3D Finger CAPE: Clicking Action and Position Estimation under Self-Occlusions in Egocentric Viewpoint

Youngkyoon Jang, Seung-Tak Noh, Hyung Jin Chang, Tae-Kyun Kim, *Member, IEEE* and Woontack Woo, *Member, IEEE*

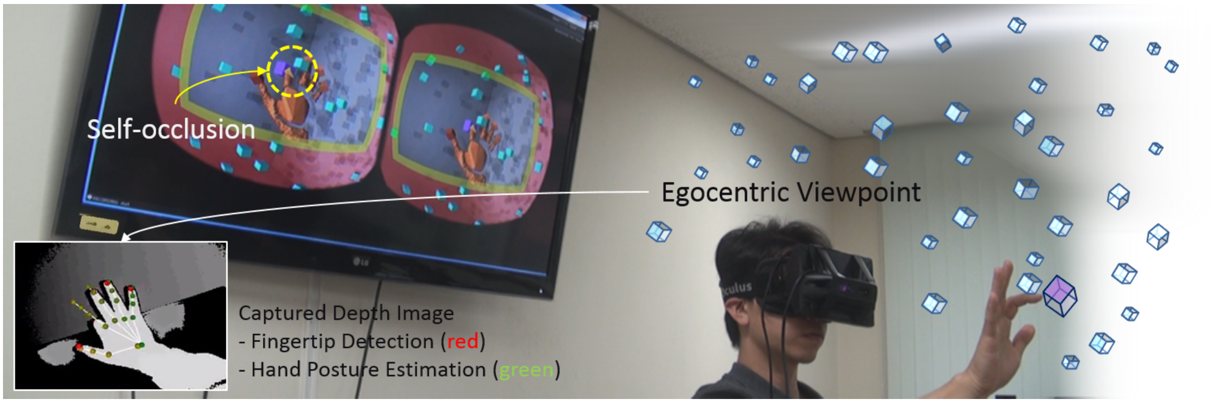

Self-occlusion
Egocentric Viewpoint
Captured Depth Image
- Fingertip Detection (red)
- Hand Posture Estimation (green)

Fig. 1. Our system, called 3D Finger CAPE, supports both 3D finger clicking action detection and clicked position estimation at the same time. In egocentric viewpoint, self-occlusion is caused when a user interacts with VR objects. The proposed spatio-temporal forest estimates 3D clicking positions (*purple cube*) when the pre-learnt action has occurred (*yellow cube*) on the screen. 3D Finger CAPE could be applied to the selection process in an arm reachable AR/VR space.

Abstract— In this paper we present a novel framework for simultaneous detection of click action and estimation of occluded fingertip positions from egocentric viewed single-depth image sequences. For the detection and estimation, a novel probabilistic inference based on knowledge priors of clicking motion and clicked position is presented. Based on the detection and estimation results, we were able to achieve a fine resolution level of a bare hand-based interaction with virtual objects in egocentric viewpoint. Our contributions include: (i) a rotation and translation invariant finger clicking action and position estimation using the combination of 2D image-based fingertip detection with 3D hand posture estimation in egocentric viewpoint. (ii) a novel spatio-temporal random forest, which performs the detection and estimation efficiently in a single framework. We also present (iii) a selection process utilizing the proposed clicking action detection and position estimation in an arm reachable AR/VR space, which does not require any additional device. Experimental results show that the proposed method delivers promising performance under frequent self-occlusions in the process of selecting objects in AR/VR space whilst wearing an egocentric-depth camera-attached HMD.

Index Terms—Hand tracking, spatio-temporal forest, selection, augmented reality, computer vision, self-occlusion, clicking action detection, fingertip position estimation

1 Introduction

Real-time 3D finger clicking action detection and clicked position estimation (3D Finger CAPE) in egocentric view whilst allowing free movement of head-mounted display (HMD) promises several possibilities for augmented reality (AR) and virtual reality (VR) interaction scenarios. 3D Finger CAPE aims to infer an occluded fingertip position as well as clicking action, based on joint points or preliminarily detected fingertip points, from a single-depth image. In AR/VR, research on bare hand tracking and gesture recognition has produced many compelling interaction scenarios, notably multi-touch interaction in mid-air [11], gesture-based input in a wearable AR environment [7, 10], VR scene navigation based on hand tracking [30], and direct object-touch interaction in VR environment by wearing HMD with hand posture estimating sensor (e.g. Leap Motion [1]) attached in the front. However, much of this work was not motivated toward the more sophisticated movements of fingers, especially occluded fingertips. Although there have been studies about hand posture recognition in the field of computer vision, the most recent methods [8, 27, 12, 25, 36] cannot be repurposed directly to 3D Finger CAPE, due to the following challenges of the task:

Self-occlusion in egocentric viewpoint. Self-occlusions are a common problem in general hand pose estimation because of the sophisticated articulations of the hand. Self-occlusions in egocentric viewpoint in particular, however, characteristically have more challenges, as visual information regarding fingertip position is hidden by the back of the hand, as shown in Fig. 1. Additionally, some data from the finger are also hidden by the other fingers when clicking action occurs. The mentioned problems are frequently observed when users are allowed to move their heads freely and utilize multiple fingers to select objects in a wearable AR/VR environment.

Variances of clicking motion. Having no restrictions on a user's finger movements and clicking actions supports a more natural human computer interaction, but leads to other challenges, even for the simple motion of finger bending. More specifically, the large variety of possible movements makes it difficult for the system to decide the target motion. There might be several different types of clicking motion because there are 4 degrees-of-freedom (DOF) for a finger. For instance,

- *Y. Jang, S-T. Noh and W. Woo are with KAIST, Daejeon, S. Korea. E-mail: y.jang@kaist.ac.kr, stnoh@kaist.ac.kr and wwoo@kaist.ac.kr.*
- *H. J. Chang and T-K. Kim are with Imperial College London, London, UK. E-mail: hj.chang@imperial.ac.uk and tk.kim@imperial.ac.uk.*

Manuscript received 18 Sept. 2014; accepted 10 Jan. 2015. Date of Publication 20 Jan. 2015; date of current version 23 Mar. 2015. For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org.
Digital Object Identifier 10.1109/TVCG.2015.2391860

especially with clicking action, a user can tap by using only one joint, bridging finger and palm, or using all joints, utilizing the full 4 DOF. The addressed problems cannot be easily solved (or generalized) by taking a heuristic algorithm because they are caused by the different combinations of the utilized joints, joint positions, speeds, and axes related to the fingers of different users.

Variances of natural posture. Having no restrictions on a user's posture guarantees more comfortable user interaction, but leads to additional challenges, such as variance of motion. More specifically, in addition to the complexity of 4 DOF of a finger, the complexity of 6 DOF (rotation and translation) of the palm of the hand is added to be invariant to the posture of the hand. Nevertheless, it is better to not force restrictions on users in order to resolve the possible problems of self-occlusions, variances of motion, or posture. By enforcing strict rules (such as having all fingers be shown at all times or having the speed of the clicking motion be consistent) on a user makes the user fatigued and the interaction techniques become unnatural.

Addressing the above challenges, a novel spatio-temporal (ST) forest for 3D Finger CAPE is proposed. The ST forest is designed to take benefits from utilizing both spatial and temporal features without any performance degradation. The ST forest not only captures temporal features (e.g. velocity and acceleration), but also utilizes spatial features (e.g. offset between joints) to both detect action and accurately estimate the position of the fingertip with high stability in egocentric viewpoint. For that, the ST forest learns from the sequence of points for fingertips and hand joints, which are detected by using the conventional methods [17, 25]. Even though both detected fingertip and joint data points based on [17, 25] are somewhat noisy due to fast motions of the hand and some fingertips being hidden, the proposed ST forest is able to define the best split function utilizing the most important joints, axes, and offset times which are necessary to consider in order to detect the clicking actions and estimate positions at each split node.

In addition, based on the results of 3D Finger CAPE testing, we both quantitatively and qualitatively experimented the proposed technology in a selection process of VR. For verifying its contribution to the AR/VR community, we made the challenging environment by referring to the prior work [4, 5], which is based on the combinations of sparse, dense, static, and dynamic objects. As far as we are aware, the proposed method is the first framework utilizing spatio-temporal information in a single random forest framework and utilizing its results to select AR/VR objects in egocentric viewpoint, causing self-occlusions and variances of motion. The main contributions are threefold:

(1) Invariant to the rotation and translation of hand: Based on the combination of 2D fingertip detection and 3D hand posture estimation, the clicking actions and occluded fingertip positions are detectable regardless of the positioning of the hand, which is a large improvement over the conventional methods based on the static gestures that require strict front-facing positioning in order for detection to be possible.

(2) Spatio-temporal information learning: The proposed ST forest utilizes both temporal and spatial information to find the best split function at every split node, efficiently handling both action detection and 3D position estimation in a single tree, whilst keeping high detection and estimation accuracies independent of the variance of motion.

(3) Robust bare hand-based selection under self-occlusions: Considering the issue of occluded fingertips in an AR/VR selection process, we suggest a novel approach for natural user interface (NUI) research in egocentric viewpoint, based on the results of the forest. Using the proposed 3D Finger CAPE especially helps to select objects in wearable AR/VR interfaces and provides better performance in accuracy, compared to the conventional methods [3, 20, 25].

2 Related Work

Hand-based interaction in AR. As more AR devices are being developed, demands for natural hand-based interactions in the AR environment, such as fingertip positioning, clicking action or hand gesture, are increasing. There have been some studies done using a device's built-in touch-panel [15], wrist [13] and finger [29]-worn sensors, and optical markers [28]. However, that research cannot be redirected toward an interaction scenario in AR. In AR, especially for user-3D graphic interaction scenarios, the 3D fingertip position in AR space has to be estimated independently of such devices or sensors. Earlier approaches for hand-based interactions in the AR environment are diversified, such as fingertip detection [17, 33] and silhouette segmentation of the hand [31]. Among the diversity, utilizing fingertip position is one of the more interesting and more recent popularly researched approaches because further possibilities still remain, such as gesture-based interaction based on the fingertips [13, 6] and intuitive selection and manipulation for AR objects [28].

The data, which can be gathered only from the visual information presented, would be of the following two types:

Distance based fingertip detection: Bhuyan *et al.* [3] utilized a fingertip position detector for sign language recognition in human-robot interaction. The method depends on the distance between the center position of the palm and contour points of the hand. The most distant points are selected as fingertips, similar to other approaches [20, 19]. The basic idea of the 2D fingertip detection methods is that it is conducted by selecting the geodesic maxima. It is also extended to the case of 3D depth images to detect fingertips [16]. The previous works, however, did not consider the egocentric viewpoint. As a result, those methods might fail to detect occluded fingertips.

3D hand posture estimation-based fingertip detection: Recently presented 3D hand posture estimation methods [8, 27, 12, 25, 35, 32] show good performances on average. Even though a finger is occluded, [12, 25] especially could solve the global optimization problem by depending on the rest of the hand, besides the occluded finger. However, the global optimization causes their obtuse tracking of fast motion, especially for the suddenly occluded fingertip. Very recent approaches [35, 32] for 3D hand posture estimation are fast and can handle slight occlusions, but our challenge of complete occlusions from an egocentric viewpoint are too difficult for them. Furthermore, they all show some difficulties at the finely detailed level, when sophisticated finger movement occurs. Hence, the sophisticated movement of fingertips, such as clicking actions in egocentric viewpoint, might not be estimated utilizing these methods.

Spatio-temporal forests for action detection and regression. There have been several approaches to analyze spatio-temporal data with a random forest framework. In order to estimate spatially and temporally varying relational data, spatio-temporal relational probability trees were proposed [23, 34] and applied to understanding weather processes [24]. The relational feature-based tree building is not rigorous enough for visual data analysis. Yao *et.al.* [37] extended 2D object detecting Hough forests [9] to multi-class action detection in spatio-temporal domain. However, the method requires many dense spatio-temporal local features of relatively long video sequences for robust Hough voting, so on-line detection is impractical. In [26] a simultaneous action recognition and localization method based on a vocabulary forest was proposed. It works on data from an uncontrolled environment, but this method also requires a large number of local features, represented in many vocabulary trees. Yu *et.al.* [38, 39, 40] proposed a random forest based voting method for action detection and search. Although local feature matching becomes much faster, its coarse-to-fine sub-volume search for action detection requires full sequences for an off-line fashion that is not suitable for on-line detection, especially for AR applications.

In our application it is necessary to have a unified framework that can process sequential data for real-time action detection and do position estimation, simultaneously. To the best of our knowledge, there is no such method that can fulfill all the requirements.

3 Methodology

3.1 Problem Formulation

In this paper, the 3D Finger CAPE is formulated, as shown in Fig. 2. In order to utilize spatio-temporal information, we assume that a sequence of frames V is given as an input. From each image included in the sequence 3D finger joint locations $\Phi_{[0:20]}$ and 3D fingertip locations $s_{[0:4]}$ (which are represented as feature vectors) are extracted by a publicly available 3D pose estimator R [25] and a modified 3D fin-

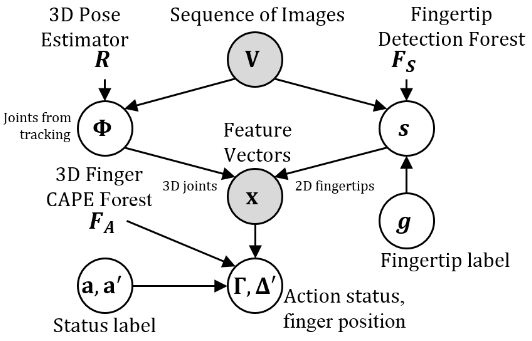
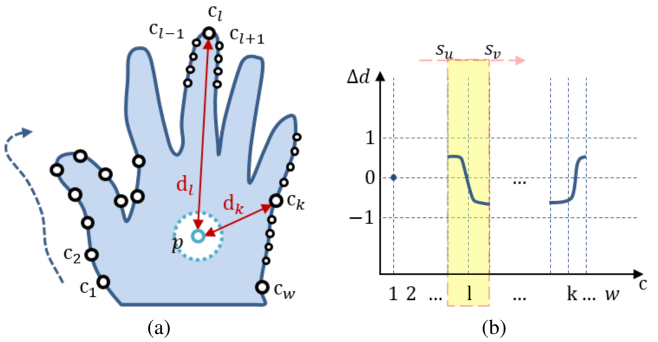


Fig. 2. Graphical representation of the proposed framework (gray nodes: input features).



(a) (b)

Fig. 4. Fingertips detection concept based on a depth image as a 2D gray image: (a) fingertip detection concept (b) differentiation graph along the contour points.

gertip detector F_S based on depth image (described in Sec. 3.2.1). The 3D fingertip detector F_S is a random forest which is using a fingertip shape g as features. By selectively combining those feature vectors at every frame, the spatio-temporal (ST) feature vectors x are composed as input vectors for the proposed framework. Our approach simultaneously learns clicking action a and fingertip position a'. The proposed ST forest F_A is trained to separately capture the spatial or temporal features at each node. After passing the ST feature vectors x through the ST forest F_A, both action status Γ and 3D fingertip position Δ' are estimated by averaging the results stored at the leaf nodes of each tree.

3.2 Spatio-Temporal Feature

The depth image-based fingertip detection method F_S using random forest generally provides fast and robust fingertip detection results. However, it easily fails to detect the fingertips when the fingertips are occluded by the back of the hand or the other fingers. Alternatively, the state-of-the-art 3D hand posture estimator R provides reasonable posture estimation performance, even if some parts of the fingers are occluded. However, the estimator R cannot finely estimate location of every joint as a real hand shape, even in the case where the fingers are all stretched out or finger movements are relatively slow. Thus, we propose to utilize a combination of the two different features in a depth image-based 3D fingertip detector and a 3D hand posture estimator at the same time. Based on a depth image, we extract 2D positions first, then, by utilizing a 3D depth value in the position, we will thus be able to determine the 3D position of the extracted fingertips. The proposed method captures sophisticated movements from the 2D image-based fingertip data s and the general stable posture from the 3D hand posture data Φ.

3.2.1 3D Fingertip Detection (based on Depth Image)

In order to take the benefits of the proven fingertip detection methods, we reimplemented [3, 20]'s method. Based on the binary classification using RF structure, it provides fast fingertip detection results from various viewpoints (mainly for the shown fingertips). For configuring a scale-invariant fingertip shape feature, we first calculate the differentiated values for configuring scale-invariant fingertip shape features when there is a given contour based on Eq. 1, as shown in Fig. 4.

$$\Delta d_l = d_l - d_{l-1} = f(c_l) - f(c_{l-1}), \qquad (1)$$

where $l = \{1, ..., w\}$. w is a number of contour points, as shown in Fig. 4(a). c_l represents the 2D position of the l-th contour point in an image and d_l represents the distance between the contour point c_l and palm center point p, described as $d_l = f(c_l) = \sqrt{(p_x - c_{l_x})^2 + (p_y - c_{l_y})^2}$. Δd of Fig. 4(b) is calculated by Eq. 1. We set Δd_1 to zero. A sliding window is determined by considering a current position of a contour point l, where $s_u = l - \text{offset}$ and $s_v = l + \text{offset}$ shown in Fig.4(b). We set the offset for the sliding window to 7, experimentally.

The fingertip shape feature vector $\mathbf{Y} = \Delta d_{[(l-\text{offset}):(l+\text{offset})]}$ is invariant to the scale changes, as shown in Fig. 4(b). With the feature configuration method, based on the standard RF classification model, we learn the fingertip priors g of Fig. 2, (represented by $\mathbf{Y}$ in this subsection) labelled as true fingertip, which are extracted from the position of the fingertip. Similarly to the two pixel test in [18], the test function compares the values specified by the two randomly chosen elements (e.g. $\mathbf{Y}(\alpha)$ and $\mathbf{Y}(\beta)$, where $\mathbf{Y}(\cdot)$ indicates an element of the feature vector $\mathbf{Y}$), so that it splits a feature dataset D_s of a current split node into two subsets D_s^l and D_s^r, s.t. $D_s^l = \{\mathbf{Y}(\alpha) - \mathbf{Y}(\beta) < \tau_s\}$ and $D_s^r = D_s \backslash D_s^l$. With the trained classifier F_S, we initially detect the candidate points of the fingertips by selecting the maximum probability between true and false choices. Then we find the center position of each fingertip, from clustering the candidate points into five groups of points, as s_i shown in Fig. 3(a).

3.2.2 3D Hand Posture Estimation

In this paper, we utilize the state-of-the-art 3D hand posture estimator [25]. The estimator is based on physics simulation, specifically magnetic properties, to track hand posture. If the joints are close to the depth value, then the joints stick to the depth points based on the physics property, similar to the behavior of magnets. Thus, it shows generally stable overall posture estimation performance. Based on the estimator, we extract hand joints $\Phi_{[0:20]}$, which is a set of 3D vectors, as 3D joints configuring 3D hand posture. Moreover, the estimator provides a basis vector for each joint. Thus, in order to make scale and rotation-invariant ST input features for learning and testing the proposed ST forest, all 3D vectors included in Φ and s are transformed to the local coordinates of the base joint Φ_0 of hand posture as Eq. 2.

$$v_l = M^{-1} v_g = \begin{pmatrix} R & T \\ 0^\top & 1 \end{pmatrix}^{-1} v_g, \qquad (2)$$

where v_l and v_g are local and global coordinates, respectively, and M^{-1} is the inverse matrix of the base joint Φ_0's transformation matrix.

3.2.3 Spatio-temporal Feature Configuration

By selectively combining the elements from those feature vectors Φ and s at every frame, spatial feature vector x is composed, as Eq. 3. In addition, by gathering the x during the predefined time (see Sec. 6 for a detailed discussion), spatio-temporal input feature vector $X = \{x_{it}\}$ is composed and the vector is used for the remaining part of the framework, as shown in Fig. 2.

$$x_{it} = \begin{pmatrix} \Phi_i \\ s_i \end{pmatrix}_t = \begin{pmatrix} \Phi_{(i*4)+1} \\ \Phi_{(i*4)+2} \\ \Phi_{(i*4)+4} \\ s_i \end{pmatrix}_t, \qquad (3)$$

where i and t represent finger and frame(time(ms)) indices, respectively. t is in the range of $[1:n]$. For instance, if the specified finger index is 3, the Φ_i is composed of $\{\Phi_{13}, \Phi_{14}, \Phi_{16}\}$ among $\Phi_{[0:20]}$, as shown in Fig. 3(a).

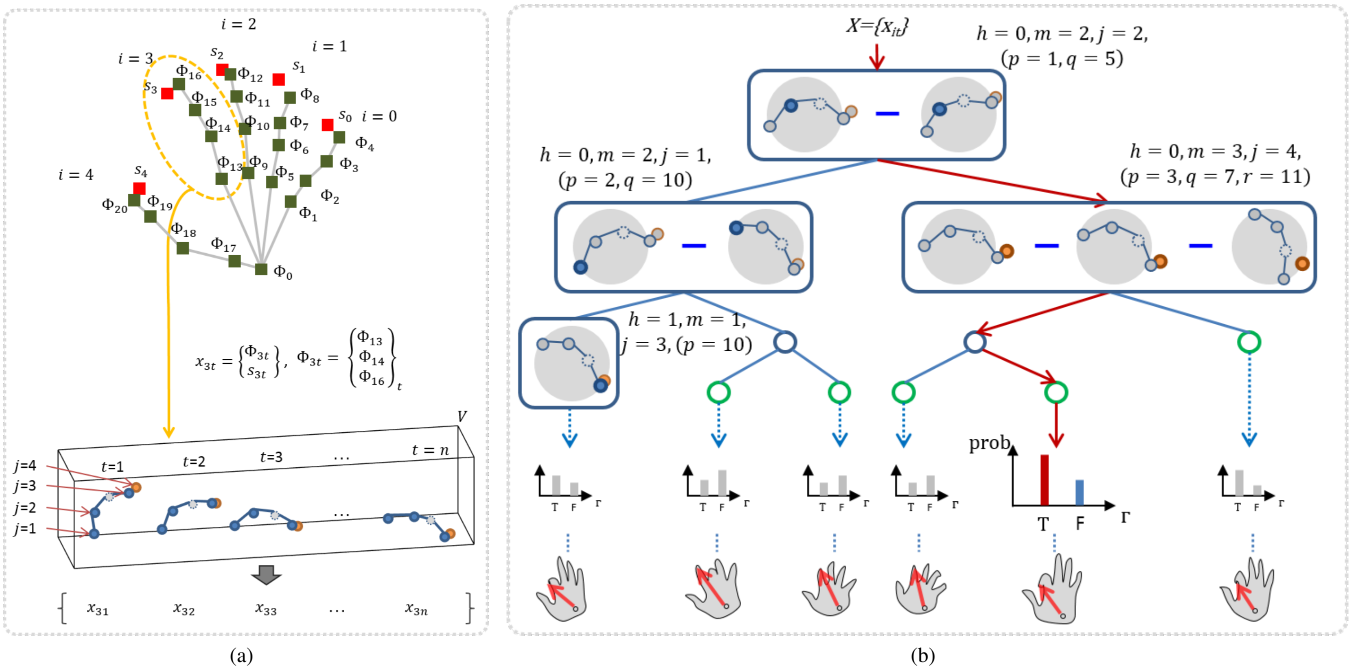

i = 2
i = 1
i = 3
i = 0
i = 4
s_0 s_1 s_2 s_3 s_4
Φ_0 … Φ_{20}
$x_{3t} = \begin{Bmatrix}\Phi_{3t}\\ s_{3t}\end{Bmatrix}$, $\Phi_{3t} = \begin{Bmatrix}\Phi_{13}\\ \Phi_{14}\\ \Phi_{16}\end{Bmatrix}_t$
V
j=4
j=3
j=2
j=1
t=1 t=2 t=3 … t = n
{ x_{31} x_{32} x_{33} … x_{3n} }
(a)
X={x_{it}}
$h = 0, m = 2, j = 2,$ $(p = 1, q = 5)$
$h = 0, m = 2, j = 1,$ $(p = 2, q = 10)$
$h = 0, m = 3, j = 4,$ $(p = 3, q = 7, r = 11)$
$h = 1, m = 1,$ $j = 3, (p = 10)$
prob
T F r
(b)

Fig. 3. The proposed clicking action detection & fingertip position estimation model: (a) an example of configuring the ST feature based on the detected fingertips and the estimated joints of the hand (b) examples of the process of the forest.

3.3 Spatio-temporal Forest

The aim of a ST forest is to detect clicking action and estimate fingertip position in a classifier, even if the fingertip is occluded. A ST forest is an ensemble of randomized binary decision trees, containing two types of nodes: *split* and *leaf*. As the ST forest is inspired by Hough forests [9, 37], split nodes perform a task-specific test function, which is determined by a randomly chosen parameter value h={0: action detection, 1: position estimation} representing a task type, on input data and makes the decision to route them either left or right. Leaf nodes are terminating nodes representing a status of clicking action, and store the probability of the status and store votes for the location of fingertip in 3D space.

3.3.1 Spatio-temporal Forest Learning

Each ST tree in the ST forest is grown by recursively splitting and passing the current training data to two child nodes. At each node, we randomly generate splitting candidates, $\Psi = \{(f_\gamma, \tau_\gamma)\}$, consisting of a function f_γ, and threshold τ_γ, which splits the input data D, into two subsets D^l and D^r, s.t. $D^l = \{I | f_\gamma(V) < \tau_\gamma\}$ and $D^r = D \backslash D^l$. A function f_γ for a splitting candidate is defined as:

$$f_\gamma(V) = \begin{cases} Pos_{j_{axis}}\left(x^V_{i(n-p)}\right), & \text{if } m = 1. \\ Vel_{j_{axis}}\left(x^V_{i(n-p)}, x^V_{i(n-q)}\right), & \text{if } m = 2. , \\ Acc_{j_{axis}}\left(x^V_{i(n-p)}, x^V_{i(n-q)}, x^V_{i(n-r)}\right), & \text{if } m = 3. \end{cases} \quad (4)$$

where $Pos(\cdot)_j, Vel(\cdot)_j$ and $Acc(\cdot)_j$ are the functions returning the position on the specified *axis*, and calculating the velocity and acceleration vector using the values on the specified *axis* of the jth element of the reconfigured spatial feature vectors x_{it}s, respectively. x^V_{in} is a spatial feature vector containing ith finger data at the last frame n of the video V (described in Sec. 3.2.3). p, q and r are random offsets in terms of preceding time (ms) from the last frame of a sequence. The function type m is determined based on the randomly chosen task type h. For instance, if h is 0, meaning action detection task, m is randomly chosen between 2 and 3, which is similar to the random channel selection among multi-feature channels of Hough forests [9, 37]. Otherwise, if h is 1, meaning position estimation task, m is determined by 1.

As mentioned above, in contrast with the standard RF, different types of the splitting candidate ψ^*_γ are stored depending on the specified task h at each split node of the ST tree. For instance, if h is 0, the splitting candidate giving the largest information gain is stored. Otherwise, if h is 1, the splitting candidate giving the smallest regression uncertainty is stored. The information gain is defined as:

$$IG(D) = H(D) - \sum_{k \in \{l,r\}} \frac{|D^k|}{|D|} H(D^k), \quad (5)$$

where $H(\cdot)$ is Shannon's Entropy as:

$$H(D) = - \sum_{u \in \{T,F\}} p(u) log(p(u)), \quad (6)$$

In terms of the regression uncertainty, we simply define it based on the variance of the fingertip position vectors in the local coordinates, as follows:

$$RU(D) = \sum_{k \in \{l,r\}} \frac{|D^k|}{|D|} \operatorname{tr}\left(\Sigma^{D^k}\right), \quad (7)$$

where Σ^{χ} is the sample covariance matrix of the set of the fingertip position vectors and $\operatorname{tr}(\cdot)$ is the trace function. The vectors indicate the offsets from the base joint position to the fingertip position, both transformed into the local coordinates based on Eq. 2. This process is then repeated recursively on each split of the data, D^l and D^r, until it meets the stopping criteria. The growing process of the tree stops when the sample number of the dataset is less than the predefined minimum number (experimentally set as 20) or the depth of the tree exceeds the predefined value (experimentally set as 10).

3.3.2 Testing

As shown in Fig. 3(b), the proposed ST trees find the optimal combinations of parameters to find the best split function at each split node by randomly selecting the value of each parameter in the learning phase, described in Sec. 3.3.1. Moreover, at the leaf node of a ST tree in the forest, clicking action probability and offset vector located in the local coordinates of the base joint of the hand are stored at the training stage.

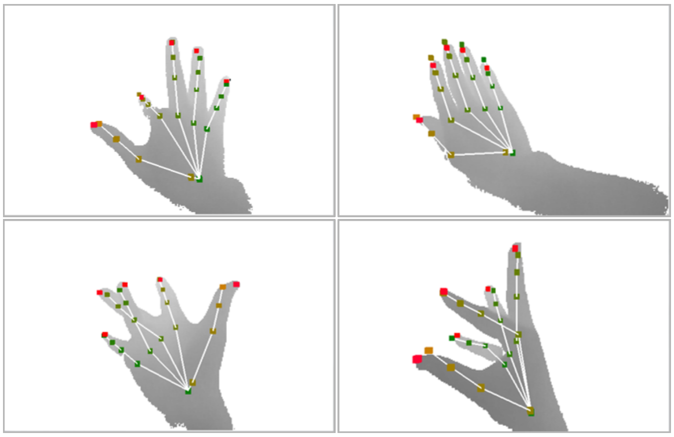


Fig. 5. Examples of the modified geodesic maxima-based detection (red dots) [3, 20] and the 3D hand posture estimation (skeletal model) [25] for the various hand posture cases.

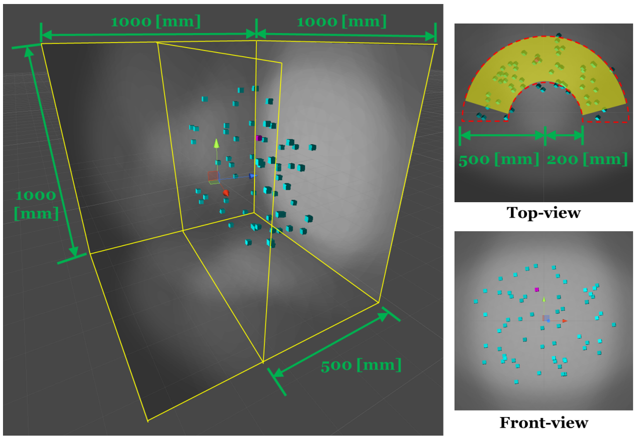


Fig. 6. VR environment setting for object selection experiments.

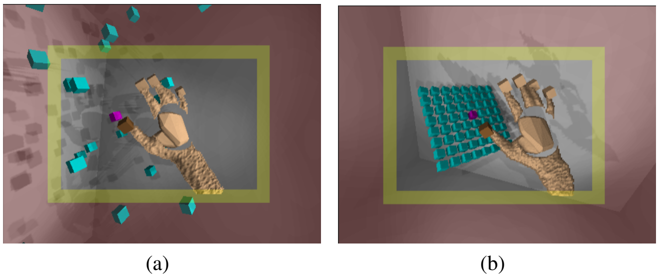


Fig. 7. Example of experimental scenarios of sparse and dense object selections in static scene: (a) objects are spaced farther apart in the sparse environment (b) objects are placed closer together in the dense environment. In dynamic scene, the objects move in the VR environment.

Thus, using the given ST feature X, based on the Eq. 8, the probability of clicking action status and the fingertip position offset in local coordinates are retrieved together from the proposed ST forest.

$$\underset{T,F\in\Gamma,\{x,y,z\}\in\Delta'}{\arg\max} \quad P(Y_\Gamma, Z_{\Delta'}|X), \qquad (8)$$

where $X=\{x_{i,t=1:n}\}$. Y_Γ is the probability of a status of clicking action Γ, where $\Gamma=\{\text{True, False}\}$. $Z_{\Delta'}$ is the local offset position x,y,z of the estimated fingertip when clicking action has occurred. The probability of clicking action Y_Γ and the fingertip position $Z_{\Delta'}$ are calculated by averaging the values stored at a leaf node of the trees in ST forest. The local positions of the fingertip are converted into the global coordinates, which depend on the current posture of the hand.

4 Implementation

4.1 Features and 3D Finger CAPE Results

In this paper, we reimplemented the geodesic maxima selection-based fingertip detector [3, 20], as shown in Fig. 5. For making the stable posture recognition configuring the dataset, we made use of the publicly available 3D hand posture estimator [25], as shown in Fig. 5.

4.2 VR Environment and Scenarios for Experiments

For experiencing a more immersive VR environment, a user is requested to wear a HMD, attached by a camera, as shown in Fig. 1. We use *Oculus Rift* (Development Kit) [21] as a HMD and Intel's *Creative Interactive Gesture Camera* [25] as a depth camera. To square the viewing angle of the HMD with the angle of the camera, we used 109 degree, which is the default viewing angle value in vertical axis of HMD. Moreover, in order to represent the possible input limitation range of the viewing angle (72 × 58 degree) of the camera, we overlaid translucent yellow guidelines of the range onto the VR viewer, as shown in Fig. 7. The VR environment, which is shown in Fig. 7, is implemented by Unity Engine [2]. The metric unit in the VR environment is homogenized as a *mm*.

4.2.1 Four Different Scenarios of VR environment

As shown in Fig. 6, a virtual camera, mapped with the real camera, is placed at the center of the origin in the virtual environment. In the environment, we applied the values of the rotation parameters, which are extracted from the built-in sensors of the HMD, to match the VR viewing direction with the head's direction, led by the user's head movements. As shown in Fig. 6, the virtual objects are enclosed by the 1000 × 1000 × 500 cube-shaped space so that the moving objects would be blocked by the walls of the space. The virtual objects are randomly positioned in the range of [200 : 500] *mm* along z-axis, which is the working range of the [25]'s solution. Moreover, on z-axis basis, the objects are also randomly distributed within the range of [−75:75] degrees on each x and y axes.

Based on the above discussed VR environment setting, we implemented four different scenarios by referring to the previous work [4, 5], which are configured by the factors of arrangement and movement. The arrangement factors consist of sparse and dense, as shown in Fig. 7. The movement factors are categorized as static and dynamic. In the static scenario, the sparse or dense objects stay in their initial position during the task. In the sparse and dynamic scenario, each object has its own initial velocity and travels with uniform motion, but changes its direction periodically so as to prevent it from going outside of the arm reachable area. In the dense and dynamic scenario, all objects are arranged in a grid, and rotating around the grid center axis at approximately 30 degree/second. The target object is colored as a purple cube. Moreover, as shown in Fig. 7, we made the objects in the scenarios all small-sized (e.g. fingertip size) so that we can prove its usefulness for elaborate interactions, such as fine point selection.

5 Evaluation

Experiments were performed to investigate the feasibility of the proposed approach in both quantitative (using our newly gathered hand dataset) and qualitative manners (through user tests in VR environment). Especially for the user test, we migrate the selection mechanism utilizing the 3D Finger CAPE into VR environment to show its practical use. In the user test, we specifically focus on the index finger for both clicking action and 3D occluded fingertip position estimation. As we mentioned above, the 3D Finger CAPE could be used for direct interactions in AR environment, where the AR objects are registered in the environment and a user can naturally approach the AR objects, so making the space arm reachable.

5.1 The 3D Finger CAPE Evaluation Dataset

Existing public hand posture and gesture datasets are inadequate to meet the main objectives of the proposed approach. Whilst benchmarks such as [22, 14] utilize the dataset which contains easily discriminable hand shapes so that they can classify the pre-determined type of gesture (rather than the natural movement of the hand or fingers) from frontal view camera, our framework focuses on the more sophisticated movements of fingers (e.g. bending fingers as well as the 3D translation and 3D rotation of the hand) captured from egocentric view camera. Moreover, the benchmarks utilize the dataset which are captured over a clean background. However, our framework focuses on the natural hand images captured over a natural background.

Dataset. To this end, we collected the *3D Finger CAPE* dataset for training and testing the proposed ST forest by using Intel's *Creative Interactive Gesture Camera* [25]. However, there is still an issue of how to make a ground truth data for the occluded fingertip position when there is no information due to the loss of the depth values of the occluded fingertip. To overcome that information loss, we propose using a paired camera, which can get synchronized frames captured from the frontal and egocentric viewpoints. By utilizing the frontal viewpoint camera, we could capture the depth information of the fingertip, which is occluded in the egocentric viewpoint.

For training, we have collected 10 sequences for the index finger from 3 different subjects with varying hand sizes by asking each subject to make various motions of clicking action utilizing different joints of each finger. Each sequence was then sampled by manually picking the last frame of the clicking action as a true action, and adjusting the pre-estimated fingertip position based on [25] in 3D space as a clicking position. When we manually picked a frame, a set of preceding frames is converted into a spatio-temporal feature of a clicking action. Each sequence, comprised of $2,000$ frames, has at least 25 clicking actions. The dataset, including only positive actions, contains 296 actions of ground truth annotated training pairs of clicking action and fingertip position. Negative actions are randomly sampled by selecting frames from the remaining frames of the training videos which were not labelled as true samples, to form the complete dataset.

Because the finger's movements, especially the more sophisticated clicking actions, are not easily discriminable, not only the manually picked frame, but also the frames nearest to the labelled frame and positions in the frames, could also be true clicking actions and clicked positions. Moreover, because we manually labelled the ground truth, we cannot assure that it is the precise ground truth. Hence, in order to make the ground truth more practical, we applied a perturbation factor by applying 5% of the number of frame n, configuring the ST feature. To establish this, we add an additional ±5% of the configured ST feature frames, and label them as another last frame of the true clicking action and clicked position in the frame. By applying the perturbation factor, we were able to collect $five$ times the number of true clicking actions and fingertip positions for the final dataset, which count as $2.9K$, including each $1.4K$ positive and negative samples, for training.

For testing, we have collected 10 sequences (306 actions in total) , which are different from the training sequences, from the 3 different subjects, capturing different clicking motions utilizing different joints and occluded fingertip positions caused by natural hand and head movements causing scale and viewpoint changes. Furthermore, as Melax's method [25] requires initialization (frontal view of an open hand), in order to do a fair comparison, both the training and testing sequences start in this way.

5.2 Experimental Results using Dataset

Number of tree selection. First, before checking the feasibility of our technical contributions as experiments, we tested to find the saturation point of the accuracy by trying a different number of trees in the ST forest so that we can get an optimal performance with a minimal number of trees. As shown in Fig. 8, we confirm that the Equal Error Rates (EER) of clicking action detection are saturated when the number of trees exceeds 11. Thus, we set 11 as the adequate number of trees to configure a forest for the rest of the experiments in this paper. In the stage of growing trees, we considered the following variables, which are an 'axis', an 'index of element' of reconfigured spatial feature vector, and preceding 'time offsets' depending on the specified 'frame number', which is determined by the 'task type' (see Sec. 3.3.1). Among them, in order to simplify the splitting criteria, we only considered x and y axes for the variable 'axis' because experiments confirmed that the x axis only minimally affects optimal split function, while it can sometimes cause degradation of accuracy due to the confusion caused by a larger number of variables. For each parameter, we tested 15 times to find an optimal value of each parameter by randomly changing the value of it, so that the ST tree learns optimal split function.

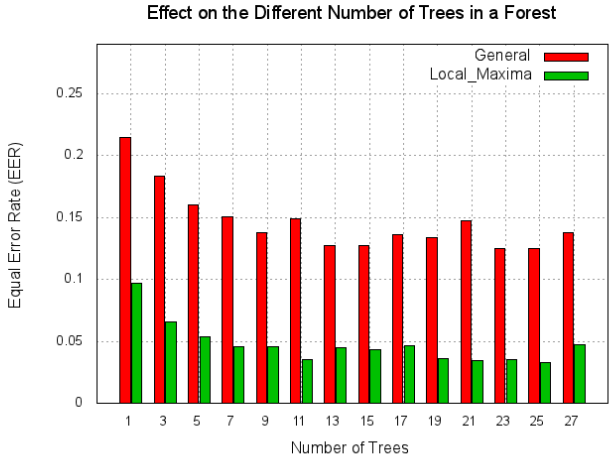


Fig. 8. Experimental results showing the Equal Error Rate (EER) of action detection for two different cases (see text).

For the clicking action detection, we determined the true action by picking the local maxima among the candidates of clicking action detection results, which are within the perturbation range that we have already applied for the making of the training dataset. Because the ST forest gives the probability result for each clicking action, we can easily pick the frame having the highest probability among the values within the perturbation range. As shown in Fig. 8, the EER of the proposed 3D Finger CAPE, picking the local maxima within the perturbation range, are lower than the general case in which it counts the success of clicking action detection only when the action is detected at the exact same frame with the labelled frame of the testing sequence. In our test dataset, experimentally, ±5% perturbation range is converted into ±2 frames.

For 3D fingertip position estimation, we determined the fingertip position by picking the highest value of voting, which is accumulated by the results of each tree. Because the leaf node of each tree of the ST forest stores the offset vector located in the local coordinates of the base joint of the hand, the quantized offset vector position is voted by every tree of the forest. Then, by picking the cell having the highest voting value, representing an offset vector, we estimated the 3D fingertip position. As a result, the measured distance errors between the ground truth values of the dataset and the 3D occluded fingertip position estimations based on the three different methods– depth image-based fingertip detection [3, 20], 3D hand posture estimation [25] and the proposed **ST forest**– were $27.02mm$ (excluding 77.09% detection failure cases), $35.03mm$ and $\mathbf{25.55}mm$, respectively. Experimentally, the average processing time of the proposed framework was 32.63 ms (30.65 FPS) in our experimental environment, which was an Intel Core-i7 3770 CPU processor with 8GB of DRAM.

We analyze the quantitative results of the 3D fingertip position estimation later in Sec. 5.3 because we confirmed that there is a consistency between the results gathered based on the dataset and the results gathered from the user test, as well as to avoid redundant descriptions.

Expanding to multiple fingers. As a second experiment, we confirmed the expandability of the proposed ST forest. Even though the addressed challenges, including self-occlusion, variance of motion and

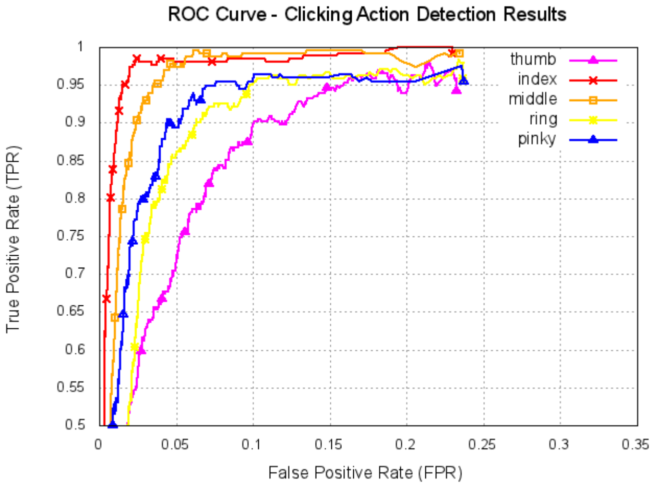


Fig. 9. ROC curves of clicking action classification for five fingers by the proposed ST forest.

variance of comfortable posture depending on each individual, are frequently observable in the case of utilizing the index finger, the case utilizing multiple fingers obviously makes a more severe challenge, especially in terms of self-occlusion. For instance, when a user bends their ring finger in egocentric viewpoint with a comfortable hand posture, the finger could be occluded by other fingers as well as by the back of the hand. Thus, we additionally collected 10 sequences-per-finger (training: 50 sequences and $16.2K$, testing: $1.6K$ actions in total). The collected *3D Finger CAPE* dataset includes those challenges, especially for the middle and ring fingers and the pinky. Based on the proposed ST forest with different finger index i (see Sec. 3.2.3), we tested to check the feasibility of using multiple fingers.

As shown in Fig. 9, the Receiver Operating Characteristic (ROC) curves show that the ST forest-based clicking action detection works accurately for the use of multiple fingers. The reason why the method still shows the reasonable result is that the depth image-based fingertip detection rarely fails to detect fingertips when the finger is bent where 3D posture estimation fails. Thus, as the ST forest has been optimized for classification during learning, the ST input features, by finding the optimal combinations of variables, achieve excellent accuracy. For instance, when 3D hand posture data is unreliable, it relies more on the depth image-based fingertip data, which is independent of the 3D posture data. The clicking action detection accuracies of ST forest utilizing thumb, **index finger**, middle finger, ring finger, and pinky show 89.80%, **96.90**%, 95.68%, 92.82% and 94.37%, respectively.

5.3 Experimental Results via User Tests

In addition to the experiments based on the dataset, to test the feasibility of the proposed 3D Finger CAPE in VR environment, we did a user test for a selection process in VR environment. The experimental environment of the user test is the same as our implementation setting of the four interaction scenarios (see Sec. 4).

Subjects. We ran 12 (male) participants with ages ranging from 22 to 39 with a mean age of 29, four of them office workers and eight graduate students. All participants are aware of the basic knowledge of AR/VR, and two-thirds of them have an experience in using the AR/VR application. Our user study took approximately 45 minutes-per-person, including the training, testing and verbal interview based on the completed post-questionnaire.

Experimental task. Before starting the experimental task, participants were supervised in training session for about 10 minutes to adapt to the VR environment. Moreover, they were asked to do natural clicking motions with natural hand posture, especially using their index finger in the four scenarios. The selection results, which were detected in the training session, were excluded from the evaluation and analysis. In advance, the proctor calibrated a user's hand size and changed the parameters to allow the 3D hand posture estimator [25] to work properly. After that, the proctor demonstrated an initialization process for the 3D hand posture estimator, to prepare for the case where the estimator failed to track a user's hand, so that the user can reinitialize the estimator by himself in a testing session.

Table 1. Post-Questionnaire

Q#-1	This method is easy to use.
Q#-2	This method is a natural way to select the virtual object.
Q#-3	I want to use this method again in the future.

After the training session, participants were asked to select target objects (purple cube) in the four scenarios, using their natural clicking motion. When the action is detected, based on the 3D Finger CAPE, the estimated fingertip position is retrieved simultaneously. Based on the distance between the estimated position and the target object, the system checks if the target object is selected or not. When the distance is closer than $20mm$, the system determines that the virtual object is selected. Moreover, as a visual feedback of the selection result, if the target is selected, the color and size of the object changes.

Experimental design and procedure. There are two types for an evaluation session. The aim of the first session is to compare the performances of the 3D fingertip position estimation triggered by three different methods, which are: depth image-based detection, 3D hand posture estimation-based detection, and 3D Finger CAPE, in a quantitative manner, as shown in Fig. 10. The aim of the second session is to analyze the characteristics of each method in a qualitative manner. Participants were asked to take a 1 minute break between the scenarios for mitigating fatigue. During the experiments, the behavior of participants and the values gathered from the HMD and camera were also recorded for post-analysis.

We used a within-methods design in the first session. In this session, participants were asked to perform the four scenarios described in Sec. 4.2. In the first session of the evaluation, all three methods were activated internally to check the distance between the target and the estimated position. If the target object is selected by at least one method among the three in the scenario, the stage is finished and the next stage showing another random arrangement of virtual objects starts. Each scenario has 18 stages. The order of the scenarios were randomized for counterbalance.

We conducted the second session for examining the difference of the user experiences, based on the three different methods. Based on a within-subjects design, participants were also asked to select the target object in the scenario composed of dense and static objects (scenario#-3), as shown in Fig. 7(b). However, at each trial, only one individual method is activated and the estimated fingertip position is visualized as clicking action is detected. Each trial for showing an individual method consists of 10 stages, and only the position of the target object is changed at every stage so that the participants only focus on the characteristics of the different methods. After the session, post-questionnaire utilizing 7 Likert scale was given (see Table 1) to get the feedback from the participants. After completing the session for testing the three different methods, participants were asked to rank their preference of each method.

3D Fingertip Position Estimation and Statistical Analysis. Based on the data recorded from the first session, we transformed the estimated 3D fingertip positions in the global coordinates into the positions in the camera coordinates. In the first session, for each clicking action, a 2 or 3-tuple including three dimensional vector is stored, because all three methods are activated at the same time. When depth image-based fingertip detection fails, only 2-tuple could be stored. Based on the gathered set of tuples through the four scenarios of the sessions with all the participants, we counted up the success and failure cases of each method for further analysis. Outliers, recorded because of the false action detection, were rejected by checking if the distance of the estimated point was farther than the predefined threshold. Experimentally, we set the threshold to 80 mm.

As shown in Fig. 10, based on the recorded data, we plotted the

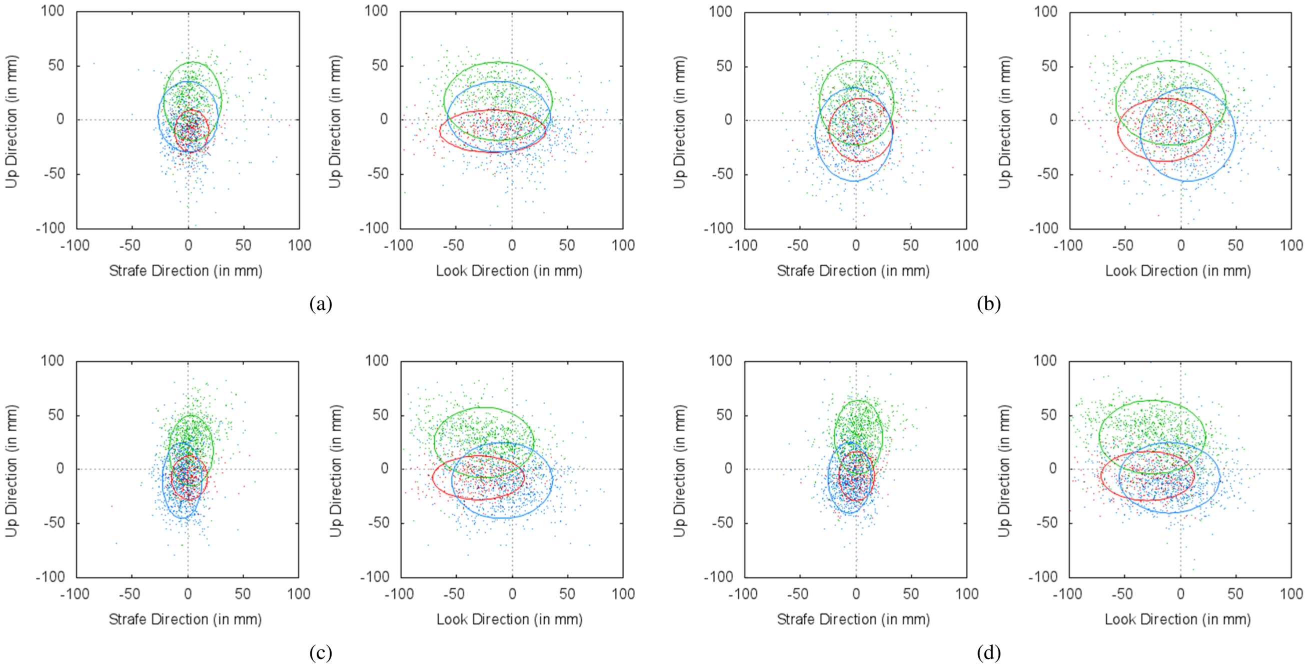


Fig. 10. The results of the distance errors between the target position $(0,0,0)$ coordinates and the estimated fingertip position, based on GM-based (red dots) [3, 20], HPE-based (green dots) [25], and the 3D Finger CAPE (blue dots), plotted in the camera coordinates gathered in the four different scenarios, composed of: (a) sparse and static objects (scenario#-1) (b) sparse and dynamic objects (scenario#-2) (c) dense and static objects (scenario#-3) (d) dense and dynamic objects (scenario#-4).

distance error points resulting from the three different methods (red points: Geodesic Maxima (GM)-based method (also represented as depth image-based method in this paper) [3, 20], green points: 3D Hand Posture Estimation (HPE)-based method [25], and blue points: the 3D Finger CAPE). When fingertips are visible during the clicking actions, GM-based method provides the most accurate estimation results. However, as shown in 11, the method easily fails to detect the fingertip when it is bent variably and occluded. Thus, as we mentioned above, in order to make it successful, GM-based method restricts the natural interaction and makes users feel pressured to keep the strict posture. In addition, even though 3D posture estimation-based detection [25] does not fail to track the posture of the hand, it does not fit the fingertips at a fine level.

Moreover, when a finger is suddenly occluded because of a clicking motion in egocentric viewpoint, the part of the skeleton model representing the finger tends to stay at the position where the depth value was in the preceding frames or is slowly bent to find optimal solution based on the structural constraints of the finger skeleton, like a guessing procedure. Because of the characteristics of the 3D HPE-based method [25], the estimation points of the fingertip are generally plotted above the target position, as shown in 10. In contrast to the state-of-the-art methods, the 3D Finger CAPE provides the most accurate and stable results, as shown in Table 2. The reason why the distance errors gathered through the user test are somewhat larger in number than the errors based on the dataset described in Sec. 5.2 is because there are human factors causing different types of error, such as depth perceptions. Thus, the error distributions, even between the scenarios, are different from each other.

Additionally, in order to check the statistical significance of the results, we utilized the Repeated Measures analysis of variance (ANOVA) test and the T-test as a post-hoc test. As a result, we confirmed that the distance error results based on the three methods are statistically significant as well as the error distribution of each method is distinguished from other methods at a 99% significance level (see Table 3), in all scenarios except the scenario composed of sparse and dynamic objects (scenario#-2). The stages of the scenario#-2 made users feel confused about the depth perception and, as a result, the three methods were not able to perform properly in that scenario. The reason why the T-test values between the GM-based method [3, 20] and the CAPE method in the scenarios of #-3 and 4 do not show statistical significance is that the analysis was done with the estimated points from the GM-based method, which were made by declassifying the failure cases.

Table 2. Mean (standard deviation) of distance error between the target object and the estimated position, based on each method in the different scenarios. (Unit: *mm*)

Scenario#	GM-based [3, 20]	HPE-based [25]	**CAPE**
#-1 (sparse&static)	33.77 (21.62)	42.58 (23.68)	**38.26** **(19.21)**
#-2 (sparse&dynamic)	38.13 (18.32)	46.09 (22.09)	**43.55** **(20.30)**
#-3 (dense&static)	38.02 (21.92)	47.52 (23.49)	**37.99** **(18.12)**
#-4 (dense&dynamic)	38.45 (22.28)	51.88 (23.73)	**37.31** **(17.52)**

Preferences based on Post-Questionnaire. Finally, through the post-questionnaire utilizing the 7 Likert scale shown in Table 1 and an informal verbal interview, we checked the order of the preferences among the three methods. As shown in Fig. 12, the proposed 3D Finger CAPE is the most preferred method in terms of the selection process in VR environment, based on the answers of the questionnaire. In order to check the statistical significance of the results based on the Likert score, which is a non-parametric value, we utilized the Friedman test and Wilcoxon Signed Rank test as a post-hoc test, which are appropriate for the analysis of non-parametric values. As shown in Table 4, we confirmed that there are statistically significant differences between 3D Finger CAPE and other methods.

Comments from the interview session. Even though some participants noticed that there was some odd tendencies with the other methods, such as the estimated position based on GM method relies on the maximally distant position of the hand shape, rather than the real posi-

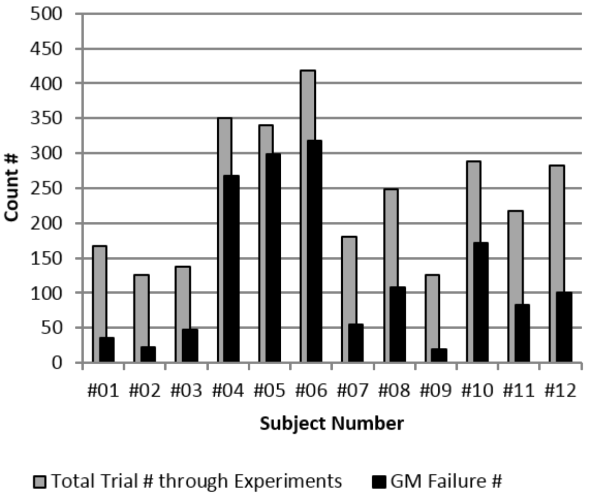


Fig. 11. Experimental results showing the number of trials from a subject and failures triggered by GM-based method [3, 20]: (gray) number of total trials (black) failure counts.

Table 3. Statistical results of the first session, based on ANOVA and T-test as a post-hoc test.

Scenario#	ANOVA	[3, 20] vs. CAPE	[25] vs. CAPE
#-1 (sparse& static)	$F_{value} = 15.892$ $F_{crit} = 3.858$ $p < 0.001$	$t_{value} = 3.046$ $t_{crit} = 2.585$ $\mathbf{p < 0.01}$	$t_{value} = 3.384$ $t_{crit} = 2.580$ $\mathbf{p < 0.001}$
#-2 (sparse& dynamic)	$F_{value} = 5.825$ $F_{crit} = 3.858$ $p < 0.05$	$t_{value} = 4.038$ $t_{crit} = 3.304$ $\mathbf{p < 0.001}$	$t_{value} = 1.987$ $t_{crit} = 1.962$ $\mathbf{p < 0.05}$
#-3 (dense& static)	$F_{value} = 122.135$ $F_{crit} = 3.852$ $p < 0.001$	$t_{value} = 0.024$ $t_{crit} = 1.964$ $p = 0.981$	$t_{value} = 9.436$ $t_{crit} = 3.297$ $\mathbf{p < 0.001}$
#-4 (dense& dynamic)	$F_{value} = 326.309$ $F_{crit} = 3.852$ $p < 0.001$	$t_{value} = 0.865$ $t_{crit} = 1.964$ $p = 0.387$	$t_{value} = 14.775$ $t_{crit} = 3.296$ $\mathbf{p < 0.001}$

tion, and the estimated fingertip position based on 3D HPE method stays in the starting position when fast finger movement occurs, most participants did not notice those tendencies. Also, most felt the 3D Finger CAPE gave better estimating response, in terms of the selection of the target object. Thus, most of the participants answered that they prefer the 3D Finger CAPE for selecting VR objects. However, a few participants answered that another method was more convenient as they could adjust their hand movements to make the method succeed, adapting to the tendency of the estimator. On the other hand, the restrictions on movements necessary to make the methods succeed might cause a user fatigue. Additionally, a user needs time to know how the method works before they are able to adapt to the tendency and use it successfully. Because of the difficulties in perceiving depth, users had some trouble to accurately select VR objects in the testing sessions, as similarly described in the prior study [21].

6 Discussion

The main differences between the methods based on the state-of-the-arts [3, 20, 25] and our proposed ST forest are our focuses on not only occlusion-invariant fingertip position estimation, but also clicking action detection in occlusion situations. The existing methods [3, 20, 25] are not able to detect the sophisticated movements of the fingers (especially self-occluded fingertips) and thus have to make a heuristic rule to interact with AR objects, like “a user has to make a fist and wait two seconds before selecting it” or “a user has to move their hand about ten centimeters in a forward direction to select a menu item”. However, those are still 2D selections and do not provide three dimensional interaction results. In contrast, our proposed algorithm detects clicking action, which is independent of the variance of motions and occlusions. Moreover, because it estimates occluded fingertip position in 3D space, it interacts with AR objects intuitively.

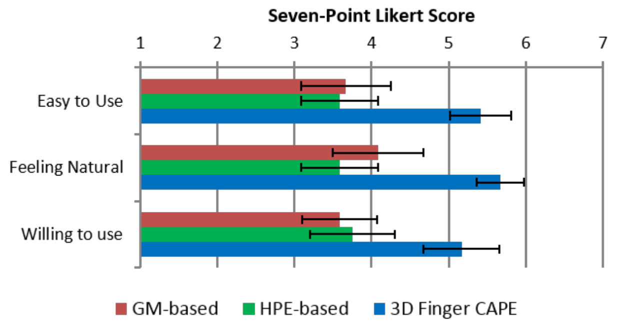


Fig. 12. Mean and standard deviation results of user’s preference on each method based on the post-questionnaire. (GM: [3, 20], HPE: [25])

Table 4. Statistical results based on the questionnaire, based on Friedman and T-test as a post-hoc test.

Question#	Friedman	GM [3, 20] vs. CAPE	HPE [25] vs. CAPE
#-1	$Q = 9.125$ $p < 0.05$	$T = 7$ $T_{crit} = 17$ $\mathbf{p < 0.05}$	$T = 7$ $T_{crit} = 9$ $\mathbf{p < 0.01}$
#-2	$Q = 9.042$ $p < 0.05$	$T = 7$ $T_{crit} = 17$ $\mathbf{p < 0.05}$	$T = 0$ $T_{crit} = 9$ $\mathbf{p < 0.01}$
#-3	$Q = 6.167$ $p < 0.05$	$T = 5$ $T_{crit} = 17$ $\mathbf{p < 0.05}$	$T = 14$ $T_{crit} = 17$ $\mathbf{p < 0.05}$

Besides the above translated quantitative and qualitative results shown in this paper, the technology has many possibilities to be extended into the field of gesture recognition and interaction applications based on the bare hand movement because the proposed ST forest is easily applied for the multi-finger case, as shown in Fig. 9. For example, various combinations of sequential bent fingers can generate various gestures, even in the case of occlusions, which means that the gesture would be invariant to the scale, translation and rotation of the hand in egocentric viewpoint. Moreover, in the case of applications encouraging direct interactions with AR objects, use of the ST forest is possible, such as when typing on an AR keyboard or piano playing on an AR piano. In addition, physics based interaction is also possible because a split function inside of the ST forest has already utilized the velocity and acceleration values. Thus, we can utilize those values to interact with AR objects in AR games.

Nevertheless, there are still some technical limitations. The proposed framework depends on the performance of the utilized methods [3, 20, 25] which are used for configuring the spatio-temporal feature. The utilized hand tracking method [25] sometimes fails to track when most of the fingers are completely occluded. When the tracking method fails, it needs to be reinitialized in order to get a proper spatio-temporal feature. Consequently, a user was requested to keep their hand naturally open so that the tracking method performs properly. However, we expect that the gesture based on the proposed method will become more realistic as the performance of the tracking technique gets better.

In addition, the performance of the proposed algorithm might be degraded by severe variances of motion. For the feature configuration described in Sec. 3.2.3, we set the maximum time period to 500 ms, which can capture a general clicking action within the given time. According to our measurement results throughout the dataset, the action occurs within 500 ms at most. As we confirmed from the experiments, our proposed algorithm is invariant to the variances of motion happening within the short time period, especially for our dataset. However, we inform that the large variances of a different gesture motion taking a longer time period might affect the performance degradation.

7 Conclusions

This paper presents the ST forest-based 3D Finger clicking action and position estimation (CAPE) under self-occlusion scenarios of egocentric viewpoint. One problem that arises from allowing a user to have free movement whilst wearing head-mounted display (HMD) is the occurrence of self-occluded (hidden) fingertip position and action detection during occlusion. Experimental results demonstrate that our approach, using a combination of depth image-based fingertip detection along with the help of 3D hand posture estimation from noisy and occluded data, results in superior performance when compared with state-of-the-art methods. Experimentally, the ST forest retains accurate clicking action detection (96.90%) and the most accurate position estimation performance (25.55 *mm*) when compared with state-of-the-arts, especially as the position estimation provides comparably more stability than the scattered estimation that can occur with previous methods. Moreover, we found that the proposed method encourages more intuitive direct interactions in the scenarios of small-sized object selection, which is directly extendable into AR environment and independent of rotation and translation of the hand.

In the future work, we plan to extend this work for more sophisticated gesture recognition, including work utilizing all fingers and different combinations of the actions, as well as increase other fingers' performance in the case of more severe self-occlusion. Moreover, we plan to integrate a method that can deal with varying action speeds (e.g. Dynamic Time Warping).

Acknowledgments

This work was supported by the Global Frontier R&D Program on <Human-centered Interaction for Coexistence> funded by the National Research Foundation of Korea grant funded by the Korean Government(MSIP)(2010-0029751)

References

[1] Leap motion. `http://www.leapmotion.com/`. Accessed Sep. 10, 2014.

[2] Unity engine. `http://unity3d.com/`. Accessed Sep. 15, 2014.

[3] M. K. Bhuyan, Neog, D. R., and M. K. Kar. Fingertip detection for hand pose recognition. *International Journal on Computer Science and Engineering*, 4(3):501–511, March 2012.

[4] J. Cashion, C. A. Wingrave, and J. J. L. Jr. Dense and dynamic 3d selection for game-based virtual environments. *IEEE TVCG*, 18(4):634–642, 2012.

[5] J. Cashion, C. A. Wingrave, and J. J. L. Jr. Optimal 3d selection technique assignment using real-time contextual analysis. In *3DUI*, pages 107–110. IEEE, 2013.

[6] W. H. Chun and T. Höllerer. Real-time hand interaction for augmented reality on mobile phones. In *IUI*, pages 307–314, New York, USA, 2013.

[7] A. Colaço, A. Kirmani, H. S. Yang, N.-W. Gong, C. Schmandt, and V. K. Goyal. Mime: Compact, low power 3d gesture sensing for interaction with head mounted displays. In *UIST*, pages 227–236, USA, 2013.

[8] M. de La Gorce, D. J. Fleet, and N. Paragios. Model-based 3d hand pose estimation from monocular video. *IEEE TPAMI*, 33(9):1793–1805, Sept. 2011.

[9] J. Gall and V. Lempitsky. Class-specific hough forests for object detection. In *CVPR*, 2009.

[10] T. Ha, S. Feiner, and W. Woo. Wearhand: Head-worn, RGB-D camera-based, bare-hand user interface with visually enhanced depth perception. In *ISMAR*, pages 219–228. IEEE, September 2014.

[11] G. Hackenberg, R. McCall, and W. Broll. Lightweight palm and finger tracking for real-time 3d gesture control. In *Virtual Reality Conference (VR), 2011 IEEE*, pages 19 –26, march 2011.

[12] N. K. Iason Oikonomidis and A. Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *BMVC*, pages 101.1–101.11, 2011.

[13] D. Kim, O. Hilliges, S. Izadi, A. Butler, J. Chen, I. Oikonomidis, and P. Olivier. Digits: Freehand 3d interactions anywhere using a wrist-worn gloveless sensor. In *UIST*, 2012.

[14] T.-K. Kim, S.-F. Wong, and R. Cipolla. Tensor canonical correlation analysis for action classification. In *CVPR*. IEEE Computer Society, 2007.

[15] S. G. Kratz, P. Chiu, and M. Back. Pointpose: finger pose estimation for touch input on mobile devices using a depth sensor. In *ITS*, pages 223–230. ACM, 2013.

[16] P. Krejov and R. Bowden. Multi-touchless: Real-time fingertip detection and tracking using geodesic maxima. *2013 10th IEEE International Conference and Workshops on Automatic Face and Gesture Recognition (FG)*, 0:1–7, 2013.

[17] T. Lee and T. Höllerer. Multithreaded hybrid feature tracking for markerless augmented reality. *IEEE TVCG*, 15(3):355–368, 2009.

[18] V. Lepetit and P. Fua. *Keypoint Recognition Using Random Forests and Random Ferns*, pages 111–124. Springer, 2013.

[19] H. Liang, J. Yuan, and D. Thalmann. 3d fingertip and palm tracking in depth image sequences. In *Proceedings of the 20th ACM International Conference on Multimedia*, MM '12, pages 785–788, New York, NY, USA, 2012. ACM.

[20] Y. Liao, Y. Zhou, H. Zhou, and Z. Liang. Fingertips detection algorithm based on skin colour filtering and distance transformation. In *QSIC*, pages 276–281. IEEE, 2012.

[21] P. Lubos, G. Bruder, and F. Steinicke. Analysis of direct selection in head-mounted display environments. In *3DUI*, pages 1–8, 2014.

[22] S. Marcel, O. Bernier, J.-E. Viallet, and D. Collobert. Hand gesture recognition using input-output hidden markov models. In *FG*, pages 456–461. IEEE Computer Society, 2000.

[23] A. McGovern, N. Hiers, M. Collier, D. Gagne, and R. Brown. Spatiotemporal relational probability trees: An introduction. In *Data Mining, 2008. ICDM '08. Eighth IEEE International Conference on*, pages 935–940, 2008.

[24] A. McGovern, D. John Gagne, II, N. Troutman, R. A. Brown, J. Basara, and J. K. Williams. Using spatiotemporal relational random forests to improve our understanding of severe weather processes. *Stat. Anal. Data Min.*, 4(4):407–429, Aug. 2011.

[25] S. Melax, L. Keselman, and S. Orsten. Dynamics based 3d skeletal hand tracking. In *Proceedings of the 2013 Graphics Interface Conference*, GI '13, pages 63–70, Toronto, Ont., Canada, Canada, 2013.

[26] K. Mikolajczyk and H. Uemura. Action recognition with motion-appearance vocabulary forest. In *CVPR*, 2008.

[27] S. Miyamoto, T. Matsuo, N. Shimada, and Y. Shirai. Real-time and precise 3-d hand posture estimation based on classification tree trained with variations of appearances. In *ICPR*, pages 453–456. IEEE, 2012.

[28] G. F. Natalia Bogdan, Tovi Grossman. Hybridspace: Integrating 3d freehand input and stereo viewing into traditional desktop applications. In *3DUI*. IEEE, 2014.

[29] M. Ogata, Y. Sugiura, H. Osawa, and M. Imai. iring: Intelligent ring using infrared reflection. In *UIST*, pages 131–136, New York, NY, USA, 2012.

[30] Z. Pan, Y. Li, M. Zhang, C. Sun, K. Guo, X. Tang, and S. Z. Zhou. A real-time multi-cue hand tracking algorithm based on computer vision. In *Proceedings of the 2010 IEEE Virtual Reality Conference*, VR '10, pages 219–222, Washington, DC, USA, 2010. IEEE Computer Society.

[31] N. Petersen, A. Pagani, and D. Stricker. Real-time modeling and tracking manual workflows from first-person vision. In *ISMAR*, pages 117–124. IEEE Computer Society, Oct. 2013.

[32] C. Qian, X. Sun, Y. Wei, X. Tang, and J. Sun. Realtime and robust hand tracking from depth. In *CVPR*, June 2014.

[33] Y. Shen, S. K. Ong, and A. Y. C. Nee. Vision-based hand interaction in augmented reality environment. *IJHCI*, 27(6):523–544, 2011.

[34] T. A. Supinie, A. McGovern, J. Williams, and J. Abernathy. Spatiotemporal relational random forests. In *2012 IEEE 12th International Conference on Data Mining Workshops*, pages 630–635, 2009.

[35] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *CVPR*, June 2014.

[36] D. Tang, T.-H. Yu, and T.-K. Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *ICCV*, December 2013.

[37] A. Yao, J. Gall, and L. V. Gool. A hough transform-based voting framework for action recognition. In *CVPR*, 2010.

[38] G. Yu, J. Yuan, and Z. Liu. Real-time human action search using random forest based hough voting. In *Proceedings of the 19th ACM International Conference on Multimedia*, MM '11, pages 1149–1152, 2011.

[39] G. Yu, J. Yuan, and Z. Liu. Unsupervised random forest indexing for fast action search. In *CVPR*, pages 865–872, 2011.

[40] G. Yu, J. Yuan, and Z. Liu. Action search by example using randomized visual vocabularies. *IEEE TIP*, 2012.