the depth image. By mapping the pixels from the depth image to the color image, the body extraction results can be easily obtained in the color image. We perform these processes for the purpose of reducing the complexity of the image and accelerating the subsequent CNN-based method.

During the preprocessing stage, the accuracy of body extraction is negatively affected by the noise of the depth image. The noise of the depth image is usually presented as mounts of null-value areas that are randomly distributed in the depth image. Therefore, we adopt pixel filtering and context filtering [36] to smooth the depth image and remove most of the noise.

3.1.2 *CNN pose regression and skeleton extraction*

The core of our body tracking method is a two-branched CNN that predicts the 2D joint positions of each user in real time. This stage takes as input the non-background color image and produces as output the 2D locations of the anatomical joints of each user in the color image. Using the architecture as proposed in [16], we first utilize a feed-forward network to simultaneously predict a set of 2D confidence maps of joints and a set of 2D vector fields of part affinities. Then, the confidence maps and part affinity fields are parsed by bipartite matching to output the individual linked joints for all users in the color image. We split this stage into several sub-stages.

In the first sub-stage, we process the input by the first 10 layers of the VGG-19 model [38] to generate a set of feature images F. According to [38], VGG-19 adopts stacked small convolutional layers, which are better than large convolutional layers [39]. Because a small convolutional layer has more non-linear transformations than a larger convolutional layer, the CNN has a stronger ability to learn image features, while the cost is relatively small (with fewer parameters).

Next, we adopt two CNNs, α and β, which use the ResNet50 network architecture of He et al. [26], to proceed with these feature images F, and simultaneously produce a set of confidence maps $C{=}\alpha\,(F)$, and a set of part affinity fields $P{=}\beta\,(F)$, in the second sub-stage. Iteratively, the feature images F, along with the results of both branches in the previous sub-stages, are concatenated to calculate refined results in subsequent sub-stages. At sub-stage t, the set of detection confidence maps C^t, and the set of part affinity fields P^t, can be defined as follows:

$$C^t = \alpha^t(F, C^{t-1}, P^{t-1}), \quad \forall t \geqslant 2, \tag{1}$$

$$P^t = \beta^t(F, C^{t-1}, P^{t-1}), \quad \forall t \geqslant 2, \tag{2}$$

where α^t and β^t are the CNNs for inference at sub-stage t.

To guide the network to iteratively predict the confidence maps of body parts and part affinities, two loss functions are applied at the end of each sub-stage, respectively. We use the L2 loss between the estimated predictions and the ground truth maps and fields. For sub-stage t, the loss functions at the two branches are

$$f_\alpha^t = \sum_{i=1}^{I}\sum_{X} \left\| C_i^t(\boldsymbol{x}) - C_i^{\mathrm{GT}}(\boldsymbol{x}) \right\|_2, \tag{3}$$

$$f_\beta^t = \sum_{i=1}^{I}\sum_{X} \left\| P_i^t(\boldsymbol{x}) - P_i^{\mathrm{GT}}(\boldsymbol{x}) \right\|_2, \tag{4}$$

where C_i^{GT} is the ground truth part confidence map, P_i^{GT} is the ground truth part affinity vector field, and $\boldsymbol{x}$ is the pixel location on the image. Similar to [15], we adopt intermediate supervision at each stage to solve the problem of the vanishing gradient by replenishing the gradient periodically. The network ultimately contains a set of detected joints and a set of part affinity fields for each user. Using maximum weight bipartite graph matching, the entire 2D body skeleton of each user is presented to the non-background color image. Finally, a bicubic interpolation is calculated to map the non-background color image to the depth image in real time. Because the depth image is a source of 3D information, all 2D body joints and associations are transformed to 3D in the depth image. In this manner, our body tracking method is capable of estimating the positions of the head and index fingertip for each user.

3.2 Region-based selection

Although our body tracking method provides the precise positions of the detected head and index fingertip for each user, how does the user utilize them to interact in a virtual scene? A simple approach is to generate an invisible virtual ray that emits from the use's head position and extends through the fingertip. The selections are identified by detecting the intersection of the ray with objects in the virtual scene. The user's head position directly corresponds with the virtual viewpoint, making the user's perception and virtual environment consistent. However, difficulties are experienced when attempting to select a certain object of a small size or selecting an object in a cluttered scene, leading to confusion or frustration. Such difficulties arise due to two limitations: (1) the method treats the fingertip as a single point, and (2) the virtual ray projects to a single point in the viewing plane.

We propose a novel selection method that attempts to overcome the negative effects of these limitations. In essence, our method adopts a selection region to select virtual objects contained inside this region. If the method detects several objects that are simultaneously inside this region, a progressive refinement proceeds to select the desired objects. The region is calculated by projecting the fingertip onto the projection plane (which means the display screen in our VE). The algorithm of our method consists of three general stages, and we describe the necessary algebraic calculations of each stage in the following part.

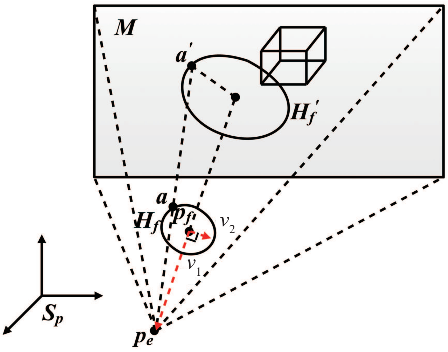


Figure 3 (Color online) The process of projecting a fingertip region onto the projection plane. The projection region of the fingertip can be considered the "shadow" of the fingertip with the eye position.

3.2.1 *Region test*

The first stage of our method is to generate a selection region by the relevant fingertip and determine which objects are inside the region. Consider the perspective relation which is shown in Figure 3. In the coordinate system of the physical space S_p, the image of the virtual scene is projected onto the projection plane M, which is located on the plane $z = z_m$, and the user's eye is positioned at p_e. Because the fingertip retains a certain size in the user's vision, we need to calculate the volume of the fingertip in this situation. To facilitate the computation, we adopt a small circle which is centered at the position of the tracked fingertip to indicate the fingertip area in the color image. In our method, the radius of the circle is 0.5 cm because the average minimum diameter of the human finger is approximately 10 mm. Considering that the fingertip is represented three-dimensionally, we map the 2D shape of the fingertip to the corresponding depth image to generate a set of 3D points, denoted by W. To simplify the calculation, we use a circular 3D plane H_f to indicate the 3D fingertip representation. The plane is centered at the fingertip position p_f with its normal vector directly pointing to the eye position p_e. To calculate the radius of H_f, a vector $\boldsymbol{v}_1 = p_e - p_f$ is defined. We then calculate the subset W' of W, in which each point p of W' satisfies $(p - p_f) \cdot \boldsymbol{v}_1 = 0$. By calculating the Euclidean distance between each point of W' and the fingertip position p_f, we adopt the point p_m, which has the largest Euclidean distance, to compute the radius $r = |p_m - p_f|$.

Then, the set of points on H_f is defined as those points a within a distance threshold of the line segment, and we obtain

$$\begin{cases} (a - p_f) \cdot \boldsymbol{v}_1 = 0, \\ |a - p_f| = r, \end{cases} \quad \forall a \in S_p. \tag{5}$$

Based on the eye position p_e, the plane H_f is projected onto the projection plane M. As shown in Figure 3, a projection region H_f' is then generated. Each point a' on H_f' is calculated by the intersection of the projection plane with a line that is defined by $p_e = (x_e, y_e, z_e)$ and a point $a = (x_a, y_a, z_a)$ of H_f, as shown in Eq. (6):

$$a' = \left\{ \frac{(z_m - z_e)(x_a - x_e)}{z_a - z_e} + x_e,\ \frac{(z_m - z_e)(\mathrm{y}_a - y_e)}{z_a - z_e} + y_e,\ z_m \right\}. \tag{6}$$

As the selection region H_f', has been correctly generated, it is simple to determine the pixels that are covered by the selection region on the image displayed on the projection plane. In this case, an object is considered a selectable object if its pixels are fully or partially contained by the selection region. Until now, we have ignored the fact that displays in VE are stereoscopic, and there is actually no single eye position. Therefore, our method takes the head position as the midpoint in the line joining both eyes and estimates the approximate eye positions with a fixed offset σ. According to [40], the average human physiological interocular distance is 6.3 cm. Therefore we set $\sigma = 3.15$. For both eyes, the region tests are respectively proceeded and provide two selectable object sets O_l and O_r. If O_l and O_r contain only

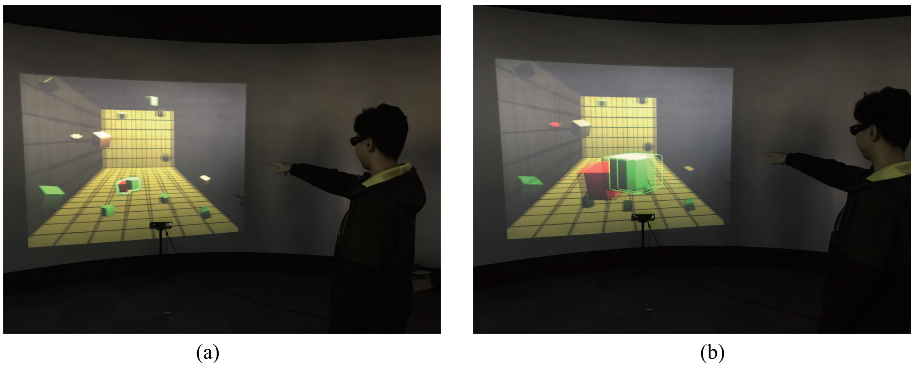


Figure 4 (Color online) The progressive refinement in selection. (a) Several selectable objects are detected; (b) the objects are zoomed in for convenience of selection.

the same single object, the object is treated as the selection result. Once our method detects multiple objects contained in O_l or O_r, a progressive refinement selection is adopted to determine the desired object.

3.2.2 *Progressive refinement selection*

The goal of this stage is to accurately select the desired object if O_l or O_r contains multiple objects. Inspired by [35], a progressive refinement process is adopted in our method. Upon completion of the region test, all selectable objects that are contained inside O_l and O_r are immediately zoomed in by the virtual camera on the screen, while the other objects in the virtual environment are all ignored. Among all these zoomed objects, the user points directly at the desired object to make the selection. By performing a region test, our method detects the selectable objects for O_l and O_r among all the zoomed objects. To make the resulting selection, the procedure is repeated each time reducing the number of selectable objects per region test until the desired object is the only one contained in both O_l and O_r. Figure 4 shows the progressive refinement of our method.

3.3 Multi-user merge

The previous subsections mainly discuss a single-user situation; here, we concentrate on the situation of multiple users. In a classic, single-user situation, the user's head position and the virtual viewpoint can make a direct agreement, allowing the correct experience for both the stereo and motion parallax depth cues and providing a co-located virtual environment. In the multi-user scenario, however, when the projection is viewed by different users, the mental 3D reconstruction of the virtual scene and the real space do not agree in their perceptions. Then, the parallax problem [21] in a stereoscopic image occurs.

We expect to make users' perceptions and interactions consistent without influencing the general viewing conditions by rendering a virtual scene into a consistent image. Inspired by the work of Simon et al. [21], we calculate an observer viewpoint that is centered at the middle position of the tracked users in the real space to make all users share the same projection image viewed from the observer viewpoint. For each user, a warp calculation is processed by warping the user's head position to the observer viewpoint. By the warp calculation, the interaction position, i.e., fingertip position, is then transformed to a warped interaction position with an affine transformation. Based on the observer viewpoint and the warped interaction position, the users are able to unerringly select the target, because the users now see the selection controlled by their true real position. Therefore, different viewpoint projections can be precisely aligned with the users' interactions while still maintaining and rendering a single consistent stereoscopic image, as illustrated in Figure 5.

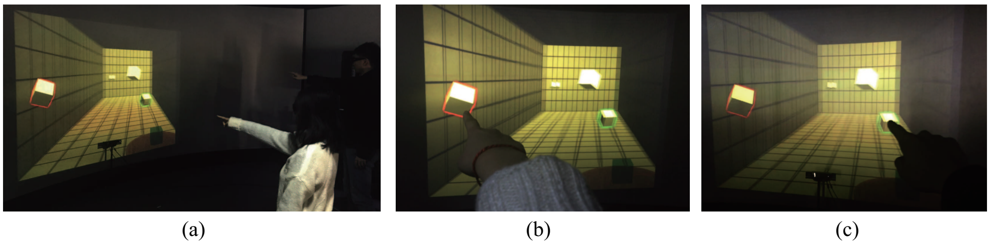


Figure 5 (Color online) Multi-users' interaction in the virtual environment is shown in (a). The individual visions of the two users are respectively shown in (b) and (c).

4 Evaluation

Our technique makes the users' fingertips an interaction tool and allows the users to directly select virtual objects by pointing at the targets in their vision. According to the requirement of our selection technique, our CNN-based body tracking method needs to accurately estimate the positions of users' joints in real-time. Therefore, we first conduct a comparative experiment to quantitatively evaluate our CNN-based body tracking method by comparing with state-of-the-art human pose estimation methods, in Subsection 4.1.

In order to verify the performance of our selection technique, we then conduct two separate user studies to test our system in Subsections 4.2 and 4.3. In study one, we evaluate the efficiency and effectiveness of our selection technique in a single-user situation and have compared our technique with three prevalent selection techniques. In study two, we evaluate the performance of our body tracking method in the multi-user interaction task. Both studies adopt the same system setup. Besides, we invite the same participants to participate in each of the studies.

4.1 Comparative experiment

Because our body tracking method is designed for interaction task, the major evaluation criteria contains two aspects: the precision and the real-time performance. We compare our body tracking method with state-of-the-art CNN-based multi-person pose estimation methods to evaluate the precision and real-time performance on different datasets. In the experiment, all comparisons are performed on an NVIDIA GeForce GTX-1080 GPU.

4.1.1 *Datasets and evaluation measure*

Our body tracking method is pre-trained and evaluated for multi-person pose estimation on two widely used datasets: the MPII multi-person dataset [41] and the COCO 2016 key points challenge dataset [42]. Moreover, to evaluate the performance of our body tracking method for estimating the position of user's index fingertip, we adopt our own dataset which labels index fingertip positions in addition to typical joints.

MPII multi-person dataset. The MPII multi-person dataset contains 3844 training and 1758 testing groups with both multiple overlapping people in highly articulated poses. These groups are taken from the test set as outlined in [25]. In the MPII multi-person dataset, the occupied areas of each group and the scale term of all people in each group are provided. However, no information is provided concerning the number of people or the scales of individual figures in this dataset. For evaluating the precision, we use the evaluation metric called mean average precision (mAP), which is outlined by [25], to calculate average precision of joint detections. In addition, we report the median running time per frame to evaluate the real-time performance of our method.

COCO key points challenge dataset. The COCO 2016 key points challenge dataset consists of around 100000 training images and more than 80000 testing human instances with annotated key points (i.e., body parts). The testing set contains four roughly equally sized subsets, namely, "test-challenge", "test-dev", "test-standard", and "test reserve". In the experiment, the performance of our body tracking

Table 1 Comparison of mAP (%) and time (s/frame) on the full testing set of MPII multi-person dataset[a)]

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Total	Time
DeeperCut [17]	78.4	72.5	60.2	51.0	57.2	52.0	45.4	59.5	485
Iqbal et al. [18]	58.4	53.9	44.5	35.0	42.2	36.7	31.1	43.1	10
CMU-Pose [16]	91.2	87.6	77.7	66.8	**75.4**	68.9	61.7	75.6	1.24
RMPE [19]	88.4	86.5	**78.6**	**70.4**	74.4	73.0	**65.8**	76.7	1.5
Ours	**91.7**	**87.9**	78.3	68.7	75.2	**74.1**	64.3	**77.2**	**1.05**

a) The optimal results are in bold.

Table 2 Comparison on the testing subset test-dev of the COCO dataset[a)b)]

Method	AP(%)	AP^{50}(%)	AP^{75}(%)	AP^{M}(%)	AP^{L}(%)	Time (s/frame)
CMU-Pose [16]	61.8	84.9	67.5	57.1	68.2	0.1
RMPE [19]	61.8	83.7	**69.8**	**58.6**	67.6	2.5
Ours	**63.3**	**85.3**	68.9	57.8	**68.8**	**0.08**

a) AP is the average of AP scores at 10 different OKS thresholds. AP^{50} is the AP at OKS = 0.50. AP^{M} is the AP for medium objects, and AP^{L} is the AP for large objects.
b) The optimal results are in bold.

method is evaluated on "test-dev". For the evaluation of the precision, the standard evaluation metric is based on object keypoint similarity (OKS). By applying OKS, the official COCO evaluation metric average precision (AP) can be computed as main competition metric. The median running time per frame is also reported in the result.

Our own dataset. Because index fingertip is one of the key joints for our body tracking method, we construct our own dataset for the training and evaluation. Our own dataset contains 2125 training images and 843 testing images that are selected from MPII and COCO datasets. For each image of this dataset, the index fingertip of each person is clearly discernible in the image. We manually add the index fingertip positions for all images in the dataset. Similar to MPII, we adopt mAP and the median running time per frame as the evaluation metrics to evaluate the precision and real-time performance of our body tracking method, respectively.

4.1.2 *Results*

Results on MPII multi-person. Our body tracking method and the compared methods are evaluated on full MPII multi-person test set. The comparison results are given in Table 1. The results illustrate that our body tracking method achieves overall 77.2% mAP and outperforms previous state-of-the-art method [16] by 1.6% mAP. Compared with RMPE [19], our method shows the similar performance for several body joints (such as elbows or hips), and achieves better results over some joints, including heads, shoulders and knees. These results indicate that our body tracking method can accurately detect joints for multi-person pose estimation. Moreover, the computational speeds of all methods are reported in Table 1. Notably, our method significantly reduces the running time for computing per image and is about 1.4 times faster than the approach [16] with state-of-the-art speed for multi-person pose estimation.

Results on COCO. We also perform evaluation on the subset "test-dev" of the COCO dataset. We compare our body tracking method with other approaches to measure the median run time and the AP with different OKS thresholds. The quantitative results on "test-dev" are given in Table 2. From the results, we find that our method outperforms previous method [16] on all AP results. In contrast with RMPE [19], our approach shows a similar performance than that of RMPE on AP^{75} and AP^{M} while achieving better performance on AP, AP^{50} and AP^{L}. However, it is noteworthy that the runtime speed of our system is better than those of all compared methods.

Results on our dataset. A comparison between our method and CMU-Pose [16] is performed on our own dataset. Different from MPII and COCO, our own dataset additionally labels the index fingertips of all people on each image. Thus, we report the mAP of index fingertip in the results, as shown in Table 3. According to the results, our method outperforms CMU-Pose in both precision and real-time performance.

Table 3 Comparison of mAP (%) and time (s/frame) on the full testing set of MPII multi-person dataset[a)]

Method	Head	Shoulder	Elbow	Wrist	Hip	Knee	Ankle	Finger	Total	Time
CMU-Pose [16]	**90.3**	86.5	74.7	64.2	**74.3**	70.2	62.3	77.4	74.9	0.56
Ours	89.8	**87.4**	**76.2**	**65.7**	73.8	**73.1**	**63.5**	**79.7**	**76.1**	**0.13**

a) The optimal results are in bold.

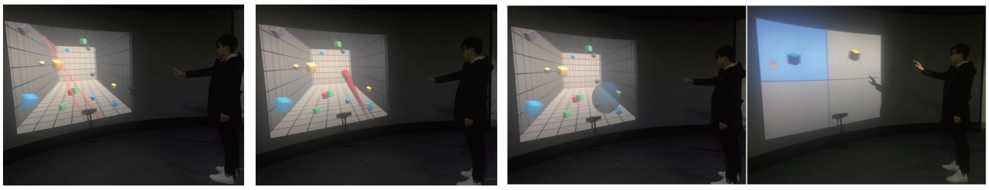

(a) (b) (c)

Figure 6 (Color online) Implementation of Ray Cast, Cone Cast and SQUAD in the virtual environment. (a) Ray Cast technique; (b) Cone Cast technique; (c) SQUAD and the Quad-menu.

4.2 Study one

Study one is conducted to evaluate our selection technique by comparing with prevalent selection techniques, in a single-user situation. Of the many possible selection techniques to compare, we choose Ray Cast [29], SQUAD [35] and Cone Cast [28] because they act as baselines for most of the current techniques. We extend Ray Cast and Cone Cast techniques with our body tracking method to enable the virtual ray (cone) to be directly controlled by the user's hand in the large display VR system. For both Ray Cast and Cone Cast techniques, the positions of the user's fingertip and the associated hand are tracked to accurately generate a visible virtual ray (cone). The virtual ray (cone) defining the pointing direction originates at the hand position and passes through the fingertip. For Ray Cast technique, the first intersection of this ray with an object indicates the selected object. With regard to Cone Cast technique, the object, which is contained in the virtual cone and nearest to the user, is treated as the selected object. Figures 6(a) and (b) illustrate the implementation of Ray Cast and Cone Cast in our system. To extend SQUAD in our system, we implement the sphere-cast in a hand-based manner similar to the implementation of Ray Cast. We also apply our body tracking method to SQUAD in our system. The selection mechanism within the Quad-menu is a 2D cursor controlled by the user's hand. The implementation of SQUAD is shown in Figure 6(c).

4.2.1 *Participants*

A total of 20 participants (10 males, 10 females), age 21–29 (M = 24.35, SD = 2.56) participate in study one. All participants are right-handed. All participants are frequent computer users, and 13 have very skilled experience with stereoscopic projections. All participants are undergraduate or Ph.D. students in computer science.

4.2.2 *System setup*

For the evaluation, we use a body tracking enabled active stereoscopic projection system. For visualization we use a 2.5 m wide, 1.88 m high, hoist-projected display, with imagery generated by one 1920 × 1080 resolution 3D projector. The virtual environment is rendered by active stereo projection. Due to the stereoscopic demand, the users wear active shutter glasses to perceive the stereoscopic images. We use a Microsoft Kinect v2 for body tracking. For convenience of calculation, the tracking senor is placed at the middle-bottom of the projection display. The system properly proceeds to track all users' heads and fingertips. All test scenes are developed by the unity game engine.

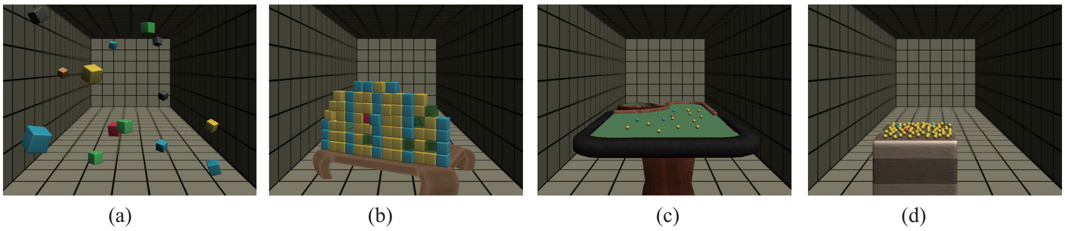


Figure 7 (Color online) The four different scenarios used in study one. (a) The scene of Scenario 1; (b) the scene of Scenario 2; (c) the scene of Scenario 3; (d) the scene of Scenario 4.

4.2.3 *Design*

To investigate the quantitative evaluation of the aforementioned different selection techniques, a comparative study is designed using a repeated-measures within-subjects factorial design including four conditions: Ray Cast, Cone Cast, SQUAD, and our technique. According to [43], two main factors that influence the performance of the selection technique can be identified: object size and object density. Therefore, the study is performed in four different scenarios relating to object size and density. How the scenario varies along the four scenarios is described below. To analyze the efficiency and effectiveness of these selection techniques, the dependent variables are completion time and error number. During the test, we counterbalance the influence of presentation order of the different techniques and the scenarios.

4.2.4 *Test scenarios*

We design four different scenarios that encompass the spectrum of potential selection situations for the evaluation. These scenarios vary in object size (from small to large) and object density (from low to high). All scenarios are illustrated in Figure 7.

Scenario 1. Large size, low density. The scenario consists of a large enclosed area that feature 15 large-sized floating cubes placed in a random manner, as shown in Figure 7(a). We design this scenario to test the basic performance of the four techniques.

Scenario 2. Large size, high density. The participants are presented with a table that features 75 large-sized cubes of varying color (see Figure 7(b)). The cubes are stacked on the table, leading to partial occlusion of the target object. We use this scenario to test the performance of the four techniques in selecting occluded objects.

Scenario 3. Small size, low density. The participants are presented with a pool table that features 15 small-sized spheres of varying color, as illustrated in Figure 7(c). This scenario is used to test the performance of the four techniques in selecting small visible objects.

Scenario 4. Small size, high density. In this scenario, a cardboard box which features 75 small-sized spheres of varying color, is presented to the participants (see Figure 7(d)). This scenario is designed to be the hardest, as the participants experience difficulty in selecting a highly occluded small-sized object.

4.2.5 *Task and procedure*

Before the participants start the formal test, the general purpose of the test is explained to them. The participants are asked to evaluate the four selection techniques in the four different scenarios. To accustom themselves to each selection technique prior to the test, the participants start with a two-minute special practice scenario before testing each scenario and selection technique combination. The special practice scenario consists of several virtual cubes which are randomly placed in the scene. Participants can end the practice scenario at any time if they feel familiar with the technique.

In the formal test, each participant is required to select the target object of each scenario using each of the four selection techniques. During the test, a random object of each scenario is appointed as the target object and is uniquely colored red in the scene. Using the assigned techniques, the participant attempts to select the target and holds his or her dominant hand in the same position for 200 ms to start the selection

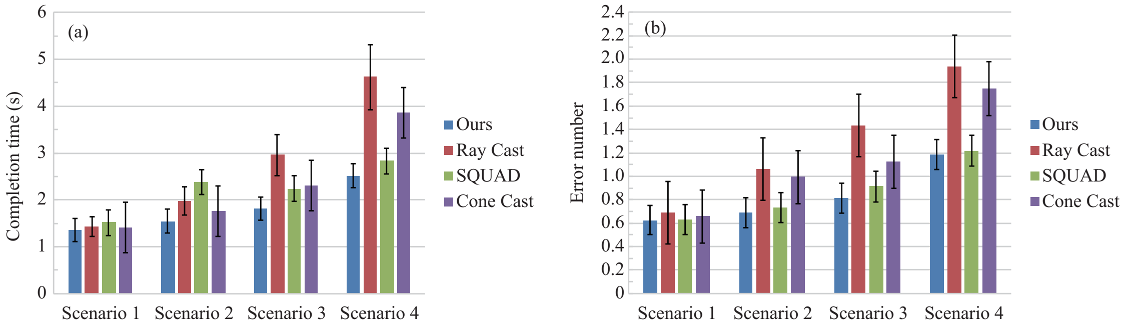


Figure 8 (Color online) The mean completion time (a) and error number (b) for the four techniques in each scenario.

detection. If the system fails to detect a selection or detects an incorrect selection, the participant has to readjust the position of his or her hand until the selection succeeds. Once the participant succeeds in selecting the target object, a notification of the correct selection is instantly displayed on the screen. Subsequently, the next scenario is shifted to the test scene and follows the same procedure described. The interactive portion of the test is accomplished after the participants have tested all four scenarios using each of the four selection techniques.

The participants are initially positioned approximately two meters away from the projection plane. During the multi-step selection process, the participants are informed that they are free to walk or rotate their heads as long as they feel comfortable with the technique. During the formal test, we require the participants to focus on selecting the target without too many misses or slowdowns.

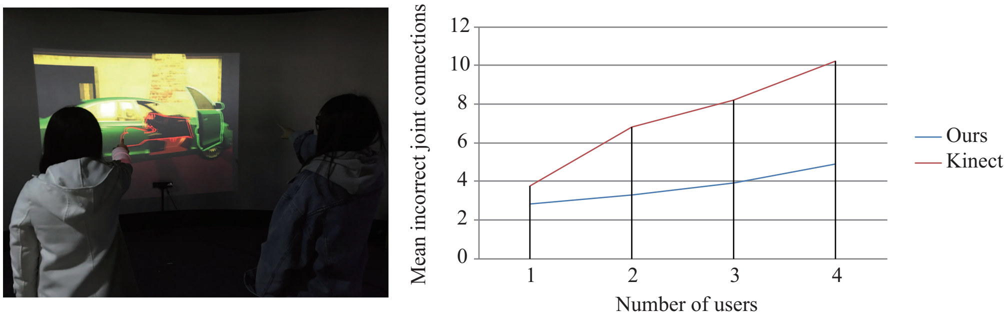
4.2.6 *Measures*

The measures used to investigate the efficiency and effectiveness of these selection techniques are completion time and error number. The completion time is a major factor for investigating the efficiency of these techniques. It is measured by the time for a participant to successfully select one target with the assigned technique in a test scenario. The error number is the factor used to evaluate effectiveness. This factor is measured by the number of wrong selections that occur when a participant adopts the assigned technique to select one target. Because each test scenario is assigned with a single target object, the wrong selection is detected if the participant selects a non-target object.

4.2.7 *Results*

The goal of study one is to compare the performance of the aforementioned selection techniques with our technique in different scenarios. Therefore, we perform Paired-samples T tests on both completion time and error number to analyze the quantitative data of each technique for each scenario.

Completion time. Figure 8(a) shows the mean completion time of each technique for each scenario. In Scenario 1, which features several large-sized cubes in an enclosed area, the mean completion time of our technique (1.36 s ± 0.19) is slightly lower than that of Ray Cast (1.43 s ± 0.25), SQUAD (1.52 s ± 0.20) and Cone Cast (1.41 s ± 0.24). The results of T test show that our technique is significantly faster than SQUAD ($t = 2.78$, $p = 0.007 < 0.01$) but not Ray Cast ($t = 0.89$, $p = 0.19$) or Cone Cast ($t = 0.63$, $p = 0.27$). It can be expected that our technique would perform similarly to Ray Cast and Cone Cast because Scenario 1 is quite simple, and the participants are able to rapidly select the target object by the virtual ray (cone) or fingertip. With regard to Scenario 2, it features many large-sized cubes stacked densely on a table. Our technique (1.55 s ± 0.19) provides a shorter completion time than Ray Cast (1.98 s ± 0.28), SQUAD (2.38 s ± 0.27) or Cone Cast (1.76 s ± 0.24). The T test reveals that our technique significantly reduces the completion time compared with Ray Cast ($t = 4.69$, $p < 0.001$), SQUAD ($t = 9.96$, $p < 0.001$) and Cone Cast ($t = 2.42$, $p = 0.014 < 0.05$). For Scenario 3, which features small-sized objects at a low density, our technique (1.81 s ± 0.19) provides a shorter completion time than Ray Cast (2.96 s ± 0.25), SQUAD (2.24 s ± 0.14) or Cone Cast (2.31 s ± 0.15). The T test results



(a) (b)

Figure 9 (Color online) Illustration of multi-user collaboration (a) in the virtual assembly application. (b) The mean number of incorrect joint connections of each method for each scenario.

show that our technique significantly reduces the completion time compared with Ray Cast (t = 14.84, p < 0.001), SQUAD (t = 7.37, p < 0.001) and Cone Cast (t = 8.21, p < 0.001). Scenario 4 features small-sized cubes at a high density. For this scenario, Ray Cast (4.63 s ± 0.30), SQUAD (2.83 s ± 0.32) and Cone Cast (3.86 s ± 0.24) provide a longer completion time than our technique (2.51 s ± 0.29). The results of T test show that our technique can significantly reduce the mean completion time (for Ray Cast, t = 18.19, p < 0.001; for SQUAD, t = 3.04, p = 0.004 < 0.01; for Cone Cast, t = 15.61, p < 0.001).

Error number. In Figure 8(b), the mean error number of each technique for each scenario is revealed in graphical form. In Scenario 1, the error number of our technique (0.63 ± 0.50) is slightly smaller than that of Ray Cast (0.69 ± 0.60), SQUAD (0.64 ± 0.62) and Cone Cast (0.66 ± 0.53). The results of T test show that our technique causes a collision number similar to that of Ray Cast (t = 0.44, p = 0.33), SQUAD (t = 0.13, p = 0.49) and Cone Cast (t = 0.24, p = 0.41). For Scenario 2, our technique (0.69 ± 0.49) reduces the mean error number compared to Ray Cast (1.06 ± 0.57), SQUAD (0.73 ± 0.53) and Cone Cast (0.99 ± 0.52). By performing the T test, the results show that our technique significantly reduces error number compared to Ray Cast (t = 2.42, p = 0.014 < 0.05) and Cone Cast (t = 2.06, p = 0.03 < 0.05). However, the results reveal that the error number of our technique is similar to that of SQUAD (t = 0.27, p = 0.39). In Scenario 3, the mean error number of each technique is as follows: our technique (0.81 ± 0.54), Ray Cast (1.44 ± 0.73), SQUAD (0.91 ± 0.53) and Cone Cast (1.06 ± 0.57). The result of T test shows that our technique significantly reduces the error number compared with Ray Cast (t = 3.10, p = 0.003 < 0.01). We find that the performance of our technique in terms of error number is similar to that of SQUAD (t = 0.54, p = 0.30) and Cone Cast (t = 1.17, p = 0.13). With regard to Scenario 4, the mean error number of our technique (1.19 ± 0.83) is smaller than that of Ray Cast (1.94 ± 0.85), SQUAD (1.22 ± 0.84) or Cone Cast (1.75 ± 0.93). We find that our technique significantly reduces the error number compared to Ray Cast (t = 2.54, p = 0.01 < 0.05) and Cone Cast (t = 1.86, p = 0.04 < 0.05) while causing an error number similar to that of SQUAD (t = 0.09, p = 0.46).

4.3 Study two

Study two is to evaluate the performance of our body tracking method when multiple users interact in our system. Currently, Kinect SDK is the most frequently used body tracking method for multi-user interaction, and therefore we compare our body tracking method with Kinect SDK in a multi-user interaction scenario. For the comparison, we implement a simplified version of our selection technique that adopts Kinect SDK for body tracking. Because we have introduced a multi-user merge approach which allows several users to interact in a large display VE, we develop a simple virtual assembly application for the study, as shown in Figure 9(a).

4.3.1 *Design*

To investigate the effectiveness of our body tracking method in multi-user interaction, we conduct a comparative study using a within-subjects factorial design including two conditions: our body tracking method and Kinect SDK. The independent variable is the number of participants who simultaneously interact in the system. Owing to the limitation of the tracking area, the number of participants varies from 1 to 4, resulting in four different values for the independent variable. For each value, we randomly select the corresponding number of participants from 20 participants during each test. Each value of the independent variable is tested 10 times. During the test, the dependent variable is the number of incorrect joint connections for our body tracking method and Kinect SDK for different numbers of participants.

4.3.2 *Task and procedure*

We explain the general purpose of study two to the participants before starting the formal test. Each group of participants is required to complete two assembly tasks using our body tracking method and Kinect SDK, respectively. The assembly tasks are all tested in a virtual assembly scene. In the scene, a virtual car, which consists of several automotive components, is contained in a large enclosed area. In each assembly task, the virtual car is split into same discrete parts, and the participants are asked to reconstruct the virtual car by assembling the discrete parts. However, for each assembly task, the discrete parts are randomly placed on the screen. To complete the assembly task, the participants are asked to select all the automotive components and drag them to a fixed target position, which is located at the center of the virtual scene. During the test, the skeletal data of the participants are visually presented to us. If we detect an incorrect skeleton connection caused by mutual occlusion, or the incorrect joint connections of a single skeleton, the participants are required to reposition until all the skeletal data are calculated correctly.

4.3.3 *Measures*

We adopt the number of incorrect joint connections to investigate the effectiveness of the two body tracking methods in multi-user interactions. This factor is measured by the total number of times that the incorrect joint connections of a single skeleton and the incorrectly connected skeletons caused by mutual occlusion occur in a single test.

4.3.4 *Results*

Figure 9(b) shows the mean number of incorrect joint connections of each method for each scenario. For the single-user situation, the results show that both our body tracking method and Kinect SDK have similar performance. However, a significant increase in incorrect joint connections can be found for Kinect SDK as the number of participants increases. In contrast, the number of incorrect joint connections for our body tracking method increases slowly when the number of participants increases. Comparing with Kinect SDK, our body tracking method can effectively reduce the incorrect joint connections when multiple users simultaneously interact in a virtual space. These results indicate that our body tracking method is robust to the processing of occlusion compared to Kinect SDK.

4.3.5 *Extension*

In addition to the effectiveness, we also evaluate the accuracy of our body tracking method by comparing with Kinect SDK. In the test, a participant is free to move around on the tracking area, and his skeleton is calculated by our body tracking method and by Kinect SDK, respectively. To verify our body tracking method, we compare the values calculated by our method with those calculated by Kinect SDK for the same joint. For the test, the positions of the participant's head are recorded in both tracking methods and compared. The comparison between the positions of the participant's head obtained by our body tracking method and those computed by Kinect SDK is shown in Figure 10. The results show that the trajectory of the head tracked by our method is more stable than that calculated by Kinect SDK, which indicates that our method has better accuracy comparing with Kinect SDK. However, when the body parts of a single user are heavily occluded, or multi-user occlusion occurs, the 3D skeletal poses tracked by Kinect SDK yield erroneous reconstructions compared to those computed by our method. As shown

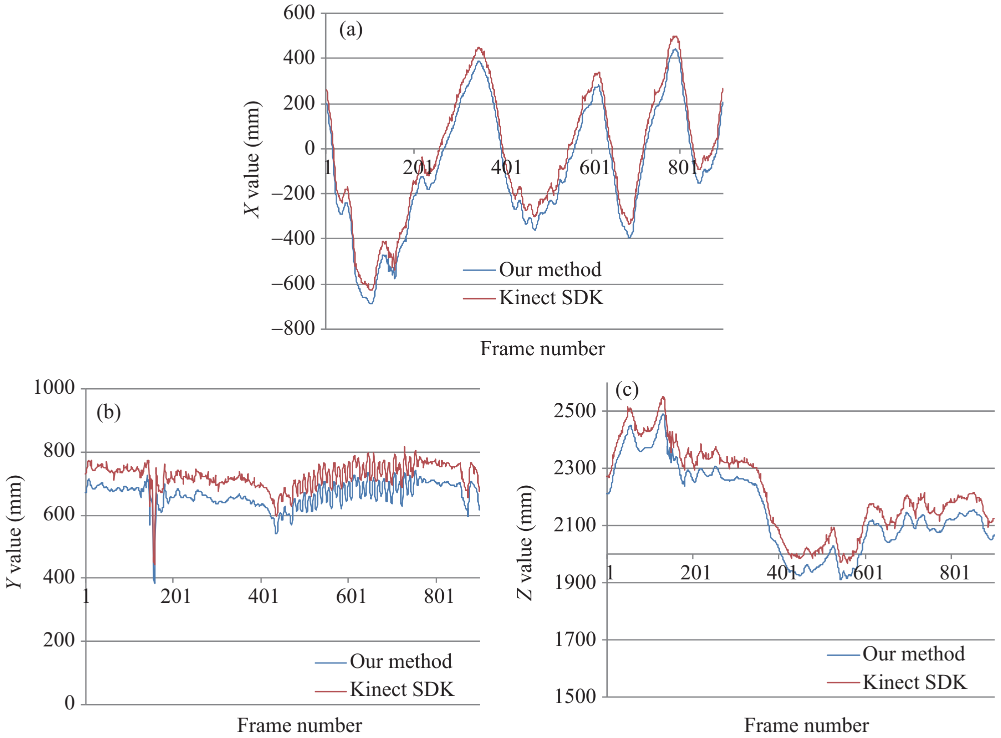
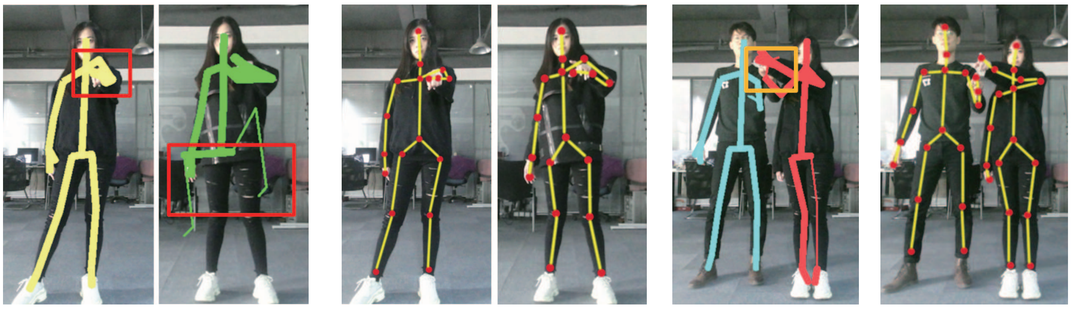


Figure 10 (Color online) Comparison between the trajectories of the user's head tracked by our body tracking method and by Kinect SDK. (a) The values of X coordinate; (b) the values of Y coordinate; (c) the values of Z coordinate.



(a) (b) (c) (d)

Figure 11 (Color online) Our method succeeds in mutual occlusion (b) and (d), while the estimates of Kinect SDK are erroneous (a) and (c).

in Figure 11(a) (red box) and Figure 11(c) (yellow box), the Kinect reconstructions are in disorder. In contrast, our body tracking method succeeds in this case.

4.4 Discussion

4.4.1 *Comparative experiment*

In the comparative experiment, we compare our body tracking method with state-of-the-art CNN-based pose estimating methods on different datasets. According to the results, our body tracking method achieves the same performance and even outperforms the compared state-of-the-art methods on the evaluation datasets. Thus, this demonstrates that our body tracking method is capable of accurately tracking users' bodies. In terms of the real-time performance, the results show that our body tracking method achieves superior speed for processing each frame in contrast with other methods. Furthermore, we investigate the computation time of our body tracking method in multi-user interaction task. When tracking 4 users, the average time cost of our method is 58.8 ms for each frame. Because 58.8 ms for each frame is equivalent to approximately 17 frames per second (fps), the results indicate that our body tracking method is feasible for real-time applications.

4.4.2 *User studies*

In study one, we use the term of completion time and error number to analyze the efficiency and effectiveness of the four techniques. The outcomes are statistically significant, which enables us to draw multiple meaningful conclusions. In Scenario 1, the results of completion time show that our technique has performance similar to Ray Cast and Cone Cast, and is significantly faster than SQUAD. Because Scenario 1 features large-sized objects and the distribution of the objects is sparse, our technique, Ray Cast and Cone cast, could rapidly select the target, while SQUAD spends more time on refinement. For the error number in Scenario 1, we find similar performance of all four techniques owing to the simplicity of the virtual scene.

With regard to Scenario 2, we find significant differences among the four techniques in terms of completion time. As we expected, our technique performs faster than the other three techniques. When looking at the error number for Scenario 2, we notice that our technique significantly reduces the error number compared to Ray Cast and Cone Cast while performing similarly to SQUAD. Owing to the dense distribution of objects in Scenario 2, users could make more wrong selections using Ray Cast or Cone Cast and spend more time on reselection. When using our technique or SQUAD, the progressive refinement effectively reduce wrong selections. However, the Quad menu of SQUAD increases the complexity of refinement, resulting in a longer completion time.

For Scenario 3, the results show that our technique significantly reduces the completion time compared with Ray Cast, SQUAD and Cone Cast. For the error number, the results reveal significant differences between our technique and Ray Cast. Compared to SQUAD and Cone Cast, our technique performs similarly in terms of error number. Considering the features of Scenario 3, the poor performance of Ray Cast in both completion time and error number is to be expected because selecting small-sized objects using the virtual ray is quite difficult. For Cone Cast, the error number result indicates that Cone Cast performs similarly to our technique. The reason is that the virtual cone of Cone Cast and the selection region of our technique are able to efficiently select sparse virtual objects. With regard to SQUAD, the progressive refinement of our technique and SQUAD ensures that both techniques effectively reduce wrong selections and preserve similar performance regarding error number. Nevertheless, the sphere-cast of SQUAD requires users to carefully adjust the position of the sphere to ensure that the desired object is contained in the sphere. This is why SQUAD needs more completion time than our technique.

In Scenario 4, the results reveal that our technique significantly reduces the mean completion time and error number compared with Ray Cast, SQUAD and Cone Cast. Because Scenario 4 is the most complex of the scenarios, Ray Cast has the worst performance in both completion time and error number. For SQUAD, the density of objects increases the steps of refinement to make a single selection, which results in a longer completion time than our technique. In consideration of Cone Cast, the dense objects increase the difficulty of selecting the target object, leading to a longer completion time and higher error number than our technique.

By analysing the mean completion time and error number across all scenarios, the stability of our technique is comparable to that of the other three techniques, and our technique achieves the best performance. Regarding error number, our technique and SQUAD are significantly accurate. However, the results of completion time reveal that our technique performs at a higher speed than SQUAD. In summary, the results indicate that our technique has better performance than Ray Cast, SQUAD and Cone Cast techniques in stationary scenes.

In study two, the effectiveness of our body tracking method in multi-user interaction is evaluated by comparing our method and Kinect SDK. When the body parts of a single user are in complex occlusion or multiple users simultaneously interact in the same tracking area, the results show that our body tracking method has better performance on correctly tracking the skeletons of the users than Kinect.

5 Conclusion

In this paper, we introduce a flexible 3D selection technique that allows multiple users to efficiently select virtual objects in a large display VE. In our selection technique, we present a real-time CNN-based body tracking method that accurately tracks multiple users. Furthermore, we propose a novel region-based selection method and a multi-user merge method to allow multiple users to efficiently select target virtual objects in a single-view stereoscopic display. The proposed body tracking method is compared with state-of-the-art CNN-based multi-person pose estimation methods. The results show that our body tracking method achieves improvements on both accuracy and real-time performance. To evaluate the performance of our selection technique, we conduct a user study to compare our selection technique with prevalent selection techniques. The empirical evidence indicates that our technique clearly outperforms the prevalent techniques in selection performance in stationary scenes. Moreover, to test the performance of our body tracking method in multi-user interaction task, a user study is performed to evaluate our body tracking method against Kinect v2. The results reveal that our body tracking method obtains better performance than Kinect v2 as the number of users increases.

Limitation. In consideration of constraints, our technique is designed for projection-based virtual spaces that require users to face the screen during the interaction. Therefore, our algorithm may degrade in accurately detecting selection if the user's head is not facing the target object. Because our body tracking method can track the proper skeleton for each of multiple users, our technique has the ability to support effective multi-user interaction. However, a limitation of our technique is that the proposed body tracking method works inefficiently when multiple users' body parts are highly overlapped. This limitation causes our body tracking method to struggle with groups of people, such as crowded scenes. In this case, our body tracking method may calculate wrong connection associating parts from two persons in serious occlusion, or fail to detect the pose of the user who is completely occluded by another user. To address this issue, a probable solution is to apply multiple body tracking devices in our system.

In the future, we plan to continue this research by designing and evaluating additional 3D selection techniques. In addition, we would like to improve the proposed body tracking method to overcome the limitation of our technique. Meanwhile, we expect to adopt more multi-user techniques, such as Multi-view [44], to improve the performance of our technique in multi-user interaction.

References

1 Cruz-Neira C, Sandin D J, DeFanti T A, et al. The CAVE: audio visual experience automatic virtual environment. Commun ACM, 1992, 35: 64–72

2 Rademacher P, Bishop G. Multiple-center-of-projection images. In: Proceedings of the 25th Annual Conference on Computer Graphics and Interactive Techniques, Orlando, 1998. 199–206

3 Simon A, Smith R C, Pawlicki R R. Omnistereo for panoramic virtual environment display systems. In: Proceedings of IEEE Annual International Symposium on Virtual Reality, Chicago, 2004. 67

4 van de Pol R, Ribarsky W, Hodges L, et al. Interaction techniques on the virtual workbench. In: Virtual Environments'99. Vienna: Springer, 1999. 157–168

5 Banerjee A, Burstyn J, Girouard A, et al. Multipoint: comparing laser and manual pointing as remote input in large display interactions. Int J Human-Comput Studies, 2012, 70: 690–702

6 Myers B A, Bhatnagar R, Nichols J, et al. Interacting at a distance: measuring the performance of laser pointers and other devices. In: Proceedings of SIGCHI Conference on Human Factors in Computing Systems, Minneapolis, Minnesota, 2002. 33–40

7 Polacek O, Klima M, Sporka A J, et al. A comparative study on distant free-hand pointing. In: Proceedings of European Conference on Interactive Tv and Video, Berlin, 2012. 139–142

8 Nancel M, Wagner J, Pietriga E, et al. Mid-air pan-and-zoom on wall-sized displays. In: Proceedings of SIGCHI Conference on Human Factors in Computing Systems, Vancouver, 2011. 177–186

9 Brown M A, Stuerzlinger W. Exploring the throughput potential of in-air pointing. In: Proceedings of International Conference on Human-Computer Interaction, Toronto, 2016. 13–24

10 Ortega M, Nigay L. Airmouse: finger gesture for 2D and 3D interaction. In: Proceedings of IFIP International Conference on Human-Computer Interaction, Uppsala, 2009. 214–227

11 Vogel D, Balakrishnan R. Distant freehand pointing and clicking on very large, high resolution displays. In: Proceedings of ACM Symposium on User Interface Software and Technology, Seattle, 2005. 33–42

12 Kim K, Choi H. Depth-based real-time hand tracking with occlusion handling using kalman filter and dam-shift. In: Proceedings of Asian Conference on Computer Vision, Singapore, 2014. 218–226

13 Zohra F T, Rahman M W, Gavrilova M. Occlusion detection and localization from Kinect depth images. In: Proceedings of International Conference on Cyberworlds, Chongqing, 2016. 189–196

14 Wu C J, Quigley A, Harris-Birtill D. Out of sight: a toolkit for tracking occluded human joint positions. Pers Ubiquit Comput, 2017, 21: 125–135

15 Wei S E, Ramakrishna V, Kanade T, et al. Convolutional pose machines. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016. 4724–4732

16 Cao Z, Simon T, Wei S E, et al. Realtime multi-person 2D pose estimation using part affinity fields. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2017. 7291–7299

17 Insafutdinov E, Pishchulin L, Andres B, et al. Deepercut: a deeper, stronger, and faster multi-person pose estimation model. In: Proceedings of European Conference on Computer Vision, Amsterdam, 2016. 34–50

18 Iqbal U, Gall J. Multi-person pose estimation with local joint-to-person associations. In: Proceedings of European Conference on Computer Vision Workshops, Crowd Understanding, 2016. 627–642

19 Fang H S, Xie S Q, Tai Y W, et al. Rmpe: regional multi-person pose estimation. In: Proceedings of International Conference on Computer Vision, 2017. 2334–2343

20 Bolas M, McDowall I, Corr D. New research and explorations into multiuser immersive display systems. IEEE Comput Grap Appl, 2004, 24: 18–21

21 Simon A. Usability of multiviewpoint images for spatial interaction in projection-based display systems. IEEE Trans Visual Comput Graph, 2007, 13: 26–33

22 Matulic F, Vogel D. Multiray: multi-finger raycasting for large displays. In: Proceedings of the SIGCHI Conference on Human Factors in Computing Systems, Montreal, 2018. 1–13

23 Ramanan D, Forsyth D A, Zisserman A. Strike a pose: tracking people by finding stylized poses. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Washington, 2005. 271–278

24 Jain A. Articulated people detection and pose estimation: reshaping the future. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Washington, 2012. 3178–3185

25 Pishchulin L, Insafutdinov E, Tang S Y, et al. Deepcut: joint subset partition and labeling for multi person pose estimation. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2016. 4929–4937

26 He K M, Zhang X Y, Ren S Q, et al. Deep residual learning for image recognition. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, 2016. 770–778

27 Liang J D, Green M. JDCAD: a highly interactive 3D modeling system. Comput Graph, 1994, 18: 499–506

28 de Haan G, Koutek M, Post F H. Intenselect: using dynamic object rating for assisting 3D object selection. In: Proceedings of Eurographics Conference on Virtual Environments, Aalborg, 2005. 201–209

29 Steed A, Parker C. 3D selection strategies for head tracked and non-head tracked operation of spatially immersive displays. In: Proceedings of the 8th International Immersive Projection Technology, Workshop, 2004. 13–14

30 Grossman T, Balakrishnan R. The bubble cursor:enhancing target acquisition by dynamic resizing of the cursor's activation area. In: Proceedings of Conference on Human Factors in Computing Systems, Portland, 2005. 281–290

31 Vanacken L, Grossman T, Coninx K. Exploring the effects of environment density and target visibility on object selection in 3D virtual environments. In: Proceedings of IEEE Symposium on 3D User Interfaces, Charlotte, 2007. 115–122

32 Frees S, Kessler G D, Kay E. PRISM interaction for enhancing control in immersive virtual environments. ACM Trans Comput-Hum Interact, 2007, 14: 369–374

33 Kopper R, Bowman D A, Silva M G, et al. A human motor behavior model for distal pointing tasks. Int J Human–Comput Studies, 2010, 68: 603–615

34 Forlines C, Balakrishnan R, Beardsley P, et al. Zoom-and-pick: facilitating visual zooming and precision pointing with interactive handheld projectors. In: Proceedings of ACM Symposium on User Interface Software and Technology, Seattle, 2005. 73–82

35 Kopper R, Bacim F, Bowman D A. Rapid and accurate 3D selection by progressive refinement. In: Proceedings of IEEE Symposium on 3D User Interfaces, Washington, 2011. 67–74

36 Shen Y J, Hao Z H, Wang P F, et al. A novel human detection approach based on depth map via Kinect.

In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Portland, 2013. 535–541

37 Kuang H, Cai S Q, Ma X L, et al. An effective skeleton extraction method based on Kinect depth image. In: Proceedings of International Conference on Measuring Technology and Mechatronics Automation, Changsha, 2018. 187–190

38 Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. In: Proceedings of International Conference of Learning Representation, San Diego, 2015. 1–14

39 Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks. In: Proceedings of the 25th International Conference on Neural Information Processing Systems, Lake Tahoe, 2012. 1097–1105

40 Rosenberg L B. The effect of interocular distance upon operator performance using stereoscopic displays to perform virtual depth tasks. In: Proceedings of IEEE Virtual Reality Annual International Symposium, Washington, 1993. 27–32

41 Andriluka M, Pishchulin L, Gehler P, et al. 2D human pose estimation: New benchmark and state of the art analysis. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Washington, 2014. 3686–3693

42 Lin T Y, Maire M, Belongie S, et al. Microsoft coco: common objects in context. In: Proceedings of European Conference on Computer Vision, Zurich, 2014. 740–755

43 Argelaguet F, Andujar C. A survey of 3D object selection techniques for virtual environments. Comput Graph, 2013, 37: 121–136

44 Kulik A, Kunert A, Beck S, et al. C1x6: a stereoscopic six-user display for co-located collaboration in shared virtual environments. ACM Trans Graph, 2011, 30: 1–12