IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received October 24, 2019, accepted November 22, 2019, date of publication December 3, 2019,
date of current version December 23, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2957365

Many-to-One Gesture-to-Command Flexible Mapping Approach for Smart Teaching Interface Interaction

SHICHANG FENG[1], ZHIQUAN FENG[2], AND LIUJUAN CAO[1]
[1]Media Analytics and Computing Laboratory, Department of Artificial Intelligence, School of Informatics, Xiamen University, Xiamen 361005, China
[2]School of Information Science and Engineering, University of Jinan, Jinan 250022, China

Corresponding author: Zhiquan Feng (ise_fengzq@ujn.edu.cn)

This work was supported in part by the National Key Research and Development Program of China under Grant 2018YFB1004901, and in part by the Natural Science Foundation of China under Grant U1705262, Grant 61772443, Grant 61572410, Grant 61802324, and Grant 61702136.

ABSTRACT Mapping between gestures and semantics is inherently challenging, especially in the aspect of defining meaningful gesture semantics that are easy to understand and remember. To address this challenge, we put forward an intent-driven approach in this paper, which is based on users' interaction intent. We implemented a many-to-one flexible mapping between multiple gestures and one semantic object. We also designed and evaluated a domain-specific gestural input technique for classroom use by considering the intent of the user. The main contributions of this work consist of: (1) a mapping between multiple gestures and one semantics and (2) an intelligent and natural interacting interface model for a three-dimensional (3D) user interfaces (UIs). The proposed algorithm is evaluated with five frequently used gestures and applied to the UIs system. The experimental results demonstrate that the gesture-based algorithm performs well and can substantially reduce the memory loads of the user.

INDEX TERMS Memory load, flexible mapping, interaction intention, gesture command, smart classroom interface.

I. INTRODUCTION

Human hand gestures provide an important means for nonverbal interaction, and touchless hand-based gestural interaction provides users new ways of interacting with computers. A gesture-based smart classroom interface (SCI) system enables a teacher in a classroom to directly manipulate lecture materials that are shown on a projected display. Teachers can rotate, scale and translate the selected objects on the screens of the SCI system via free-hand gesture input. This type of manipulation supports operations at a far distance anywhere in the classroom instead of only close to the projected display. To design an effective SCI, most approaches focus on building the most commonly used and natural gesture set and on mapping a gesture to a fixed command. In contrast, we focus on the recognition reliability for the gestural input systems. Therefore, two basic problems are highlighted: users cannot freely choose the gestures, and users do not have free choice regarding the semantic meaning. In most cases, the issue of semantic meaning is not addressed because the systems are stopped at the recognition stage. One challenge in designing effective gestural input is to distinguish the gestures that are expected by the system among all gestures from the users. It is difficult to create, maintain, and modify gestural input with a reliable recognition rate and for users to easily remember the commands. In this work, we designed and evaluated domain-specific gestural input techniques for classroom use by considering the object attributes and the user intent. Current gesture UI (User Interface) imposes a memory load on the teachers memorizing gesture functions for manipulation, which distracts their attention from the content they operate. To address all these issues together, this paper assumes that the mapping between gestures and semantics is flexible. A flexible mapping (F-M) from multiple gestures to a single semantic object is designed and evaluated on common gesture groups. The results demonstrate that the proposed method substantially reduces the user's memory load and outperforms state-of-the-art approaches.

The associate editor coordinating the review of this manuscript and approving it for publication was Shuihua Wang.

II. RELATED WORKS

Human-computer interaction has regained popularity under the development of interaction techniques of devices, such as

tablet PCs, smartphones, and even smart houses [1]. Touchless hand-based gestural interaction provides users new ways of interacting with computers [2], [3]. During gestural interaction, several key points are addressed: the performance of the gesture recognizer, the approaches for defining the mapping between gestures and commands, and the user workload. However, gestures are error-prone, according to Elin [4]; for interacting with the whiteboard, it is most important to be unselfconscious (so as not to draw the attention of the participants from their interactions with one another) and to be fluid (to allow unhindered expression of ideas).

To address the issues that are discussed above, gesture recognizers are designed to distinguish the gestures. Gestures can be classified as hand posture (static) [5] and hand movement (dynamic) gestures [6]. Pisharady and Saerbeck [7] reviewed vision-based hand gesture recognition algorithms that were reported in the last two decades. The methods that use RGB and RGB-D cameras are reviewed with quantitative and qualitative comparisons of algorithms. Ding *et al.* [8] proposed an automatic feature extraction method for similar gesture recognition that overcomes the confusion that arises among similar gestures. Liu *et al.* [9] presented a novel method for human-computer interaction that is based on finger motion detection, which can accurately identify the characteristic quantities of the marked figure and realize cursor movement and mouse clicking. Zeng *et al.* [10] presented an HCI system and discuss its applications in medical assistance. The hand gesture vocabulary in the system consists of five key hand postures and three compound states, and its design strategy covers the minimal hand motions, distraction detection and user-friendly design. Recently, hand posture and gesture recognitions have utilized neural network approaches [11]. A finger detection algorithm [12] was proposed for detecting extensional and flexional fingers via salient hand edges, which are extracted based on the parallel edge characteristics, and the angular projection is centered on the wrist position for obtaining the angle, orientation, and length of each finger. This method can directly extract high-level hand features and estimate hand poses in real time. Raheja *et al.* [13] demonstrated the use of two learning techniques, namely, dynamic time warping (DTW) and the hidden Markov model (HMM), and compared them for real-time implementations. Their experimental results demonstrated that both the DTW and the HMM approaches realize high classification rates of approximately 90% and that DTW outperforms the HMM-based approach when time is constrained. An effective method was proposed in [14] for encoding a joint's trajectories to JTMs (joint trajectory maps), where the motion information can be encoded into texture patterns, and the convolutional neural networks are used to exploit the discriminative features for real-time hand gesture recognition for the MSRC-12 Kinect gesture dataset (MSRC-12). Recently, Sang *et al.* [15] used micro dynamic hand gestures for recognition to realize human–computer interaction. They proposed a state-transition-based HMM method and realized a comparable classification accuracy of 89.38%. Furthermore, to effectively recognize gesture trajectories, Ibaez *et al.* [16] encoded the movements of the joints of the hand during a hand gesture as sequences of characters by utilizing approximate string matching with a Kinect input device. Recently, Liu *et al.* [17] constructed a new dataset of complex hand activities and made it publicly available, in which the latent structures of complex activities are shaped by constructing probabilistic interval-based networks with temporal dependencies.

Gesture is a 'meta-stroke' interpreted as a command. Kita *et al.* [18] proposed a syntagmatic rule system based on inter-coder reliability of segmentation and identification of movement phase and provided useful guidelines for segmenting gestures into phases, whereas they did not take gesture semantics into consideration. In fact, most methods define the mapping between gestures and commands based on the frequency ratio. According to this way, the users are asked to derive a gesture for each command. Once the gestures are collected from the users, similar gestures for each command are gathered together based on the physical shapes and motions of the gestures. Then, one of the gestures with high frequency for each command is selected. But this method is likely to neglect the meaningful gestures due to their low frequency. Hence, Choi et al. [19] developed and verified two hypotheses: (a) users may change their selection after observing other gestures and (b) a gesture that is derived from only a few users might be a higher performing gesture. Their experimental results also demonstrate that the frequency could not guarantee the selected gestures because the users have only limited sets of gestures in their minds in the first step. Wang *et al.* [20] mapped between nine gestures and commands; for example, 'MOVE' is defined by a single finger movement and 'SELECTION' is defined by changing the number of fingers from one to two. These approaches obey to the rule that the mapping between gesture and command is one-to-one.

The widely used hand gestures are limited to a carefully selected vocabulary of symbolic gestures that are mostly used for issuing commands [20]. These methods require users to remember the vocabularies and the responding gestures, thereby leading to heavy loads on the users. Rempel *et al.* [21] studied 24 professional sign language interpreters who reported discomfort during using hand gestures which were associated with 47 characters and words and 33 hand postures. To reduce the memory loads of the users and to improve recognition speed of the system, many researchers propose algorithms and approaches for various application scenarios. Katsuragaw *et al.* [22] examined the effects of bi-level thresholding on the workload and acceptance of end-users. Researchers [23] conducted a guessability study eliciting end-user motion gestures for invoking commands on a smartphone device and demonstrated that consensus exists among users on parameters of movement and on mappings of motion gestures onto commands. They develop a taxonomy for motion gestures and specify an end-user-inspired motion gesture set. For the in-vehicle environment, new interaction techniques are investigated. They aim

at facilitating interaction with infotainment systems while driving. The authors in [24] suggest utilizing the steering wheel as an additional interaction surface. Lai [25] presented low-complexity algorithms and gestures for reducing the gesture recognition complexity, and their experimental results demonstrated that the proposed gesture-based interaction is more suitable for controlling real-time computer systems. A new event-driven service-oriented framework, namely, GS-CPE was proposed [26] for personalized gesture recognition. In dialogue, repeated references contain fewer gestures than initial references. In work [27], the researchers described three experiments studying the extent to which gesture reduction is comparable to other forms of linguistic reduction. Jeanne *et al.* [28] used a visual metaphor to guide trainees' gestures by showing trajectory errors instead of showing the path to follow. In recent years, interactive training has been utilized to realize a user-independent application via on-line supervised training. For example, the work [29] introduces an on-line training method that is embedded into the recognition process, is interactively controlled by the user, and adapts to his/her gestures. An algorithm for flexible mapping between gestures and semantics is presented for the first time for reducing the cognitive and operating loads of the user by using gesture commands [30]. With the same objective regarding the relationship between semantics and actions, Eshuis *et al.* [31] introduced an approach for automatically and flexibly constructing complex, executable compositions of semantic services by deriving the links between semantics and specifying data dependencies among the services. According to Kok K. [32], many current gesture functional classification systems are rigid and implicitly assume that gestures perform only one function at any time, and they are designed to be open to the potential for complex multifunctionality of gestural expression and have realized convergence between ecological and experimental views on gesture functionality. Their work exploits the observation that in the same situational context, there exists a many-to-one mapping between different gestures and the same semantics. Unfortunately, they fail to provide a mapping from different gestures to the same semantics.

To the best of our knowledge, few studies have been done on how to map many gestures to one command or semantics in the same situational context. In this work, we focus on the issues of recognition reliability for the gestural input systems, in other word, the issue of semantic meaning. In contrast to the state-of-the-art algorithms, this paper assumes that the mapping between multiple gestures and single command is flexible and attempts to investigate the flexible mapping between gestures and semantics.

III. BASIC DEFINITIONS FOR MAPPING BETWEEN MULTIPLE GESTURES AND SINGLE COMMAND

The basic application objective is the developing an SCI system in which the teachers can rotate, scale and translate the selected objects on the screens through free-hand gesture input. Moreover, the teachers can freely choose the gestures in SCI system, and the teachers have free choice regarding the semantic meaning.

A. FLEXIBLE MAPPING

The concept of flexible mapping (F-M) is proposed for addressing the problem of requiring users to memorize gesture commands in the same interaction scenario. A single semantic object can be expressed with multiple gestures, that is to say, in the same scenario there is a many-to-one mapping between multiple gestures and single semantic object, which is called a flexible mapping from many gestures to one semantic object.

B. CONTEXT-RELATED INTENT

In the proposed algorithm, all object semantic operations are stored in a set $\boldsymbol{F}$ and all gesture semantics (user intent) are stored in a set $\boldsymbol{G}$. To determine the objective of the user's gesture input (represented the actual user's intent from $\boldsymbol{G}$), the proposed algorithm determines the operating functions of the current object and calculates the corresponding gesture semantic set $\boldsymbol{F}$ based on the usage context. The user's intent or the final gesture semantics must be in the intersection of $\boldsymbol{G}$ and $\boldsymbol{F}$, or $\boldsymbol{\Omega}$.

We define the following terms (**Definition *D_1***):

D_1*: The interaction intent x of the user's gesture is in the intersection of the semantic set* $\boldsymbol{G}$ *of this gesture and the semantic set* $\boldsymbol{F}$ *that is required by the functions of the current operation object, or* $\boldsymbol{\Omega} \in \boldsymbol{G} \cap \boldsymbol{F}$.

For example, suppose that the semantic set of a gesture g, namely, 'Fetching hand', is the set $\boldsymbol{G} = \{$'fetch', 'scale-down'$\}$ and that the semantic set that corresponds to the functions of the object is $\boldsymbol{F} = \{$'scale-down', 'rotate', 'scale-up'$\}$. Then, the final gesture will be $\boldsymbol{\Omega} \in \boldsymbol{G} \cap \boldsymbol{F} = \{$'scale-down'$\}$.

C. GESTURE GROUP (GG)

If there is only one semantic object in the intersection set $\boldsymbol{\Omega}$ ($\boldsymbol{\Omega} \in \boldsymbol{G} \cap \boldsymbol{F}$), then this semantic object is the user's interaction semantic object. When more than one semantic object exists in the intersection set $\boldsymbol{\Omega}$, the users can choose another gesture for issuing the same command. This decision is made based on the observation and analysis of many users' behavioral models.

Assuming that the alternative gesture is g_1 and that its semantic set is $\boldsymbol{G_1}$, in which both $\boldsymbol{G_1}$ and $\boldsymbol{\Omega}$ reflect the user's interaction intent. Consequently, the actual interaction intent must belong to the intersection of the two, and $||\ \boldsymbol{\Omega} \cap \boldsymbol{G_1}\ || \leq ||\ \boldsymbol{\Omega}||$. This process is repeated. Thus, the number of elements in the intersection set decreases. Eventually, the actual gesture semantics will be uniquely determined. Therefore, **Definition *D_2*** is obtained:

D_2*: The interaction intent x of the user's current gesture g must be in the semantic intersection of all alternative gestures in* $\boldsymbol{GG} = \{g_1, g_2, \ldots, g_m\}$, *or* $x \in \cap s_i$, *where* $s_i =$ *semanticset*(g_i). *Here, semanticset(i) refers to the semantic set of gesture i.*

Definition ***D_2*** demonstrates the feasibility of constructing ***GG*** in which alternative gestures are formed for each semantic object in ***GG***.

For example, suppose that we have gestures g_1 = 'grab with fingers 1 and 2', g_2 = 'grab with fingers 1, 2 and 3', and g_3 = 'grab with all 5 fingers'. Their corresponding semantic sets are as follows: s_1 = {'zoom an object', 'close window', 'reduce the volume of sound'}, s_2 = {'zoom an object', 'reduce the illumination of display'}, and s_3 = {'lift an object', 'zoom an object'}. Therefore, $x \in \cap s_k$ ={'zoom an object'}.

D. NECESSARY CONDITIONS FOR SHAPING GG

Because the gestures in the same ***GG*** are related to a common semantic object, they should share a common feature. To investigate the common features, a survey was conducted. Fifty college students (with an average age of 23 years old) were invited to answer the questionnaire. Based on the survey results, the alternative gestures that most participants selected are regarded as the natural gestures for specified semantics and are grouped into a single ***GG***.

All gestures in the same ***GG*** share the same movement direction or trajectory. In addition, the users' operational experiences in expressing the same or similar semantics under various interaction scenarios also affect their gesture selection.

Based on the above experiment, Definition ***D_3*** is formulated:

D_3*: The condition for constructing a* ***GG*** *of a semantic object is that given the same situational context of the interaction, the gestures in a* ***GG*** *that share the same semantics always have the same direction or trajectory of gestures, and the necessary condition for the gesture set* $\{g_1, g_2, \ldots, g_m\}$ *to form a* ***GG*** *is* $\cap Tra\,(i) \neq \emptyset$ *or* $\cap Dir\,(i) \neq \emptyset$ *for all gestures* $g_i \in$ ***GG***. *Here, Tra (i) and Dir (i) refer to trajectory and direction of gesture* $i \in$ ***GG***.

Definition ***D_3*** expresses that the ***GG*** is constructive.

In the above experiment, the following interesting observations are made:

O_1: *In a camera-based gesture-oriented human-computer interface, if the user forgets the gesture that can express the requested semantics or his/her gesture operation does not work, he/she prefers to try another gesture (alternative gesture) for expressing the same interaction intent.*

O_2: *A* ***GG*** *can also be constructed based on the user's life experience or the user's operating experience on smart devices, such as intelligent TVs, smartphones or computers, which can be transferred to the construction of the* ***GG***.

IV. GESTURE-SEMANTIC FLEXIBLE MAPPING (F-M) ALGORITHM

A. FLEXIBLE MAPPING ALGORITHM

The F-M algorithm is composed of four parts and operates as follows (FIGURE 1): The first part is error recovery and self-repair for motor gestures. After a motor gesture has been recognized, error recognition is performed and errors are repaired with our proposed algorithm. The second part is context extraction and representation, with which the gesture semantics are narrowed down to a smaller set. The third part is error recovery and self-repair for semantic gestures, and the last part is interaction application, in which the recognized gesture semantics are used to manipulate the object in the interactive scene.

We have implemented a novel motor gesture recognition method and have used a set of mis-recognition rules to automatically correct errors. The semantics of a motor gesture g are recognized from the Motor Gesture (MG) dataset. For a motor gesture, there are always many semantics. To decrease the mis-recognition probability of the semantics, the context of the current object $\boldsymbol{O}$ to be manipulated is extracted and represented using the Semantics Gesture (SG) dataset. After the object semantic set $\boldsymbol{F}$ and the gesture semantic set $\boldsymbol{G}$ are obtained, we can narrow the user's context-related intent via $\boldsymbol{\Omega} \leftarrow \boldsymbol{G} \cap \boldsymbol{F}$ according to Definition ***D_1***. If $||\,\boldsymbol{\Omega}|| = 1$ (only one semantic object exists), then the object $\boldsymbol{O}$ is manipulated with this unique semantic object in $\boldsymbol{\Omega}$. Otherwise, ***GG*** of gesture g is constructed according to Definition ***D_3*** and Observation ***O_2***. According to Observation ***O_1***, alternative gestures in ***GG*** are used to express the same interaction intent. According to Definition ***D_2***, (a) the user can express the assumptions that the semantic set of the alternative gesture is $\boldsymbol{G_1}$, which has the same operational intent as an alternative gesture, and (b) the algorithm computes $\boldsymbol{\Omega} \leftarrow \boldsymbol{\Omega} \cap \boldsymbol{G_1}$. Steps (a) and (b) are repeated until $||\boldsymbol{\Omega}|| = 1$. If $||\boldsymbol{\Omega}|| = 1$, the semantics of the gesture g are predefined for the object $\boldsymbol{O}$.

According to Definition ***D_2***, the so-called alternative gesture is defined as follows: if the operating result of the current gesture is not usable, the user can continue the operation until another gesture is obtained for performing the same interaction. All alternative gestures constitute a gesture group with a common semantic object. The current set of gesture semantics is repeatedly intersected with the semantic set of alternative gestures to obtain a smaller set of semantics until a unique semantic object is obtained.

The MG dataset is composed of two parts: gesture number and features. The SG dataset is composed of four parts: gesture number, object number, functions of the object and gesture semantics of the functions.

According to the above analysis, the proposed F-M algorithm is described in detail as Algorithm 1.

B. EXAMPLE

Assuming that the functions that are applicable to the current object are scale-up, scale-down, and grasp, that the user's current gesture g_1 is 'grasping with five fingers' and that the semantic set of gesture g_1 is $\boldsymbol{G}$ = {displace, scale-down, split, grasp}, how does the computer determine which of the four operations the user wants to apply to the current object? First, the functions of the current object are converted to gesture semantics through retrieval of the *MG* dataset and

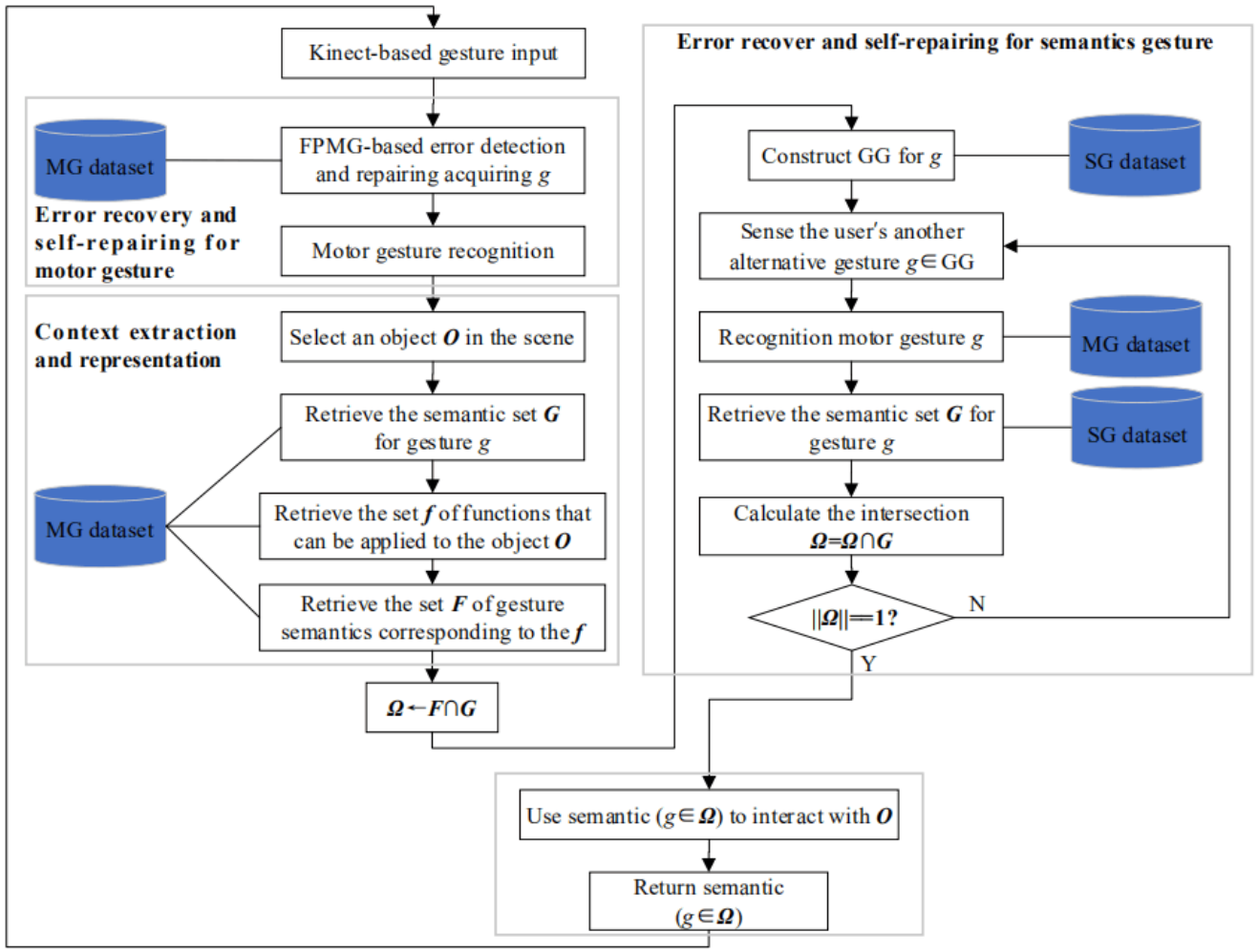


FIGURE 1. Overview of the four parts in the proposed F-M algorithm: (1) Error recovery and self-repairing for motor gestures, (2) Context extraction and representation, (3) Error recovery and self-repairing for semantics gestures, and (4) Interactive application. The two datasets are designed: Motor Gesture (MG), Semantics Gesture (SG).

the *SG* dataset, and the gesture semantics that are required by the current object are further assumed to be $\boldsymbol{F} = \{$scale-up, scale-down, rotate, grasp, release$\}$. Second, the intersection of sets $\boldsymbol{G}$ and $\boldsymbol{F}$ is calculated to be $\{$scale-down, grasp$\}$. In this case, nothing will be done because $\|\boldsymbol{\Omega}\| \neq 1$. After the user observes that nothing happens to the object as s/he makes gesture g_1, s/he will naturally try another gesture g_2 (which is defined as 'pinch with thumb and index fingers') to manipulate the object. Suppose the semantic set of gesture g_2 consists of operations rotate, scale-down, and descend. The user's interaction intention should be within the scope of the intersection of the set $\{$scale-down, grasp$\}$ and the set $\{$rotate, scale-down, descend$\}$, and the user's operational intention will be ''scale down''. As such, the semantic 'scale-down' is exerted on the selected object, which causes it to scale down, as shown in Figure 2.

The proposed F-M differs from FM [30]. The former provides many-to-one gesture-to-command mapping while the latter introduces a flexible mapping from one gesture to multiple semantics in the same situational context.

V. GESTURE RECOGNITION

A. GESTURE DATABASE CONSTRUCTION

A polar coordinate system is utilized with the current gesture feature point as the pole to divide the entire plane evenly in E directions, whereas the radius is evenly divided into D parts, so that the whole plane space is divided into U natural regions. Along the same circle, the area of each region is the same. Then, the number of points in array sumpoints[M] that fall in each region is counted. Via this approach, with the feature point of the current gesture as the pole and the maximum distance, namely, max distance, as the radius, the plane space is divided into U regions. The U attribute values of the i$^{\text{th}}$ gesture will constitute a sequence $(a_{i,1}, a_{i,2}, \ldots, a_{i,U})$. Therefore, the shape of the image can be described by an $N \times U$ shape matrix $(a_{i,j})$ (U = 60 in this paper). In $a_{i,j}$, i is the i_{th} feature point and j is the j_{th} region out of U regions. If a coordinate system is established with the i_{th} feature point as the pole, then the number of points that fall within the j_{th} region is $a_{i,j}$. The value of N is the total number of feature points. This matrix represents the contextual features

Algorithm 1 F-M Algorithm

1. Recognition, Error Recovery and Self-Repair for Motor Gestures:
(1) Input a gesture from the Kinect device.
(2) The gesture images are segmented based on depth information from the background.
(3) Recognize the gesture using the motor gesture recognition algorithm and obtain the gesture g.
(4) IF the gesture g' is mistaken for g, THEN $g \leftarrow g$'.
2. Context Extraction and Representation:
(1) The user selects the operation object O in the scene using the gesture.
(2) Form GG for g by retrieving the dataset SG.
(3) Retrieve the function set f of the object O from the dataset SG.
(4) Retrieve the gesture semantic set F of the set f.
3. Recognition, Error Recovery and Self-Repair for Semantic Gestures
(1) $\mathbf{\Omega} \leftarrow F$
(2) WHILE ($\|\mathbf{\Omega}\| \neq 1$) DO
(a) Use step 1, namely, recognition, error recovery and self-repair for motor gestures, to acquire gesture $g \in GG$.
(b) Retrieve the semantic set G of gesture g from dataset MG.
(c) $\mathbf{\Omega} \leftarrow \mathbf{\Omega} \cap G$.
4. Interaction Application:
(1) Apply the semantics in $\mathbf{\Omega}$ to object O.
5. If NOT all interactive tasks are accomplished, then Goto step 1.

of the image shape. The values of this matrix are stored in a two-dimensional array: FeatureNo[N][U]. If the x- and y-coordinates of these gesture feature points are both 0, then all U attribute values of the feature points are set to 0.

Based on the strategy discussed above, the detailed algorithm for constructing the gesture database is as Algorithm 2.

B. GESTURE RECOGNITION ALGORITHM THAT IS BASED ON THE SHAPE CONTEXTUAL FEATURES

The main objective of gesture recognition is to determine the representative feature and the image similarity descriptor. In this paper, the feature is extracted based on the shape context and the distance between features, which is denoted as χ^2, is utilized as the image similarity descriptor. The gesture recognition algorithm is as follows.

Gesture recognition algorithm based on shape contextual features Algorithm 3.

C. IMPLICIT INPUT AND DETECTION OF 'CONFIRMING' AND 'CORRECTING' BEHAVIORS

'Confirming' and 'correcting' are the two basic operations in the proposed F-M algorithm, which require the user to respond to the computer's feedback. If the computer interprets the user's gesture semantics or interaction intent correctly, then the user can use a gesture that represents the 'confirming' semantics to respond; otherwise, the user can use the gesture with the 'correcting' semantics to respond. We start from the user's behavioral models to solve this problem. According to multiple observations, when completing gesture input, users often maintain the last posture of the gesture for a period if the computer's feedback is consistent with the user's intent [33], [34]. If the current gesture has been completed and the gesture posture remains for a period, then the user is 'confirming' the computer's interpretation. It is also observed that whenever the user disapproves of the semantics that are interpreted by the computer, s/he subconsciously moves the gesture toward the left chest, which is the basis for the perception that the user wants to 'correct' the semantics.

Algorithm 2 Construct the Gesture Database

1. Read the video streams of each gesture, and acquire an image for each gesture g.
2. Count the number of gesture points in each image. Traverse the entire image (black is considered the background and the gesture points are other points), and record the coordinates and the number of gesture points.
3. Determine the center of gravity by calculating the gesture points and compute the maximum distance between the center of gravity and the gesture points.
4. Regard the calculated maximum distance as the maximum radius and divide this radius evenly into D parts for determining D circles. Count the number of gesture points that fall into each circle and calculate the center of each circle, which is used as a feature point of the gesture.
5. Extract the shape contextual features based on the gesture feature points and gesture points. The gesture points at this time include the gesture points and the gesture feature points that were counted previously.
6. Write the obtained shape contextual features into a text file as the gesture database.

D. FPMG-BASED ERROR DETECTION AND REPAIR

First, based on the conclusion above, we set up the feature pool for mis-recognized gestures (FPMG). This feature pool consists of two matrices: The first is the matrix $\boldsymbol{P}$ for the rate of false acceptance, and each element $p_{i,j}$ of this matrix constructs the model of the probability distribution for gesture i being mis-recognized as gesture j. The other is the matrix E for the error between the features of the target gesture and the features of the misrecognized gesture, and each element e_{ij} of this matrix constructs a model for the distribution of the error between the feature of gesture i and the feature of mis-recognized gesture j. Here, the feature represents the eigenvector in the final pooling layer of the CNN model that has been trained. The error between the features of the

Algorithm 3 Shape-Context-Feature-Based Gesture Recognition Algorithm

1. Take v consecutive image frames from the video stream as the gesture images to be recognized.
2. Conduct the following operation on each image frame to be recognized:
(1) Count the number of gesture points in each frame. The procedures involve traversing the entire image, where a pixel is considered background if the pixel value is black and a gesture point otherwise. At the same time, record the coordinates and the number of gesture points.
(2) If the number of gesture points in a frame is 0, then this frame is discarded and no additional calculations will be conducted on this frame. If the number of gesture points is not 0, calculate the coordinates of the gravity center using the gesture points and calculate the maximum distance between the gravity center and the gesture points.
(3) Use the maximum distance as the maximum radius and divide this radius into D parts for determining D concentric circles. Then, count the center of each circle, which is utilized as a gesture feature point.
(4) Extract the shape contextual features, the gesture feature points, and the gesture points.
(5) Calculate the distance between the shape contextual feature of each gesture feature point and that of all gestures in the gesture database, namely, χ^2.

$$C_{ij} = \frac{1}{2}\sum_{k=1}^{K}\frac{\left[h_i(k) - h_j(k)\right]^2}{h_i(k) + h_j(k)} \tag{1}$$

where $h_i(k)$ is the shape contextual feature value of the i_{th} gesture feature point to be recognized and $h_j(k)$ is the shape contextual feature value of the j_{th} gesture feature point in the user gesture database. (6) For gesture j, the matching cost of the k_{th} image is:

$$d_{jk} = \sum_{i=1}^{n} C_{ij}^{k} \tag{2}$$

where n is the number of gesture feature points and is the number of gestures in the gesture database.
3. The matching cost of gesture j is:

$$Cost^{(j)} = \min_{k}\left\{d_{jk}\right\} \tag{3}$$

4. Return the gesture with the lowest matching cost:

$$\min_{j}\left\{Cost^{(j)}\right\} < \lambda \tag{4}$$

in which λ is the non-negative experience threshold.

target gesture and the features of the mis-recognized gesture is defined as:

$$e_{ij} = \|Feature\_Map(i) - Feature\_Map(j)\| \tag{5}$$

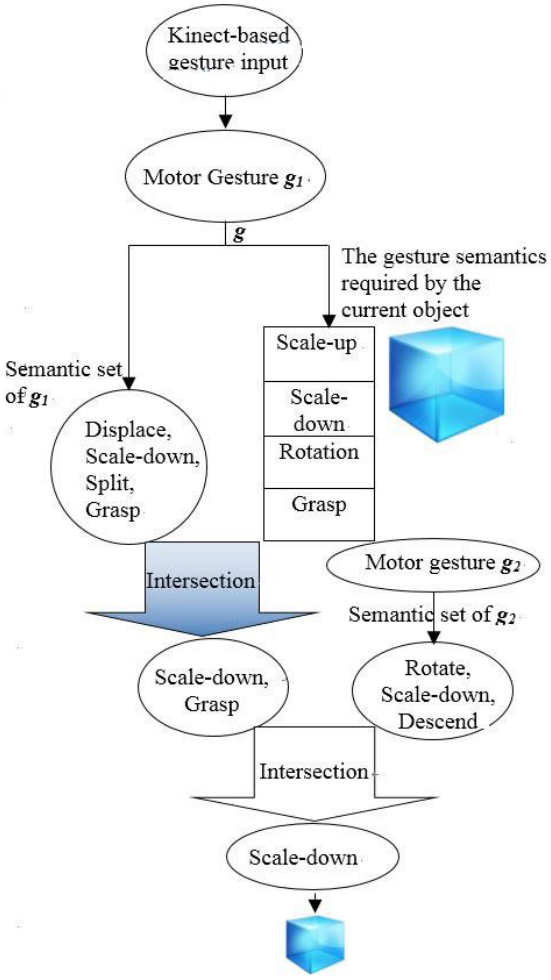


FIGURE 2. Example of the F-M algorithm. The user uses natural gestures to scale-down the 3D (three dimensional) objects in the interaction scene.

where Feature_Map(i) represents the image eigenvector of the last pooling layer in the CNN model of gesture i, and the eigenvector is a one-dimensional vector of the pixels of the final pooling layer in order from top to bottom and from left to right.

We can detect a false gesture and correct it with FPMG, and this algorithm is described as follows:

Algorithm 4 Detection and Correction of Error Gestures

Input: (a) gesture i; (b) gesture j.
Output: (a) identification the correction of gesture j; (b) return the correct value if gesture j is incorrect.
(1) Calculate the feature error e between the gesture image input and gesture
image j;
(2) Find $e_{i,j}$ in matrix E that most closely approximates e;
(3) If
$\|e_{ij}\| < \sigma_1$ and $\|p_{ij}\| < \sigma_2$ then return (TRUE, j)
else
return (FALSE, i), and return to step (2).
end if

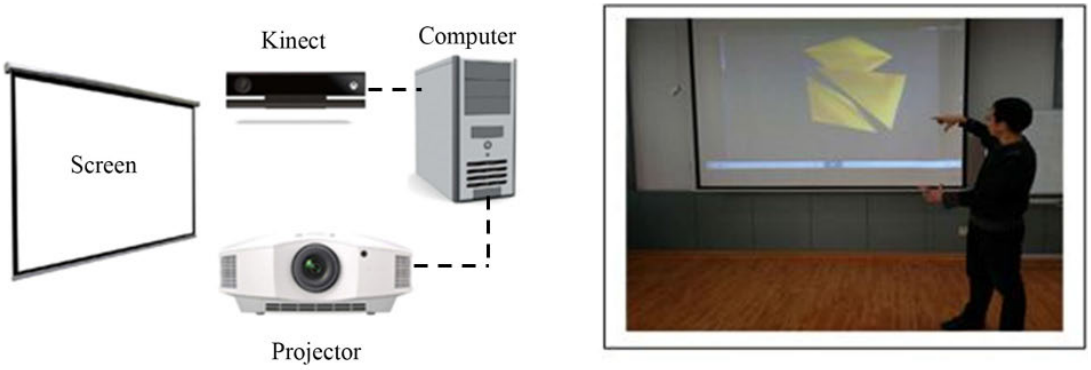


FIGURE 3. In our geometry teaching system, a teacher manipulates an object on the big screen using the F-M algorithm. (a) The hardware structure of the SCI system. (b) The teacher is rotating and scaling three pyramids for students.

VI. EXPERIMENTAL RESULTS AND ANALYSIS

A. THE APPLICATION OF F-M IN GEOMETRY TEACHING SYSTEM

1) IMPLEMENTATION

Our gesture-based smart classroom interface (SCI) is composed of a computer that is running the F-M-based gesture interface, a projection screen, a projector and a Kinect. The computer used in this study was equipped with Intel® Core™ 2 Quad CPU 2.66 GHz processor and 4 GB memory. The 3D geometrical objects on the big screen are manipulated with free gestures at a distance of 2-2.5 meters from the Kinect (FIGURE 3).

When the user scales up an object, s/he performs the gesture 'release thumb and first finger' (g_1). If there is no response on the screen, s/he performs the next gesture, namely, 'release thumb, first finger and middle finger' (g_2). If there is still no response on the screen, s/he further performs another gesture, namely, 'release five fingers' (g_3). This process continues until feedback is provided on the screen. Here, $\boldsymbol{GG} = \{g_1, g_2, g_3\}$ and the semantic object = 'Scale-up the selected object'. This process is independent of the sequence of g_1, g_2 and g_3, which lowers the user's memory workload. Hence, the user need not remember gestures and their corresponding semantics. Thus, s/he can focus most of his/her attention on the main task.

Five semantics groups are defined and experimented on for F-M (as listed in TABLE 1).

The participants were asked to perform operations in Table 1 using gestures from GG. If the semantic object of a user's gesture was not in the set of gesture semantics, the interface would not respond and would display 'Invalid gesture'. If the user wanted to scale-down the current scene, s/he could use the gesture 'fetch with 5 fingers'; if the user forgot the gesture, s/he could use the gesture 'fetch with fingers 1 and 2' or the gesture 'fetch with fingers 1, 2 and 3'. If an error occurred, the system would restore the scene back to the status prior to the current operation. Then, the user was allowed to use an alternative gesture to complete the operation.

TABLE 1. Definition of the five semantics groups.

Semantics	GGs
Scale-up	{release 5 fingers; release fingers 1 and 2; release fingers 1, 2 and 3}
Scale-down	{fetch with 5 fingers; fetch with fingers 1 and 2, fetch with fingers 1, 2 and 3}
Move leftward	{move hand leftward, move hand downward}
Move rightward	{move hand rightward, move hand upward}
Turn page	{move hand rightward, move hand downward}

'Release 5 fingers' refers to expanding five fingers of a fist outward into an open hand, 'Release fingers 1 and 2' refers to expanding the thumb and index finger of a fist; and 'Release fingers 1, 2, and 3 refers to expanding the thumb, index finger, and middle finger of a fist.

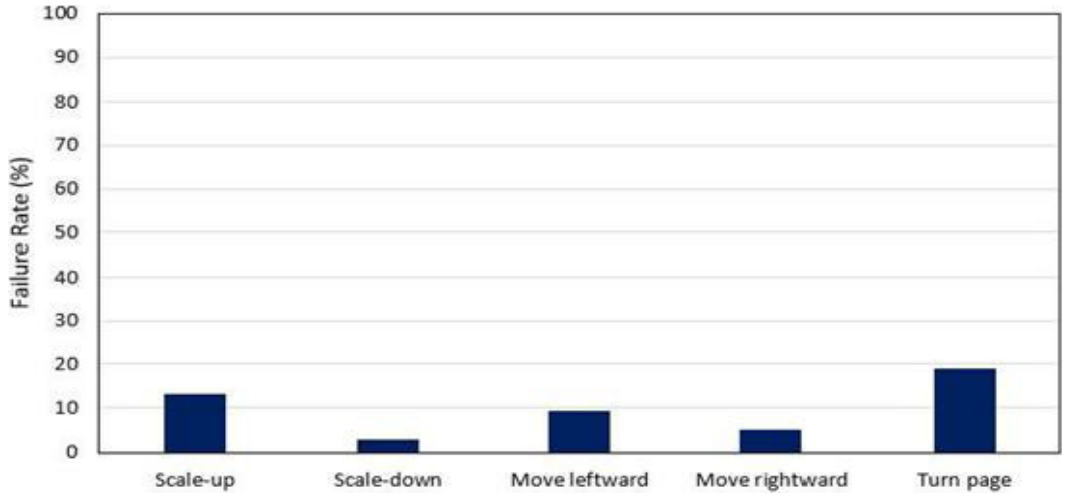
2) RESULTS AND ANALYSIS

The experimental results demonstrate that the users did not need to deliberately memorize gesture commands to successfully perform all functions in SCI. The users often chose gestures for control that were based on their operational experiences. For example, a user often attempts to use the 'release 5 fingers' gesture to scale-up the object. In addition, when a gesture failed, alternative gestures were explored until the operation succeeded.

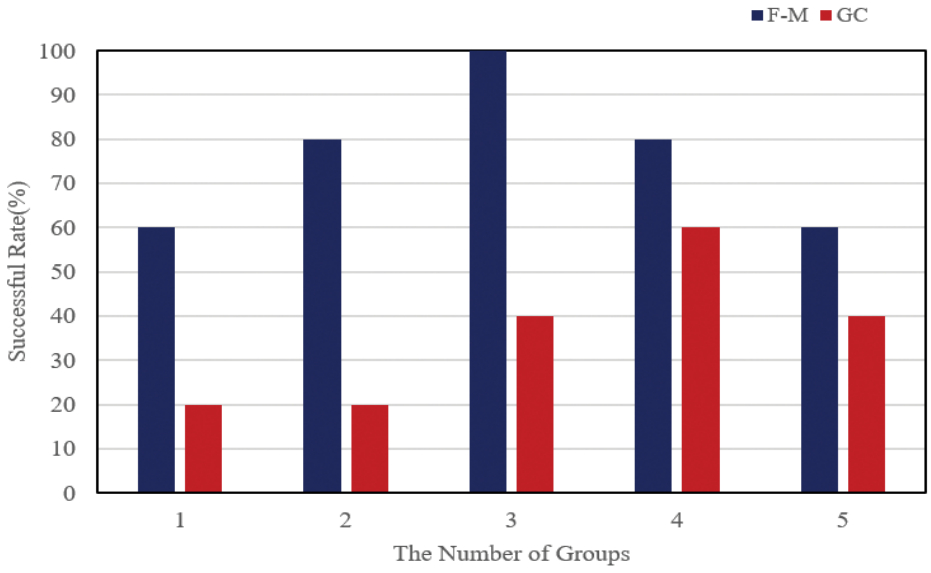
To evaluate the performance of the proposed F-M algorithm, a series of comparison experiments were conducted between F-M-based gesture recognition and the approach based on one-to-one mapping between gestures and commands (named GC) [33]. For GC, a gesture is mapped to one command and a command is mapped to one gesture compulsorily in the same context.

a: COMPARISON OF THE SUCCESS RATES OF F-M AND GC WITH UNTRAINED USERS

With a novice user group, we deem an operation successful if the results of gesture command execution accord with the expectation or intent of the user. If out of N attempts by the user to operate the application system there are n single operations that can successfully complete all the functions,



(a)



(b)

FIGURE 4. Averaged performance statistics of the F-M algorithm for the 100 untrained subjects. (a) Failure rate statistics of single operations for F-M. (b) A group of comparative experiments: Overall success rate comparison between F-M and GC.

then the success rate is defined as:

$$\alpha = n/N \tag{6}$$

One hundred college students volunteered for the study. They performed the five operations in Table 1. According to the results, the operation failed to be performed 13, 3, 9, 5, and 19 times (FIGURE 4(a)). The five basic operations failed 10 times on average. Subsequently, we invited the subjects to form an experimental group for five experiments on the success rates of operations with F-M algorithms and GC. Each test subject independently performed each function of the F-M algorithm interface (a total of five functions). For the F-M algorithm, all functions were successfully performed 3, 4, 5, 4, and 3 times on average; for the GC algorithm, all functions were successfully performed 1, 1, 2, 3, and 2 times (FIGURE 4(b)). In this experiment, the users were not trained to use the F-M or GC prior to the statistical experiments. We found that the users were highly capable of transferring their smartphone operation experience to the new system.

b: TRAINING DURATIONS FOR SKILLED OPERATORS

Ten college students of ages 19−23 volunteered to participate in the experiment. They were trained on the F-M-based system and the results are presented in FIGURE 5. The correction or recognition rate reached 100% after training for 1 minute for those with an average age of 22.7, for 1.5 minutes for those with an average age of 21.5, and for 2 minutes for those with an average age of 19.8. All participants could

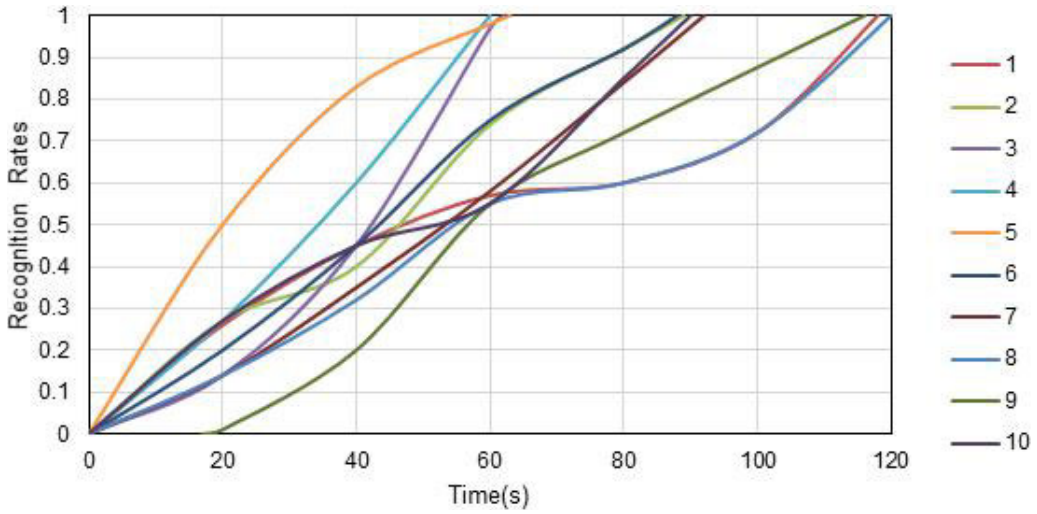


FIGURE 5. Success rate versus training time for five gesture groups (GGs) for 10 college students.

skillfully operate the interface and the older students seemed to be more efficient at learning our system.

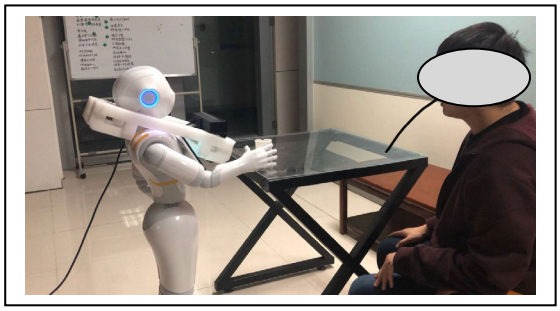
In summary, the F-M algorithm can reduce the level of effort that is required for completing a task. Moreover, the implicit error detection and correction and the detection of 'confirming' and 'correcting' behaviors in the F-M algorithm effectively reduce the frustration and anxiety of the user that are caused by possible mis-operations.

B. THE APPLICATION OF F-M IN ROBOT DEMONSTRATION TEACHING SYSTEM

Chinese tea art can be displayed in a smart teaching system through human-robot intelligent interaction. In this intelligent teaching system, the robot can respond to human gesture commands and cooperate with human to complete the tea art exhibition. In response to different commands, the robot enables to perform some basic actions, such as lifting the teacup, moving the teacup, and pouring the tea, etc. The main problem of this kind of system is that the robot cannot always correctly recognize the gesture commands or misunderstands the user's intention, therefore interrupting the interaction between human and robot. This paper presents an F-M algorithm which enables the human to issue commands to the robot utilizing gestures formed by daily life experience without memorizing new gesture commands. The experimental results show that the gesture-control system based on the proposed F-M algorithm can always make the robot understand the user's intention correctly, ensuring a smooth interaction between human and robot.

Twenty volunteers of ages 20-25 were invited to participate in the experiment and asked to perform the same semantics "Would you please have a cup of Chinese tea?". Each volunteer used the gesture in $\boldsymbol{GG} = \{$fetch with 5 fingers; fetch with fingers 1 and 2; fetch with fingers 1, 2 and 3$\}$ to control the Pepper Robot to fetch a teacup (Figure 6). The task was repeated 10 times for each volunteer and the number of alternative gestures was counted. Here, alternative gestures

(a)



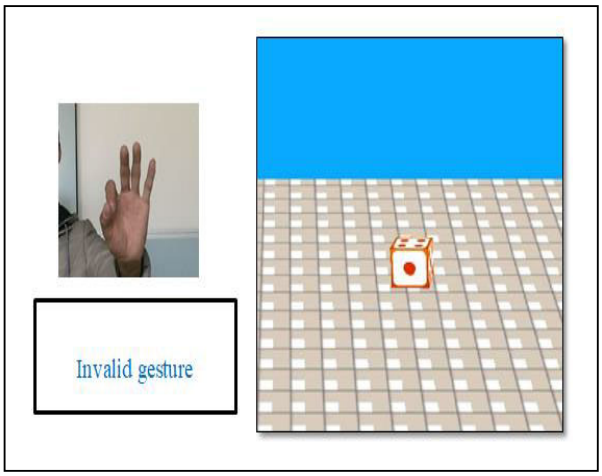
(b)

FIGURE 6. Screens hots of F-M algorithm runs in our Pepper Robot demonstration teaching system. (a) a volunteer was issuing a command using hand gesture. (b) The robot was fetching a teacup for the volunteer.

refer to the gestures in ***GG.*** For example, if the robot failed to respond to the gesture command from a volunteer, the volunteer should use an alternative gesture in ***GG*** to repeat this process until the command is correctly identified by robot. We further defined the ratio of the alternative gestures as follows:

$$\beta_m = r_m / M \tag{7}$$

in which M represents the repeated times of the tasks, and r_m refers to the number of the fact that the alternative gestures



(a)

(b)

(c)

FIGURE 7. Screens hots of F-M algorithm runs in this study. (a)–(b): In the same interaction scenario, the user used an alternative gesture to handle the issue of forgetting gestures. (c): Experimental scenarios.

used during the M repeated tasks is m. The detailed statistical results of r_m for each volunteer are shown in Table 2.

For the twenty volunteers, the β_{m} on average were calculated by the former formula (7), and it turned out that

TABLE 2. The times r_m that the alternative gestures were use. in our f-M-based robot demonstration teaching system.

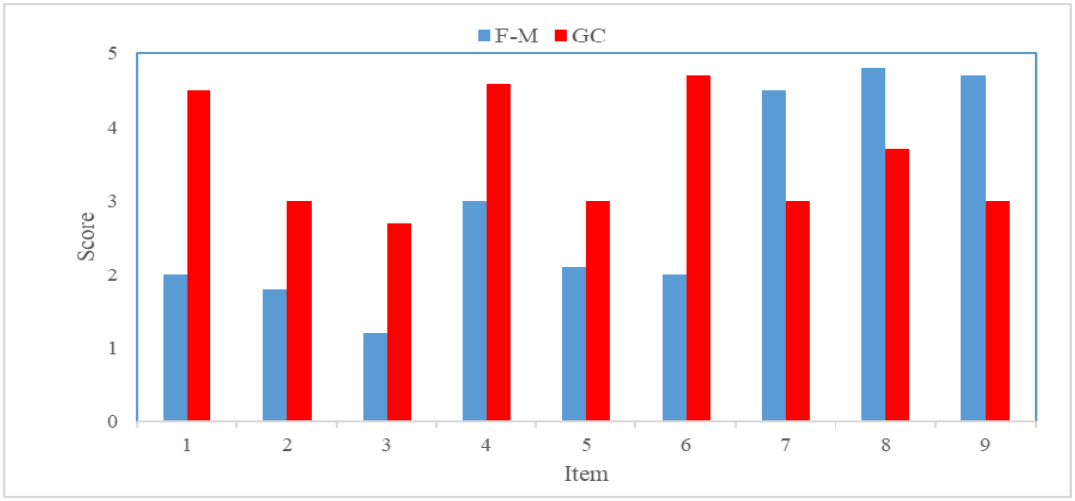
The NO. of the volunteers	r_m when m =0	r_m when m =1	r_m when $m \geq 2$
1	9	1	0
2	6	2	2
3	5	4	1
4	8	1	1
5	3	5	2
6	8	0	2
7	9	0	1
8	10	0	0
9	7	2	1
10	5	5	0
11	5	2	3
12	8	1	1
13	6	3	1
14	4	2	4
15	6	4	0
16	8	2	0
17	8	2	0
18	6	1	3
19	10	0	0
20	5	3	2

$\beta_{\mathrm{m}=0} = 68\%$, $\beta_{\mathrm{m}=1} = 20\%$, $\beta_{\mathrm{m}\geq 2} = 12\%$, which showed a fact that alternative gesture is used with a probability of 32%. The experimental results also demonstrate that the proposed gesture-based F-M algorithm works well in our robot demonstration teaching system and can substantially reduce the memory loads of the user.

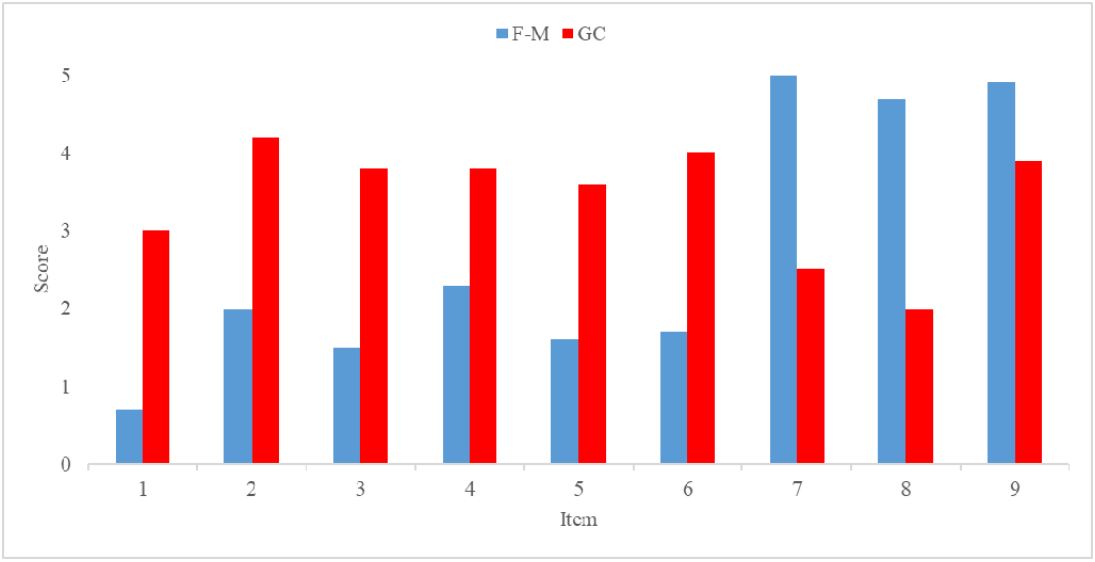
C. USER STUDY

A comparative experiment was conducted to determine whether or not the interface could reduce the cognitive and operation loads. Volunteers were invited to participate in the experiment. Each participant performed the required interaction functions using the F-M and GC algorithms independently and repeated the operation ten times using each algorithm.

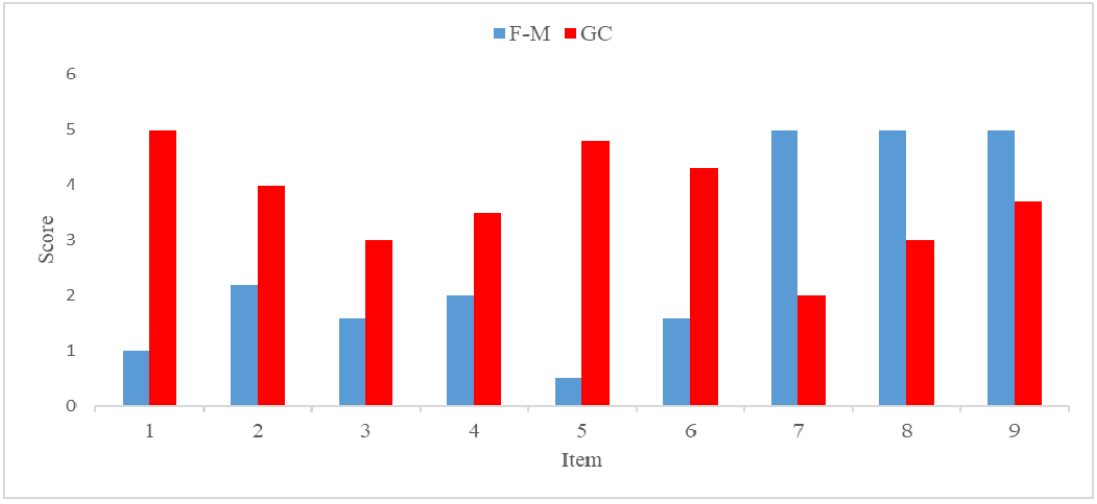
The participants were asked to perform operations, such as zooming in and zooming out of a cube in a 3D scene using natural gestures and were required to use the same type of gesture to perform rotation of, enclosing of, and zooming in on the cube in the same context (Figure 7). If the semantic object of the user's gesture (e.g., "Ok gesture") was not in the set of gesture semantics that were required by the cube's function set, the cube would not respond and the interface would display "Invalid gesture" (Figure 6a). If the user wanted to zoom in on the cube in the current scene,



(a)



(b)



(c)

FIGURE 8. Comparison of the cognitive load and the usability test for F-M and GC. 1-Mental demand. 2-Physical demand. 3–Temporal demand. 4-Performance. 5-Effort. 6-Frustration. 7-Convenience. 8-Joviality. 9-Workability. (a) For group #1. (b) For group #2. (c) For group #3.

TABLE 3. For the three groups, the distribution parameters (μ, δ) of F-M for the nine evaluation indicators are compared with GC in detail.

Group	Method	MD	PD	TD	OP	E	DF	J	C	W
Group #1	**F-M**	(2, 0.43)	(1.8,0.28)	(1.2,0.34)	(3, 1.33)	(2.1,0.29)	(2, 1.07)	(4.5,1.61)	(4.8,0.89)	(4.7,0.76)
	GC	(4.5,1.03)	(3, 1.36)	(2.7,0.82)	(4.6,0.97)	(3, 0.72)	(4.7,1.08)	(3, 1.28)	(3.7,0.88),	(3,0.39)
Group #2	**F-M**	(0.7, .51)	(2, 0.34)	(1.5,0.31)	(2.3,1.46)	(1.6,0.58)	(1.7,0.89)	(5, 1.29)	(4.7, 1.05)	(4.9,1.22)
	GC	(3, 0.98)	(4.2,1.33)	(3.8,1.07)	(3.8,0.44)	(3.6,0.69)	(4, 0,97)	(2.5,0.48)	(2, 0.27)	(3.9,0.882)
Group #3	**F-M**	(1.0,0.81)	(2.2,0.55)	(1.6,0.91)	(2.0,1.22)	(0.5,1.27)	(1.6,1.08)	(5.0,0.88)	(5.0, 1.74)	(5.0, 1.92)
	GC	(5.0,0.48)	(4.0,1.26)	(3.0,0.87)	(3.5,0.84)	(4.8,0.99)	(4.3,0.91)	(2.0,0.86)	(3.0, 0.84)	(3.7, 1.08)

he/she could use the gesture ''open all fingers of a fist''; if the user forgot the gesture, he/she could use the gesture ''release fingers 1 and 2'' to enlarge the cube (Figure 6b). If the gesture was still misunderstood by the system, he/she could use the gesture ''release 1, 2 and 3 fingers''. When an error (in the case of mis-operations, e.g., if the gesture was not in the gesture database or was a misunderstood gesture) occurred, the system restored the scene back to the status prior to the current operation and the user was allowed to use an alternative gesture to complete the operation. The experimental results demonstrated that the user essentially did not need to learn or deliberately memorize any gesture command to successfully achieve all the functions for controlling the cube. We observed that the users often chose to initially use gestures for control based on their experience. For example, the users often attempted to use the ''release 5 fingers'' gesture to enlarge the cube. If this method failed, they would often use another gesture to enlarge the object. In addition, if a gesture failed, alternative gestures would be explored until the operation succeeded.

The volunteers were divided in three groups according to age: Group #1 was composed of the 100 pupils of ages 10-17 (average age 15.3), group #2 was composed of 110 college students of ages 19-24 (average age 22.7), and group #3 was composed of 30 teachers of ages 35-43 (average age 40.9). For each volunteer, the task is to enlarge the cube and to shrink it, without prior training, using F-M and GC. Each volunteer was allowed to attempt the task 10 times.

For each group, the operation was averagely scored using the NASA Task Load Index (NASA-TLX) [35] evaluation indicators on a five-point scale. The NASA-TLX was utilized to quantitatively analyze and evaluate the load on users in terms of mental demand (**MD**), physical demand (**PD**), temporal demand (**TD**), operating performance (**OP**), effort (**E**) and degree of frustration (**DF**). After finishing her/his task, each volunteer was asked to independently provide scores for the NASA-TLX index. Then, the evaluations of the corresponding indicators were averaged. In addition, each volunteer was requested to evaluate the joviality (**J**), convenience (**C**), and workability (**W**) of F-M and GC. Joviality describes the degree of amusement the user feels, convenience describes the quality of being suitable for the user's objectives, and workability describes the extent to which the initialization approach is feasible. The results of the quantitative comparative experiments are presented in Figure 8. Based on user experiences and questionnaires, we used the Gaussian distribution $x \sim \mathrm{N}(\mu, \delta^2)$ to analyze the evaluation index x. Here, μ and δ denote the mean and mean variance, respectively. The distributions of the nine indicators for F-M and GC are shown in Table 2. For example, in Table 3, $\mathrm{MD}^{\mathrm{F-M}} \sim \mathrm{N}(2, 0.43^2)$ and $\mathrm{MD}^{\mathrm{GC}} \sim \mathrm{N}(4.5, 1.03^2)$.

It is figured out from Table 2 that, compared with GC, for the group #1, the average user load of F-M is reduced by 46.22%, and the valuation indicators, joviality, convenience and workability, are increased by 50%, 29.73%, and 56.67% respectively. For the group #2, the average user load of F-M is reduced by 56.25%, and the valuation indicators, joviality, convenience and workability, are increased by 100%, 135%, and 25.64% respectively. For the group #3, the average user load of F-M is reduced by 63.82%, and the valuation indicators, joviality, convenience and workability, are increased by 150%, 66.67%, and 35.14% respectively.

An interesting finding is that group #3 has the largest reduction in the user load for F-M, whereas group #2 has the highest evaluation in terms of the three indicators, namely, joviality, convenience and workability, out of the three groups. Moreover, almost all experiment volunteers believed that no additional memory load or operational load will be imposed on users in the F-M-based interactive system.

Compared with the GC-based interface, success in using the F-M-based interface to complete the interaction task mainly depends on the user's past operating experience and this interface almost eliminates the need for users to

memorize gesture commands or operating methods or gesture order in a ***GG***. There are similarities among the gestures in the GG. It is easy to transition from one gesture to another gesture, thereby decreasing the physical efforts that are required by the user. The required time depends on the degree of proficiency with the operation and the correction rate of mis-operations. Consequently, the F-M algorithm ensures a lower time requirement. Using the F-M interface, users are not required to employ gestures with which they are not familiar or that are dificult to perform. In addition, the error correction strategies accord with cognitive-behavioral principles. For example, in the same context, gestures 'release 5 fingers' and 'release fingers 1 and 2' could define the same semantic object, namely, 'Scale-up an object'. As such, users could flexibly select different gestures based on their life experiences for expressing the same interaction intent.

VII. CONCLUSION AND FUTURE WORK

With the application background of the SCI system that is based on gesture sensing and interaction, which aims at flexibly mapping from many gestures to one semantic object under the same situational context, this paper makes the following main contributions. (a) A flexible mapping algorithm, namely, F-M between multiple gestures and one semantic object under the same situational context, is proposed. (b) It is demonstrated that the proposed F-M algorithm substantially reduces the user's memory loads. (c) An intent-driven approach is presented for F-M.

A flexible mapping between multiple gestures and one semantic object is proposed based on the observation that multiple gestures for an interaction can represent the same interaction intent under the same situational context. A method is presented for establishing a gesture group (GG) based on the behavioral models of the user by utilizing multiple gestures to express the same semantics in the same interaction scenario. An intelligent and natural interaction interface model for 3D platforms has been implemented. The performance and cognitive mechanism of the F-M between the users' gestures and interaction semantics have been evaluated with five frequently used gestures and applied to the SCI system. It is demonstrated that the proposed F-M algorithm substantially reduces the user's memory loads.

There exist limitations in this work. The one issue is that we neglected ethical context surrounding because of no ethical comity is available at our institution. Another one is that we overlook the relationship between many-to-one gesture-to-command mapping and user intentions.

In our future work, we will consider fusing the continuous production of gestures, signs and speech such that the computer will locate the potentially meaning-bearing phases in the continuous production and understand the user's intentions [18], [34], [36] and upon which build more smart many-to-one gesture-to-command flexible mapping models and algorithms.

REFERENCES

[1] R. D. Vatavu and I. A. Zaiti, "Leap gestures for TV: Insights from an elicitation study," in *Proc. ACM Int. Conf. Interact. Experiences TV Online Video*, 2014, pp. 131–138.

[2] B. Sumak, M. Pusnik, M. Hericko, and A. Sorgo, "Differences between prospective, existing, and former users of interactive whiteboards on external factors affecting their adoption, usage and abandonment," *Comput. Hum. Behav.*, vol. 72, pp. 733–756, Jul. 2017.

[3] X. Yin and M. Xie, "Finger identification and hand posture recognition for human–robot interaction," *Image Vis. Comput.*, vol. 25, pp. 1291–1300, Aug. 2007.

[4] E. R. Pedersen, K. McCall, T. P. Moran, and F. G. Halasz, "Tivoli: An electronic whiteboard for informal workgroup meetings," in *Proc. Readings Hum. Comput. Interact.*, Jan. 1995, pp. 509–516.

[5] D. Y. Huang, W. C. Hu, and S. H. Chang, "Gabor filter-based hand-pose angle estimation for hand gesture recognition under varying illumination," *Expert Syst. Appl.*, vol. 38, no. 5, pp. 6031–6042, 2011.

[6] J. F. Hessam, M. Zancanaro, M. Kavakli, and M. Billinghurst, "Towards optimization of mid-air gestures for in-vehicle interactions," in *Proc. 29th Austral. Conf. Comput.-Hum. Interact.*, 2017, pp. 126–134.

[7] P. K. Pisharady and M. Saerbeck, "Recent methods and databases in vision-based hand gesture recognition: A review," *Comput. Vis. Image Understand.*, vol. 141, no. 5, pp. 152–165, Dec. 2015.

[8] Z. Ding, Y. Chen, Y. L. Chen, and X. Wu, "Similar hand gesture recognition by automatically extracting distinctive features," *Int. J. Control Autom. Syst.*, vol. 15, pp. 1770–1778, Jun. 2017.

[9] Y. Liu, L. Tang, K. Song, S. Wang, and J. Lin, "A multicolored vision-based gesture interaction system," in *Proc. 3rd Int. Conf. Adv. Comput. Theory Eng. (ICACTE)*, vol. 2, 2010, p. V2-281.

[10] J. Zeng, Y. Sun, and F. Wang, "A natural hand gesture system for intelligent human-computer interaction and medical assistance," in *Proc. 3rd Global Congr. Intell. Syst. (GCIS)*, Nov. 2012, pp. 382–385.

[11] C. Oz and M. C. Leu, "Human-computer interaction system with artificial neural network using motion tracker and data glove," in *Proc. Int. Conf. Pattern Recognit. Mach. Intell.* Berlin, Germany: Springer, 2005, pp. 280–286.

[12] Y. Zhou, G. Jiang, and Y. Lin, *A Novel Finger and Hand Pose Estimation Technique for Real-Time Hand Gesture Recognition*. Amsterdam, The Netherlands: Elsevier Science, 2016.

[13] J. L. Raheja, M. Minhas, D. Prashanth, T. Shah, and A. Chaudhary, "Robust gesture recognition using kinect: A comparison between DTW and HMM," *Optik*, vol. 126, nos. 11–12, pp. 1098–1104, Jun. 2015.

[14] P. Wang, Z. Li, Y. Hou, and W. Li, "Action recognition based on joint trajectory maps using convolutional neural networks," in *Proc. ACM Multimedia Conf.*, 2016, pp. 102–106.

[15] Y. Sang, L. Shi, and Y. Liu, "Micro hand gesture recognition system using ultrasonic active sensing," *IEEE Access*, vol. 6, pp. 49339–49347, 2018.

[16] R. Ibaez, L. Soria, A. Teyseyre, G. Rodriguez, and M. Campo, "Approximate string matching: A lightweight approach to recognize gestures with Kinect," *Pattern Recognit.*, vol. 62, pp. 73–86, 2017.

[17] L. Liu, S. Wang, B. Hu, Q. Qiong, J. Wen, and D. S. Rosenblum, "Learning structures of interval-based Bayesian networks in probabilistic generative model for human complex activity recognition," *Pattern Recognit.*, vol. 81, pp. 545–561, Sep. 2018.

[18] S. Kita and G. I. V. D. H. H. Van, "Movement phases in signs and co-speech gestures, and their transcription by human coders," in *Gesture and Sign Language in Human-Computer Interaction* (Lecture Notes in Computer Science), vol. 1371, I. Wachsmuth and M. Fröhlich, Eds. Berlin, Germany: Springer, 1998, pp. 23–35.

[19] E. Choi, S. Kwon, D. Lee, H. Lee, and K. C. Min, "Towards successful user interaction with systems: Focusing on user-derived gestures for smart home systems," *Appl. Ergonom.*, vol. 45, no. 4, pp. 1196–1207, 2014.

[20] K. Wang, B. Xiao, J. Xia, D. Li, and W. Luo, "A real-time vision-based hand gesture interaction system for virtual EAST," *Fusion Eng. Des.*, vol. 112, pp. 829–834, Nov. 2016.

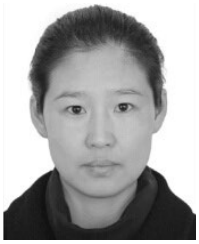
[21] D. Rempel, M. J. Camilleri, and D. L. Lee, "The design of hand gestures for human–computer interaction: Lessons from sign language interpreters," *Int. J. Hum. Comput. Stud.*, vol. 72, pp. 728–735, Oct./Nov. 2014.

[22] K. Katsuragawa, A. Kamal, and E. Lank, "Effect of motion-gesture recognizer error pattern on user workload and behavior," in *Proc. Int. Conf. Intell. User Interfaces*, 2017, pp. 439–449.

[23] J. Ruiz, Y. Li, and E. Lank, "User-defined motion gestures for mobile interaction," in *Proc. Sigchi Conf. Hum. Factors Comput. Syst.*, 2011, pp. 197–206.
[24] D. Kern, P. Marshall, M. Pfei er, V. Gruhn, and A. Schmidt, "Gestural interaction on the steering wheel: Reducing the visual demand," in *Proc. SIGCHI Conf. Hum. Factors Comput. Syst.*, 2011, pp. 483–492.
[25] C. H. Lai, "A fast gesture recognition scheme for real-time human-machine interaction systems," in *Proc. Int. Conf. Technol. Appl. Artif. Intell.*, Nov. 2011, pp. 212–217.
[26] Y. Lou, W. U. Wenjun, R. D. Vatavu, and W. T. Tsai, "Personalized gesture interactions for cyber-physical smart-home environments," *Sci. China*, vol. 60, Oct. 2017.
[27] M. Hoetjes, R. Koolen, M. Goudbeek, E. Krahmer, and M. Swerts, "Reduction in gesture during the production of repeated references," *J. Memory Lang.*, vols. 79–80, pp. 1–17, Feb./Apr. 2015.
[28] F. Jeanne, Y. Soullard, A. Oker, and I. Thouvenin, "EBAGG: Error-based assistance for gesture guidance in virtual environments," in *Proc. IEEE Int. Conf. Adv. Learn. Technol.*, Jul. 2017, pp. 472–476.
[29] A. Licsár and T. Szirányi, "User-adaptive hand gesture recognition system with interactive training," *Image Vis. Comput.*, vol. 23, pp. 1102–1114, Nov. 2005.
[30] Z. Feng, B. Yang, T. Xu, X. Yang, W. Xie, C. Ai, and Z. Chen, "FM: Flexible mapping from one gesture to multiple semantics," *Inf. Sci.*, vol. 467, pp. 654–669, Feb. 2018.
[31] R. Eshuis and N. Mehandjiev, "Flexible construction of executable service compositions from reusable semantic knowledge," *ACM Trans. Web*, vol. 10, no. 1, pp. 1–27, 2016.
[32] K. Kok, K. Bergmann, and A. Cienki, "Mapping out the multifunctionality of speakers," *Gesture*, vol. 15, no. 1, pp. 37–59, 2016.
[33] Z. Feng, B. Yang, Y. Li, Y. Zheng, X. Zhao, J. Yin, and Q. Meng, "Real-time oriented behavior-driven 3D freehand tracking for direct interaction," *Pattern Recognit.*, vol. 46, no. 2, pp. 590–608, 2013.
[34] K. Sun, Z. Feng, C. Ai, Y. Li, J. Wei, X. Yang, X. Guo, H. Liu, Y. Han, and Y. Zhao, "An intelligent discovery and error correction algorithm for misunderstanding gesture based on probabilistic statistics model," *Int. J. Performability Eng.*, vol. 14, no. 1, pp. 89–100, 2018.
[35] E. R. Muth, J. D. Moss, P. J. Rosopa, J. N. Salley, and A. D. Walker, "Respiratory sinus arrhythmia as a measure of cognitive workload," *Int. J. Psychophysiol.*, vol. 83, no. 1, pp. 96–101, 2012.
[36] K. Wang, Z. Feng, J. Li, and R. Han, "A structural design and interaction algorithm of smart microscope embedded on virtual and real fusion technologies," *IEEE Access*, vol. 7, pp. 152088–152102, 2019, doi: 10.1109/ACCESS.2019.2945330.

SHICHANG FENG is currently pursuing the master's degree with the School of Informatics, Xiamen University. His main research interests include human–computer interaction and 3D vision.

ZHIQUAN FENG received the master's degree from Northwestern Polytechnical University, China, in 1995, and the Ph.D. degree from the Computer Science and Engineering Department, Shandong University, in 2006. He is currently a Professor with the School of Information Science and Engineering, University of Jinan. He has published more than 50 articles on international journals, national journals, and conferences in recent years. His research interests include human hand tracking/recognition/interaction, virtual reality, human–computer interaction, and image processing.



LIUJUAN CAO is currently an Associate Professor with the School of Informatics, Xiamen University, China. Her current research interests include image understanding and image processing. She has published more than 30 articles, including Elsevier *Information Science*, *Neurocomputing*, and *Signal Processing*, and the IEEE International Conference on Computer Vision and Pattern Recognition and International Joint Conference on Artificial Intelligence.