Accepted Manuscript

Immersive human computer interactive virtual environment using large-scale display system

Xiuhui Wang, Ke Yan

PII: S0167-739X(17)31622-9
DOI: http://dx.doi.org/10.1016/j.future.2017.07.058
Reference: FUTURE 3588

To appear in: *Future Generation Computer Systems*

Received date : 19 July 2016
Revised date : 22 June 2017
Accepted date : 23 July 2017

Please cite this article as: X. Wang, K. Yan, Immersive human computer interactive virtual environment using large-scale display system, *Future Generation Computer Systems* (2017), http://dx.doi.org/10.1016/j.future.2017.07.058

Immersive Human Computer Interactive Virtual Environment using Large-Scale Display System

Xiuhui Wang and Ke Yan*
College of Information Engineering,
China Jiliang University,
258 Xueyuan Street, Hangzhou, China, 310018.
Tel.: (+86) 153-9700-8303 Fax.: (+86) 571-8691-4580
*Corresponding author: Ke Yan, E-mail: yanke@cjlu.edu.cn

Abstract: Large-scale display system with immersive human computer interaction (HCI) is an important solution for virtual reality (VR) systems. In contrast to the traditional human-computer interactive VR system that requires the user to wear heavy VR headsets for visualization and data gloves for HCI, the proposing method utilizes a large-scale display screen (with or without 3D glasses) to visualize the virtual environment and a bare-handed gesture recognition solution to receive user instructions. The entire framwork is named as an immersive HCI system. Through a virtual 3D interactive rectangular parallelepipe, we establish the correspondence between the virtual scene and the control information. A bare-handed gesture recognition method is presented based on extended genetic algorithm. An arm motion estimation method is designed based on the fuzzy predictive control theory. Experimental results showed that the proposed method has lower error rates than most existing solutions with acceptable recognition frequency rates.

Keywords: Human computer interaction; Virtual reality; Gesture recognition; Motion estimation

1. Introduction

The technology of large-scale display system using multi-projector system presents an important solution for human computer interaction (HCI) [1, 2]. In addition, remote controls using bare hand gestures, body poses and limbs motions are more preferable for virtual reality (VR) experience, compared with wearing heavy VR devices, such as VR headsets and data gloves [3]. In this study, users will experience the virtual environment by roaming in front of a large-scale screen with multi-projector system. Instructions and orders can be received by bare-handed gesture recognition solutions [4].

The experimental setup of this study is illustrated in Fig. 1, which consists of a large-scale display, fifteen projectors, fifteen client PCs, two cameras, and one server. The user's hand gesture and arm motion are captured by the two cameras. The hand gestures are translated to operating commands as grabbing, releasing, rotating etc. And the arm motions create navigate commands, such as move left, move right, move forward etc. By combining the operating commands and the navigation commands, we enable the users to experience the immersive HCI in the virtual environment.

Fig.1: A large screen presenting the virtual environment which is produced by a multi-projector system.

Gesture recognition from a video camera is a challenging problem. First, the tightly coupled rotation, inclination and motion produce a large number of variables for computation. Second, hand region segmentation is difficult without professional devices, such as data gloves, long-sleeves shirt [5] and hand-held LED light pen [6]. Third, it remains difficult to track the head, body or limbs and combine the tracking information with bare-handed commands recognition techniques [3, 7]. In addition, integrating arm motion estimation into gesture recognition is one way to stabilize the motion tracking done by the cameras, resulting in a more robust HCI system.

In this study, we propose an immersive HCI VR framework based on computer vision techniques where the users are not required to wear extra sensors, clothing or equipment (only markers are available). The users can perform editing or roaming in a virtual 3D environment built by a multi-projector system. Comparing with the existing related works in the literature, we summarize the main contributions of this study as follows:

1) *A novel simplified skeletal hand model.* A simplified skeletal model is introduced, which uses an ellipsoid palm with strip fingers to approximate the hand. Compared with the existing hand models, the skeletal model reduces the recognition errors caused by hand rotation and occlusion.

2) *A novel hand gesture recognition algorithm.* A bare-handed gesture recognition algorithm is designed using extended genetic algorithm (GA). The extended GA naturally avoids local extremes, which increases the robustness of the gesture recognition algorithm. Results show that our method produces higher correct recognition rate comparing with existing methods.

3) *An novel arm motion estimation method.* An arm motion estimation method based on a rectangular parallelepiped for virtual interaction (RPVI) and fuzzy predictive control (FPC) is proposed. Compared to the existing motion estimation methods, our method achieves more accurate arm motion recognition results.

4) An immersive HCI VR framework. Combining all the techniques that we have used, the proposed HCI VR framework successfully accomplishes scene editing and walkthrough using bare-handed interactive commands .

2. Related Works

The proposed framework mainly contains two important techniques, namely, the hand gesture recognition and the arm motion estimation. In the section, we review the related works of the two techniques respectively.

2.1 Hand Gesture Recognition Methods

Hand gesture recognition methods received tremendous attention during the past decade for its naturalness and flexibleness in the field of HCI. Binh and Ejima [5] presented a real-time hand gesture recognition system that utilized the pseudo 2D hidden Markov models. It was limited to 2D hand tracking by a single camera. Wu et al. [8] proposed an interactive hand gesture model based on information processing model of human attention, and solved the Midas Touch Problem. Using their method, static and operator-dependent gestures could be recognized efficiently, but defects appeared on operator-independent gestures. Roomi et al. [9] proposed an improved method for gesture recognition, which employed Gaussian Mixture Model to extract foreground from the input hand images, and used star skeletonization to calculate edge information from the segmented hand region. Wan et al. [6] proposed a hand gesture interaction scheme with one LED light pen, which improved the robustness of existing hand tracking and segmentation techniques. Annamária and Balázs [10] proposed a hand gesture modeling and recognition system that converted the received hand information into a fuzzy hand-posture feature model using fuzzy neural networks, and recognized the gestures by fuzzy inference. However, in [10], the gesture recognition process did not take the position of the hand into consideration; only the hand shape was considered. Dardas and Georganas [11] presented a real-time HCI system, which detected and tracked hands in cluttered background by skin detection and hand posture contour comparison. The method was efficient and practical, but it was susceptible to quality of the cameras. Lee et al. [12] presented a method that distinguishes the starting and ending point of a gesture from a series of continuous actions. Xie and Cao [13] presented an accelerometer-based sensing device and a related gesture recognition algorithm. The method in [13] improved both user-dependent and user-independent recognition accuracy rates, but still required special assistances from the hardware aspect. In [14], a dynamic gesture recognition system with the depth information was proposed. The static gestures are classified using support vector machine (SVM). Pu et al. [15] proposed a gesture recognition method based on wireless signals, which offered a whole-home gesture recognition system without any extra instrumentations, such as cameras. Krishnan and Sarkar [16] introduced a gesture matching algorithm based on a level building approach. They

modeled each gesture sequence as a curve; and each curve is a data point in a high-dimensional space formed by gesture classes. The method could deal with both isolated and continuous gestures, but the complexity of the algorithm is high. In [17], a high-level hand feature extraction method for real-time gesture recognition was presented. The method extracted both extensional fingers and flexional fingers with high accuracy. However, if the salient hand edge is not well detected, false detection of flexional fingers could occur. Hohn et al. [18] proposed a vision-based hand gesture recognition system for intelligent vehicles, which increased the drivers' comfort without affecting their safety.

2.2 Arm Motion Estimation Methods and Related Works

There are also related works about tracking arm motion and the rest parts of the body. Sileye and Odobez [19] proposed a novel geometric model to recognize head poses without training data. Cheng and Trivedi [20] proposed a real-time vision-based user determination system that tracked the hand movements to improve the safety of vehicles. The system proposed in [20] alleviated driver distraction and maximized the passenger infotainment experience. However, it might fail to identify the active user when both the passenger’s and driver’s hands were in the region of interest. Reale et al. [7] presented a vision-based HCI system that integrated multiple modalities, including eye gaze, head pose, hand pointing, and mouth motions. The system could only recognize the pointing gesture. Sanchez and Puig [21] proposed a method solve the body gesture recognition problem in the field of HCI. Asque et al. [22] presented two haptic-assistive techniques to help motion-impaired computer users. Suau et al. [23] proposed a real-time algorithm for both head and hand tracking. The ambiguities and overlaps are resolved by using a range camera. In their solution, the hand tracking algorithm was fully dependent on the head position estimation. Therefore, a tiny error in head position estimation might result in great deviation in hand tracking. Masters et al. [24] combined accelerometer and gyroscope measurements to track limbs. A sensor fusion algorithm was implemented using commercial inertial measurement units (IMU). Tran and Trivedi [25] proposed a novel posture and gesture recognition algorithm with multiple cameras. The experimental results showed good classification rates, but the system did not handle the gesture spotting issue. In [26], a temporal tracking technique, which is used to detect and track the arm of a pointing agent, was proposed. A probabilistic method was used to weight different features for estimating the target objects.

3. A Novel Immersive Human-computer Interactive VR System using Large-scale Screen

Aiming at developing a next-generation HCI VR system, a large screen with multi-projector system is employed to increase the immersive user experience. The practical virtual environment setup is shown in Fig. 2. Two synchronized cameras are used to capture the user movements in front of the large screen displaying virtual scenes. Five markers are purposely placed for camera calibration, where four marks

are placed on the four corners of the floor rectangle and one point is placed on the top plane of the rectangular parallelepiped for virtual interaction (RPVI). A pin-hole camera model is used to recover the extrinsic and intrinsic parameters of two synchronized cameras for output images.

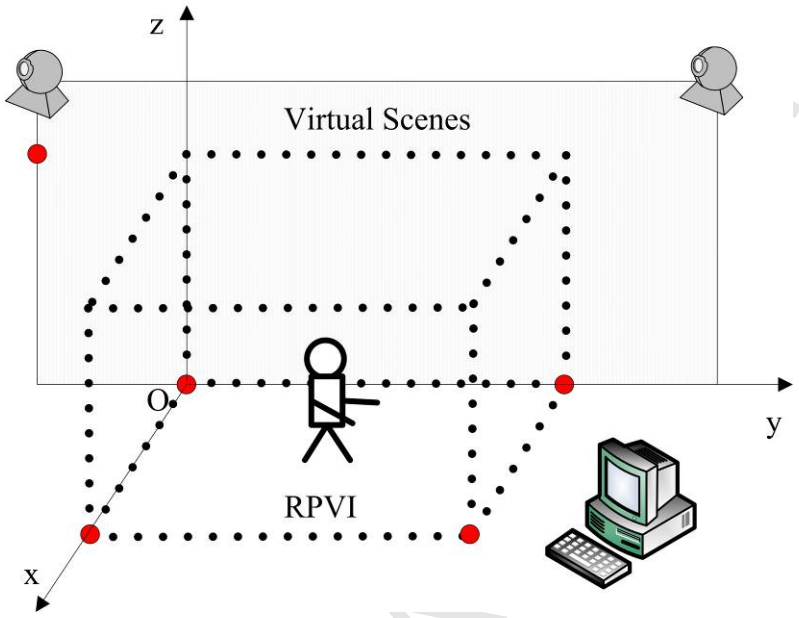


Fig.2: System framework of the immersive HCI system. Red circles represent markers.

The flowchart of the proposed immersive HCI framework is depicted in Fig. 3. The inputs are video images containing hand, arm and two marks. There are two large blocks in the flowchart, marked using dash lines, which indicate the two main topics of this study, namely, the hand gesture recognition and the arm motion estimation. The hand gesture recognition requires a GHSML and a finite state machine as inputs to build the command system; and the arm motion estimation incoportrates the camera and RPVI parameters to generate the navigation commands. In the desired immersive HCI virtual environment, users are allowed to perform instructions, such as grabbing, releasing and rotation, with arm motions moving left, right, etc. The focuses of this study are marked in green color, which are the GA-based static hand gesture recognition and FPC based arm motion estimation.

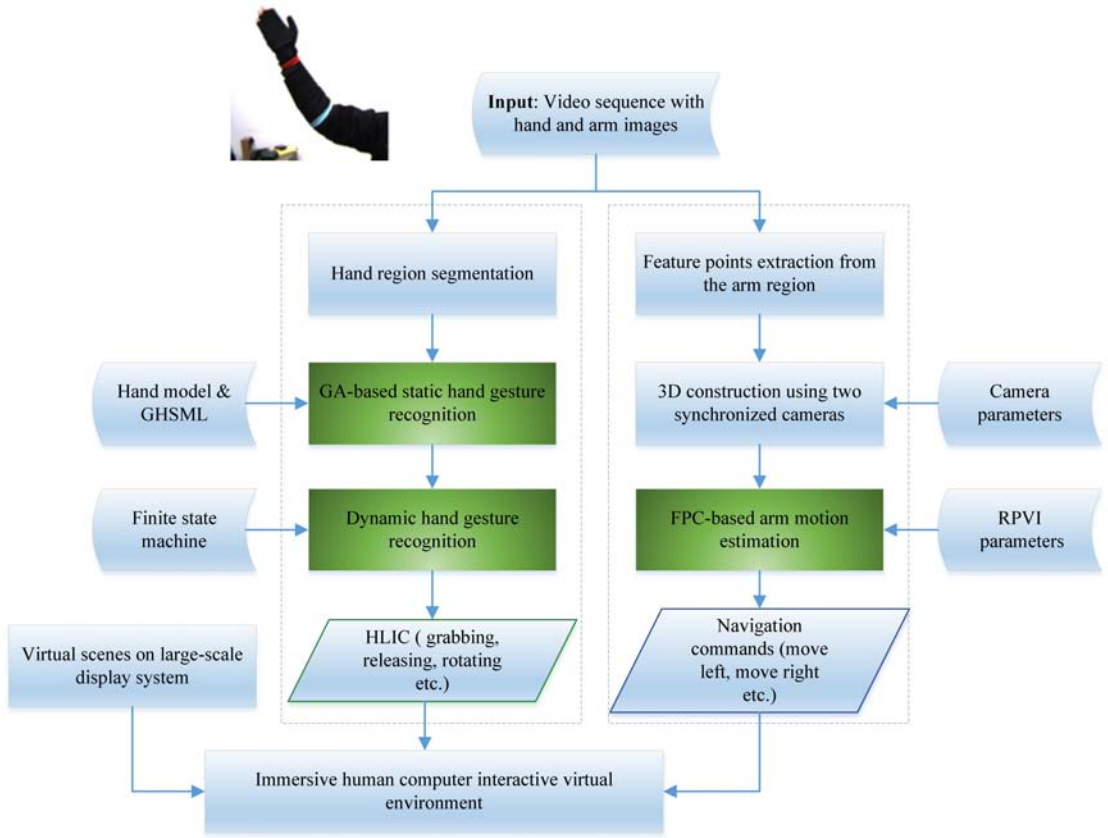


Fig.3: Main steps of the proposed immersive HCI framework.

3.1 Hand Gesture Recognition

Hand gesture recognition refers to the procedure mapping the input hand image segment with the hand model in library/database. In this subsection, first, we describe the procedures of building the hand model libraries using a simplified skeletal hand model. Second, a static hand gesture recognition algorithm is introduced integrating an extended genetic algorithm (GA). Last, we explain the dynamic hand gesture recognition that is used to generate the high-level interaction commands. A finite state machine is introduced to formalize the transit rules between different interactive commands.

3.1.1 Hand Gesture Feature Extraction

Hand gesture features include the number of fingers, the length and width of each finger, the angle between wrist and each finger and the skin color. A simplified skeletal hand model is designed to build the general hand shape model library (GHSML) consisting of six basic hand shapes. The GHSML is trained using the actual hand images of the user to obtain the special hand shape model library (SHSML) for the current user. The hand features are extracted from the practical interaction images to match the gestures in SHSML. Based on the matching results, a finite state machine is designed utilizing interactive semantics, such as “selection”, “translation”, and “rotation”.

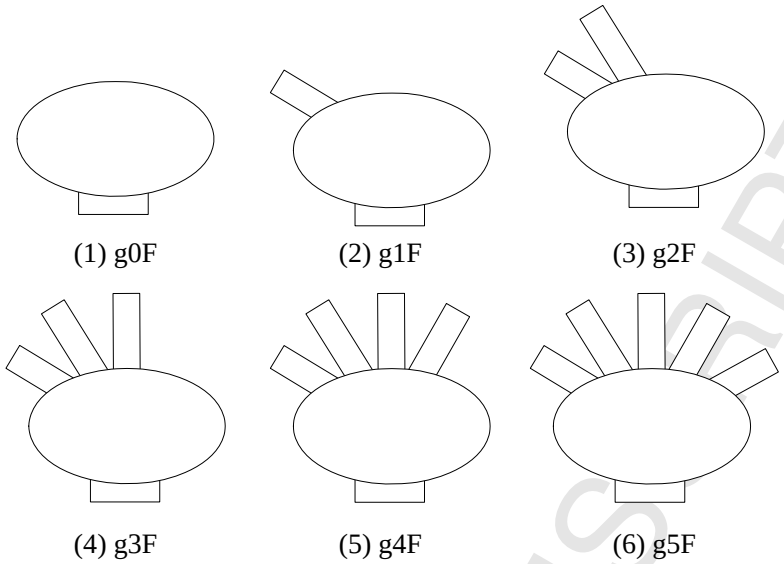


Fig.4: Six basic hand shapes in the GHSML.

A general hand shape model library with a simplified skeletal hand model is depicted in Fig.4, and can be expressed as:

$$y = f(r_1, r_2, n, L_1 \cdots L_n, W_1 \cdots W_n, \theta_1 \cdots \theta_n, R, G, B), \tag{1}$$

$$\text{satisfying: } \begin{cases} r_1 \geq 1.5 r_2 \\ n \in [0,5] \\ 1.2 r_1 \geq L_i \geq 0.3 r_1, i = 1 \cdots n \\ r_1 \geq 3.0 W_i, i = 1 \cdots n \\ \theta_i \in [0,90], i = 1 \cdots n \\ R \in [0,255], G \in [0,255], B \in [0,255] \end{cases}, \tag{2}$$

where r_1 and r_2 are the radii of palm and wrist respectively; n represents the number of fingers; $L_1 \cdots L_n$ are the length of fingers; $W_1 \cdots W_n$ are the width of fingers; $\theta_1 \cdots \theta_n$ are the angles between the wrist and fingers ($\theta_i \in [0,90)$, $i = 1 \cdots n$); R, G, B are the three channels of skin color.

The simplified skeletal hand model has three advantages:

(1) The simple ellipsoid palm model can speed up the matching process for hand shapes; and the symmetry of the ellipsoid shape can reduce the false matching rates caused by rotation.

(2) The simple strip models for fingers simplify the matching calculation for figures. The false matching of fingers due to occlusions can be reduced by counting the total number of fingers from different images.

(3) Different gestures can be distinguished using the relative size of the palm

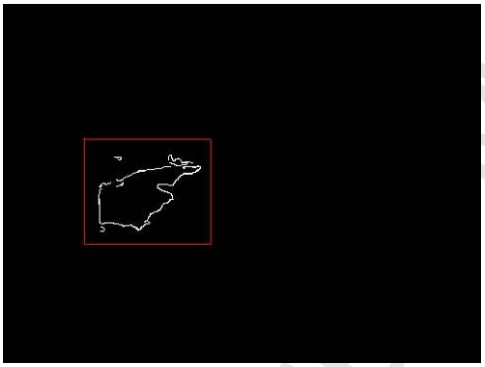
model and finger models. The relative size checking reduces the number of false gesture matching instances caused by the scaling.

The SHSML is built based on the GHSML with the current user. The hand region segmentation can be done by integrating the coherence of the hand movement, the color of the skin and the basic shape of the hand model (Fig. 5). Utilizing the coherence of the hand movement, a hand image segment can be found in the initial search window (Fig. 5a). The skin color and the basic shape of the hand model help refine the extracted hand image (Figs. 5b and 5c). The contour lines of the hand image segment is extracted using the canny operator, which can be further refined using morphological operations (Figs. 5d and 5e).

a. Initial window

b. Refined result according to skin color

c. Refined result according to hand model



d. Result after edge detecting

e. Result after filtering and connecting

Fig.5: Segmenting the hand region by coarse-to-fine method.

3.1.2 Static hand gesture recognition using extended genetic algorithm

In this subsection, we employ an extended genetic algorithm (GA) to find the optimal gesture in the SHSML. The genetic algorithm (GA) is a heuristic searching algorithm that mimics the natural process of selection and fittest survival to find the optimal solution.

The gesture recognition process can be viewed as a fitness optimization problem, and formulated as minimizing:

$$F=(\hat{n}-n)^2+\sqrt{\frac{(\hat{r}_1-r_1)^2}{(\hat{r}_2-r_2)^2}}+\sum_{i=1}^{n}\sqrt{\left((\frac{\hat{L}_i}{\hat{r}_1}-\frac{L_i}{r_1})^2+(\frac{\hat{W}_i}{\hat{r}_1}-\frac{W_i}{r_1})^2\right)\bullet\left|\hat{\theta}_i-\theta_i\right|}+\sqrt{(\hat{R}-R)^2+(\hat{G}-G)^2+(\hat{B}-B)^2}\ , \quad (3)$$

where the variables with $\wedge$ are initial values of the parameters; and F is the fitness function.

Denoting r_1, r_2 as the radii of the palm and the wrist respectively, and L_i as the length of the *i*th finger, the fitness function for the *i*th individual in SHSML can be defined as:

$$F(i) = (\hat{n} - n)^2 + \sum_{j=1}^{n} \sqrt{\frac{(\hat{r}_1 - r_1)^2}{(\hat{r}_2 - r_2)^2} + \left(\frac{\hat{L}_j}{\hat{r}_1} - \frac{L_j}{r_1}\right)^2}, \quad (4)$$

where n is the number of fingers.

After defining the fitness functions, the static hand gesture recognition algorithm based on extended GA can be described as in Algorithm 1; and the flowchart of the algorithm is depicted in Fig. 6.

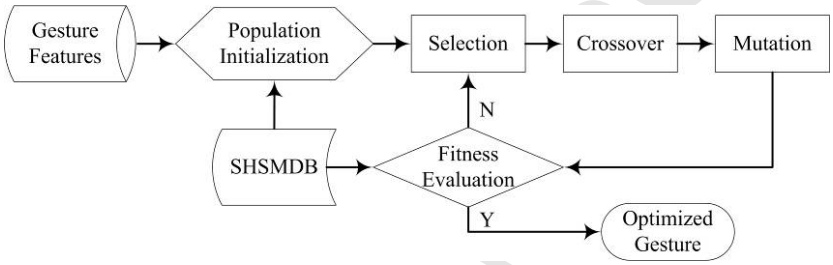


Fig.6: Steps of the static hand gesture recognition algorithm based on extended GA.

Algorithm 1 The static hand gesture recognition algorithm based on extended GA

Input: The extracted hand features and the fitness function.

Output: The selected optimized individual from library.

Step 1: Encode the hand features using binary coding.

Step 2: Construct the initial population containing M individuals. The initial population is created with several pre-defined gestures from the SHSML randomly.

Step 3: Compute fitness values. Both the fitness value of each chromosome and the total fitness value of the population are calculated.

Step 4: Selection. The fittest individual is selected and passed to the next generation directly. The rest individuals are selected based on the scale priority probability P_S. The selection probability of the *i*th individual can be expressed as

$$p_S(i) = 1 - \frac{F(i)}{\sum_{k=1}^{n} F(k)}, \quad (5)$$

where n is the overall population size, $F(i)$ is the fitness value of the *i* individual.

Step 5: Crossover. A new chromosome can be created based on general crossover rules using two old chromosomes [27]. The individuals with large differences in coding are given the priority for the crossover operation to avoid local extreme cases.

Step 6: Mutation. After the crossover, the mutation operation is performed on the newly derived child individual according to the mutation probability P_M which is

defined as:

$$P_M(i) = \left| 1 - \frac{1}{n} \sum_{k=1}^{n} F(k) / F(i) \right|, \quad (6)$$

where n is the population size. To speed up the convergence rate of the matching algorithm, we construct the select mutation probability based on the fitness deviation, to assure that the best capabilities are inherited. If the number of newly derived individuals reaches M, a new population is formed; and the algorithm will go to Step 3. Otherwise, the algorithm goes to Step 4 and continues the genetic iteration, until that the pre-defined maximum iteration number is reached.

Since the selection probability is determined by the fitness value, the fitness probability is proportional to the selection probability. The crossover enables the chromosome to exchange information and assure that the useful information is preserved. The mutation is a bit-changing operation in the gene code level, which produces new individuals. The extended genetic algorithm is a oriented stochastic algorithm, which makes the proposed algorithm naturally avoid the local extremes, and potentially increases the robustness of the overall HCI framework.

3.1.3 A dynamic hand gesture interaction model

A finite state machine is used to create high-level interaction commands (HLIC) for a dynamic hand gesture interaction model. The HLIC set includes 15 interaction commands, which are listed in Table 1. The six gestures from g0F to g5F refer to the six basic gestures shown in Fig. 4.

Table 1: Interaction commands in HLIC.

Name	Abbreviation	Implementation
Leisure	sLeisure	Continuous g5F
Grabbing	sGrabbing	From g5F to g0F
Releasing	sReleasing	From g0F to g5F
Rotating along X	sRotatingX	From g1F to g2F
Rotating along minus X	sRotatingMX	From g2F to g1F
Rotating along Y	sRotatingY	From g1F to g3F
Rotating along minus Y	sRotatingMY	From g2F to g3F
Rotating along Z	sRotatingZ	From g1F to g4F
Rotating along minus Z	sRotatingMZ	From g2F to g4F
Zooming in along X	sZoomingIX	From g3F to g1F
Zooming out along X	sZoomingOX	From g4F to g1F
Zooming in along Y	sZoomingIY	From g3F to g2F
Zooming out along Y	sZoomingOY	From g4F to g2F
Zooming in along Z	sZoomingIZ	From g3F to g4F
Zooming out along Z	sZoomingOZ	From g4F to g3F

The commands in the HLIC set can be categorized into four types. 1) The sLeisure command, achieved by continuously keeping g5F gesture, is the only command implemented by a single gesture, which terminates all other commands. 2)

The sGrabbing and sReleasing commands are mainly used for translation operations, which can be integrated with the navigation commands created by arm motions. 3) The sRotating and sZooming commands are composite commands, which are implemented by switching gestures from g1F to g2F and from g3F to g4F, respectively. They are dependent commands, which require at least one selected object in the virtual scene. The complete finite state machine of all gesture interaction operations is shown in Fig.7.

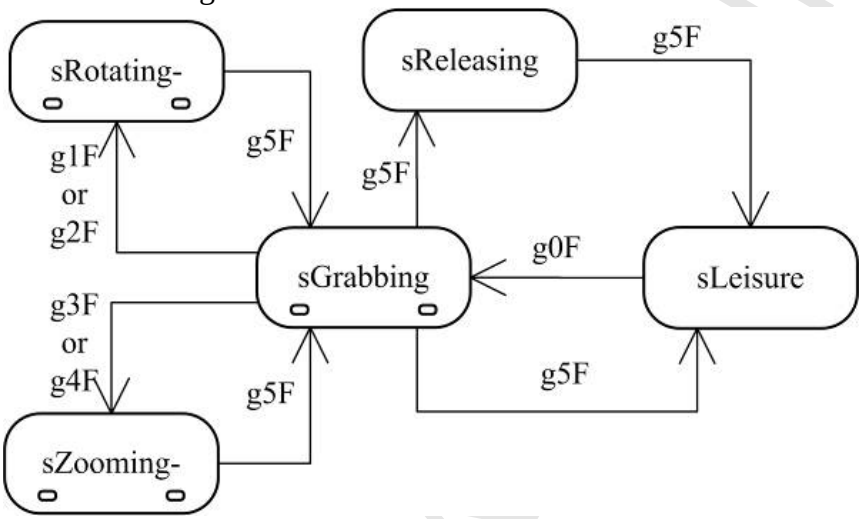


Fig.7: The dynamic hand gesture interaction model.

3.2 Arm Motion Estimation

In this subsection, we introduce the arm motion estimation algorithm. Two markers are attached on the elbow and wrist of the arm. The arm motions are estimated by calibrating two synchronous cameras to calculate the 3D movements of the two markers. The navigation commands are deduced based on the arm motion estimation in the virtual scenes. The main difference between our method and traditional motion estimation methods is that our method utilizes the rectangular parallelepiped (RPVI) to infer selection and navigation operations. Furthermore, A fuzzy predictive control (FPC) algorithm is utilized to stabilize the relative movements between the arm and the front plane of the RPVI, which leads to the robustness of the proposed navigation operations.

3.2.1 Feature points extraction

The feature points extraction algorithm can be described by the following four steps:

1) The two markers' center coordinates are calculated based on the input images captured by the cameras: $P_W(x1, y1)$ and $P_E(x2, y2)$.

2) With the extrinsic and intrinsic parameters of the cameras, we construct the projection matrix:

$$P_C^i = K_i \left[R_i \middle| t_i \right], \tag{7}$$

where **K** denotes the intrinsic matrix, **R** is the rotation matrix, *t* is the translation

vector and *i* is 1 or 2. The 3D coordinates of the two points, $P_W(x1, y1)$ and $P_E(x2, y2)$, are projected into the coordinates system as shown in Fig.2, to obtain the new coordinates $P_W^{'}(x_1^{'}, y_1^{'}, z_1^{'})$ and $P_E^{'}(x_2^{'}, y_2^{'}, z_2^{'})$.

3) The intersection point $P_X^{'}\left(x_3^{'}, y_3^{'}, z_3^{'}\right)$ is calculated by intersecting the extended line of $P_W^{'}P_E^{'}$ and the front plane of the RPVI (Fig.8). The point $P_X^{'}$ links to the virtual object on the display wall, which can be edited according to specific commands.

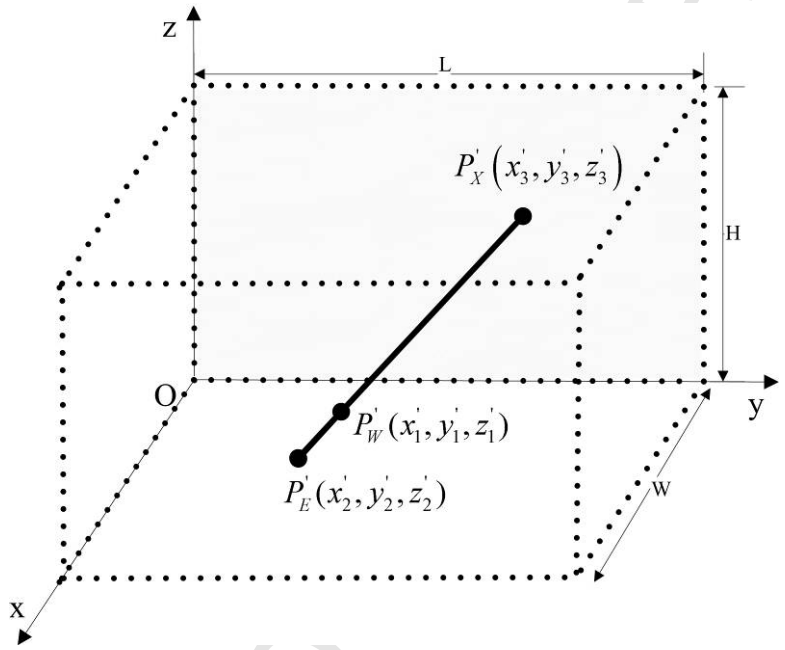


Fig.8: Intersect between the fore arm and the front plane of the RPVI.

4) The midpoint $P_M^{'}(x_4^{'}, y_4^{'}, z_4^{'})$ of the line segment $P_W^{'}P_E^{'}$ records the displacement information in the six directions, i.e. left, right, upward, downward, forward and backward. it is also the reference point navigating the walkthrough of moving arms, walking in the RPVI, or both.

3.2.2 Arm motion estimation based on fuzzy predictive control

An arm motion estimation algorithm based on fuzzy predictive control is proposed to correct the possible errors in camera calibration and feature point extraction. The objective of the proposed arm motion estimation algorithm is to 1) predict the movement tendency by pre-defined fuzzy control rules; 2) use the prediction to revise the position of the input intersection points and the reference points.

Referring to the situation in Fig. 8., the three angles between $P_W^{'}P_E^{'}$ and three coordinate axes are denoted as α, β and χ. The ranges of all variables are defined as:

$$\begin{cases} x_i^{'} \in [0, W] \\ y_i^{'} \in [0, L] \\ z_i^{'} \in [0, H] \\ \alpha, \beta, \chi \in [0, 90] \end{cases}, \tag{8}$$

where $\left\{ x_i^{'}, y_i^{'}, z_i^{'} \mid i = 1, 2, 3 \right\}$ denote the coordinates of $P_W^{'}$, $P_E^{'}$ and $P_X^{'}$. The fuzzy control variables related to $P_W^{'}$, $P_E^{'}$ and $\{\alpha, \beta, \chi\}$ are denoted as $\left\{ (\hat{x}_i, \hat{y}_i, \hat{z}_i) \mid i = 1, 2 \right\}$ and $(\hat{\alpha}, \hat{\beta}, \hat{\chi})$ respectively. The control rules are defined as

$$\begin{cases} \text{IF } (\hat{x}_1, \hat{y}_1, \hat{z}_1) = (A_1, A_2, A_3) \; AND \; (\hat{x}_2, \hat{y}_2, \hat{z}_3) = (A_4, A_5, A_6) \; THEN \; \hat{\beta} = A_0 \\ \text{IF } (\hat{x}_1, \hat{y}_1, \hat{z}_1) = (B_1, B_2, B_3) \; AND \; (\hat{x}_2, \hat{y}_2, \hat{z}_3) = (B_4, B_5, B_6) \; THEN \; \hat{\beta} = B_0 \\ \text{IF } (\hat{x}_1, \hat{y}_1, \hat{z}_1) = (C_1, C_2, C_3) \; AND \; (\hat{x}_2, \hat{y}_2, \hat{z}_3) = (C_4, C_5, C_6) \; THEN \; \hat{\chi} = C_0 \\ \text{IF } (\hat{x}_1, \hat{y}_1, \hat{z}_1) = (D_1, D_2, D_3) \; AND \; (\hat{x}_2, \hat{y}_2, \hat{z}_3) = (D_4, D_5, D_6) \; THEN \; \hat{P}_X = (0, 0, D_0) \end{cases}, \tag{9}$$

where $\hat{P}_X$ denotes the fuzzy control variable of $P_X^{'}$; and $\left\{ A_i, B_i, C_i, D_i \mid i \in [1, 6] \right\}$ are the linguistic terms related to the linguistic variables of the input and output.

The membership function of mapping the input points into the fuzzy sets is given by:

$$\varphi(x) = \frac{e^{-\frac{x - x_0}{2\sigma^2}}}{\Delta r + 1}, \tag{10}$$

where σ denotes the standard deviation of the input variable x; Δr is the predictive feedback factor according to the current output; and x_0 is the initial value whose default value is the offset of the RPVI center in x direction.

In addition, the Mamdani minimum operation rules and the fuzzy control rules in Eq. (5) are used to calculate the predictive values of the state variables for arm positions. After fuzzy reasoning, we employ weighted mean method to calculate the clarifying control variables at t time by

$$\mathbf{P} = \left\{ \alpha^t, \beta^t, \chi^t, P_X^t \right\} = \frac{\sum_{k=1}^{n} \omega_k \hat{\mathbf{P}}_k}{\sum_{k=1}^{n} \omega_k}, \tag{11}$$

where $\hat{\mathbf{P}}_k$ denotes the fuzzy control variable $\left\{ \hat{\alpha}^t, \hat{\beta}^t, \hat{\chi}^t, \hat{P}_X^t \right\}$, and ω_k is the weight of the kth control rule. The predictive feedback variable Δr at $(t+1)$ time is:

$$\Delta r = \frac{\left| \alpha^t - \bar{\alpha} \right| + \left| \beta^t - \bar{\beta} \right| + \left| \chi^t - \bar{\chi} \right|}{\frac{1}{n} \sqrt{\sum_{i=1}^{n} \left(\omega_i \cdot \left(\alpha_i^2 + \beta_i^2 + \chi_i^2 \right) \right)}}, \tag{12}$$

where $\bar{\alpha}, \bar{\beta}, \bar{\chi}$ is the historical mean value of α, β, χ respectively; and $\omega_i (i = 1, \ldots, n)$ denotes the weights of the corresponding historical data samples.

4. Result and Comparison

The proposed immersive HCI framework is tested using the same experimental setup with two typical interaction examples. The experimental setup of the two experiments includes:

1. Cameras calibration: two synchronous cameras are calibrated by the pin-hole camera model.

2. Display virtual scenes on the large-scale display wall: a virtual 3D city model is displayed using the VRML rendering engine.

3. Motion capture: the 3D coordinates of the operator's arms are calculated according to steps in subsections 3.1 and 3.2.

4. Scene walkthrough: Move virtual scenes to create navigation effect according to the results of arm motion estimations.

The first example (Experiment 1) enables the user to roam in the virtual scene with navigation commands created by arm motion estimation. In the second example (Experiment 2), the arm motion estimation is combined with the hand gesture recognition to generate interactive commands in front of the large scale display wall. Our experiments are conducted on PC equipped with a 4-core Xeon E5450 3.0GHz CPU and 8G DDR3 memory.

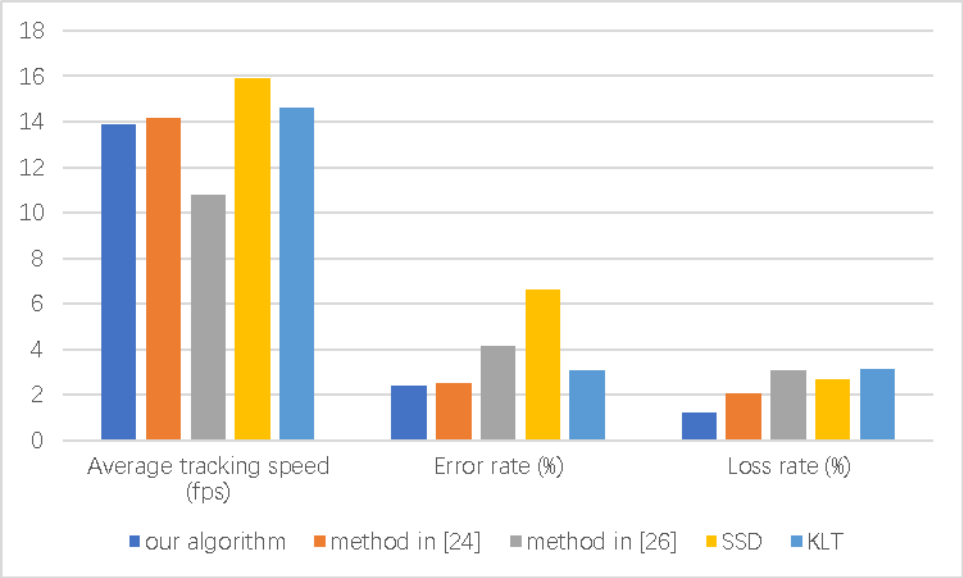
4.1 Experiment 1 Results

We compared our method with four other existing motion estimation algorithms: SSD (sum of squared difference) [28], KLT (Kanade-Lucas-Tomasi) [29], sensor-fusion based method [24], temporal arm tracking method [26]. The results of error rates, loss rates and average tracking speed are shown in Table 2 and Fig. 9.

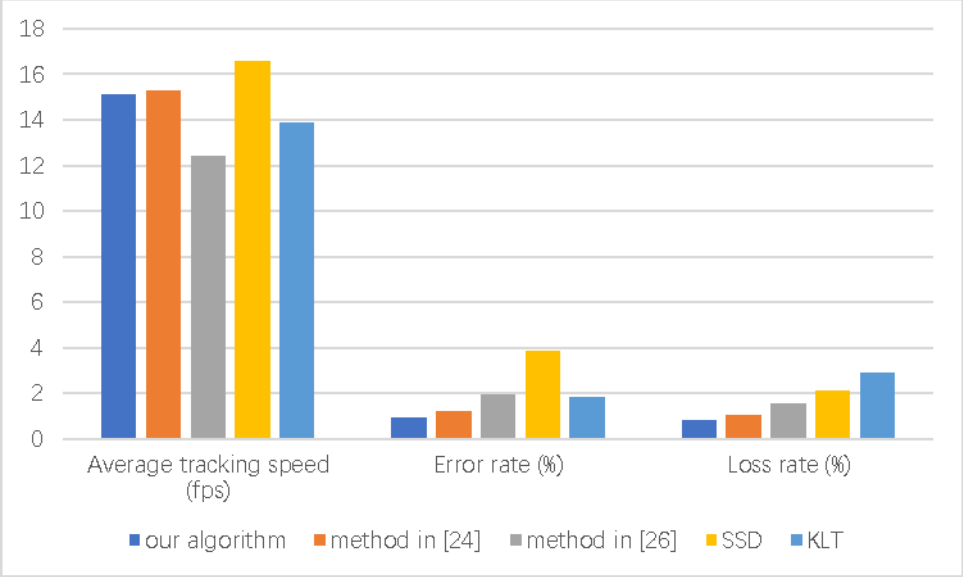
From the results, we can see that, under the same camera frame rate, our algorithm has lower error rates and loss rates with similar average tracking speed comparing with the four existing methods. The fuzzy predictive control (FPC) effectively increased the estimation accuracy and reduced the loss rate.

Table 2: Result comparison of Experiment 1.

Methods	Camera frame rate (fps)	Average tracking speed (fps)	Error rate (%)	Loss rate (%)
our algorithm	15	13.91	2.44	1.21
method in [24]	15	14.15	2.55	2.08
method in [26]	15	10.77	4.17	3.11
SSD	15	15.93	6.65	2.67
KLT	15	14.62	3.10	3.14
our algorithm	30	15.11	0.97	0.83
method in [24]	30	15.32	1.23	1.07
method in [26]	30	12.43	1.98	1.55
SSD	30	16.57	3.88	2.15
KLT	30	13.90	1.85	2.93



(a) With camera frame rate 15 fps



(b) With camera frame rate 30 fps

Fig.9: Illustrations of the results obtained from Experiment 1

4.2 Experiment 2 Results

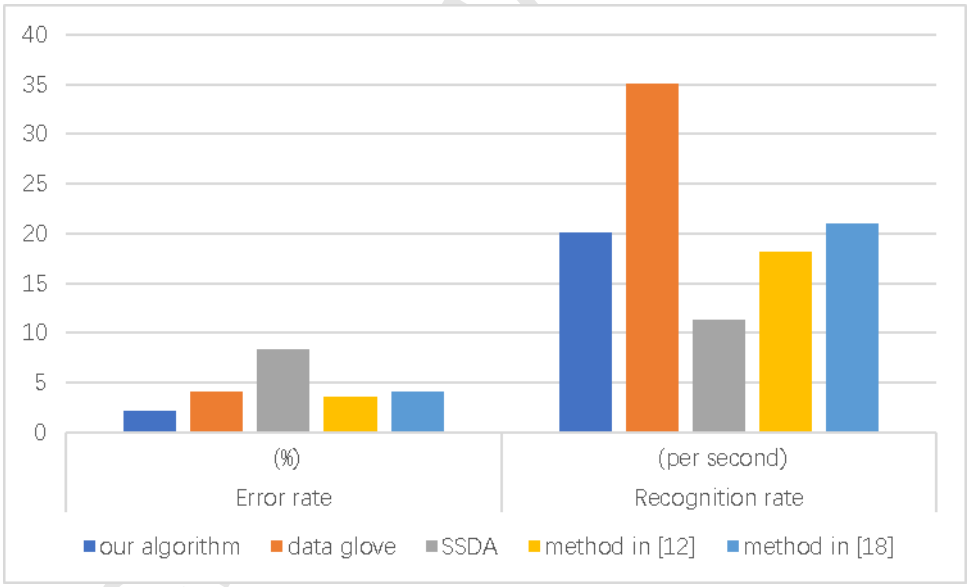
In Experiment 2, five different gesture recognition methods are compared (Table 3 and Figure 10). Comparing with gesture recognition methods based on hardware sensors (such as data gloves), although the recognition speed is slower, our method achieves much higher recognition accuracy, especially for some small distinguishing-degree gestures, since the sensitivity of data glove largely depends on operator's hand size with intrinsic sensitivity limit.

Compared to the classical equence similarity detection algorithm (SSDA) method [30], our algorithm obtains lower error rates and higher recognition speed. In comparison with the methods in [12] and [18], our algorithm has lower error rates and similar recognition speed. The lower error rates are achieved by utilizing the extended GA that converges to the global optimal effectively.

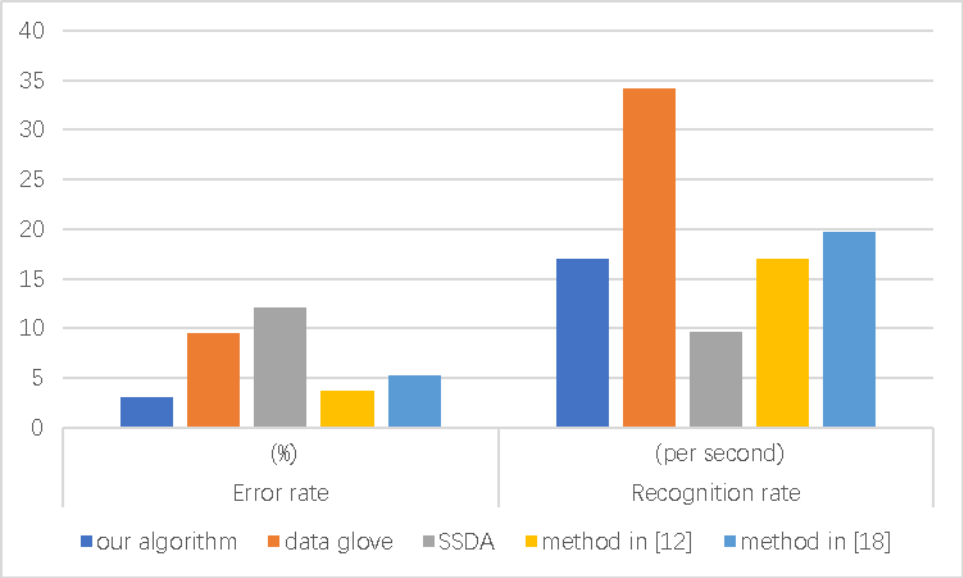
Comparing with the simple selection operation, the translation operation is much more complex. For the translation testing case, all methods have higher error rates with lower recognition rates. The reason is that the translation operation is composed by multiple basic gestures, which involves multiple gesture recognition processes. The high computational load decreases the overall recognition accuracy.

Table 3: Results comparison of Experiment 2.

Methods	Testing cases	Testing times	Error rate (%)	Recognition rate (per second)
our algorithm	selection	1000	2.15	20.11
data glove	selection	1000	4.17	35.10
SSDA	selection	1000	8.33	11.35
method in [12]	selection	1000	3.54	18.17
method in [18]	selection	1000	4.11	21.02
our algorithm	translation	1000	3.09	17.02
data glove	translation	1000	9.50	34.21
SSDA	translation	1000	12.07	9.67
method in [12]	translation	1000	3.78	16.96
method in [18]	translation	1000	5.26	19.76



(a) With the selection testing case



(b) With the translation testing case

Fig.10: Illustrations of the results obtained in Experiment 2

5. Conclusion and Future Work

In this study, we proposed a framework for immersive human-computer interactive VR system. The virtual environment was produced by a large-scale display system, with or without 3D effects. The user was allowed to roam in front of the large display wall and give commands by bare-hand gestures and arm motions. In this study, we focus on two main blocks of the whole framework, which are hand gesture recognition and arm motion estimation.

The hand gesture recognition process consists of three major steps. First, we designed a simplified skeletal model for human hands. A GHSML was built using six basic hand gestures with general initial values; and a follow-up SHSML was built by combining the GHSML with the current user gesture data. Second, we extracted the hand features from the camera captured images, and matched them with the gesture features in SHSML by solving an optimization problem using an extended GA. The matching results were of the six basic gestures which represent the original interactive semantics.

The arm motion estimation method utilized a virtual interactive rectangular parallelopiped in front of a large scale display wall. The 3D coordinates of the arms were calculated dependent on two synchronous cameras. Based on the fuzzy predictive control theory, a set of navigation commands were generated. A finite state machine was built to complete the immersive human-computer interactive VR system using interactive semantics, such as “selection”, “translation”, and “rotation”. Occlusions were handled by counting the number of fingers by calibrating multiple cameras.

As a future work, we are about to extend the current work to next generation web applications such as online games, painting exhibitions and shopping websites. The six basic hand gestures are able to provide basic instructions for online games. More hand gesture instructions are desired for more complex online games. For the online

painting exhibition and shopping experiences, users will experience the virtual environment effect, where selection, wearing, purchases and so on can be done virtually by bare-hand gesture instructions.

Conflict of Interests

The authors declare that there is no conflict of interests regarding the publication of this paper.

Acknowledgements

This work is supported by the NSF of China (No. 61303146 and No. 61602431), and is performed under the auspices by the AQSIQ of China (No.2010QK407).

Author Contributions

Conceived and designed the models: Xiuhui Wang and KeYan.
Performed the simulations: Xiuhui Wang.
Analyzed the data: Xiuhui Wang and KeYan.
Wrote the paper: Xiuhui Wang and KeYan.
Provided ideas to improve the systems modeling: Ke Yan.

References

1. Mitra, S., Acharya, T.: Gesture recognition: a survey. IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews **37**(3), 311-324 (2007)
2. Khan, R.Z., Ibraheem, N.A.: Hand gesture recognition: a literature review. International Journal of Artificial Intelligence & Applications **3**(4), 161-174 (2012)
3. Moustakas, K., Tzovaras, D., Dybkjaer, L., Bernsen, N.O., Aran, O.: Using modality replacement to facilitate communication between visually and hearing-impaired people. IEEE MultiMedia **18**(2), 26-37 (2011)
4. Lukowicz, P., Amft, O., Roggen, D., Cheng, J.Y.: On-body sensing: from gesture-based input to activity-driven interaction. IEEE Computer **43**(10), 92-96 (2010)
5. Binh, N.D., Shuichi, E., Ejima, T.: Real-time hand tracking and gesture recognition system. In: GVIP 05 Conference (2005)
6. Wan, H.G., Xiao, H.Y., Zou, S.: Hand Gesture Interaction for Next-Generation Public Games. Journal of Computer-Aided Design & Computer Graphics **23**(7), 1159-1165 (2011)
7. Reale, M.J., Canavan, S., Yin, L.J., Hu, K.N., Hung, T.: A multi-gesture interaction system using a 3-D iris disk model for gaze estimation and an active appearance model for 3-d hand pointing. IEEE Transactions on Multimedia **13**(3), 474-487 (2011)
8. Wu, H.Y., Zhang, F.J., Liu, Y.J., Dai, G.Z.: Research on key issues of vision-based gesture interfaces. Chinese Journal of Computers **32**(10), 2030-2041(2009)
9. Roomi, S.M.M., Priya, R.J., Jayalakshmi, H.: Hand gesture recognition for human-computer interaction. Journal of Computer Science **6**(9), 1002-1007 (2010)

10. Annamária, R.V.K., Tusor, B.: Human–computer interaction for smart environment applications using fuzzy hand posture and gesture models. IEEE Transactions on Instrumentation and Measurement **60**(5), 1505-1514 (2011)
11. Dardas, N.H., Georganas, N.D.: Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. IEEE Transactions on Instrumentation and Measurement **60**(11), 3592-3607 (2011)
12. Daeha Lee, Hosub Yoon, Jaehong Kim: Continuous gesture recognition by using gesture spotting. In: the 16th International Conference on Control, Automation and Systems(2016)
13. Renqiang Xie; Juncheng Cao: Accelerometer-Based Hand Gesture Recognition by Neural Network and Similarity Matching. IEEE Sensors Journal 16(11) 4537 - 4545 (2016)
14. Chih-Hung Wu, Wei-Lun Chen, Chang Hong Lin: Depth-based hand gesture recognition. Multimedia Tools and Applications 75(12) 7065-7086(2016)
15. Qifan Pu, Sidhant Gupta, Shyamnath Gollakota, Shwetak Patel: Gesture Recognition Using Wireless Signals. ACM GetMobile 18(4) 15-18 (2015)
16. Ravikiran Krishnan, Sudeep Sarkar: Conditional distance based matching for one-shot gesture recognition.Pattern Recognition 48(4) 1302-1314 (2015)
17. Yimin Zhou,Guolai Jiang, Yaorong Lin. A novel finger and hand pose estimation technique for real-time hand gesture recognition. Pattern Recognition 49 (2016) 102-114
18. Vijay John, Ali Boyali, Seiichi Mita, Masayuki Imanishi, Norio Sanma: Deep Learning-Based Fast Hand Gesture Recognition Using Representative Frames. In: International Conference on Digital Image Computing: Techniques and Applications (2016)
19. Ba, S.O., Odobez, J. M.: Recognizing visual focus of attention from head pose in natural meetings. IEEE Transactions on Systems, Man, and Cybernetics—Part B: Cybernetics **39**(1), 16-33 (2009)
20. Cheng, S.Y., Trivedi, M.M.: Vision-based infotainment user determination by hand recognition for driver assistance. IEEE Transactions on Intelligent Transportation Systems **11**(3), 759-764 (2010)
21. Sanchez, T.G., Puig, D.: Real-time body gesture recognition using depth camera. Electronics Letters **47**(12), (2011)
22. Asque, C.T., Day, A. M., Laycock, S.D.: Haptic-assisted target acquisition in a visual point-and-click task for computer users with motion impairments. IEEE Transactions on Haptics **5**(2), 120-130 (2012)
23. Suau, X., Javier, R.H., Casas, J.R.: Real-time head and hand tracking based on 2.5D data. IEEE Transactions on Multimedia **14**(3), 575-585 (2012)
24. Matthew Masters, Luke Osborn, Nitish Thakor, Alcimar Soares: Real-time arm tracking for HMI applications. In: IEEE 12th International Conference on Wearable and Implantable Body Sensor Networks(2015)
25. Tran, C., Trivedi, M.M.: 3-D posture and gesture recognition for interactivity in smart spaces. IEEE Transactions on Industrial Informatics **8**(1)178-187 (2012)
26. Polychronis Kondaxakis, Khurram Gulzar, Ville Kyrki. Temporal arm tracking and probabilistic pointed object selection for robot to robot interaction using deictic gestures. In: the 16th International Conference on Humanoid Robots(2016)
27. Jin D J, Zhang J Y.: A new crossover operator for improving ability of global searching. In: Proceeding of the 6th International Conference on Machine Learning and Cybernetics, Hong Kong, 2328-2332(2007)

28. Daniel Scharstei, Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms[J]. International journal of computer vision, 2002, 47(1-3): 7-42.
29. Jianbo Shi, Carlo Tomasi. Good features to track[C]//Computer Vision and Pattern Recognition, 1994. Proceedings CVPR'94., 1994 IEEE Computer Society Conference on. IEEE, 1994: 593-600.
30. Guangmeng Wu and Peijing Chen. An Improved SSDA in Image Registration [J]. Computer Engineering and Applications, 2005, 33: 025.

***Biographies (Text)**

Dr Xiuhui Wang was awarded PhD and Msc (Research) degrees in 2007 and 2003 from Zhejiang University. His research and teaching interests are focused on computer graphics, computer vision, and computer networks. He commenced working as an academic staff in the college of information engineering, China Jiliang University in 2007, firstly as a Lecturer then an associate professor in 2009.

***Biographies (Text)**

Dr. Ke Yan completed both his Bachelor and Ph.D. degrees in National University of Singapore (NUS). He received his Ph.D. certificate in computer science in 2012 under the supervision of Dr. Ho-Lun Cheng. During the years between 2013 and 2014, he was a post-doctoral researcher in Masdar Institute of Science and Technology in Abu Dhabi, UAE. Currently, he serves as a lecturer in China Jiliang University, Hangzhou, China. His main research interests include computer graphics, computational geometry, data mining and machine learning.

*Biographies (Photograph)

Click here to download high resolution image

***Biographies (Photograph)**

Click here to download high resolution image

- This paper presents a novel approach to utilize a large-scale screen and HCI techniques for the users to experience the virtual reality (VR).
- The user instructions are learned by the combination of gesture recognition techniques (based on extended genetic algorithm) and motion estimation techniques (using fuzzy predictive control).
- A framework and flowchart is designed for the semantics of interactive HCI.
- Compared to traditional VR headsets and data gloves approaches, the proposing method is more effective, robust and revolutionary.