Research on Initialization of 3D Hand Pose Based on User and Computer Interaction

Shichang Feng[1], Zhiquan Feng[2(✉)], and Xiaohui Yang[2]

[1] School of Information Science and Engineering,
Shandong University of Science and Technology, Qingdao 266590, China
[2] Provincial Key Laboratory for Network Based Intelligent Computing,
Jinan 250022, China
fzqwww@263.net

Abstract. Finding 3D hand models corresponding to the user's 3D hand pose in the initial frames makes the initialization for the 3D hand model much more significant in 3D human hand tracking. Blending computer interaction techniques and cognition theories, a novel initialization approach for 3D hand model is put forward in the present paper to make the initialization process more human-oriented. The proposed initialization process is primarily divided into three steps. The first step is the approximate classification of the user's poses as dominated by a computer. The second step is to adjust by freehand as dominated by the user. The third step is to modify the 3D hand model as dominated by the computer. The present study attempts to describe and shape the user's behavioral model, upon the initialization algorithm is designed and optimized. To improve the performance of the initialization algorithm, User experience, time cost, and accuracy are fused into an evaluation criterion for the optimization of the proposed algorithm. The main contributions of the present work consists of modeling the operator's cognitive behavior, attempting to answer why and how the cognitive behavioral model guides the proposed algorithm in assigning tasks for the initialization between human and computer. The experimental results demonstrate good performance by the proposed method and its potential applications. In addition, the proposed approach could provide the user with an easier, more pleasurable, and more satisfactory experience. The developed initialization system is successfully applied to several application systems.

Keywords: 3D human hand gesture model · Features extraction · Initialization · 3D hand tracking · Human–computer interaction

1 Introduction

The reconstruction of a 3D hand pose aims to reconstruct the initial 3D hand models, in accordance with the users' poses in the first frames of an online hand video, from which recursive hand trackers can work [1–4].

Most of the prevailing tracking approaches, such as Kalman filter (KF), extended Kalman filter (EKF), unscented Kalman filter (UKF), and particle filtering (PF) [2], are featured with recursions. The next state of the tracked freehand is calculated based on the last state. These approaches do not work without the initial states. As a result,

Z. Pan et al. (Eds.): Transactions on Edutainment XII, LNCS 9292, pp. 53–67, 2016.
DOI: 10.1007/978-3-662-50544-1_5

research on the general approach to initializing the states of tracked freehand has become significant.

Although initialization of the visual tracking system is critical in the performance of freehand tracking systems, not much is known about its process. Most of the currently available algorithms assume that the initialization is done manually.

Additionally, initialization for 3D hand models is a very complicated issue. First, the recovery of a 3D hand structure from a single 2D hand image remains to be a challenge. Second, the human hand is a typical articulated and elastic object with high dimensionality; finding the real 3D hand model from nearly unlimited hand poses is almost impossible. The present study attempts to solve these initialization issues by the interaction between human and computer.

Motivated by the human–computer interface in 3D application systems, such as the drag-and-drop system, 3D virtual assembly system, and pointing devices, freehand tracking requires a reliable initial pose in the first frame with an easier, more pleasurable, and more satisfactory user experience.

2 Related Work

A person-independent recognition method for hand postures against complex backgrounds is proposed in [5] by combining different feature types at the graph nodes. To estimate the arbitrary 3D human hand postures, Shimada [6] accepts not only pre-determined hand signs, but also arbitrary postures in a monocular camera environment. The estimation is based on 2D image retrieval. More than 16,000 possible hand appearances are originated from a given 3D shape model by rotating the model's joints. The images are then stored in an appearance dataset. Rosales [7] proposed the specialized mappings architecture (SMA) approach to map image features to likely 3D hand poses using a machine learning architecture. The hand is tracked and its 3D configuration on every frame is tracked. No any restrictions are imposed on the hand shape and no manual initialization is required. The chamfer distance, edge orientation histogram, and moment are used in [8] to estimate the 3D hand shape and orientation by retrieving appearance-based matches from a large dataset of synthetic views, which are rendered by 26 predefined prototype shapes. The hand shape in the input image is assumed to be close to one of the 26 predefined shapes. A tree-based representation [9], in which the leaves define a partition of the state space with piecewise constant density, can be applied effectively to track 3D-articulated and non-rigid motion.

The single frame pose estimation approach [10] is based on a local search and keeps track of only the best estimation at each frame. This type of tracker is expected to work well at the initialization phase, because no previous data is used. One of the distinct features of the single-frame pose estimation approach is the retrieval of hand poses from a hand image dataset. The advantage of using appearance-based matching for 3D parameter estimation is that the estimation is done indirectly, by looking up the ground truth labels of the retrieved synthetic views. This method avoids the ill-posed problem of recovering the depth information directly from the input image.

Particle filtering is a well-known technique for implementing the recursive Bayesian filters using Monte Carlo simulations. The basic idea of particle filtering is to

represent an arbitrary probability density using weighted samples drawn from another easy-to-sample density called the importance density. The weights represent the probability of occurrence of each sample and the weighted samples are usually called particles. In case of tracking, the particles of the hand configuration distribution are to be updated at each frame. In [11], the Stochastic Meta-Descent (SMD) algorithm was employed and resulted in an eight-particle tracker. This tracker tracks high-dimensional articulated structures using far fewer samples than the previous methods. Additionally, it can handle multiple hypotheses, clutter, and occlusion, with which pure optimization approaches have problems.

The current authors proposed a new method to initialize the 3D pose and position of the freehand by fusing the three techniques—interaction between human and computer, modeling cognitive behaviors for operators, and visualizing information in the initialization process [12]. Focus was placed on the development of a method for 3D hand models. A model behavior of the operator's hand was created, upon which the design of the present initialization algorithm is based. The previous research shows that behavioral models are not only beneficial to the reduction of the cognitive burden of operators, because it allows computers to cater to changes in the operators' hand poses, but also helpful in addressing the high dimensionality of an articulated 3D hand model. The experimental results also show that the method provides an easier, more pleasurable, and satisfactory experience for operators. Currently, the developed initialization system has been successfully applied to the proposed 3D freehand tracking system. A distinctive disadvantage of this method is that it does not present a way to determine the best temporal points for the interaction between human and computer, which affects the interaction's degree of harmoniousness. In fact, the boundary point between human and computer is set by trial and error. The present paper deepens and expands the previous method [12]. The main contributions of the current work lies in modeling the operator's behavior, illustrating why and how the Cognitive Behavioral Model (CBM) guides the developed algorithm in assigning tasks for the initialization between human and computer.

3 Overview

3.1 Modeling Problem

The present research attempts to find a solution of S, V, and R for the following problem (See Fig. 1):

$$\underset{V_S, R_S}{Min}\{H(V, R)\} \tag{1}$$

where $\boldsymbol{R}$ is a real hand pose and $\boldsymbol{V}$ is a virtual synthesized 3D human hand model (or 3D hand model). In expression (1), $\boldsymbol{H}(.)$, denoted as the undirected Hausdorff ditance, is the cost function used to evaluate the similarity of $\boldsymbol{V}$ and $\boldsymbol{R}$. Formula (1) means that the user's real 3D hand pose should be fitted into the 3D hand model at frame 0, starting from the frame $-S + 1$.

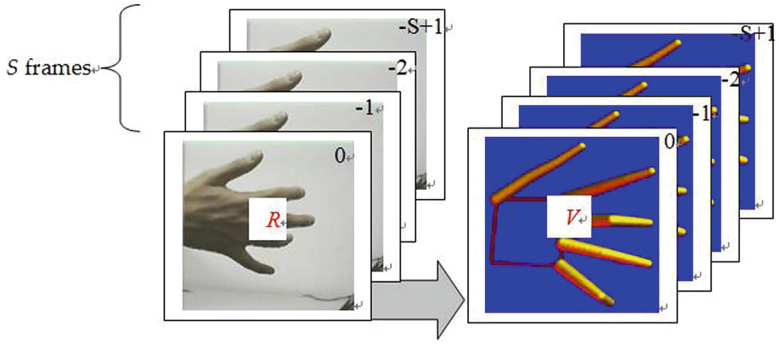

Fig. 1. This figure shows how to acquire the 3D hand model during the initial S frame images in an online video with an easy, pleasurable, and satisfactory user experience, which is the main objective of the present paper. The frame images ***R*** are to the left and the right images are the corresponding recovered 3D hand model ***V***.

3.2 Overview

Freehand tracking using a PF tracker [2] is implemented in the present study. As soon as the initialization of hand pose is performed, the recursive system begins to work automatically, frame by frame (See Fig. 2).

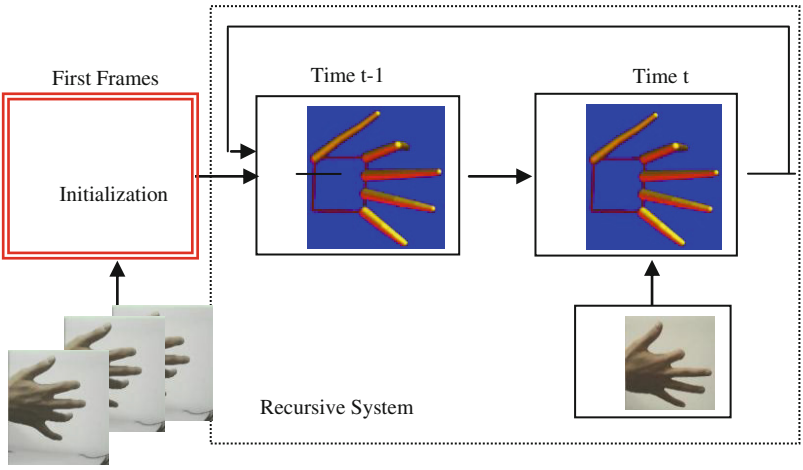

Fig. 2. Global framework of freehand tracking system, a typical recursive system. Acquisition of hand state during first frames is important. Focus in placed on initialization.

The objective of initialization for the 3D hand model is to determine a 3D hand model corresponding to a 3D hand posture during the first frame images.

As shown in related works, most of the known initialization approaches require the users' initial hand poses to be the same as those in a hand pose dataset, which means that users are asked to reconstruct the same predefined poses.

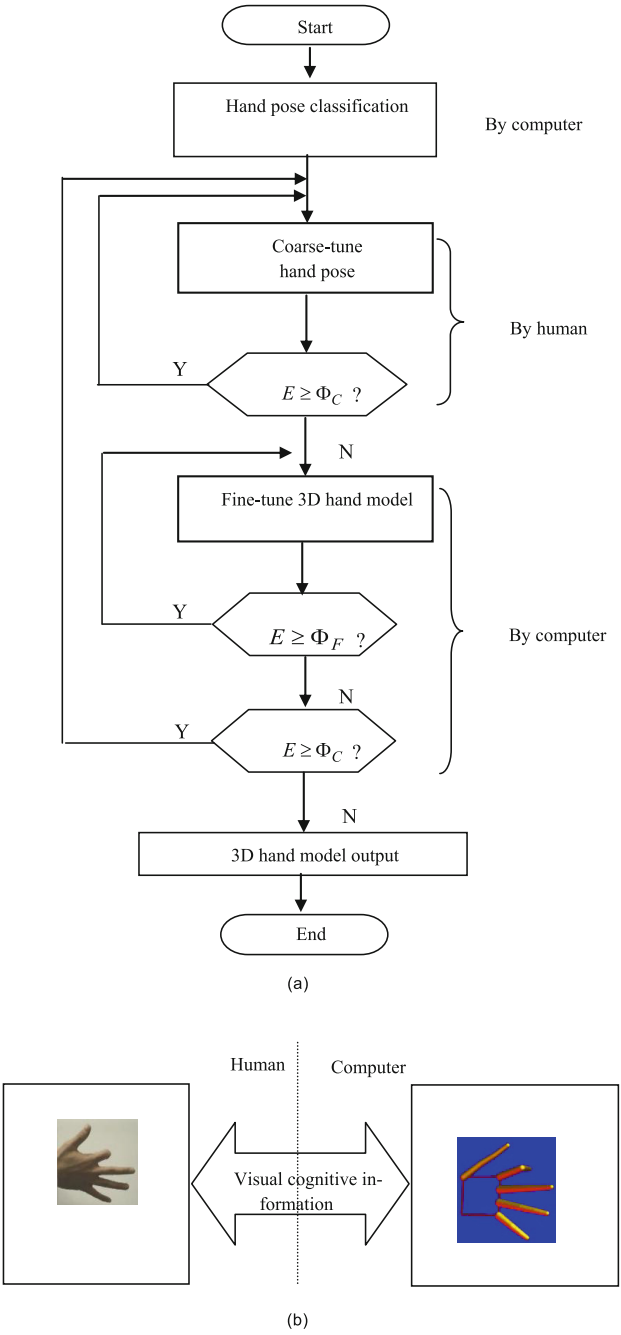

Fig. 3. Overview of OM. (a) Φ_C and Φ_F are threshold values used as temporal points between user and computer. After a hand pose is classified from the rough 3D hand model, OM performs the interaction between human and computer. (b) Framework for how human, computer, and cognitive information are integrated.

The algorithm framework of the proposed initialization method (referred to as OM) is depicted in Fig. 3. OM is composed of three phases, namely, hand pose classification, coarse-tuning the user's hand pose, and fine-tuning the 3D hand model. Figure 3 also shows how the 3D hand model is shaped by the three phases and how an initialization task is approximately assigned between user and computer for effective interaction.

The objective of hand pose classification is to determine a rough 3D hand model similar to the frame image by the retrieval from a hand posture dataset. Given a dataset of many poses, a pose that best matches the object in the input image is identified. However, the rough 3D hand model is only an initial value, $V - S + 1$, of V in Eq. (1). The objective of coarse-tuning the hand pose is to bring the user's hand image close to the projection of the 3D hand model while keeping the hand model fixed. The objective of fine-tuning the hand model is to making the hand model and 3D hand pose similar, while the hand pose is kept fixed.

To provide a new approach to initializing the 3D hand pose, the operation of which is human-oriented and convenient for online use, the three core techniques, namely, HCI in the initialization process, the visualization of cognitive information, and modeling cognitive behaviors to reduce the cognitive burden, are blended together in the proposed method. The states of temporary 3D hand models and image features from videos are fed back onto screen in the form of interactive graphics and imaging. The states are used as cognitive information upon which users adjust their behaviors. On the other hand, the computer fine-tunes the 3D hand models according to users' responses.

In Fig. 3a, both Φ_C and Φ_F are the Hausdorff distances between the projections of the 3D hand models onto the frame image and the frame image features. Φ_C determines the accuracy of coarse-tuned human hand pose and Φ_F determines the accuracy of the fine-tuned 3D hand model. In most cases, $\Phi_C \geq \Phi_F$.

To determine Φ_C and Φ_F, a behavioral model is built for users.

4 Modeling Users' Behavior

Research on cognitive behavioral model is helpful in exploring the cognitive mechanism and ways for the interaction between users and computer.

4.1 Experiment for CBM

The participators (users) were equipped with a data glove and position tracker on their hands. They would be requested to perform some initialization tasks. A synthesized 3D hand model based on data from the data glove and position tracker was displayed in the scene and each participator was requested to adjust his/her hand until it fit into the 3D hand model. In Fig. 4a, a user is doing an experiment for CBM and further analysis of the experiment is shown in Fig. 4b.

According to Fig. 4, the joint angles acutely change in one period of time, and placidly change in another period of time. The former is regarded as the coarse-tuning process, the latter as the fine-tuning process. This observation guides the introduction

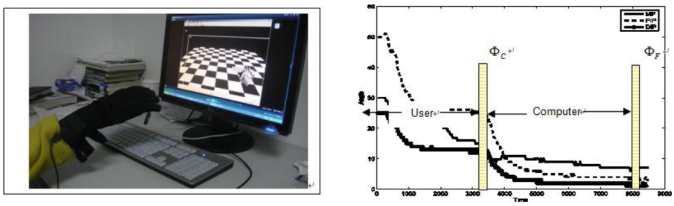

Fig. 4. An investigation of CBM for the user. (a) To reveal the CBM, the user was equipped with a data glove and a position tracker on the right hand to fulfill initialization for the 3D hand model. (b) The angle curves over the frames for the little finger in the process of grasping a virtual object; it describes the three angles per frame; the measurement unit in the y axis is the degree and the time unit is in ms. The time determined by Φ_C and Φ_F are the temporal points between user and computer; in most cases, $\Phi_C \infty \geq \Phi_F$ is satisfied. Φ_C and Φ_F are the Hausdorff distances.

of the technique of interaction between human and computer, and provides cues to assign interactive and cooperative tasks between user and computer. The CBM models are stated in the form of CBM features.

4.2 CBM Features

The main findings, also called CBM features, in the present study are presented as follows:

- CBM-0: The process of adjusting the pose can be divided into different sections, in which one features a large range regulation in the frontal period of time and another features a small range regulation in the back period of time.
- CBM-1: There are two paths that keep the hand pose consistent with the 3D hand model to maintain low cognitive burdens.
  Path-1: Initial pose → adjust the hand pose orientation → translate the hand pose → 3D hand model is slightly tuned, or
  Path-2: Initial pose → translate the hand pose → adjust the hand pose orientation → 3D hand model is slightly tuned.
- CBM-2: The variables set, x, in hand model X can be described with the unchanging variables as

$$\boldsymbol{X} = \boldsymbol{U} \cup (\boldsymbol{I} - \boldsymbol{U}) \tag{2}$$

where U refers to the set composed of unchangeable variables, I refers to the set composed of all variables in hand model X. Taking the fist-style pose as an example, in the whole process of initialization, all local angles are kept fixed. Set U can be described manually in the hand pose dataset.

- CBM-3: Compared with the approach asking users to imagine and shape their required hands according to required regulations or constraints, the approach allowing users to actively adjust their hand poses based on visual cognitive information onscreen is more concurrent with the users' cognitive customs and costs less cognitive burdens [12].

The CBM features lead to the best way to blend both HCI and visualization into the initialization process, with the purpose of maintaining low cognitive burdens for users.

5 Tune Users' Hand Poses

Based on CBM-0, the initial hand pose may be far from the 3D hand model, even if the rough pose is acquired from the retrieval hand pose dataset in the classification process. That is, finding the solution for formula (1) may require a computationally expensive search. Accordingly, the very interesting issue of how to use the CBM Features to guide the interaction between human and computer is discussed to adjust the user's hand pose for low cognitive burden and design the 3D hand model using cheap computation.

The superiority of the user over the computer is that he/she can flexibly make the decisions for all situations in completing a cognitive task, which is the reason why the HCI technique is introduced into the developed initialization system. The frontal period of time is assigned to a user. That is, the user adjusts his/her hand's position and pose in agreement with the CBM-1 Feature to superpose the online hand images onto the projection of the given initial 3D hand model while keeping the 3D hand model fixed.

The effectiveness of the HCI, to some extent, depends on the path of feedback and the visualization style of the hand image and the 3D hand model. The present study uses many approaches for visualization. For example, the 3D hand model is rendered and outputted by OpenGL, and the new hand images are displayed in real time with a visual style.

Clearly, the developed initialization process starts from an initial pose similar to that in a dataset. However, the final 3D hand model may not be limited to the initial pose.

6 Tune 3D Hand Model

The present study does not rely only on the user to align his/her hand to the hand shape displayed. In fact, the back period of time is assigned to the computer. Once the user's hand is close enough to the 3D hand model, the computer provides the user with feedback, flickering and highlighting, and begins to fine-tune the 3D hand model using the approach similar to PF [2]. Taking Fig. 5b as an example, the duration from 0 to 3700 ms is called the frontal period of time, and the duration from 3700 ms to 8500 ms is called the back period of time. Φ_C is the Hausdorff distance at time 3700 ms and Φ_F is the Hausdorff distance at time 8500 ms.

In this process, the CBM-2 feature follows. Although a coarse pose is obtained by the coarse-tuning process, because of high dimensionality problem, the computer

would have difficulty further fine-tuning the 3D hand model in rapid speed until the latter is the same as the user's 3D hand pose. Fortunately, the CBM-2 feature can help to alleviate this problem. For example, if the computer knows or predicts that some part of the variables in a hand model vector would change over time, it will focus on determining the values of these variables and pay little attention to the unchanged variables. This approach is equivalent to reducing the dimensionality of the hand pose vector.

Suppose X0 is the retrieved virtual initial 3D hand model from hand pose dataset, I is the hand frame image at the current frame, Ω is the feature set extracted from image I, which is composed of the contour, fingertips, roots of fingers, joints, and the intersection of the knuckle on the fingers. According to the proposed method, the feature extraction process is composed of two steps, namely, the coarse location phase (CLP) and the refined location phase (RLP), from coarseness to refinement. In the CLP phase, the hand contour is approximately described by a polygon with concave and convex. An approach to obtaining the hand shape polygon using locating points and locating lines is meticulously discussed. Subsequently, using a coarse location algorithm, the contour, fingertips, roots of fingers, joints and the intersection of the knuckle on different fingers can be extracted. In the RLP phase, a multi-scale approach is applied to the extracted features in the CLP phase by defining the response strength of different types of features; the accurate features can then be obtained.

The algorithm for adjusting the 3D hand model based on the CBM is presented as follows.

(A) The empirical values of Φ_C are determined by trial and error.
(B) For the user: Change hand shapes and hand positions.
(B.1) The user moves his/her hand along path-1 or path-2, as stated in CBM-1, while the 3D hand model is unchanged. The 3D hand model and hand image features are synchronously visualized onscreen.
(B.2) The user adjusts his/her hand postures according to the visualized cognitive information, such as the projection of the 3D hand model, hand image features, and other feedback.
(B.3) E is evaluated by

$$E = Hausdorff(X_0, \Omega) \tag{3}$$

where Hausdorff ($\boldsymbol{X_0}$, $\boldsymbol{\Omega}$) is the Hausdorff distance [13] between the 3D hand model $\boldsymbol{X_0}$ projection onto the hand image and the hand image features.
(B.4) If ($E \geq \Phi C$) the computer feedbacks with a flicker and highlight, and goes to step (B) else X1 ← X0.
(C) For the computer: Fine-tune the 3D hand model with PF tracker.
(C.1) Superpose the 3D hand model projection onto the hand image.
(C.2) Sample
Generate N particles X_1^i of X_1, i = 1, 2, …, N, using the Gaussian distribution in agreement with CBM-2.
(C.3) Compute weight ω for each particle
(C.4) State Updating

$$X = \sum_{i=1}^{N} \omega_i X_1^{(i)} \tag{4}$$

(C.5) Evaluate E by

$$E = Hausdorff(X, \Omega) \tag{5}$$

If ($E \geq \Phi F$), X1 ← X and go to step (C).
If ($E \geq \Phi C$), X0 ← X and go to step (B).
Output X.

By looping between steps (B) and (C), the optimization function (1) could be satisfied approximately. Φ_C and Φ_F are threshold values used to control the accuracy of the coarse-tuning and fine-tuning processes, respectively. One of the hardest part in the pose recovery process is tracking under the pose, which is tackles by PF tracking. The role of CBM-1 in the OM is to reduce the cognitive burdens for the user. CBM-2 is used into the OM to avoid sampling for the unchanging parts in the 3D hand model.

7 Optimization of OM

One of the important issues in the OM is the way to choose the parameters Φ_C and Φ_F, by which the OM will be optimized in this section.

Suppose L algorithms, M1, M2, …, ML, are applied to the same initialization task. Ta is the average time cost of the L algorithms; Ak and Tk are the accuracy and time cost of the kth algorithm, where $1 \leq k \leq L$; α_k is the degree of harmoniousness (DOH) determined by the users, which reflects the users' cognitive burden.

Cognitive burden is evaluated by four factors in the present study: tiredness, joviality, freedom, and workability. Tiredness describes the extent of toil a user feels in the initialization process; joviality describes the degree of amusement the user feels; convenience describes the suitability to the user's purposes; and workability describes the extent to which the initialization approach is feasible. The four factors have a scale between 0 and 100, and are scored by the users.

The performance, as one of the evaluation criterions, is defined as formula (6):

$$\lambda_k = \alpha_k T_a \frac{A_k}{T_k} \tag{6}$$

where α_k is the average of J, the joviality, convenience, and workability of the algorithm k. Here, J is the difference between 1 and degree of tiredness.

One of the key issues in evaluating system performance is the availability of ground-truth data. Obtaining ground truth data for the 3D hand pose estimation is a difficult problem [4]. The widely used method to evaluate the 3D hand model is to project the hand model onto the input image to show how well the projection matches the image data, which can be measured by Hausdorff distance.

Accuracy is defined as

$$A_k = e^{-\beta H_k} \tag{7}$$

where Hk is the Hausdorff distance of the algorithm k and β is an experiential constant, which is set at 0.01 in the present experiments.

According to the definition of λ in formula (7), if the time cost of an algorithm is equal to the average time cost, the algorithm is evaluated by accuracy if the parameter α is ignored. On the whole, a large λ_K means high accuracy and low time cost.

The objective of the OM's optimization is to find the threshold values Φ_C and Φ_F to maximize λ_{OM} or

$$\arg(\max(\lambda_{OM})). \tag{8}$$

8 Experimental Results

8.1 Experimental Settings

The present study uses a color CCD camera ZT-QCO12 with a 4 mm lens that captures a 640 × 480 video at 30 Hz. The computer used has an Intel(R) Core(TM) 2 Quad CPU with a 2.66 GHz processor and 3.25 GB memory. A 26 DOF 3D hand model is used, with 6 DOFs for the global transformation and four DOFs for each finger. The length of each knuckle on the fingers and the size of the palm are fixed.

8.2 Experimental Procedures and Results

The experimental process is composed of three phases, namely, hand pose recognition, coarse-tuning, and fine-tuning. There are many hand pose recognition approaches, but most of them have difficulty effectively differentiating between two similar hand poses with the same protrudent fingers. Recognition involves finding the location in the input image that best match occurs. Density distribution features (DDF) [14] are used to describe the hand image features upon which the initial pose is retrieved from the dataset. DDF is defined by the following formula

$$DDF = (\rho_1, \rho_2, \ldots, \rho_N; \delta_1, \delta_2, \ldots, \delta_N) \tag{9}$$

in which the first feature vector represents the relative density of object pixels within each sub-image and the second represents the difference of relative density in the direction of radial coordinates. DDF is invariant to translation, scale, and rotation. In formula (9),

$$\rho_i = n_i/n \tag{10}$$

$$\delta_i = \begin{cases} |\rho_2 - \rho_1| & i = 1 \\ |2\rho_i - \rho_{i-1} - \rho_{i+1}| & 1 < i < N \\ |\rho_N - \rho_{N-1}| & i = N \end{cases} \tag{11}$$

where n_i, (i = 1, 2, …, N) is the total number of pixels of the hand skin in the ith circumcircle and n is the maximum of all n_i.

The OM is compared with two widely used methods, namely, the single frame (hereafter simplified to SF) pose estimation approach [10] and the SMD approach [11]. The first experiment is done using the OM. The number of particles used is 50. The proposed algorithm is evaluated using the same user, and different users. An experiment suing an optimized OM then follows.

Figure 5 is the $\lambda_{OM} - \Phi_C$ curve of the OM algorithm. Based on Fig. 5, the threshold value Φ_C has an effect on performance λ of the OM, as defined in formula (8). If Φ_C is too small or too big, the performance of the OM will decreases, and a maximum exists on the $\lambda_{OM} - \Phi_C$ curve. In the current experiment, when Φ_C = 36.96, λ_{OM} reaches its maximum.

The reason for the above fact is that if Φ_C becomes smaller, the cognitive burdens become bigger; as a result, the parameter α_k becomes smaller. If Φ_C becomes bigger, the accuracy will become smaller; as a result, the parameter Ak becomes smaller. Thus, the performance parameter λ_k becomes smaller under either of the two situations.

Thus, the OM can be optimized by the optimized threshold value Φ_C. The performances of the SMD, SF, and optimized OM by Φ_C are shown in Fig. 6.

Clearly, of the three algorithms, the optimized OM by Φ_C has the best performance.

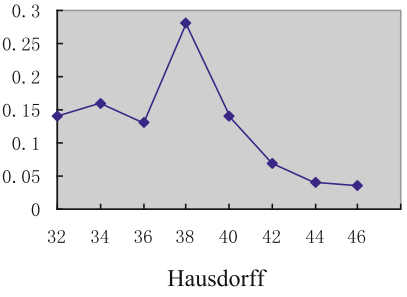

Fig. 5. Hausdorff-λ curve of OM. Here, the Hausdorff distance is the threshold value Φ_C used in OM. When Φ_C = 36.96, λ_{OM} reaches its maximum.

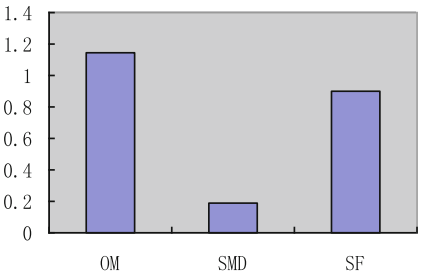

Fig. 6. Performance of three algorithms after OM is optimized by Φ_C.

For further comparison of the OM and optimized OM by Φ_C, the two algorithms are run 50 times each with the same user. The average results are shown in Table 1. The bigger the λ, the smaller the cognitive burden.

For 50 different users, the performance comparison between the OM and optimized OM by Φ_C are shown in Table 2.

Table 1. Performance comparisons of OM and optimized OM by Φ_C. The results are an average of 50 times with the same user.

Performance / Algorithm	Time Cost (ms)	Accuracy	λ
OM	1265.953	0.698	1.361
OM and optimized OM by Φ_C	955.200	0.715	1.430

Table 2. Performance comparisons of OM and optimized OM by Φ_C. The results are an average of 50 times with the same user.

Performance / Algorithm	Time Cost (ms)	Accuracy	λ
OM	1915.3	0.6896	0.7052
OM and optimized OM by Φ_C	2023.59	0.7146	0.80316

Table 3. Comparison of OM and optimized OM by Φ_C and Φ_F

Performance / Algorithm	Time Cost (ms)	Accuracy	λ
OM	1915.30	0.720	0.7368
OM and optimized OM by Φ_C	2023.59	0.744	0.8364

According to Tables 1 and 2, the optimized OM by Φ_C has better performance than the OM.

After the Φ_C is fixed, Φ_F can be determined by a similar method as Φ_C.

The comparisons for the time cost, accuracy, and λ of the OM and optimized OM by both Φ_C and Φ_F are shown in Table 3. The DOHs of the different methods are shown in Table 4.

Table 3 shows that the time cost by the optimized OM by Φ_C and Φ_F increased, and its global performance also increased. Table 4 shows that the optimized OM by Φ_C and Φ_F imposes the least cognitive burden for users.

Table 4. Comparison of DOH of the several algorithms

Algorithm	Optimized OM by Φ_C and Φ_F	OM	SMD	SF
α(DOH)	0.85	0.732	0.624	0.647

At the end of this section for the OM, the dependency of Φ_C and Φ_F on the participants and on the complexity of the pose are discussed. Φ_C and Φ_F are determined by formula (9). For different participants or different poses, the parameters Ta, Ak, Tk, and α_k are different. Thus, Φ_C and Φ_F are dependent on the participants and the complexity of the pose. In fact, let 50 different participants do the same or not the same poses for initialization, the variances of Φ_C and Φ_F are large and change in a large scale. However, for the same participant, the variances of Φ_C and Φ_F are small.

The OM and optimized OM by Φ_C and Φ_F expands the idea proposed in the present paper [12]. In the same experimental conditions, the performance is listed in Table 5.

Table 5. Comparison OM with the approach presented in paper [12]

Performance / Algorithm	Time Cost (ms)	Accuracy	λ
OM	1915.3	0.6896	0.7052
OM and optimized OM by Φ_C and Φ_F	1823.59	0.7146	0.80316
Previous Method [32]	2535.3	0.6392	0.6988

The OM and optimized OM by Φ_C and Φ_F perform the same initialization task with less time, higher accuracy, and higher λ, compared with the previous work [12].

9 Discussion and Conclusions

Compared with the previous related work [12], the main contributions of the present study involves the attempt to theoretically explore the fundamental "why, how, when (WHW) problems" in the process of fusing the three core techniques, namely, the HCI for initializing, visualization of the 3D hand model, and modeling the operator's cognitive behavior. Thus, the present study shows why and how the CBM guides the

proposed algorithm in assigning tasks for the initialization between human and computer, and informs when the temporal points between users and computer occur. These benefits are summarized as follows.

Acknowledgments. This paper is supported by the Science and technology project of Shandong Province (No. 2015GGX101025).

References

1. Erol, A., et al.: Vision-based hand pose estimation: a review. Comput. Vis. Image Underst. **108**, 52–73 (2007)
2. Julier, S.J., Uhlmann, J.K.: A new extension of the Kalman filter to nonlinear systems. In: Proceedings of of AeroSense: The 11th International Symposium on Aerospace/Defence Sensing, Simulation and Controls, pp. 182–193, SPIE, Orlando, Florida, USA (1997)
3. Salih, Y., Malik, A.S.: Comparison of stochastic filtering methods for 3D tracking. Pattern Recogn. **44**(10–11), 2711–2737 (2011)
4. Erol, A., Bebis, G., Nicolescu, M., Boyle, R., Twombly, X.: A review on vision-based full DOF hand motion estimation. In: Proceedings of the IEEE Workshop on Vision for Human-Computer Interaction (V4HCI), pp. 15–22, San Diego, California (2005)
5. Triesch, J., von der Malsurg, C.: A system for person-independent hand posture recognition against complex background. IEEE Trans. Pattern Anal. Mach. Intell. **23**(12), 1449–1453 (2001)
6. Shimada, N., Kimura, K., Shirai, Y.: Real-time 3-D hand posture estimation based on 2-D appearance retrieval using monocular camera. In: Proceedings of International Workshop RATFG-RTS, pp. 23–30 (2001)
7. Rosales, R.: The specialized mappings architecture with applications to vision-based estimation of articulated body pose. Ph.D. thesis, BOSTON University Graduate School of Arts and Sciences (2002)
8. Athitsos, V., Sclaroff, S.: Estimating 3D hand pose from a cluttered image. In: Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition, vol. 2, pp. 432–439 (2003)
9. Stenger, B., Thayananthan, A., Torr, P.H.S., Cipolla, R.: Filtering using a tree-based estimator. In: Proceedings of IEEE International Conference on Computer Vision, vol. 2, pp. 1063–1070 (2003)
10. Tomasi, C., Petrov, S., Sastry, A.: 3D Tracking = Classification + Interpolation. In: Ninth IEEE International Conference on Computer Vision, pp. 1441–1448 (2003)
11. Bray, M., Koller-Meier, E., Gool, L.V.: Smart particle filtering for 3D hand tracking. In: Sixth IEEE International Conference on Automatic Face and Pose Recognition, pp. 675–680. IEEE Computer Society, Los Alamitos, CA, USA (2004)
12. Feng, Z., Zhang, M., Pan, Z., Yang, B., Xu, T., Tang, H., Li, Y.: 3D-freehand-pose initialization based on operator's cognitive behavior models. Vis. Comput. **26**(6–8), 607–617 (2010)
13. Huttenlocher, D.P., Klanderman, G.A., Rucklidge, W.J.: Comparing images using the hausdorff distance. IEEE Trans. Pattern Anal. Mach. Intell. **15**(9), 850–863 (1993)
14. Liu, H., Feng, S., Zha, H.: Document image retrieval based on density distribution feature and key block feature. In: Proceedings of 8th International Conference on Document Analysis and Recognition, pp. 1040–1044, Seoul, Korea (2005)