Information Sciences 369 (2016) 368–387

Contents lists available at ScienceDirect

Information Sciences

journal homepage: www.elsevier.com/locate/ins

An HCI paradigm fusing flexible object selection and AOM-based animation

Zhiquan Feng[a,b], Bo Yang[a,b,*], Hong Liu[c,d], Na Lv[a,b], Xiaohui Yang[a,b], Jianqin Yin[a,b], Yuan Zhang[a,b], Xiuyang Zhao[a,b]

[a] School of Information Science and Engineering, University of Jinan, 250022, China
[b] Shandong Provincial Key Laboratory of Network-based Intelligent Computing, Jinan, 250022, China
[c] School of Information Science and Engineering, Shandong Normal University, 250014, China
[d] Shandong Provincial Key Laboratory for Novel Distributed Computer Software Technology, 250014, China

ARTICLE INFO

Article history:
Received 12 August 2015
Revised 14 June 2016
Accepted 28 June 2016
Available online 30 June 2016

Keywords:
Gestural UI
3D human–computer interaction
Freehand tracking
User interface

ABSTRACT

The use of three-dimensional (3D) gesture input devices is important and necessary in 3D systems, but such devices face considerable challenges posed by the high dimensionality of dexterous hand motion. The objective of this study is to achieve real-time interaction in object selection and direct manipulation in 3D application systems by capturing and visualizing the interaction intentions and probing the cognitive behavior models of users. An interactive operation procedure is divided into three stages: object selection, manipulation and reset. Trajectory scene interaction (TSI) is proposed for object selection starting from a fixed position called a forward point (FP). The manipulations exerted on the selected object include grasping and translation. After these manipulations, the gesture is reset to the FP. This work offers four novel contributions. First, flexible object selection and atomic operation model (AOM)-based animations are fused to form a uniform, real-time human-computer interaction (HCI) paradigm. Second, a cognitive behavior model is proposed for recognizing and reacting to hand gestures as captured by a monocular camera. Third, an approach to capturing, expressing, and probing a user's interaction intention is presented. Fourth, a 3D real-time gesture input interface is achieved. The use of the proposed HCI interface, which offers fast speed, satisfactory accuracy and a responsive user experience, is demonstrated in virtual assembly, a game of chess, dialing a cell phone number and menu operation.

1. Introduction

Touching, picking up and manipulating objects are the main ways in which humans interact with the physical world. From the moment we are born, we learn to manipulate objects around us using our hands. By adulthood, we are able to unconsciously perform extremely complex manipulations [31]. Two-dimensional (2D) gestures have been extensively used in input devices for 3-dimensional (3D) application systems. However, 2D gestures lack intuitiveness and responsiveness. Thus, the use of 3D gesture input devices is important and necessary in 3D systems. Humans use their hands in most activities of everyday life. Therefore, the development of technical systems that can track the 3D position, orientation and

* Corresponding author.
E-mail addresses: ise_fengzq@ujn.edu.cn, fzqwww@263.net (Z. Feng), yangbo@ujn.edu.cn (B. Yang).

http://dx.doi.org/10.1016/j.ins.2016.06.052
0020-0255/

full articulation of human hands based on markerless visual observations is of fundamental importance for supporting a diverse range of applications. The direct manipulation offered by 3D hand gestures is one of the primary methods of interaction in the virtual world. In this context, direct manipulation in 3D results in successful user–interface interactions with a qualitative feeling of engagement, that is, the feeling that the user is directly manipulating the object [11]. Direct 3D manipulation offers the following advantages: users can immediately observe whether their hand-gesture-based manipulations are furthering their goals, novice users can quickly learn basic functionalities from experienced users, and the interaction style required for the interactive semantic expressions of users is natural and clear. Bare-hand interactions enable immersive interactive experiences within a 3D environment. Although researchers have expended considerable effort in obtaining accurate 3D gesture data for each frame of a captured video using various tracking techniques, no effective direct manipulation technique has yet been developed because of a number of difficulties and challenges, such as the high dimensionality of dexterous hand motion, the ambiguity of identifying different parts of the hand because of their color uniformity, and the interruption of observations when the fingers occlude one another or the palm. Most importantly, an interactive process is inherently dependent on its users, but research on the extraction of users' interaction intentions is still lacking.

Motivated by the need to develop a 3D real-time gesture input interface, this work focuses on capturing users' interaction intentions and extracting users' cognitive behavioral models to achieve real-time interaction for object selection and direct manipulation in 3D application systems. Our fundamental idea is that as soon as the user's interaction intentions have been identified, time-consuming 3D gesture tracking can be replaced with the corresponding real-time gesture animations. The advantage of our method is that it enables the development of a gesture-based, task-oriented, flexible, natural 3D human-computer interaction (HCI) system with fast, accurate and robust gesture input recognition using a simple monocular camera without the assistance of gloves or markers.

2. Related work

To discuss gesture recognition, we must first define the concepts of gesture and pose. A pose is the specific combination of the position, orientation and flexion of the hand observed at a particular instance of time, whereas the term 'gesture' refers to dynamic gestures, which are sequences of poses connected by motions over a short time span [18].

The development of a freehand tracking method without the need for markers is hindered by the high dimensionality of the hand structure. Researchers have directed considerable efforts toward overcoming this obstacle in 3D hand tracking. A wide variety of hand-tracking methods are available. Monte Carlo methods estimate the required probability densities based on a set of discrete weighted samples. One of the most prevalent implementations of Monte Carlo methods is particle filtering (PF), which has attracted increasing attention in articulated object tracking as a means of addressing non-Gaussian and non-linear problems, as the high dimensionality of the hand structure leads to an exponential increase in the number of particles [10]. Raskin et al. [32] combined an annealed particle filter with a Gaussian process dynamical model to obtain a Gaussian process annealed particle filter (GPAPF) to reduce the dimensionality of the state vector. They used GPAPF to generate trajectories in latent space that could be classified based on the Fréchet distance or distance transform. However, this method is unreliable in 2D latent space because it may result in gaps between sequential poses. Deutscher et al. [8] used a continuation principle based on annealing to incorporate the influence of narrow peaks into the fitness function. To increase the speed of convergence, Deutscher et al. [9] first developed a hierarchical search strategy that automatically partitions the search space without any explicit representation of partitions by using gradient descent with adaptive and parameter-specific step sizes. Thereafter, Deutscher et al. [9] introduced a crossover operator to improve the ability of the tracker to search different partitions in parallel. Lin et al. [24] proposed a divide-and-conquer approach for estimating both global and local hand motions. For global motions, they approximated the palm as a rigid planar object, and they tracked the local articulation of the fingers using a sequential Monte Carlo technique. However, their method requires manual initialization for gesture tracking. Bray et al. [2,3] proposed smart particle filtering (SPF) to improve the tracking speed by using the stochastic meta-descent (SMD) method to solve high-dimensional problems. In their approach, the body pose is estimated based on depth data, and the iterative closest point (ICP) algorithm is used to find the necessary correspondences. Using SMD, a small number of points on the model surface are randomly selected in each iteration, thereby reducing the computational cost and allowing the algorithm to escape local minima. Hand model constraints are also carefully considered by applying an additional step at each iteration. Morshidi et al. [27] presented a gravity-optimized particle filter (GOPF) to attract nearby particles and replicate new particles, thereby improving the sampling efficiency. Singh et al. [35] proposed a novel gesture recognition method based on a discrete wavelet transform (DWT) and mel-frequency cepstral coefficients (MFCCs), in which the dimensionality of the data is reduced by the DWT. Enrique Yeguas-Bolivar et al. [12] used an evolutionary algorithm to achieve markerless human motion capture, and these authors also evaluated the performance of three of the most competitive algorithms in the covariance matrix adaptation evolution strategy for solving continuous optimization problems. Kulkarni et al. [20] proposed a novel static hand gesture recognition method using depth information and used random projection (RP) and kernel principal component analysis (KPCA) for dimensionality reduction. Subsequently, classification was performed in the lower-dimensional space. To address the high dimensionality of articulated hand models, Feng et al. [16] proposed a practical Task-Phase-Trajectory-Mentality (TPTM) model for the Selection-Move-Release (SMR) system.

Another type of method that reduces dimensionality is based on the correlation analysis or statistical analysis of 3D hand model variables. Wu et al. [41] proposed a novel dimension-reduction method to learn the low-dimensional intrinsic motion manifold of an articulated object. Their proposed framework is based on the generation of a mapping from the image

feature space to the embedding space and a mapping from the embedding space to the configuration space. Agarwal and Triggs [1] took advantage of the local correlations between motion parameters by partitioning the space into contiguous regions and obtaining the individual local dynamical behavior within reduced-dimensional manifolds. Chen [5] described a new approach that combines statistical and syntactic analysis. They divided the problem into low and high levels, of which the purpose of the former is to detect hand gestures with Haar-like features and that of the latter is to analyze hand motions. Vaswani [38] stated that most of the state change at any given time occurs in a small number of dimensions, whereas the change in the rest of the state space is small. To recover 3D hand structures in real time, Oikonomidis et al. [28] used particle swarm optimization (PSO) to minimize the objective functions and developed a graphics processing unit (GPU)-based implementation for improved running efficiency. Thereafter, Oikonomidis et al. [29] used a Kinect sensor and a PSO variant to observe the 3D positions, orientations and articulations of the entire human hand.

In addition to recovering the 3D hand structure through frame-by-frame gesture tracking, gesture recognition is another important issue in gesture-based HCI. Many gesture recognition algorithms have been proposed in recent years [30].

Data gloves can be used to capture precise 3D inputs for real-time control [37]. However, these systems are usually expensive and unwieldy. As an alternative method, Dorner [7] used a glove with color-coded rings to recognize the sequences of the sign language alphabet (consisting of 6–10 frames). Wang et al. [39] used a single camera to track a hand wearing an ordinary cloth glove. The pose database was generated by sampling recordings of natural hand poses and was indexed by rasterizing images of the glove in these poses. A pose sample was compared with each pose in the database using *k*-nearest neighbor (KNN) lookup based on a robust Hausdorff-like distance metric. Although the Hausdorff-like distance metric improved the robustness to alignment problems or minor distortions of the pose images, it was still necessary for the user to wear a colored glove. To obtain a gesture descriptor that is invariant to rotation, translation and scaling, Yang et al. [42] proposed a density distribution feature (DDF)-based pose recognition algorithm. In their method, the DDF vector is used for pose feature extraction. In the DDF-based approach, a pose image is first divided into concentric sub-regions (SRs), and the relative object pixel density within each SR is computed. However, this method is not effective in separating one pose from other similar poses (e.g., distinguishing a gun-type pose from a scissors-type pose). Recently, Takahashi et al. (2014) used 4-dimensional (4D) trajectory features, including horizontal, vertical, time and depth information, to achieve gesture recognition. By contrast, our application system uses only a typical universal serial bus (USB) camera. With the intent of resolving the problems faced by the methods discussed above, we propose a new dynamic gesture recognition algorithm.

Schlattmann et al. [34] proposed an innovative technique for symmetric object manipulations that enables efficient and intuitive two-handed grasping, moving and releasing of objects without postural changes. Romero et al. [33] considered the object shape in 3D hand reconstruction. Wang et al. [40] proposed a bi-manual hand-tracking system that provides physically motivated control in six degrees of freedom (DOFs) for 3D assembly and created precise and memorable gestures using metaphors that correspond closely to physical actions. Two depth cameras and a database of hand poses that match a downscaled version of the depth maps are used in this bi-manual hand-tracking system. Despite the advancements that have been made, it remains challenging to provide a convenient, natural and effective HCI interface based on gesture input. Direct and natural interaction in 3D scenes is one of the dominant directions for future studies. The existing interaction methods do not provide an effective means of capturing user intentions. Thus, we present a simple method of detecting gestures using a single-gesture channel without the use of any other channel, such as the face or body.

Stefanov et al. [36] combined PF with annealing and variable-length Markov models (VLMMs) to develop an innovative hand-tracking algorithm for HCI. This hand-tracking algorithm is a mathematical framework that can be used to model complex non-linear activities and dependencies at variable temporal scales in a simple and efficient manner. An interaction is primarily composed of structured behavior. Thus, a high-level structure governs hand movements and the temporal ordering of different gestures. Li et al. [23] created a library of communication gestures and their variants to learn gesture models using the Gaussian Process Dynamic Models (GPDM) statistical approach. Nevertheless, the question of how a cognitive behavior model can be used to construct interactive applications remains an open and interesting issue.

J. Yu et al. (2013) presented a novel approach for recovering 3D human poses from silhouettes with locality-sensitive sparse retrieval by imposing locality constraints. In this way, the semantic gap problem was effectively mitigated. Recently, to address the problem of inferring 3D human poses from monocular video frames, the same authors proposed a supervised dimension-reduction algorithm that incorporates the information from the pose space for finding the optimal projection [43]. Furthermore, they fused machine learning algorithms and signal processing techniques to achieve human pose recovery and behavior analysis (J. [44]). Depth imagery was used in a hand-gesture-based recognition system, and compressive sensing was adopted for dimensionality reduction [25]. Maqueda et al. [26] developed a robust vision-based hand-gesture recognition system in which a video descriptor called Volumetric Spatiograms of Local Binary Patterns (VS-LBP) is computed and delivered to a bank of SVM classifiers for gesture recognition. Laskar et al. [21] proposed an approach for detecting hand gestures based on stereo vision; they used the disparity-map-based movement of the centroid and the changes in its intensity as features for recognizing gestures using a conditional random field (CRF) classifier. Zhang et al. [45] proposed a descriptor named the Histogram of 3D Facets (H3DF), in which the 3D shape information from depth maps is explicitly encoded; this descriptor can effectively represent both the 3D shapes and structures of various depth maps. Kılıboz et al. [19] proposed a real-time trajectory-based dynamic hand gesture recognition method for HCI, in which a six-DOF position tracker is used to collect trajectory data and represent gestures. Zhou et al. [46] modeled the fingers as cylindrical objects because of their parallel edge features and proposed a high-level hand feature extraction method for real-time gesture recognition.

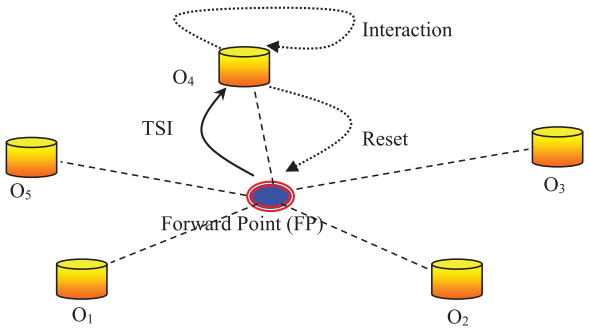


Fig. 1. Proposed FHCI interface paradigm. The dotted lines with arrows indicate that the corresponding operations are performed using AOM-based animations.

In summary, all existing 3D hand-gesture-tracking systems strive to accurately acquire the 3D hand structure in each frame, whereas the system proposed here does not require the hand to be tracked over time. This work offers several novel contributions. First, flexible object selection and atomic operation model (AOM)-based animations are fused to form a uniform, real-time HCI paradigm. Second, a cognitive behavior model is proposed for recognizing and reacting to hand gestures as captured by a monocular camera. Third, an approach to capturing, expressing, and probing a user's interaction intention is presented, and finally, a 3D real-time gesture input interface is achieved.

3. Overview

This study aims to provide a method for extracting and visualizing the interaction intentions of users in an intuitive 3D gesture-based HCI system that offers fast speed, satisfactory accuracy and an exciting user experience. AOMs and their correlations are used to develop the behavioral model. The behavioral model can interpret hand motions even in the case of poor tracking accuracy (TA). Actions, rather than hand direction trajectories, are smoothly rendered when recognized. An AOM database is used to recognize and map the intended gestures for specific tasks. The system is capable of retrieving the full 3D configuration of the observed hand (i.e., the 3D hand model in this study) based on the AOM database. The proposed user interface provides online real-time mapping from real space to virtual space. Users can flexibly control the interaction speed and freely select objects. This type of HCI is called flexible HCI (FHCI) and is composed of three stages. The point with the minimum sum of the Euclidean distances between that point and each object in the scene is called the forward point (***FP***) and is one of the endpoints of the gesture trajectory. Each point ***P*** has a parameter $\boldsymbol{D_P}$ that corresponds to the sum of the distances between that point and the other points (objects) in the scene, and the point ***P*** with the minimum $\boldsymbol{D_P}$ is the ***FP***. The first stage of FHCI involves the arbitrary selection of an object in the scene starting from the ***FP*** using the trajectory scene interaction (TSI) algorithm (Section 4.1). The second stage is the execution of the interaction between the gesture and the object, and the third stage is the resetting of the gesture to the FP (Fig. 1). With the exception of the TSI stage, all basic manipulations and translations are represented by an animation technique with the following features. First, the basic operation gestures, or atomic operation gestures, of the user are recognized. Second, the corresponding AOMs of these gestures are retrieved from the AOM database, which stores the image features and corresponding 3D hand model for each atomic operation. Third, the hand motions of the user are mapped to the AOMs by an animation technique for hand gestures. Finally, through synchronization with the live gestures of the user, the mapped gestures are visualized in an innovative manner, and the mapped AOMs are linked with the user's individual operations.

The last two stages are performed using AOM-based animations. Fig. 1 shows that the FP is a 3D point that satisfies the condition

$$\underset{\mathrm{FP}}{Minimize}\left(\sum ||\mathrm{FP}-\mathrm{O}_i||\right), \tag{1}$$

where $\boldsymbol{O_i}$ refers to the 3D position of an interactive object in the scene. ***FP*** can be dynamically updated when the objects change. The problem expressed by (1) could be solved using Weiszfeld's algorithm [4], but its time complexity is $O(n^2)$ (where n is the number of objects in the scene). To reduce the time complexity, we replace ***FP*** with the center of gravity of all points $\boldsymbol{O_i}$.

The proposed FHCI algorithm is described as follows:

Generally, the FHCI algorithm differs from other state-of-the-art algorithms in the following respects: (a) the existing systems focus on tracking with fast speed and high accuracy, whereas the proposed system focuses on interactive tasks and the expression of user cognitive models to enhance the user experience. (b) the existing systems acquire 3D hand structures frame by frame, whereas the proposed system extracts the user's behavioral model. (c) the existing systems track continuous hand gesture streams, whereas the proposed system tracks basic atomic operations (entire gesture streams are divided into several basic atomic operations). (d) the existing systems strive to recover the ground truth data of the 3D hand structures for each frame, whereas the proposed system retrieves 3D hand models from a database and visualizes them using animations. Specifically, to track a gesture, the first step is to recognize the atomic operations executed by the

Algorithm 1 FHCI algorithm.

While NOT(all tasks are performed)
{
(1) Object selection
The user executes a gesture toward an object to select that object via TSI.
(2) Manipulation
(2.1) If the AOM is "Grasp an object", a pseudo collision is executed.
(2.2) Object manipulation.
(2.3) The atomic operation executed by the user is recognized.
(2.4) The AOM is retrieved.
(2.5) The AOM is visualized with an animation.
(2.6) Interaction is performed.
(3) Gesture reset
The gesture is reset to the FP by means of an animation.
}

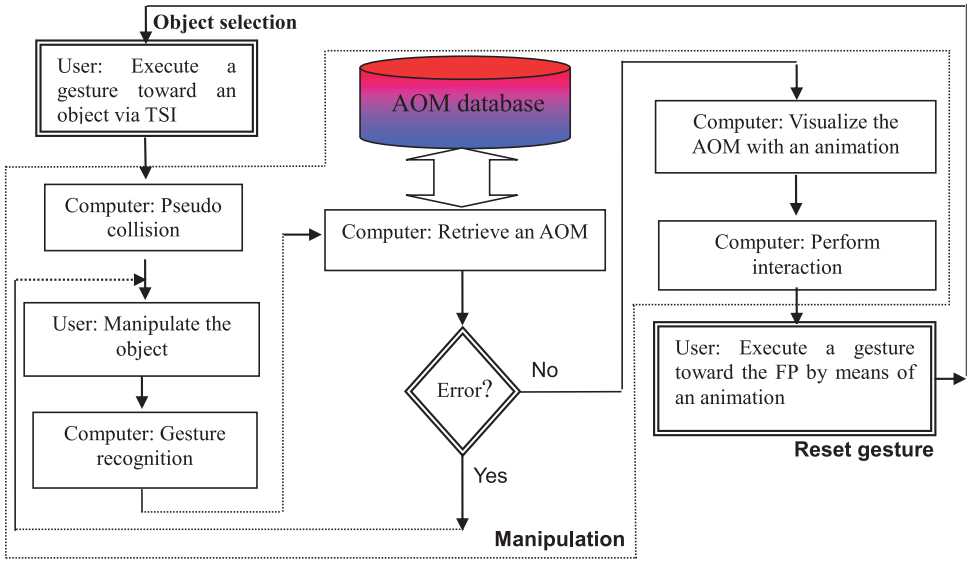


Fig. 2. Implementation details of Algorithm 1.

user and to acquire the corresponding 3D hand models from the AOM database. The second step is to estimate the user's speed in performing the atomic operations, and the third step is to drive the 3D hand models at the appropriate speed in accordance with the user's speed. Finally, the last step is to detect collisions with other objects in the scene and perform the interactive task. The main original contributions of this work are as follows: (1) a flexible object selection method is fused with an AOM-based animation approach to form a uniform HCI paradigm. (2) a cognitive behavioral model is proposed for recognizing and reacting to hand gestures as captured by a monocular camera. (3) an approach to capturing, expressing and probing a user's interaction intention is developed.

Further details regarding the implementation of Algorithm 1 are shown in Fig. 2. An in-depth discussion of Algorithm 1 is provided in the following sections.

4. TSI: object selection

We propose a new method for object selection. When a user's hand moves toward an object in a 3D scene, the motion trajectory is predicted and visualized in real time, thus allowing the user to dynamically adjust his or her selection intention. We first compute the intersection point between the 3D curve of the predicted motion trajectory and the scene. Thereafter, we determine a circular region centered at the intersection point, the radius of which changes dynamically depending on the distance d between the 3D hand model and the intersection point. The trajectory $\boldsymbol{C_k}$ of the center of the 3D hand model is predicted at time k. The intersection $\boldsymbol{G_k}$ between the 3D curve $\boldsymbol{C_k}$ and the scene is computed. When only one object is located within the projected circular region, that object is selected (Fig. 3).

A smaller d corresponds to a smaller radius r of the circle. The circular region becomes smaller as the 3D hand model approaches the target object, thus causing fewer objects to be located in that region. The relation between d and r is expressed as follows:

$$r = \exp(d). \tag{2}$$

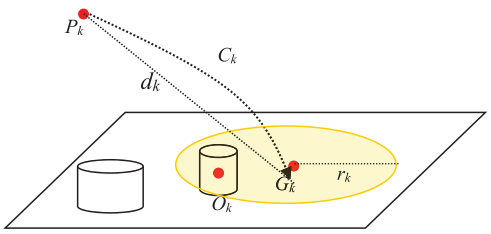


Fig. 3. Concepts related to dominant selection. The trajectory of the 3D hand model is dynamically updated. Objects within the circular region are candidate objects for selection. When only one object is located in the region, the object to be selected is determined.

Algorithm 2 TSI.

1. The dynamic model of a motion gesture is described as follows:
$$\boldsymbol{X}_k = \mathbf{f}(k-1, \boldsymbol{X}_{k-1}) + \eta_{k-1}, \tag{3}$$
$$\boldsymbol{Z}_{\boldsymbol{k}} = \boldsymbol{h}(k, \boldsymbol{X}_{\boldsymbol{k}}) + V_k \tag{4}$$
where $\boldsymbol{f}$ represents the dynamic model of $\boldsymbol{X}_k$, $\boldsymbol{h}$ represents the observation model of $\boldsymbol{X}_k$, and $\boldsymbol{\eta}$ and $\boldsymbol{V}$ are random white-noise vectors.
2. The coarse location phase (CLP) is used to extract pose features from the current image frame [14].
3. $\boldsymbol{X}_{\boldsymbol{k}}$ is estimated by PF [15], expressed as follows:
$$\hat{\boldsymbol{X}}_{\boldsymbol{k}} = \boldsymbol{f}(k-1, \boldsymbol{X}_{k-1}), \tag{5}$$
$$\boldsymbol{X}_k^{(i)} = \hat{\mathbf{X}}_k + \boldsymbol{\eta}_k, \tag{6}$$
$$\boldsymbol{X}_{\boldsymbol{k}} = \sum_{i=1}^{N_k} \omega_k^{(i)} \boldsymbol{X}_{\boldsymbol{k}}^{(\boldsymbol{i})}, \tag{7}$$
$$A_k^i = e^{-\lambda H_m}, \tag{8}$$
$$\omega_k^{(i)} = \frac{A_k^i}{\sum_{s=1}^{N_k} A_k^s}, \tag{9}$$
where $\hat{\boldsymbol{X}}_{\boldsymbol{k}}$is the predicted value of $\boldsymbol{X}_{\boldsymbol{k}}$ at time k; $\mathrm{X}_k^{(i)}$ is the ith generated particle derived from $\hat{\boldsymbol{X}}_{\boldsymbol{k}}$; $\boldsymbol{\eta}_k$ is a random white-noise vector; N_k is the number of particles; $\omega_k^{(i)}$is the weight value of the ith particle, which reflects the goodness of fit between the particle and the observed image frame; and λ is an empirical constant that is set to 0.01. H refers to the Hausdorff distance [17] between the observed posture values $\boldsymbol{Z}_{\boldsymbol{k}}$ and the projections of the 3D hand model into the image frame.
4. $\boldsymbol{X}_{\boldsymbol{k}}$ is rendered and displayed.
5. The intersection point $\boldsymbol{G}_k$ between $\boldsymbol{C}_k$ and the scene as well as the radius r are computed.
6. A circular region of radius r_k and centered at $\boldsymbol{G}_k$ is projected onto the scene.
7. If the user's movement shows an alternating pose, the object centered at $\boldsymbol{O}_i$ that satisfies the condition
$$\underset{\mathrm{O}_i \in \mathrm{Circle}}{Min}_{\mathrm{Region}} \; ||\mathrm{O}_i - \mathrm{G}_k|| \tag{10}$$
is selected; otherwise, go to Step 1.

Suppose that $\boldsymbol{X}_{\boldsymbol{k}}$ is a 3D point on $\boldsymbol{C}_{\boldsymbol{k}}$ and that $\boldsymbol{Z}_{\boldsymbol{k}}$ is the observed state of the 3D hand model at time k. Then, the TSI algorithm can be stated as follows:

In Formula (4), $\boldsymbol{h}$ represents the observation model of $\boldsymbol{X}$, which is described using the camera projection equation as follows:

$$\begin{bmatrix} u & v & 1 \end{bmatrix}^T = \mathrm{P}\begin{bmatrix} \mathrm{x} & \mathrm{y} & \mathrm{z} & 1 \end{bmatrix}^T, \tag{11}$$

where $\boldsymbol{Z} = (u, v)$, $\boldsymbol{X} = (x, y, z)$, and $\boldsymbol{P}$ is the camera projection matrix.

Moreover, based on the extracted hand-shaped polygon, the CLP is used to extract the pose features, such as the fingertips, finger roots and hand image contours.

Finally, it should be noted that although only one camera is used to track the hand motion trajectory, the 3D gesture position can be accurately predicted in accordance with the operating principle of the PF tracker. Because the particles are sampled in 3D space, tracking the 3D trajectory curve is possible. Moreover, when a particle is more consistent with the observed image frame, that particle is assigned a greater weight and makes a larger contribution to the estimation $\boldsymbol{X}_k$.

5. AOM-animation-based manipulation

Once the user has fulfilled his or her object selection intention, he or she may execute a manipulation intention. In this section, we propose a new method for predicting a user's behavioral model and expressing a cognitive model of the user to enhance the user experience.

5.1. Behavioral model database

Definition: In a hand-gesture-based HCI system, any operation can be divided into sub-operations until those sub-operations cannot be divided any further. An operation that cannot be further divided is called an atomic operation. An

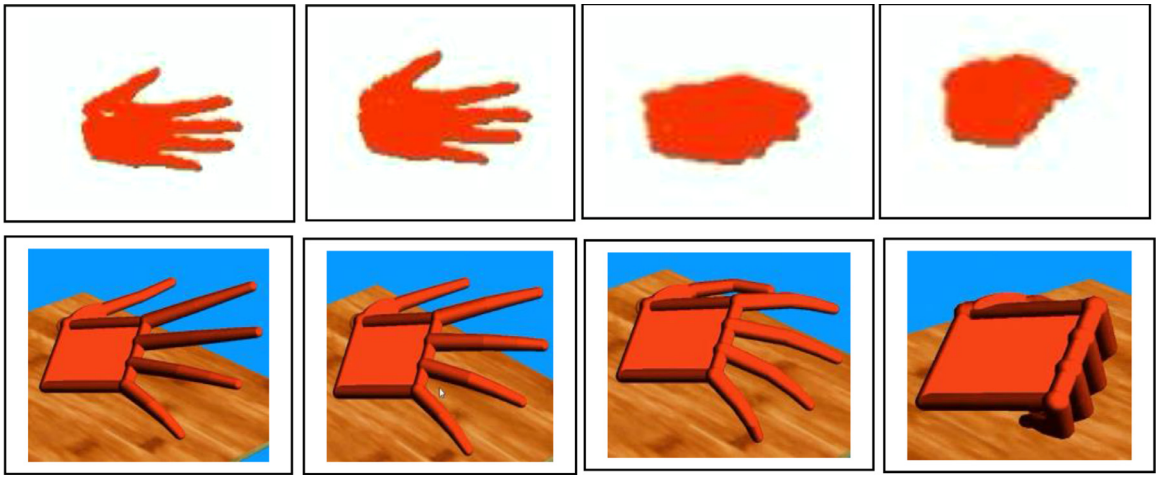


Fig. 4. Atomic operation of grasping with five fingers and the corresponding AOM. The first and second four-image sequences show 2D images of the atomic operation and the corresponding frames of the 3D hand model executing the AOM, respectively.

atomic operation is described by a sequence of features of 2D hand gesture images and their corresponding animated 3D hand models. Such a sequence of animated 3D hand models is called an AOM, which is a hand gesture of a 3D avatar that is constructed frame by frame with the aid of sensors, such as a data glove and a position tracker, by inputting the sensor data into the 3D hand model. In a task-oriented application, a dynamic gesture corresponds to a series of AOMs, and the AOMs of all possible atomic operations form a database called the AOM database or the behavioral model database. Each data entry in the AOM database is composed of the following items: an identification number, a feature sequence of 2D hand gesture images (which are also called sample images), an AOM and an AOM frame length. An AOM frame is expressed as follows:

$$\boldsymbol{X_k} = \left(x_{1,k}, x_{2,k}, \ldots, x_{26,k}\right) \tag{12}$$

Thus, any AOM that is composed of a sequence of 3D hand models can be appended to the AOM database. When the user inputs an atomic operation, the computer captures a sequence of 2D hand gesture images of that atomic operation and retrieves the corresponding AOM from the AOM database based on the image sequence. The 3D hand model sequences relevant to a particular application can be specified beforehand. The atomic operation of grasping with five fingers and the corresponding AOM are shown in Fig. 4.

5.2. Recognition of atomic operations

The prediction of a user's behavioral model involves the retrieval of AOMs from the AOM database. We have observed that the differences between the first image frames and between the last image frames of two different atomic operations are always sufficiently large to distinguish atomic operations from each another. As such, we use the sum of the differences (or errors) between the first-frame poses and the last-frame poses to distinguish between two gestures. Thus, we focus on the recognition of static pose images. To improve the robustness of the pose rotation and the accuracy of recognition, we first divide each hand pose into SRs, namely, the regions between two adjacent concentric circles. The center of these concentric circles is the center of gravity of the hand silhouette. Subsequently, we compute the Hausdorff-like distances between the corresponding SRs in the query image from the input and the sample image from the AOM database. To allow for rotation in the user's pose, for each pose, five image frames acquired from different viewpoints are included in the AOM database.

Algorithm 3 describes in detail the method used to retrieve an AOM from the AOM database.

Fig. 5 shows an example of the comparison of two hand pose images.

Compared with the DDF [42], the Hausdorff-like distance used in the AOR algorithm offers improved recognition. In fact, the DDF only compares the number of skin pixels between two corresponding SRs. Thus, the ability of the DDF to distinguish between similar poses is weaker than that of the AOR algorithm. Compared with KNN lookup, the circumcircle-based SRs computed in the AOR algorithm are more robust to pose rotation, and as a result, the number of nearest-neighbor poses is reduced.

5.3. Animation speed of the AOMs

Suppose that ∇S_{tra} represents the difference between the centers of gravity of the bounding boxes of two adjacent frame images and that ∇S_{gra} and ∇_{rot} represent the difference between their radii in grasping and rotating motion. The instantaneous translation speed, grasping speed and rotation speed of an atomic operation are represented by v_{tra}, v_{gra}

Algorithm 3 Atomic Operation Recognition (AOR).

1. Separate the hand and forearm regions using a hand detection method [22].
2. Resize the pose image to 40×40.
3. Compute the center of gravity $\boldsymbol{Z}$ for the hand silhouette.
4. Compute the maximum distance D_{max} between any pixel in the hand silhouette and the point $\boldsymbol{Z}$.
5. Compute the circumcircle of center $\boldsymbol{Z}$ and radius D_{max}.
6. Subdivide the region bounded by the circumcircle into K concentric circular bands bounded by two concentric circles with radii of $(i-1)*(D_{max}/K)$ and $i*(D_{max}/K)$, $i = 1, 2, \ldots, K$. Thus, K SRs are obtained.
7. Count the pixels in each SR.
8. Compute the Hausdorff-like distance [39] $d(\boldsymbol{SR_{q,i}}, \boldsymbol{SR_{q,j}})$ between $\boldsymbol{SR_{q,i}}$ and $\boldsymbol{SR_{q,j}}$ as follows:

$$\tilde{d}\left(\boldsymbol{SR_{q,i}}, \boldsymbol{SR_{q,j}}\right) = \sqrt{\frac{1}{\left\|\boldsymbol{SR_{q,i}}\right\|} \sum_{(x,y)\in \boldsymbol{SR_{q,i}}} \min_{(u,v)\in \boldsymbol{SR_{q,j}}} \left((u-x)^2 + (v-y)^2\right)}, \tag{13}$$

$$d\left(\boldsymbol{SR_{q,i}}, \boldsymbol{SR_{q,j}}\right) = \tilde{d}\left(\boldsymbol{SR_{q,i}}, \boldsymbol{SR_{q,j}}\right) + \tilde{d}\left(\boldsymbol{SR_{q,j}}, \boldsymbol{SR_{q,i}}\right), \tag{14}$$

   where $q = 0, 1, \ldots, K$ and $\boldsymbol{SR_{q,i}}$ is the set of skin pixels in the qth SR of the sample image.
9. Compute the error E between two static pose images as follows:

$$\boldsymbol{E_{pose}} = \sum_{q=0}^{K} d\left(\boldsymbol{SR_{q,i}}, \boldsymbol{SR_{q,j}}\right). \tag{15}$$

10. Compare the query gesture with a sample gesture from the AOM database by obtaining the error $E_{gesture}$ between the dynamic gestures as follows:

$$E_{gesture} = E_{pose}^{first} + E_{pose}^{last}, \tag{16}$$

    where E_{pose}^{first} and E_{pose}^{last} refer to the errors between the first frames and between the last frames, respectively, of the dynamic gestures.
11. The AOM that corresponds to Min($\mathbf{E}_{\text{gesture}}$) is recognized as the correct one.

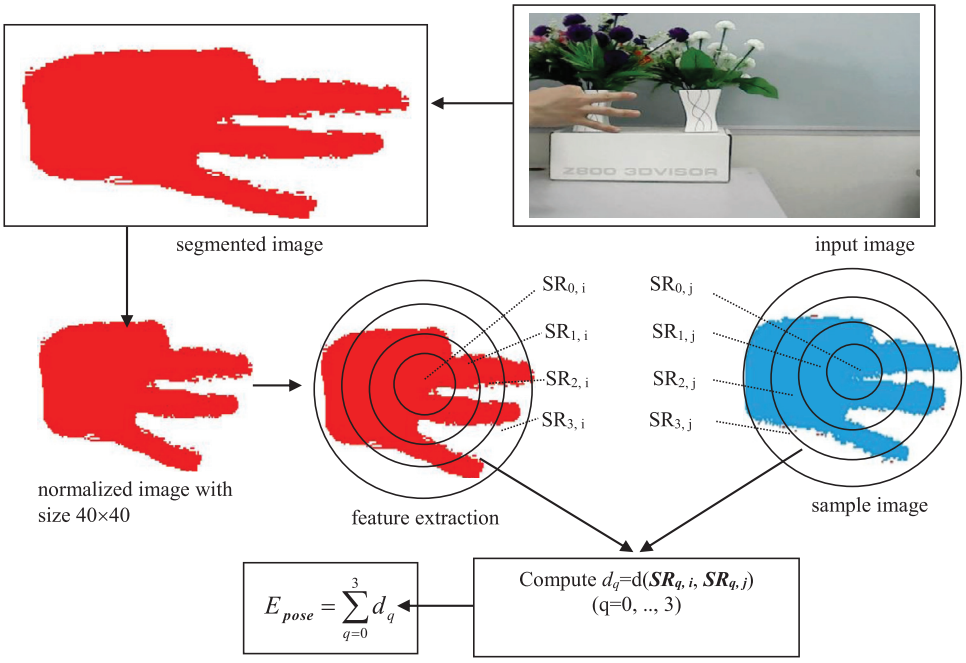


Fig. 5. An example to illustrate the detailed process of the AOR algorithm. The distance between each pair of adjacent concentric circles is the same, and the regions between two adjacent concentric circles are called SRs.

and v_{rot}, respectively. Larger ∇S_{tra}, ∇S_{gra} and ∇_{rot} values correspond to larger values of v_{tra}, v_{gra} and v_{rot}, respectively. The animation speed of an AOM can be approximately estimated as follows:

$$v_{tra} = k_{tra}|\nabla S_{tra}|, \tag{17}$$

$$v_{gra} = k_{gra}|\nabla S_{gra}|, \tag{18}$$

$$v_{rot} = k_{rot}|\nabla_{rot}|, \tag{19}$$

where k_{tra}, k_{gra} and k_{rot} are constants.

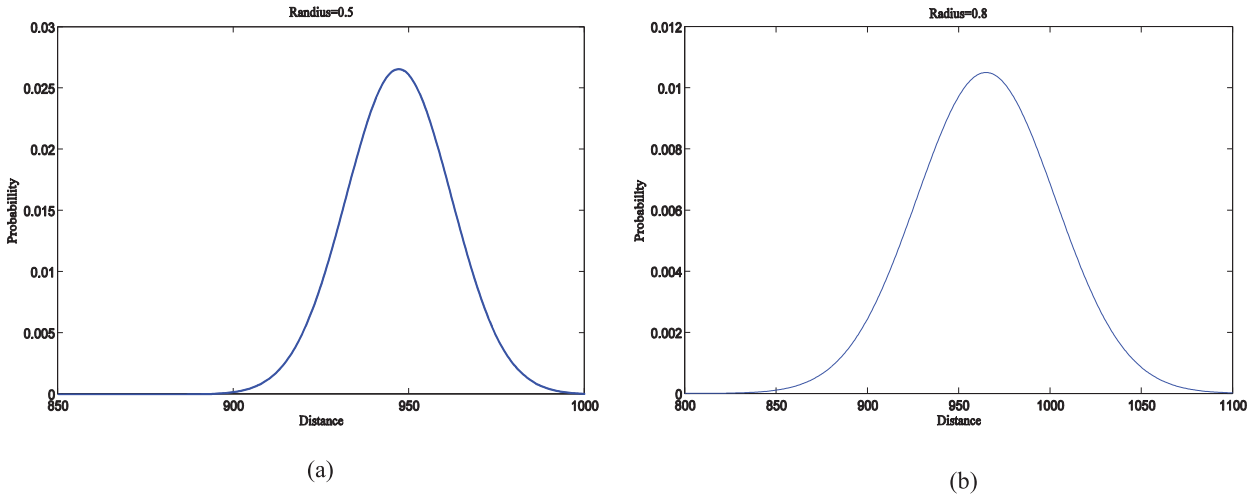


Fig. 6. Distributions of R for different r obtained with the aid of a glove and a position tracker: (a) $r = 0.5$ and (b) $r = 0.8$. The appropriate distance for a pseudo collision follows a Gaussian distribution. The acquired distribution rule is utilized in Algorithm 1, for which the data glove and position tracker are no longer needed.

5.4. Pseudo collision

We focus on the time at which the AOM of the grasping gesture is executed. We observe that a person begins to grasp an object when his or her hand is near that object. However, a person will maintain an appropriate distance between his or her hand and the object to allow for ample space to modify his or her initial gesture to a grasping gesture. For example, a person may change from a fist pose to an open-palm pose. During this transition, the 3D model of the transformational gesture is easy to discern if the distance between the hand and the object is small. To allow for this behavior, the collision detection between the 3D hand model and an object is considered successful when the 3D hand model falls within a certain proximity of the object. We refer to this collision as a pseudo collision because no true collision between the hand model and the object occurs. In pseudo collision detection, the distance between the 3D hand model and the object is denoted by R. Once pseudo collision occurs, the object will be highlighted or visualized with radial waves as if the object were in contact with the hand.

In this study, to determine R for each object, the object was replaced with a 3D sphere of radius r. The physical significance of the 3D sphere was that it represented the bounding sphere of the object. Then, a user was asked to wear a data glove. A tracker was positioned in the user's hand, and the user was asked to naturally grasp the 3D sphere 100 times. The distributions of R obtained for $r = 0.5$ and $r = 0.8$ are shown in Fig. 6. Recall that the proposed FHCI algorithm is based solely on a camera, without the need for gloves or a position tracker. Here, we used the gloves only to extract the user's behavioral model, namely, to explore the rule governing the distribution of R with different r, which is directly utilized in Step 2.1 of Algorithm 1.

The observations presented in Fig. 6 show that the distribution of R follows a Gaussian distribution related to the size r of the object. Further research reveals that the general relation between R and r can be expressed as

$$R = ar^b + c + \xi, \tag{20}$$

where ξ is a random variable, $\xi \sim N(0, 10r)$, and a, b and c are constants that can be estimated with the aid of a position tracker. Once the pseudo collision is successful, the user can easily modify his or her pose, move toward the object and perform a grasping gesture.

The pseudo collision identifies the time at which the AOM of the grasping gesture should begin.

5.5. Error handling

A user may either try again or exit the system when a retrieved AOM is not consistent with the user's intention. The system will exit if the number of failed recognition attempts exceeds a user-defined value (which was defined as 3 in our experiment). An error occurs when the user does not approve the AOM visualized on the screen.

The interaction intentions of operators are complex, and mistakes are inevitable. In fact, even under multi-channel input conditions, it is still very difficult for an operator to accurately convey his or her interaction intentions to a computer, and in the present study, only a single-channel input (gesture) is considered. We nevertheless begin from operator behavioral models to solve the "Approve" problem. Multiple observations have shown that when a gesture input is completed, if the retrieved AOM is consistent with the operator's actual intent, the operator will often maintain the last posture of that gesture

for a certain period. That is, if the current gesture has been completed and the gesture posture remains unchanged for a certain period, this means that the user "Approves" the computer's interpretation. We have also observed that whenever the operator disapproves of the AOM as interpreted by the computer, he or she will subconsciously gesture toward the left side of the chest; this observation serves as the basis for error handling.

6. Experimental results

6.1. Settings

The proposed system works without markers or colored gloves and, for cost efficiency, uses only one regular video camera as the input device. The proposed system was tested on a typical PC with an Intel® Core™ Quad CPU 2.66 GHz processor and 4 GB of random access memory. The 3D freehand model was designed based on a previously developed kinematic model [13]. The orientation of the palm is described using 32 DOFs. The fingers and thumb each have 2 DOFs at the anchoring joint to account for the angle of flexion and spreading movement. All experiments were conducted on a single processor thread.

6.2. Evaluation criteria

Time cost: The time cost was evaluated as the total time elapsed after the user had performed the specified interactive tasks.

TA: For a frame during the object selection stage, the accuracy of the 3D hand model is defined as

$$TA = \exp(-kH), \tag{21}$$

where the constant k is set to 0.01 and H represents the Hausdorff distance [36] between the observed features of the hand image and the projection of the tracked 3D hand model onto the image frame.

Positioning accuracy (PA): The PA is defined as

$$PA = \exp\left(-k\left(\left|P_{expected} - P_{real}\right|\right)\right), \tag{22}$$

where $P_{expected}$ and P_{real} are the user's expected position and the real position, respectively, of the object after manipulation.

User experience: Users evaluated the interface in terms of the cognitive burden it imposed. The extent to which the FHCI process reflects the users' mental models was also investigated.

6.3. Experimental results

Experiment 1: AOR

Based on the number of extended fingers and the motions for changing from the extended to the bent state and from the bent to the extended state, we chose 27 commonly used gestures that involve transitioning between extended and bent finger states and constructed an AOM database from the corresponding $27 \times 2 = 54$ gestures. (Here, the reason we used 27 gestures related to only one type of motion is that we simply wanted to verify the efficiency of our AOR algorithm. In practice, only some of these gestures will be requested, although we provide the operators with more choices.)

As determined from 50 runs performed for each gesture in the AOM database, the average recognition rates of the AOR, DDF and KNN methods are 94%, 35% and 84%, respectively (see Table 1). Because of limited space, we present only the 27 'grasping gestures' in Table 1. The other 27 gestures are the 'releasing gestures' that correspond to the 'grasping gestures'. The experimental results show that the recognition of a 'grasping gesture' is the same as that of its corresponding 'releasing gesture' and that the AOR algorithm is robust to rotated poses and scaled poses.

Similar poses, such as the pose with only the forefinger extended and the pose with only the middle finger extended, cannot be easily distinguished using the DDF method. This method's poor robustness to rotated poses and its lack of comparison of local features lead to low recognition for some of the poses in our database. With AOR, the recognition rates for 5 of the 54 gestures are $< 90\%$ because these gestures are too similar to be perfectly recognized. There are several reasons for these results. The first is that uncommon gestures may be difficult to distinguish from each other. For example, gesture #5 is difficult for many people to perform and therefore is often confused with gesture #11. The second reason is that in the gesture image processing, the gesture segmentation is still affected to some extent by effects related to the skin, the background and high brightness.

The recognition time costs for each pose using the AOR, DDF and KNN methods are 850, 1 and 7500 ms, respectively. From these results, it is clear that the computation of the Hausdorff-like distance is rather expensive.

We observed in our experiments that the length of a gesture of an atomic operation, when performed at normal speed, is approximately 15 frames. By comparison, a gesture can be recognized by the AOR algorithm based on only the first five frames, which enables the visualization of each atomic operation by means of the 3D hand model in real time.

Experiment 2: Diesel Engine Assembly

We asked users to assemble 3D machines, such as diesel engines, internal combustion engines, computers and television sets, in a 3D computer-aided teaching system (3D-CTS) implemented with a monocular camera and a personal computer.

Table 1
Recognition comparisons of the AOR, DDF and KNN methods. Only half of gestures, the 27 'grasping gestures', are shown because of manuscript length limitations. The recognition rate for each 'releasing gesture' is the same as that for its corresponding 'grasping gesture'.

Index	Gesture description	Number of runs	Successful instances			Recognition rate		
			AOR	DDF	KNN	AOR	DDF	KNN
1		50	48	21	45	96%	42%	90%
2		50	50	12	48	100%	24%	96%
3		50	47	26	46	94%	52%	92%
4		50	46	26	44	92%	52%	88%
5		50	42	15	40	84%	30%	80%
6		50	45	5	23	90%	10%	46%
7		50	46	12	28	92%	24%	56%
8		50	43	16	38	86%	32%	76%
9		50	49	11	26	98%	22%	52%
10		50	43	31	46	86%	62%	92%
11		50	44	8	48	88%	16%	96%
12		50	49	6	30	98%	12%	60%
13		50	48	27	46	96%	54%	92%
14		50	49	6	46	98%	12%	92%
15		50	46	9	46	92%	18%	92%
16		50	44	13	40	88%	26%	80%
17		50	48	4	35	96%	8%	70%
18		50	50	3	36	100%	6%	72%
19		50	49	24	48	98%	48%	96%
20		50	46	31	46	92%	62%	92%
21		50	50	2	47	100%	4%	94%
22		50	46	3	46	92%	6%	92%
23		50	50	28	50	100%	56%	100%
24		50	50	46	50	100%	92%	100%

(*continued on next page*)

Table 1 (*continued*)

Index	Gesture description	Number of runs	Successful instances			Recognition rate		
			AOR	DDF	KNN	AOR	DDF	KNN
25		50	49	25	44	98%	50%	88%
26		50	47	24	46	94%	48%	92%
27		50	46	38	46	92%	76%	92%
	Average recognition					94%	35%	84%

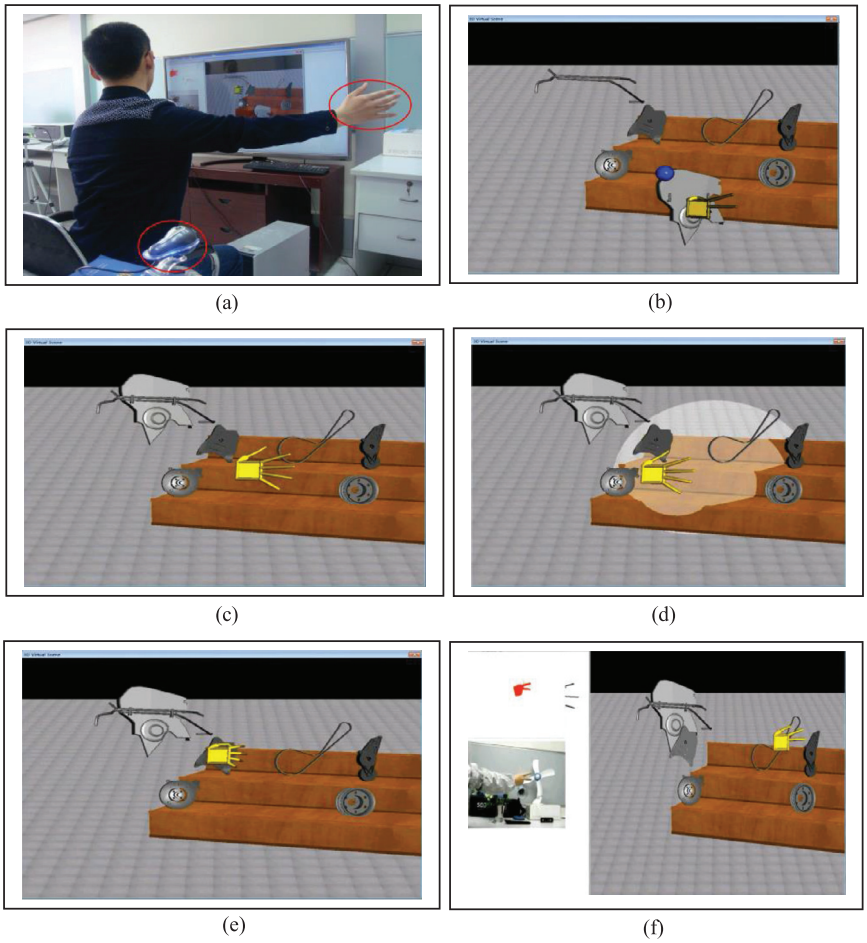


Fig. 7. Screenshots of the proposed user interface on a 3D large-scale display using a standard USB camera and freehand gesture recognition without the aid of markers or colored gloves. The screenshots were acquired during the diesel engine assembly task. (a) The user is remotely interacting with the computer by means of a large display. The ellipses indicate the locations of the camera and the freely gesturing hand. As seen from the image, the user appears to be reaching too far to his right, in a position that would be rather uncomfortable to maintain or work in, but this positioning was used only for the pictured tests. In the real evaluation, the users could work in a comfortable manner. (b) The 3D sphere indicates the location of the FP. (c) Each instance of object selection using the TSI algorithm begins from the FP. (d) The objects within the circular region are the candidate objects for selection. The circular region changes with the motion of the user's gesture. (e) The user begins to grasp an object when a pseudo collision is successful. (f) AOM recognition.

The assembly tasks required issuing natural gesture inputs in the correct order within a specified time. In this context, real-time tracking and interaction with the gesture input are among the main obstacles facing the application of the proposed system. The existing methods of achieving this purpose are limited to offline gesture inputs [6,10] or are subject to other problems. The slow speed of the 3D-CTS hinders the ability of users to complete assembly tasks in a specified time. Most gesture-based 3D tracking systems use one or more depth cameras, whereas the 3D-CTS operates with a typical monocular

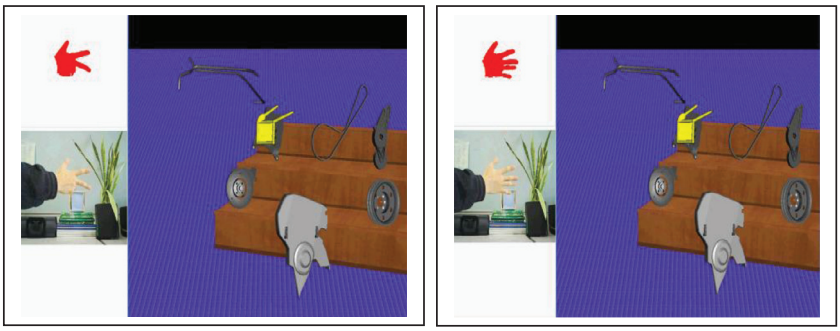


Fig. 8. Error handling. (a) The user wants to perform the 'nipping gesture with the thumb, index and middle fingers'. However, this gesture is incorrectly recognized as a 'nipping gesture with the thumb and index fingers'. (b) Therefore, the user repeats the gesture.

camera. Thus, state-of-the-art techniques do not enhance user performance in the 3D-CTS. Direct manipulation in 3D space allows the 3D-CTS system to provide a user experience that is much more similar to the real scenario being simulated than that offered by 2D operations with a mouse and keyboard. Moreover, the operational speed of such 2D operations is also slow. Our investigations yielded the following results: the use of a mouse as an input device during the assembly process leads to poor user experience, the use of a data glove as an input device significantly increases the cost of the system, and the need for colored gloves or markers on the user's hands creates inconvenience in interactions [39].

The proposed FHCI system was implemented in a 3D-CTS with a 3D remote interface for large displays. Most 3D virtual assembly systems improve their processing speeds by using multiple GPU threads depending on the depth of the image data [40]. By contrast, our virtual assembly system can run on a single thread with common red, green and blue video. Several 3D hand-based interactive systems, such as those proposed by Oikonomidis et al. [28] and Wang et al. [40], have demonstrated high real-time accuracies and have been applied for the dexterous assembly of 3D parts in computer-aided design applications. However, few attempts have been made to extract and express the users' interaction intentions in these applications.

Using the proposed system, users were asked to assemble a diesel engine remotely in a 3D scene using the FHCI-based HCI interface and a large display (Fig. 7). Users were asked to assemble seven parts of a diesel engine. The users could arbitrarily select a part and place it into the diesel engine at their own pace. We used improved PF to track the 3D hand trajectories. PF is effective in coping with non-Gaussian and non-linear problems. The resolution of the camera is low, but the results of the hand image segmentation are satisfactory under stable light conditions. With accurate observations of the positions of the 3D hand models, our TSI algorithm can accurately track and extract 3D hand trajectories from a complex background. The experiment shows that the proposed HCI paradigm works well with a monocular USB camera in the presence of complex backgrounds under typical interior lighting conditions and satisfies the needs of our 3D-CTS application.

The proposed HCI paradigm can also process certain interaction errors (Fig. 8). The user may repeat his or her gesture to correct an error if the displayed 3D hand model is not acceptable to the user.

To evaluate the proposed system, the system was tested 100 times for the performance of the same assembly tasks. Fig. 9(a) and (b) show the average performance in each run; the mean and standard deviation of the time cost per frame (including the time cost for gesture segmentation) are 69.21 ms and 4.47 ms, respectively, and the mean and standard deviation of the TA per frame are 0.77 and 0.0023, respectively. The average time cost may reach approximately 30 ms/frame. The time cost statistics for each component were obtained from the 100 iterations of the virtual assembly procedure. The time costs for object selection, grasping, release and rotation of the diesel engine parts are shown in Fig. 9(c). Among these different stages, object selection required 3403 ms, and the average time costs for grasping, release and rotation were 957.57, 824.82 and 1721.90 ms, respectively. These findings indicate that the introduction of the AOM-based animations significantly increased the speeds of grasping, release and rotation. For example, in a traditional PF tracker, the time cost of translation is 5790.68 ms.

Five experiments were conducted for performance comparisons. The FHCI, GPAPF, GPDM, PF, SPF and VLMM techniques were each used 10 times to perform the same assembly task following the same procedure under the same experimental conditions. The assembly tasks and computer environment were identical for all algorithms. In the GPAPF experiment, we applied non-linear dimension reduction to the previously observed poses corresponding to different types of motion, such as grasping, releasing and translating, to generate the latent space. The number of annealing layers was set to 8, and the number of particles used was 11. For the GPDM experiment, we created a gesture library comprising 32 basic gestures. We first mapped the local parameters of these gestures to a latent space, after which we identify the corresponding dynamical model in the latent space. In the PF experiment, the number of particles was set to 10. In the SPF experiment, we used four SMD particles, and the scalar metastep size in the SMD algorithm was set to 3. In the VLMM experiment, the maximum memory length was set to 3, and the threshold was set to 0.00005. Each VLMM was expressed in the form $\mathrm{VLMM} = (Q, \Sigma, \tau, \gamma, s)$, where Q represents the set of 32 basic gestures used to create the library mentioned above and Σ is the set of tokens representing the finite VLMM alphabet.

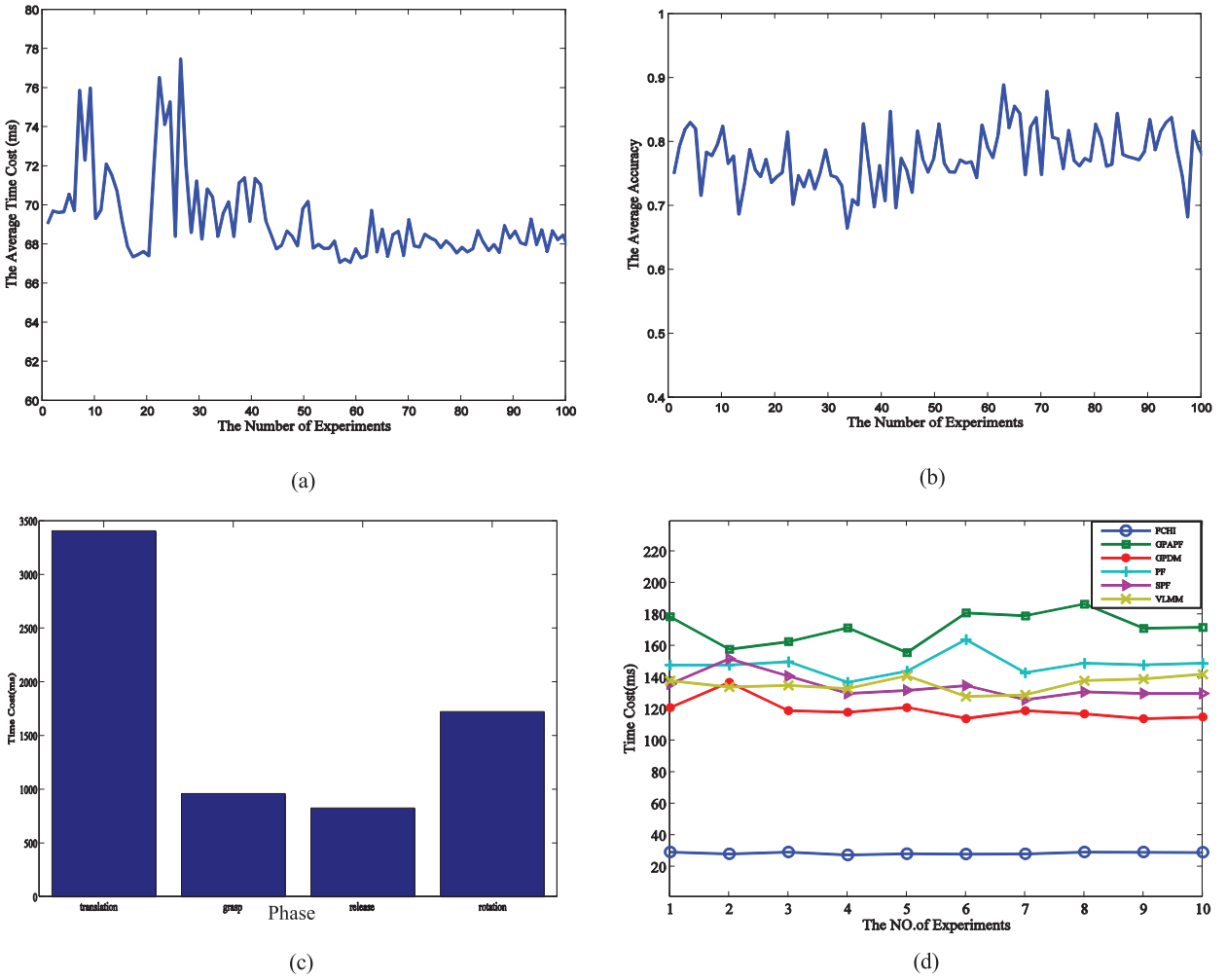


Fig. 9. Average FHCI performance based on 100 iterations. (a) Time cost per frame. (b) TA per frame. (c) Time costs of individual phases of the complete assembly process: translation (object selection) with TSI, grasping, release and rotation. (d) Comparison of time costs. The average time costs of FHCI, GPAPF, GPDM, PF, SPF and VLMM over 10 out of 50 runs were determined. The horizontal axis represents the number of runs, whereas the vertical axis represents the average time cost for each run (ms/frame).

The results show that the proposed FHCI method incurs the minimum time cost among the six algorithms (Fig. 9(d)), with an average of 26 ms/frame or 38.5 frames/s. Thus, the proposed FHCI system achieves interaction via the real-time tracking and identification of free-gesture-based inputs. The average time costs of the GPAPF, GPDM, PF, SPF and VLMM methods are 172, 120, 148, 134 and 136 ms/frame, respectively. Compared with PF, the speed of the FHCI-based interface is increased by a factor of 5.7. The TSI technique is not used in the PF algorithm, nor is the guidance provided by the projected circular region.

Finally, we conducted experiments to assess the recognition rate of Algorithm 2. In 50 runs of Algorithm 2, rotated gestures were successfully recognized 46 times, corresponding to a recognition rate of 92%. The rotated gestures included gestures that were rotated to the left or right around the central and vertical axes of the wrist. Two of 50 runs failed to be recognized because of the rotation of a gesture around the central axis to the vertical axis of the wrist. The rate of successful AOM retrieval from the AOM database was 87%.

Experiment 3: A Game of Chess

The proposed FHCI system was also applied as the interface for a chess game. We first demonstrated that the user could fetch and place chess pieces anywhere on the chessboard using the FHCI-based interface (Fig. 10(a)–(d)). Then we conducted experiments to assess the recognition rate of Algorithm 2. Fig. 10(e)–(h) illustrate the use of the TSI method to fetch or place a chess piece and the guidance of the destination of the translation gesture provided by the projected circular region. The motion of the circular region is projected from that of the 3D hand model recovered from the video of the user's gesture. Thus, the directions of motion of the circular region and the user's gesture are consistent.

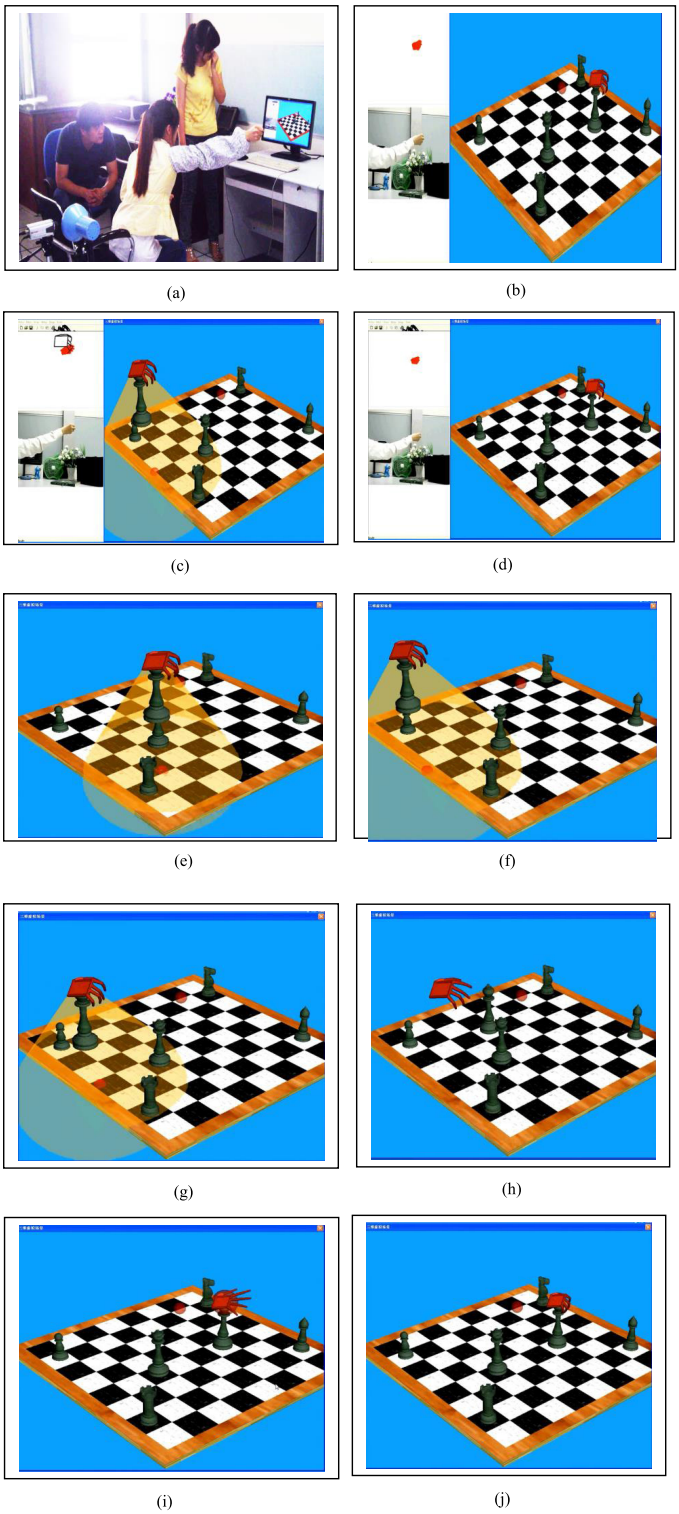
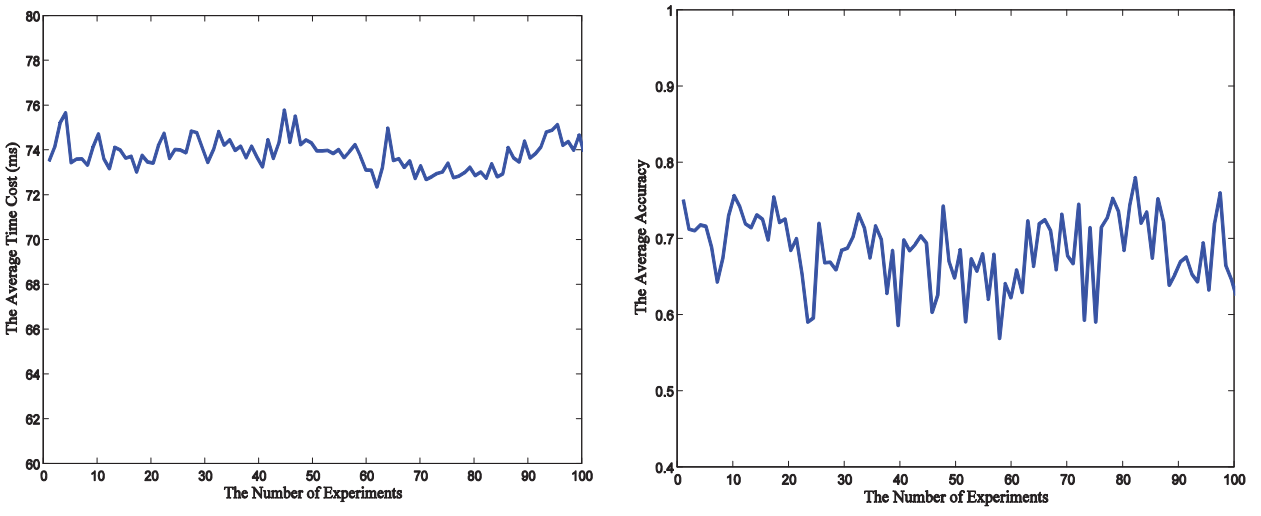
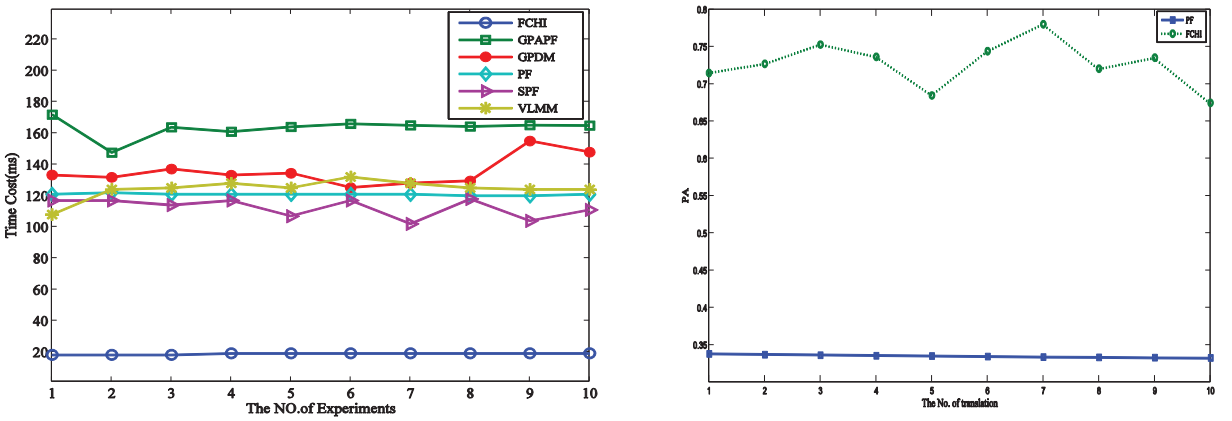


Fig. 10. Screenshots of a chess game. (a) The user is playing chess with the computer. (b) The 3D sphere indicates the location of the FP, from which object selection begins. (c) The objects within the circular region are the candidate objects for selection, and the circular region changes with the motion of the user's gesture. (d) The AOM-based grasping animation. (e)–(h) Images of the TSI-based object selection process. (i) and (j) Images of the AOM for the atomic operation 'grasp'.



(a) (b)

Fig. 11. Average time cost and TA of 100 iterations of the proposed FHCI method in a chess game: (a) average time cost and (b) average TA. The time cost includes that for gesture segmentation.



(a) (b)

Fig. 12. (a) Time cost comparison. The average time costs were determined for 10 runs using the FHCI, GPAPF, GPDM, PF, SPF and VLMM methods. The horizontal axis represents the number of runs, whereas the vertical axis represents the average time cost for each run in units of ms/frame. The time cost excludes that for gesture segmentation. (b) Comparison of the PAs in the FHCI-based and PF-based chess games.

The time cost distributions of the basic operations for the chess game were further analyzed. In 10 experiments performed for the same chess game, the average time costs for translation, grasping, release and rotation were 4566, 335, 827 and 989 ms, respectively. We also observed that skill comes with experience and that a more skillful user incurs lower time costs. TSI-based translation helps to guide user operations and improves the accuracy of determining the target positions of gestures or the objects to be manipulated. Additionally, the use of AOM-based animations significantly improves the speed of grasping, release and rotation.

The same chess game (with the same moves) was executed 100 times. The average time cost and accuracy are shown in Fig. 11(a)–(b). The mean and standard deviation of the time cost per frame (including the time cost for gesture segmentation) are 73.88 ms and 0.48 ms, respectively. The mean and standard deviation of the TA are 0.69 and 0.0021, respectively, which are superior to those for PF tracking.

The FHCI, GPAPF, GPDM, PF, SPF and VLMM methods were also applied to the chess game, and the obtained time costs are presented in Fig. 12(a). The average time cost for the FHCI method is 17 ms, with a standard deviation of 0.11 ms over 10 runs. The mean time cost for the VLMM method is 124.3 ms, with a standard deviation of 35.41 ms. The mean time cost for the GPAPF method is 163.38 ms, with a standard deviation of 34.41 ms. The mean time cost for the PF method is 120.9 ms, with a standard deviation of 0.29 ms.

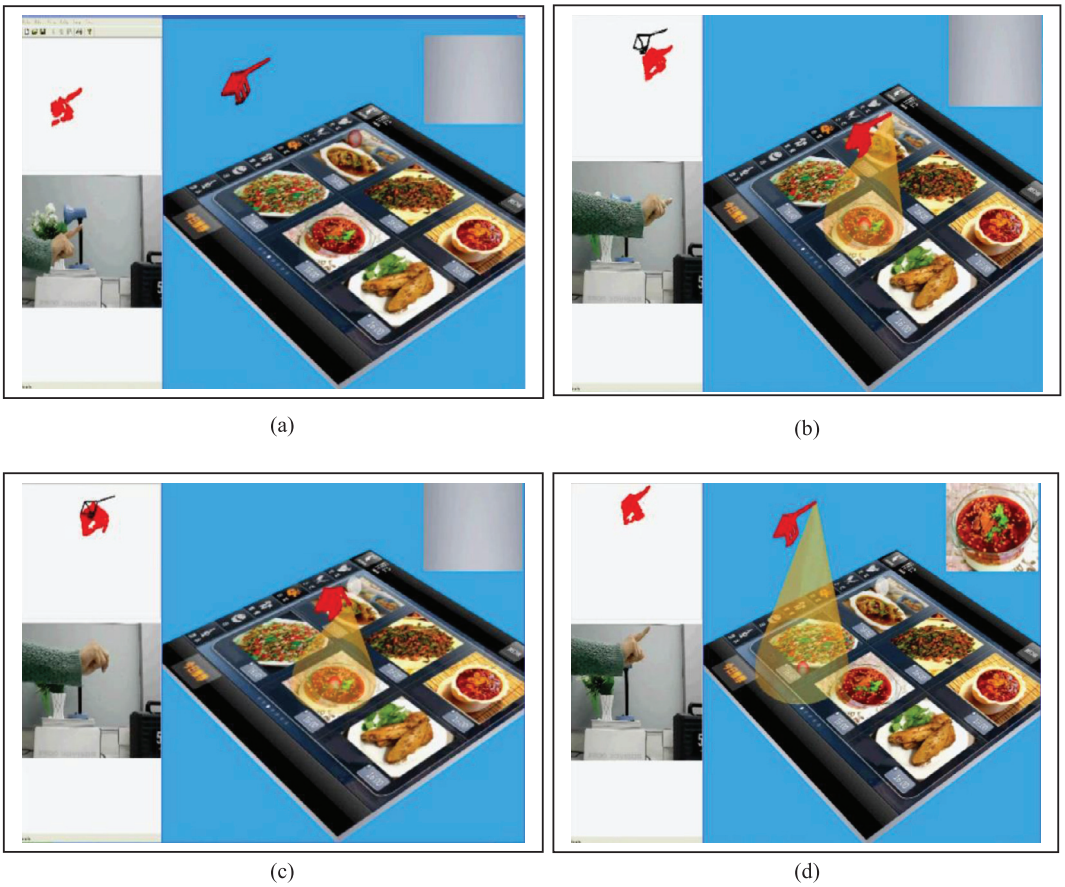


Fig. 13. Screenshots of direct menu manipulation using the FHCI-based interface. (a) The user is ready to manipulate the menu. (b) The user is indicating her intention. (c) The user is confirming her selection. (d) The user confirms her selection and begins another selection.

The mean time cost for the SPF method is 112.4 ms, with a standard deviation of 32.84 ms. The mean time cost for the GPDM method is 135.51 ms, with a standard deviation of 76.24 ms. The speed of the FHCI-based interface is higher by factors of 9.61, 7.97, 7.11, 6.61 and 7.31 compared with the GPAPF-, GPDM-, PF-, SPF- and VLMM-based interfaces, respectively. The PAs for the FHCI- and PF-based interfaces are compared in Fig. 12(b). The PA of the FHCI-based interface for placing a chess piece in a target position on the chessboard is higher than that of the PF-based interface: the average PAs for the two interfaces are 0.73 and 0.34, respectively, indicating that the PA of the FHCI algorithm is twice that of the PF algorithm.

Experiment 4: Direct Menu Manipulation

The FHCI-based interface was also applied for freehand menu manipulation by a user. First, the interaction intentions of the user were captured and visualized using the TSI algorithm. Then, the user confirmed his or her selection by bending his or her forefinger (see Fig. 13).

We asked 33 college students (18–25 years of age) and 7 teachers (28–36 years of age) to use the FHCI-based menu system, and each subject performed the experiment 10 times. The average performance in each run is shown in Fig. 14(a). The average time costs for the students (corresponding to the interval [1, 330] on the X axis in Fig. 14(a)) and the teachers (corresponding to the interval [331, 400] on the X axis in Fig. 14(a)) are 85.78 ms/frame and 85.98 ms/frame, respectively. The total average time cost for all 400 runs is 83 ms/frame, and the average PA for the students is approximately 75% (Fig. 14(b)).

User Study. We then validated the performance of the system in terms of satisfaction and cognitive burden using the National Aeronautics and Space Administration Task Load Index (NASA-TLX). We first asked 30 college students (20–24 years of age) to manipulate the FHCI-based menu interface; each subject performed the experiment 1 time. The subjects were allowed to perform manipulations at different speeds while operating the system. We asked the students to individually score the following six questions: (1) mental demand: Was the task easy or demanding? (2) physical demand: What was the intensity of the physical activity? (3) temporal demand: How much time pressure were you under because of the pace of the tasks or task elements? (4) overall performance: To what extent were you satisfied with your performance? (5) frustration level: How much irritation or stress did you feel during the task? (6) effort: How much effort did you exert to achieve your level of performance? The results are shown in Fig. 15. The scores ranged from 1 to 5, with higher scores indicating

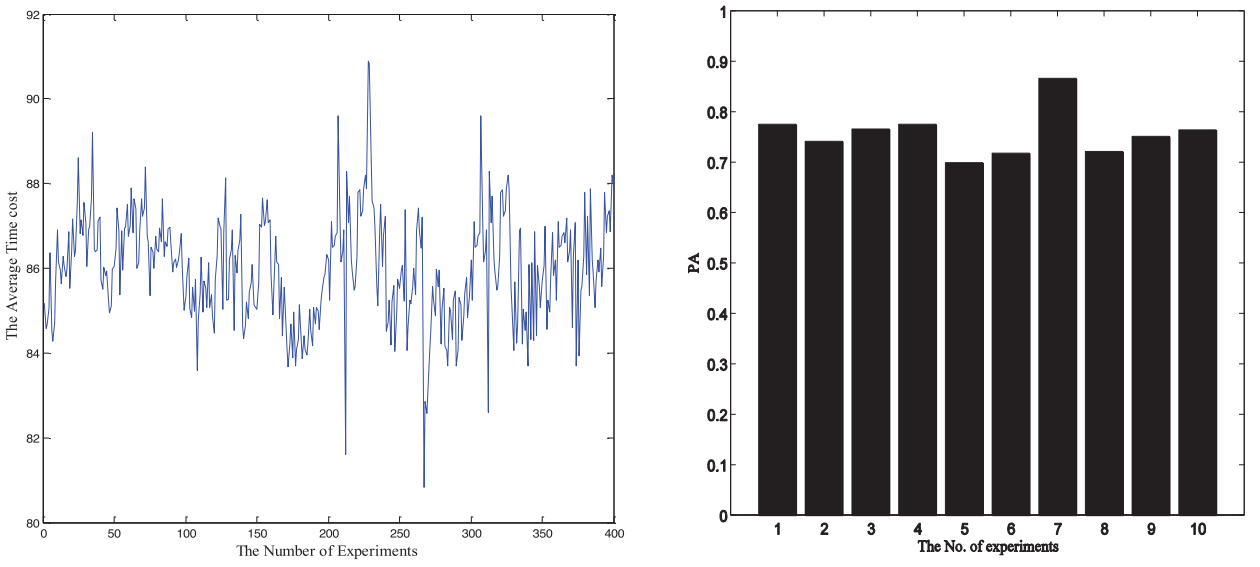


Fig. 14. Average time costs and PAs of the proposed FHCI interface for a menu system. (a) Average time costs over 400 runs. The intervals [1, 330] and [331, 400] on the X axis correspond to the time costs of the students and the teachers, respectively. (b) Average PA of the operator in each of 10 randomly selected runs. The time cost includes that for gesture segmentation.

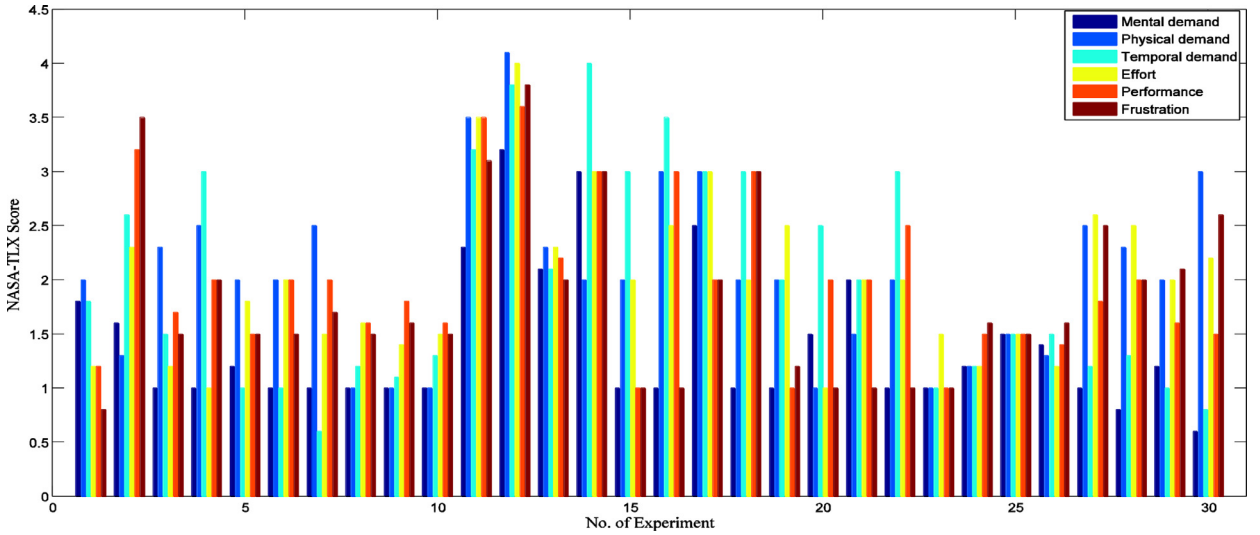


Fig. 15. NASA-TLX scores. Higher scores indicate higher levels of workload. In each experiment, each subject was required to operate the menu 10 times, and the average NASA-TLX scores were computed.

higher levels of workload. During the tasks, the subjects were not required to remember many gesture commands, which somewhat reduced the mental demand. Most of the subjects were satisfied with the speed of the interface and believed that the system was interesting. Therefore, the overall performance score given by the subjects was positive. However, most of the subjects complained of fatigue because of the long period of continuous manipulation in front of the camera. As shown in Fig. 15, the score for physical demand is high.

Limitations. Our system has several limitations. First, the proposed system is not appropriate for situations in which high-accuracy interactions are required; it is suitable only for low-accuracy systems, such as virtual assembly and game interfaces, with low TA or PA requirements. Second, the recognition rate is affected by the lighting conditions, and if the features of the background are too similar to those of the hand, the AOR algorithm will not work well. To overcome this issue, we could consider increasing the range of information acquired by using a Kinect sensor or a depth sensor.

7. Conclusions

The use of 3D hand gestures to perform interactions while providing an interesting, natural, convenient and harmonious user experience is still an open issue. Can the interaction intentions of users be captured or expressed in simple ways using a single-gesture channel? How should cognitive behavioral models guide hand tracking? Is the frame-by-frame reconstruction of 3D hand models with high accuracy necessary? Based on a thorough review of the state of the art of related work, this study proposes solutions to these interesting issues and contributes significant preliminary results.

To provide a gesture-based, task-oriented, flexible, natural 3D HCI interface with fast, accurate and robust gesture inputs based on a common monocular camera without the aid of gloves or markers, an FHCI algorithm that performs online real-time mapping from a real 3D space to a virtual 3D space is proposed for the first time. The proposed FHCI algorithm offers an innovative mode of object selection and manipulation, supports users' perceptions of their virtual interactions and reinforces users' subjective understanding of the HCI process. Little research has previously been conducted on this issue. Thus, we developed the proposed method to address this research gap.

The proposed system works well in the presence of complex backgrounds for direct 3D manipulation systems, such as a virtual assembly interface, a chess game interface and a direct menu manipulation system. The mapped hand models can move at any speed while performing interactive tasks. Based on the experimental results, a satisfactory PA can be achieved in TSI-based object fetching or placement, and a subjective understanding of the user's intent in the HCI process is expressed by means of AOM-based animations. The average interaction speed can reach 45 frames/s, and the average TA and PA can reach 0.73. Furthermore, the proposed system provides an interesting, natural, convenient and harmonious user experience that is promising not only for gesture-based interaction but also for body-based or face-based interactions.

We note that because the hand is allowed to move freely, the captured hand shapes may not be sufficiently similar to those stored in the database; in this case, gesture recognition performance will be affected. In addition, in the machine assembly experiment, the normal directions for the machine parts were all the same. These issues require further study.

Acknowledgments

This research was supported by the National Natural Science Foundation of China (Nos. 61472163, 61472232, 61572230, 61573166, 61572231, 61373054 and 61203341), the National Key Research and Development Plan (No. 2016YFB1001400), the Science and Technology Project of Shandong Province (No. 2015GGX101025) and, in part, by the Natural Science Foundation of Shandong (No. ZR2013FM004) and Doctoral Research Foundation Project of University of Jinan (No. XBS1534).

References

[1] A. Agarwal, B. Triggs, Tracking articulated motion with piecewise learned dynamical models, in: 2004 European Conference on Computer Vision. Springer– Verlag LNCS 2004, 2004, pp. 54–65.
[2] B. Bray, E. Koller-Meier, M. Muller, L. Van Gool, N.N. Schraudolph, 3D Hand tracking by rapid stochastic gradient descent using a skinning model, in: 1st European Conference on Visual Media Production (CVMP), 2004, pp. 231–237.
[3] M. Bray, E. Koller-Meier, L. Van Gool, Smart particle filtering for high-dimensional tracking, Comput. Vision Image Understanding 106 (1) (2007) 116–129.
[4] R. Chandrasekaran, A. Tamir, Open questions concerning Weiszfeld's algorithm for the Fermat-Weber location problem, Math. Program.. Series A 44 (1989) 293–295, doi:10.1007/BF01587094.
[5] Q. Chen, Real-Time Vision-Based Hand Tracking and Gesture Recognition PhD thesis, University of Ottawa, 2008.
[6] R. Chen, G. Liu, G. Zhao, J. Zhang, H. Li, 3D human motion tracking based on sequential Monte Carlo method, J. Comput. Aided Des. Comput. Graphics 17 (1) (2005) 85–92.
[7] B. Dorner, Chasing the Colour Glove: Visual Hand Tracking (Master's thesis), Simon Fraser University, 1994.
[8] J. Deutscher, A. Blake, I. Reid, Articulated body motion capture by annealed particle filtering, in: IEEE Conference on Computer Vision and Pattern Recognition (CVPR'00), 2, 2000, pp. 1144–1149.
[9] J. Deutscher, A. Davison, I. Reid, Automatic partitioning of high dimensional search spaces associated with articulated body motion capture, in: Proceedings of Conference on Computer Vision and Pattern Recognition, 2001, pp. 187–193.
[10] A. Erol, G. Bebis, M. Nicolescu, R. Boyle, R. Twombly, Vision-based hand pose estimation: a review, Comput. Vision Image Understanding 108 (2007) 52–73.
[11] E.L. Hutchins, J.D. Hollan, D.A. Norman, Direct manipulation interfaces, Hum.-Comput. Interact. 1 (4) (1985) 311–338.
[12] Y.B. Enrique, M.S. Rafael, M.C. Rafael, C.P. Angel, Comparing evolutionary algorithms and particle filters for Markerless Human Motion Capture, Appl. Soft Comput. 17 (2014) 153–166.
[13] Z. Feng, M. Zhang, Z. Pan, B. Yang, T. Xu, H. Tang, Y. Li, 3D-Freehand-Pose Initialization Based on Operator's Cognitive Behavior Models, Visual Comput. 26 (6-8) (2010) 607–617.
[14] Z.Q Feng, B. Yang, Y.H. Chen, Y.W. Zheng, T. Xu, Y. Li, T. Xu, D.L. Zhu, Features extraction from hand images based on new detection, Pattern Recognit. 44 (5) (2011) 1089–1105.
[15] Z. Feng, B. Yang, Y. Li, Y.W. Zheng, X. Zhao, Q. Meng, Real-time Oriented Behavior-driven 3D Freehand Tracking for Direct Interaction, Pattern Recognit. 46 (2) (2013) 590–608.
[16] Z.Q. Feng, B. Yang, H.T. Tang, N. Lv, Q. Meng, J.Q. Yin, S.C. Feng, Behavioral-model-based freehand tracking in a Selection-Move-Release system, Comput. Electr. Eng. 40 (6) (2014) 1827–1837.
[17] D.P. Huttenlocher, G.A. Klanderman, W.J. Rucklidge, Comparing images using the Hausdorff distance, IEEE Trans. Pattern Anal. Mach. Intell. 15 (9) (1993) 850–863.
[18] H. Hasan, S. Abdul-Kareem, Static hand gesture recognition using neural networks, Artif. Intell. Rev. 41 (2014) 147–181.
[19] N. Kılıboz, U. Güdükbay, A hand gesture recognition technique for human-computer interaction, J. Visual Commun. Image Represent. 28 (4) (2015) 97–104.
[20] M. Kulkarni, J. Butala, V. Udpikar, Static gesture recognition using PMD ToF camera, in: Proceedings of IEEE International Conference on Advances in Computing, Communications and Informatics, ICACCI, 2014, pp. 2712–2718.
[21] M. Laskar, A. Das, A. Talukdar, K. Sarma, Stereo vision-based hand gesture recognition under 3D environment, Procedia Comput. Sci. 58 (2015) 194–201.

[22] H. Liang, J. Yuan, D. Thalmann, Z. Zhang, Model-based hand pose estimation via spatial-temporal hand parsing and 3D fingertip localization, Visual Comput. 29 (2013) 837–848.
[23] Z. Li, P. Horain, A.M. Pez, C. Pelachaud, Statistical Gesture Models for 3D Motion Capture from a Library of Gestures with Variants, in: Gesture Workshop, LNAI, 5934, 2010, pp. 219–230.
[24] J. Lin, Y. Wu, T.S. Huang, Capturing Human Hand Motion in Image Sequences, in: Workshop on Motion and Video Computing (WMVC02), 2002, pp. 99–104.
[25] T. Mantecón, A. Mantecón, C.R. Del-Blanco, F. Jaureguizar, Enhanced gesture-based human-computer interaction through a compressive sensing reduction scheme of very large and efficient depth feature descriptors, in: Proceedings of IEEE International Conference on Advanced Video and Signal Based Surveillance, AVSS, 2015, pp. 1–6.
[26] A. Maqueda, C.R. del-Blanco, F. Jaureguizar, N. García, Human–computer interaction based on visual hand-gesture recognition using volumetric spatiograms of local binary patterns, Comput. Vision Image Understanding 141 (2015) 126–137.
[27] M. Morshidi, T. Tjahjadi, Gravity optimized particle filter for hand tracking, Pattern Recognit. 47 (1) (2014) 194–207.
[28] I. Oikonomidis, N. Kyriazis, A.A. Argyros, Markerless and Efficient 26-DOF Hand Pose Recovery, in: Proceedings of the 10th Asian Conference on Computer Vision, ACCV'2010, Part III, LNCS, 6494, 2010, pp. 744–757.
[29] I. Oikonomidis, N. Kyriazis, A.A. Argyros, Efficient model-based 3D tracking of hand articulations using Kinect, in: Proceedings of the 22nd British Machine Vision Conference, BMVC'2011, 2011.
[30] Prashan Premaratne, Human Computer Interaction Using Hand Gestures, Cognitive Science and Technology, Springer, 2014.
[31] I. Poupyrev, 3D Manipulation Techniques, 3D User Interface Design, Lecture Slides. SIGGRAPH'2000, 2000.
[32] L. Raskin, E. Rivlin, M. Rudzsky, 3D human tracking with gaussian process annealed particle filter, in: Proceedings of the 2nd International Conference on Computer Vision Theory and Applications (VISAPP '07), 2, 2007, pp. 459–465.
[33] J. Romero, H. Kjellstrm, D. Kragic, Hands in action: real-time 3D reconstruction of hands in interaction with objects, in: Int. Conf. on Robotics and Automation (ICRA), 2010, pp. 458–463.
[34] M. Schlattmann, R. Klein, Efficient bimanual symmetric 3D manipulation for markerless hand-tracking, Virtual Reality International Conference (VRIC), 2009.
[35] N. Singh, N. Baranwal, G.C. Nandi, Implementation and evaluation of DWT and MFCC based ISL gesture recognition, in: Proceedings of IEEE International Conference on Industrial and Information Systems, ICIIS, 2014, pp. 1–7.
[36] N. Stefanov, S. Galata, R. Hubbold, A real-time hand tracker using variable-length Markov models of behavior, Comput. Vision Image Understanding 108 (1-2) (2007) 98–115.
[37] Thomas G. Zimmerman, L. Jaron, B. Chuck, B. Steve, H. Young, A hand gesture interface device, in: Human Factors in Computing Systems (CHI), 1987, pp. 189–192.
[38] N. Vaswani, Particle Filtering For Large Dimensional State Spaces with Multimodal Observation Likelihoods, IEEE Trans. Signal Process. 56 (10) (2008) 4583–4597.
[39] R.Y. Wang, J. Jovan Popoviæ, Real-Time Hand-Tracking with a Color Glove, ACM Trans. Graphics (SIGGRAPH 2009) 28 (3) (2009) 1–8.
[40] R.Y. Wang, S. Paris, J. Popoviæ, 6D hands: markerless hand-tracking for computer aided design, in: Proceedings of the 24th annual ACM symposium on User interface software and technology (UIST '11), New York, NY, USA, ACM, 2011, pp. 549–558.
[41] X. Wu, W. Liang, Y. Jia, Tracking articulated objects by learning intrinsic structure of motion, Pattern Recognit. Lett. 30 (3) (2009) 267–274.
[42] B. Yang, X. Song, Z. Feng, X. Hao, Gesture Recognition in Complex Background Based on Distribution Features of Hand, J. Comput.-Aided Design Comput. Graphics 22 (10) (2010) 1841–1851.
[43] J. Yu, Y.K. Guo, D. Tao, J. Wan, Human pose recovery by supervised spectral embedding, Neurocomputing 166 (2015) 301–308.
[44] J. Yu, H.Y. Zhou, X.B. Gao, Machine Learning and Signal Processing for Human Pose Recovery and Behavior Analysis, Signal Process. 110 (2015) 1–4.
[45] C. Zhang, Y. Tian, Histogram of 3D Facets: A depth descriptor for human action and hand gesture recognition, Comput. Vision Image Understanding 139 (10) (2015) 29–39.
[46] Y. Zhou, G. Jiang, Y. Lin, A novel finger and hand pose estimation technique for real-time hand gesture recognition, Pattern Recognit. 49 (1) (2016) 102–114.