IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 13, 2020, accepted August 4, 2020, date of publication August 11, 2020, date of current version August 21, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3015834

Collaborative Differential Evolution Filtering for Tracking Hand-Object Interactions

DONGNIAN LI, YANG GUO, CHENGJUN CHEN, (Member, IEEE), AND ZHENGXU ZHAO, (Senior Member, IEEE)

School of Mechanical and Automotive Engineering, Qingdao University of Technology, Qingdao 266525, China

Corresponding author: Chengjun Chen (chencj@qut.edu.cn)

This work was supported in part by the National Natural Science Foundation of China under Grant 51705273.

ABSTRACT Human hands engage in interactive activities in many practical working scenarios, among which the interactions between human hands and objects are the most common. Tracking the movement of the human hand during hand-object interactions is an important research task that is also challenging due to the high-dimensionality and occlusions. In this paper, we track hand-object interactions from depth observations with a model-based method. To overcome the difficulties of optimum searching in the hand-object high-dimensional space, we propose a new algorithm — collaborative differential evolution filtering (CoDEF) — for tracking hand-object interactions. The proposed CoDEF algorithm integrates the differential evolution (DE) algorithm into a particle filtering (PF) framework to accelerate the convergence of particles. Particles are driven to the regions with a high probability by optimizing the matching error under the current observation with DE. To decompose the state space and decrease the complexity of optimum searching, CoDEF tracks the movement of the hand and object by using two collaborative trackers. Based on the proposed CoDEF algorithm, we develop a model-based tracking system with 3D graphic techniques. According to the experimental results, the proposed CoDEF algorithm can achieve robust tracking of hand-object interactions using fewer particles.

INDEX TERMS Differential evolution, depth image, hand tracking, object tracking, particle filtering.

I. INTRODUCTION

Tracking the movement of the human hand is an important task in many applications, such as the perception of human grasping, movement capture for animation, and human-machine interfacing. In many practical working scenes, the human hand engages in interactive activities. Interactions between the human hand and objects are the most common. Therefore, it is important to track the movement of the human hand during hand-object interactions. Nevertheless, tracking hand-object interactions is limited by several complicated factors. First, it is a high-dimensional problem. Next, occlusions occur frequently during hand-object interactions, including hand-object mutual occlusions and self-occlusions of the hand. However, useful contextual information with the manipulated object can promote the recognition and estimation of human hand movement.

Currently, hand-object tracking methods based on vision can generally be divided into two types: appearance-based methods and model-based methods. Appearance-based methods [1]–[11] estimate hand-object poses directly from image features via a learned mapping. They require no initialization and have a quick tracking speed. However, accurate estimations of poses need a well-trained mapping. Kjellström *et al.* [1] proposed a method for recognizing the movement of the hand and the manipulated object by expressing their relationship with a conditional random field model. However, this method does not provide detailed information about the movement of the human hand. Romero *et al.* [2], [3] reconstructed the 3D gestures of the human hand that interacted with objects using a real-time nonparametric appearance-based method. The method searches for the hand pose that best matches the input image from a large template database with nearest-neighbor searching. Gupta *et al.* [4] proposed a Bayesian approach to integrate multiple perception tasks in human-object interactions. The method searches for consistent semantic expressions by applying space limitations to perception elements. This method not only allows for the recognition of the object and corresponding actions when their appearance cannot be completely distinguished,

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

but it also allows for the recognition of the actions of the human body from static images. However, this method does not produce detailed information about body gestures. Yao and Fei-Fei [5], [6] applied a new random field model for the modeling of objects and body gestures. They estimated the degree of connection among objects, body gestures, and different parts of the human body through a structure learning method. The method calculates the parameters of the model using a new max-margin algorithm. Under this mode, object detection provides strong prior knowledge for the estimation of body gestures, and the estimation of body gestures helps the system conduct more accurate detections of objects interacting with the human body. However, this method only produces 2D estimates for body poses. Recently, some researchers [12]–[28] have introduced deep learning methods to estimate hand poses. Tompson *et al.* [12] trained a convolutional network to extract hand heat-map features from depth images. Then, they recovered hand poses from the heat-map representation with inverse kinematics. Ge *et al.* [13] acquired volumetric representations of hands from depth images. By using the volumetric representations as the input, they regressed the 3D hand joint locations by using a trained 3D convolutional neural network (3D CNN). However, these methods assume an isolated free-moving hand that is not interacting with objects.

Model-based methods [29]–[39] use prebuilt models to generate hypotheses. These methods compare the features extracted from the models with those extracted from visual observation and evaluate the similarity between them. They search for a set of hand-object state parameters that best matches the visual observation in the model state space using an optimization method. However, the tracking process involves a search task in a high-dimensional space, which is challenging. Moreover, the tracking needs to be initialized. Hamer *et al.* [30] searched for the optimal configuration of the hand states through belief propagation (BP). They connected different parts of the multijointed human hand through pairwise Markov random fields. However, they did not construct a model for the manipulated object. Oikonomidis *et al.* [31] regarded the hand-object tracking problem as a sequential optimization problem. They used particle swarm optimization (PSO) to search for the solution. Their system uses multiview RGB image sequences as the input. Kyriazis and Argyros [32] acquired the observation input using a depth camera and only searched for hand pose parameters. They deduced the object pose according to the hand pose and the hand-object interaction model. Zhang and Seah [33] performed a hybrid particle-based search that derives from PSO and differential evolution (DE) to track human body poses. They used a voxel model for the human body. Some researchers [40]–[42] have combined learning-based methods with model fitting for estimating hand poses. Sharp *et al.* [40] used a multilayered random forest to predict a hand pose distribution. Using the hand pose hypotheses sampled from the distribution for the initialization, they performed a model fitting process by minimizing the error between the hand model and the observation with PSO. Their method focused on tracking a single hand. When tracking a hand manipulating an object, failures occurred for their method. Sridhar *et al.* [41] modeled the hand and object with Gaussian mixtures. They performed object segmentation using color information and then carried out hand part classification from the depth input with a multilayered random forest. By using the hand part classification for guidance, they tracked the hand manipulating an object with a 3D Gaussian mixture alignment method. However, the hand part classification did not perform well under situations of severe hand-object occlusions.

Many researchers have performed model-based tracking of the movement of the hand [43]–[46] or body [47], [48] by using a particle filtering (PF) framework. PF has the ability to express a multipeak distribution through the propagation of multiple samples along time. Nevertheless, the standard PF requires a large sample size, especially for high-dimensional problems such as hand movement tracking. A small sample set will lead to particle divergence and tracking failures. For this problem, many researchers [43]–[46] have tracked hand movement by combining optimization methods with PF. Based on a PF framework, the particles predicted by a dynamic model are used as the initial values, and an optimization method is then applied to optimize the particles and accelerate the convergence of the particle set. In a related work [45], Gaussian PSO is combined with PF to track hand-object interactions. However, since the segmented images include a small amount of forearm pixels adjacent to hand pixels, the estimated hand pose slides up and down the arm from frame to frame. Another related work [46] integrates DE into PF for tracking. However, the method considered an isolated hand that was not interacting with objects.

As in [45], we track hand-object interactions from depth observations with a model-based method under a PF framework. However, in this paper, the constructed 3D hand model includes a part of the forearm that can be scaled, enabling the observation model to explain the forearm pixels adjacent to hand pixels in segmented depth images. To accelerate the convergence of particles and improve the distribution of particle samples, we integrate DE into the PF framework to track hand-object interactions. By optimizing the matching error under the current observation with DE, particles are moved towards the regions with high-likelihood probability. However, due to the high-dimensionality of the problem and the occlusions during hand-object interactions, there are many local optima around the global optimum in the hand-object space, making the optimum searching process still challenging. To decrease the complexity of optimum searching, we track the movement of the human hand and the object using two collaborative trackers. The resulting new algorithm — collaborative differential evolution filtering (CoDEF) — assigns one tracker to the hand and another tracker to the object. The two trackers exchange information frequently during the tracking process. Such a

collaborative tracking scheme decomposes the state space with multiple trackers, decreasing the complexity of optimum searching. We develop a model-based tracking system based on the proposed CoDEF algorithm with 3D graphic techniques. The experiments demonstrate that CoDEF can achieve the robust tracking of hand-object movement by using fewer particles. The main contributions of this paper are as follows:

- We propose a new algorithm — CoDEF — for tracking hand-object interactions. To overcome the difficulties of searching in the hand-object high-dimensional space, CoDEF integrates DE into the PF framework and applies two collaborative trackers for the hand and object.
- We construct a 3D hand model including a part of the forearm that can be scaled. In this way, we make the observation model able to explain the forearm pixels adjacent to hand pixels in segmented depth images.
- We develop a model-based prototype system for tracking hand-object interactions based on the proposed CoDEF algorithm with 3D graphic techniques.

The reminder of this paper is organized as follows: Since we track hand-object interactions with a model-based method, we first introduce the constructed hand-object models in Section II. Then, we describe the matching error function and observation model in Section III. In Section IV, we describe the proposed tracking algorithm — CoDEF. In Section V, we describe the model-based tracking system that we have developed based on CoDEF with 3D graphic techniques. Section VI provides the experimental results on real and synthetic data. Section VII presents the conclusion of this paper.

II. HAND-OBJECT MODEL

We track the hand-object interactions by using a model-based method. The human hand is an articulated object, and each joint of the hand has one or more degrees of freedom (DOFs) in rotation. From the application perspective, it is not necessary to capture the movement of all bones in the hand. Therefore, the kinematics modeling of some joints is usually simplified by some approximations. Lee and Kunii [49] introduced a 27-DOF model, which has been widely used. In this paper, we build a hand kinemics model that is similar to [49], which is shown in Fig. 1. However, different from [49], we model the MCP joint of the thumb with only 1 revolute DOF. In addition, since our model includes a part of the forearm, we add a wrist joint to the hand kinemics model. The resulting hand state vector $\boldsymbol{x}_h$ covers 29 DOFs, including 6 DOFs for global hand motion, 20 DOFs for local finger motion, and 3 DOFs for the wrist joint. The CMC joints of all fingers are fixed. The movement of the palm corresponds to 6 global DOFs of the human hand. Each finger is connected to the palm by a 2-DOF (1 flexion-extension DOF and 1 abduction-adduction DOF) joint. In addition, each finger consists of three parts that are connected by two 1-DOF joints. These 1-DOF joints are only capable of flexion-extension motion. The wrist joint has 1 flexion-extension DOF, 1 abduction-adduction DOF, and 1 scaling DOF. We use human anatomy to establish the movement constraints of the finger joints and the wrist joint. The object state vector $\boldsymbol{x}_o$ covers 6 DOFs of the manipulated object.

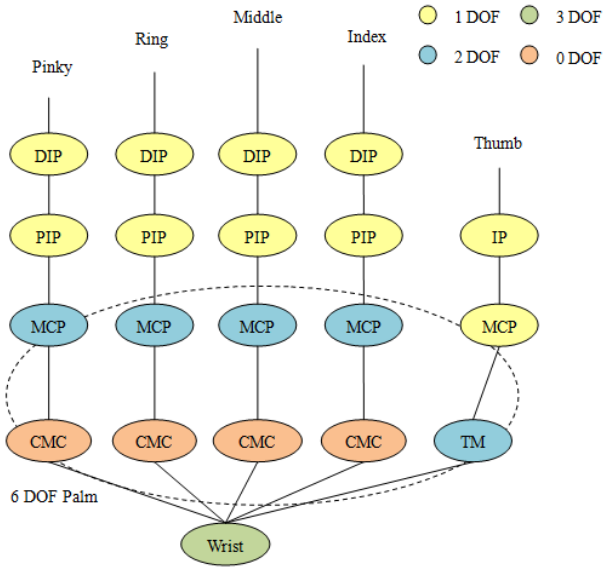

FIGURE 1. Kinematics model of the human hand.

By using the PTC Pro/Engineer[1] and Multigen-Paradigm Creator[2], we build a unified 3D model for the human hand and the manipulated object with parametric geometric primitives. The model has local coordinates and DOF nodes for hand-object pose updating. Moreover, the 3D hand model built in this paper involves a part of the forearm of the human body, which makes the model able to describe the forearm pixels adjacent to hand pixels in segmented depth images. The wrist joint has 1 scaling DOF, which makes the forearm model able to extend or retract. This paper mainly focuses on the interactions of the human hand with a sphere and the interactions with a cylinder. Fig. 2 shows the corresponding models. However, this method can also be used to track the interactions between the human hand and more complex shapes of objects.

III. OBSERVATION MODEL

In this paper, we construct a matching error function and observation likelihood function to evaluate the hand-object hypotheses. The hand-object foreground regions are segmented by a simple threshold from the current depth observation $\boldsymbol{z}$, generating a depth image $z_d(\boldsymbol{z})$. Given a hand-object pose vector $\boldsymbol{x}_{h-o} = (\boldsymbol{x}_h, \boldsymbol{x}_o)$, a depth image $r_d(\boldsymbol{x}_{h-o})$ is generated correspondingly with graphic rendering techniques by a virtual depth camera under the given calibration. Then, two

[1]https://www.ptc.com/en/products/creo/pro-engineer
[2]https://www.presagis.com/en/product/creator/

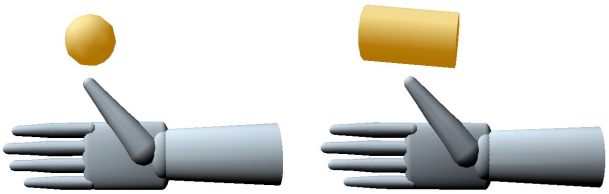

FIGURE 2. Hand-object models. (a) Sphere, (b) Cylinder.

binary silhouette images $z_s(\boldsymbol{z})$ and $r_s(\boldsymbol{x}_{h-o})$ are derived from $z_d(\boldsymbol{z})$ and $r_d(\boldsymbol{x}_{h-o})$ respectively, with a value of 1 in hand-object foreground regions and a value of 0 in the background regions.

By comparing the features extracted from the hypotheses with those extracted from visual observation, a matching error function is defined as follows:

$$E(\boldsymbol{z}, \boldsymbol{x}_{h-o}) = \lambda_d E_d(\boldsymbol{z}, \boldsymbol{x}_{h-o}) + \lambda_s E_s(\boldsymbol{z}, \boldsymbol{x}_{h-o}) + \lambda_h E_h(\boldsymbol{x}_h) \tag{1}$$

where λ_d, λ_s and λ_h are the normalization factors.

E_d measures the depth differences between the pose hypothesis $\boldsymbol{x}_{h-o}$ and the observation $\boldsymbol{z}$. E_d is defined as follows:

$$E_d(\boldsymbol{z}, \boldsymbol{x}_{h-o}) = \frac{\sum \min\left(|z_d(\boldsymbol{z}) - r_d(\boldsymbol{x}_{h-o})|, T_d\right)}{\sum \left(z_s(\boldsymbol{z}) \vee r_s(\boldsymbol{x}_{h-o})\right)} \tag{2}$$

The pixelwise depth differences are calculated and accumulated over the whole image. The accumulated sum is normalized by dividing by the total pixel area of the hand and the manipulated object. Any significant difference in depth will cause significant changes in the functional values, thus influencing the performance of the search method. For this reason, the maximum constant T_d for depth differences is introduced, and the depth differences of all pixels are limited within the range of $[0, T_d]$.

E_s describes the incompatibility of silhouette images based on the area of the nonoverlapping regions between $z_s(\boldsymbol{z})$ and $r_s(\boldsymbol{x}_{h-o})$. It is defined as follows:

$$E_s(\boldsymbol{z}, \boldsymbol{x}_{h-o}) = \frac{\sum z_s(\boldsymbol{z})\left(1 - r_s(\boldsymbol{x}_{h-o})\right)}{\sum z_s(\boldsymbol{z})} + \frac{\sum r_s(\boldsymbol{x}_{h-o})\left(1 - z_s(\boldsymbol{z})\right)}{\sum r_s(\boldsymbol{x}_{h-o})} \tag{3}$$

The first part in E_s calculates the pixel area that belongs to $z_s(\boldsymbol{z})$ but does not belong to $r_s(\boldsymbol{x}_{h-o})$, whereas the second part calculates the pixel area that belongs to $r_s(\boldsymbol{x}_{h-o})$ but does not belong to $z_s(\boldsymbol{z})$. Both parts are normalized independently.

To punish the mutual penetration of adjacent fingers, the matching error function $E(\boldsymbol{z}, \boldsymbol{x}_h)$ involves an additional prior part, which is the penalty term $E_h(\boldsymbol{x}_h)$. It is defined as follows:

$$E_h(\boldsymbol{x}_h) = \sum_{p \in J} -\min\left(\varphi\left(\boldsymbol{x}_h, p\right), 0\right) \tag{4}$$

where J refers to three pairs of adjacent fingers, except the thumb. φ refers to the difference between the abduction-adduction angles of the MCP joints between a pair of adjacent fingers in the hand pose hypothesis $\boldsymbol{x}_h$.

The observation likelihood function is defined as follows:

$$p(\boldsymbol{z}|\boldsymbol{x}_{h-o}) \propto \exp\left(-\lambda_e \cdot E(\boldsymbol{z}, \boldsymbol{x}_{h-o})\right) \tag{5}$$

where λ_e is a normalization factor.

IV. THE TRACKING ALGORITHM

We propose a new tracking algorithm — collaborative differential evolution filtering (CoDEF) — for tracking hand-object interactions. CoDEF integrates the differential evolution (DE) algorithm into a particle filtering (PF) framework. The distribution of the PF samples is improved by optimizing the matching error under the current observation with DE. In addition, CoDEF uses two collaborative trackers to track the movement of the hand and object. In this way, the hand-object space is decomposed and the complexity of the optimum searching is decreased.

A. PARTICLE FILTERING

Particle filtering (PF) can express a multipeak distribution through the propagation of multiple samples along time [50]. The basic idea of PF can be summarized as follows. According to the particle samples $\{(\boldsymbol{x}_{t-1}^i, w_{t-1}^i)\}_{i=1}^N$ of time $t-1$, PF searches for a group of samples $\{(\boldsymbol{x}_t^i, w_t^i)\}_{i=1}^N$ to represent the posterior probability distribution of time t, by using the transition prior $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ and the observation likelihood $p(\boldsymbol{z}_t|\boldsymbol{x}_t)$. $\boldsymbol{x}_t^i$ denotes the i-th sampled state particle at time t, and w_t^i denotes its weight. However, the transition prior which ignores the latest observation value $\boldsymbol{z}_t$ is used as the importance distribution. Therefore, the importance sampling process of particles is suboptimal. For PF, a small sample set will lead to particle divergence and tracking failures. To address this problem, some kind of optimization method is often introduced into the PF framework to accelerate the convergence of the particles.

B. OPTIMIZATION WITH DIFFERENTIAL EVOLUTION

In this paper, we use the differential evolution (DE) algorithm to optimize the matching error. DE is an efficient swarm intelligence optimization algorithm for nonlinear and non-differentiable objective functions [51]. After initialization, DE searches for the optimal global solution in a continuous space through iterative evolutions of N D-dimensional vectors $\{\boldsymbol{x}_g^i\}_{i=1}^N$. Population evolution is completed through mutation, crossover, and selection. Mutation and crossover are used to generate new candidates, whereas selection is used to determine whether the new candidate can survive the next generation.

During mutation, DE selects three different individuals randomly from the previous generation for each individual index i of the population, which are combined to generate a

mutant individual:

$$\boldsymbol{v}_{g+1}^{i} = \boldsymbol{x}_{g}^{r_1} + F(\boldsymbol{x}_{g}^{r_2} - \boldsymbol{x}_{g}^{r_3}) \tag{6}$$

where individual indexes r_1, r_2 and r_3 are selected randomly within the range of $[1, 2, \cdots, N]$. These three individual indexes are different from each other and different from i. F is the scaling factor of the differential vector $(\boldsymbol{x}_{g}^{r_2} - \boldsymbol{x}_{g}^{r_3})$, and it controls the convergence speed during the search process. The scaling factor F of the standard DE algorithm is constant. To improve the convergence of the algorithm, in this paper, F is adjusted on each dimension by using a "jitter" [52] factor. Therefore, $F = F_C \cdot N(0, 1)$, where F_C is a constant and $N(0, 1)$ is a Gaussian random number that has a mean of 0 and a variance of 1. In this paper, F_C is set to 0.5.

Then, a candidate $\boldsymbol{u}_{g+1}^{i} = \{u_{g+1}^{j,i}\}_{j=1}^{D}$ is generated by combining the mutant individual $\boldsymbol{v}_{g+1}^{i}$ and the old individual $\boldsymbol{x}_{g}^{i}$ through the crossover operation:

$$u_{g+1}^{j,i} = \begin{cases} v_{g+1}^{j,i} & \text{if } rand^{j} \le CR \text{ or } j = r_{g+1}^{i} \\ x_{g}^{j,i} & \text{otherwise} \end{cases} \tag{7}$$

where $rand^{j} \sim U(0, 1)$ is a random number, which follows a uniform distribution over the interval [0,1]. The crossover parameter CR determines the probability for each element in a candidate to inherit from the mutant individual. In this paper, CR is set to 0.9. r_{g+1}^{i} is a random number in the range of $[1, 2, \cdots, D]$, which ensures that candidates choose at least one element from the mutant individual.

After the mutation and crossover operations are completed, a one-to-one greedy selection operation is conducted:

$$\boldsymbol{x}_{g+1}^{i} = \begin{cases} \boldsymbol{u}_{g+1}^{i} & \text{if } f(\boldsymbol{u}_{g+1}^{i}) \le f(\boldsymbol{x}_{g}^{i}) \\ \boldsymbol{x}_{g}^{i} & \text{otherwise} \end{cases} \tag{8}$$

The generated candidate $\boldsymbol{u}_{g+1}^{i}$ and the old individual $\boldsymbol{x}_{g}^{i}$ are compared to determine which one should be retained in the next generation. If $\boldsymbol{u}_{g+1}^{i}$ has a better objective function value than $\boldsymbol{x}_{g}^{i}$, it will replace $\boldsymbol{x}_{g}^{i}$ in the next generation. Otherwise, $\boldsymbol{x}_{g}^{i}$ is retained.

The basic steps of the DE algorithm can be summarized as follows:

(1) Initialization: The population $\{\boldsymbol{x}_{0}^{i}\}_{i=1}^{N}$ is initialized randomly. The individuals in the population are evaluated according to the objective function, and the corresponding objective values are recorded. The individual with the best objective value in $\{\boldsymbol{x}_{0}^{i}\}_{i=1}^{N}$ is duplicated into the global optimum $\boldsymbol{b}_0$ of the population, with its corresponding objective value recorded.
(2) Mutation: Mutation is carried out on individuals in the population according to Equation (6) to generate the mutant individual $\boldsymbol{v}_{g+1}^{i}$.
(3) Crossover: The old individual $\boldsymbol{x}_{g}^{i}$ and its corresponding mutant individual $\boldsymbol{v}_{g+1}^{i}$ are crossed over according to Equation (7) to generate the candidate $\boldsymbol{u}_{g+1}^{i}$.
(4) Evaluate all candidates: The generated candidates $\{\boldsymbol{u}_{g+1}^{i}\}_{i=1}^{N}$ are evaluated according to the objective function, and their corresponding objective values are recorded.
(5) Selection: The old individual $\boldsymbol{x}_{g}^{i}$ or the candidate $\boldsymbol{u}_{g+1}^{i}$ is selected to be retained in the new population according to Equation (8).
(6) Update the global optimal: The objective values of all new individuals $\{\boldsymbol{x}_{g+1}^{i}\}$ are compared with the global optimum $\boldsymbol{b}_g$ to generate a new global optimum $\boldsymbol{b}_{g+1}$.
(7) Determine whether the algorithm is over: If it is, output $\boldsymbol{b}_{g+1}$ and its corresponding objective value and quit the algorithm. Otherwise, return to Step (2).

In this paper, two DE populations are assigned to the hand and object, respectively, for their pose optimization. Specifically, the hand pose $\boldsymbol{x}_h$ and object pose $\boldsymbol{x}_o$ under the current frame are respectively optimized by these two populations. Here, we denote these two populations as population h and population o, respectively. Population h conducts the iterative optimization of the hand pose $\boldsymbol{x}_h$ while regarding object pose $\boldsymbol{x}_o$ as static for the current frame. In population h, $\boldsymbol{x}_o$ is determined by the optimization result of population o for the previous frame. In contrast, population o conducts the iterative optimization of the object pose $\boldsymbol{x}_o$ while regarding hand pose $\boldsymbol{x}_h$ as static for the current frame. In population o, $\boldsymbol{x}_h$ is determined by the optimization result of population h for the previous frame.

C. COLLABORATIVE DIFFERENTIAL EVOLUTION FILTERING

We integrate the DE algorithm into the PF framework for tracking hand-object interactions. After the new positions of the particles are predicted, the DE algorithm is carried out to conduct the iterative evolution of the particles, by using the matching error function under the latest observation $\boldsymbol{z}_t$ as the objective function. Particles are moved to regions with higher observation likelihoods in the state space via DE. The particle optimization process can be regarded as an importance sampling process, whereas the new particle swarm after optimization can be regarded as an approximation of the optimal importance distribution $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1}, \boldsymbol{z}_t)$ [50]. The optimization process based on DE improves the distribution of PF samples and accelerates the convergence of the particle set, thus enabling robust hand-object tracking using fewer particles.

As Equation (9) shows, the transition prior $p(\boldsymbol{x}_t|\boldsymbol{x}_{t-1})$ is defined as a first-order dynamics model to propagate particles along time:

$$\boldsymbol{x}_{t,0}^{i} = \boldsymbol{x}_{t-1,G}^{i} + \boldsymbol{r}_{t-1}^{i} \tag{9}$$

where $\boldsymbol{r}_{t-1}^{i}$ is a Gaussian random number. $\{\boldsymbol{x}_{t-1,G}^{i}\}_{i=1}^{N}$ are the final positions gained from particle convergence after G generations of iterative optimization via DE at time t–1. The newly obtained particle set $\{\boldsymbol{x}_{t,0}^{i}\}_{i=1}^{N}$ is used to initialize the DE population at time t. The improved algorithm — differential evolution filtering (DEF) — is summarized as follows:

For time $t > 0$:

(1) Resampling: Particles are resampled from the particle set $\{(\boldsymbol{x}_{t-1}^i, w_{t-1}^i)\}_{i=1}^N$ according to the weights $\{w_{t-1}^i\}_{i=1}^N$, generating a new equal-weighted particle set $\{(\boldsymbol{x}_{t-1}^i, 1/N))\}_{i=1}^N$.
(2) Prediction: According to Equation (9), the position of each particle at time t is predicted from its position at time t-1, thus obtaining a new particle set $\{(\boldsymbol{x}_{t,0}^i, 1/N))\}_{i=1}^N$.
(3) Optimization: Using the matching error function under the latest observation $\boldsymbol{z}_t$ as the objective function, run the DE algorithm to optimize $\{(\boldsymbol{x}_{t,0}^i, 1/N))\}_{i=1}^N$.
(4) Weight updating: The particle weight $w_t^i \propto p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$ is updated according to the observation likelihood $p(\boldsymbol{z}_t|\boldsymbol{x}_t^i)$, and a weighted particle set $\{(\boldsymbol{x}_t^i, w_t^i)\}_{i=1}^N$ is obtained. Then, the weights $\{w_t^i\}_{i=1}^N$ are normalized to make $\sum_{i=1}^N w_t^i = 1$.
(5) State estimation: Output the estimates of the system state by using the maximum posterior criteria.

In this paper, two collaborative DEF trackers are applied for hand-object movement tracking and we propose a new algorithm — collaborative differential evolution filtering (CoDEF). The proposed CoDEF algorithm assigns two trackers to the hand and object to track the hand pose $\boldsymbol{x}_h$ and object pose $\boldsymbol{x}_o$. The two trackers are not independent of each other and they exchange information frequently during the tracking process. The hand tracker regards object pose $\boldsymbol{x}_o$ as static during the iterative optimization of hand pose $\boldsymbol{x}_h$ at the current frame, while $\boldsymbol{x}_o$ is determined by the tracking result of the object tracker for the previous frame. The object tracker regards hand pose $\boldsymbol{x}_h$ as static during the iterative optimization of object pose $\boldsymbol{x}_o$ at the current frame, while $\boldsymbol{x}_h$ is determined by the tracking result of the hand tracker for the previous frame. As soon as one tracker gains the solution for the current frame, the solution is transmitted to the other tracker, and the corresponding pose values are kept static during the iterative optimization for the next frame by the other tracker. Such a collaborative tracking scheme not only models occlusions between the hand and object, but it also decomposes the unified state space with multiple trackers, decreasing the complexity of optimum searching.

V. DEVELOPMENT OF THE TRACKING SYSTEM

We develop a prototype system for tracking hand-object interactions using the proposed CoDEF algorithm with the graphic rendering engine OpenSceneGraph (OSG)[3]. A prebuilt 3D hand-object model with DOF nodes is loaded into OSG. During the tracking process, the movement of the hand and the object is controlled by using osgSim::DOFTransform nodes. The depth images of the hand-object model are generated by OSG off-screen rendering, which are then compared with the observed images to calculate the matching errors and observation likelihood values for different particles. The state parameters for the minimum matching error are searched for within the hand space and the object space using the CoDEF algorithm.

OSG organizes spatial data in a scene graph tree for efficient graphic rendering. Headed by a root node on the top, the scene graph tree is composed of many group nodes and leaf nodes. The group nodes organize the geometries and their rendering states in a scene, whereas leaf nodes contain the actual geometric data for rendering. As an object-oriented rendering engine, OSG provides various group node types by using inheritance, such as transform nodes and camera nodes, allowing for many different functionalities. In our system, we create a camera node to render the hand-object pose hypotheses into depth images for matching error calculations. The camera node has a child, the hand-object model node, which is created by reading the corresponding model file. In addition, to allow for off-screen rendering, we connect a buffer object with the camera. Then, the hand-object model will be rendered onto the buffer object by the virtual camera per OSG frame. For each rendered frame, OSG performs three traversals: the update, cull and draw traversals. In the update traversal step, updates are made to the scene graph to enable dynamic scenes. Our system updates the model poses with a callback object (NodeCallback) assigned to the model node in this traversal. In the cull traversal step, OSG tests the bounding volumes of all nodes and culls the nodes that are not in the view. For our system, no special operations are added to this traversal. In the draw traversal step, OSG traverses the list of geometries created by the cull traversal and invokes drawing commands to render the geometries. In our system, for each OSG frame, after the pose-updated model is rendered into a depth image by the virtual camera, the matching error for the new pose hypothesis is calculated in this traversal through a callback object (DrawCallback) assigned to the camera.

This system calculates new hand-object pose parameters iteratively using the CoDEF algorithm. As shown in Fig. 3, the system sets up a DE population h for the optimization of the hand poses and a DE population o for the optimization of the object poses. After getting a new input frame from the depth observation, the system propagates the hand poses of all particles in population h over time for the population initialization of the new frame. The object poses of the particles in population h are set to the best object pose attained by population o for the last input frame. In addition, to initialize population o, the system propagates the object poses of all particles in population o over time. The hand poses of the particles in population o are set to the best hand pose attained by population h for the last input frame. After initialization, the two DE populations iteratively optimize their particles based on the new observation. When a new candidate is generated through mutation and crossover, the system updates the model pose in the NodeCallback object of the model node according to the position of the new candidate. A new OSG frame is set up, and the updated model is rendered into a depth image by the virtual camera. In the DrawCallback object assigned to the camera, the system calculates the

[3]http://www.openscenegraph.org/

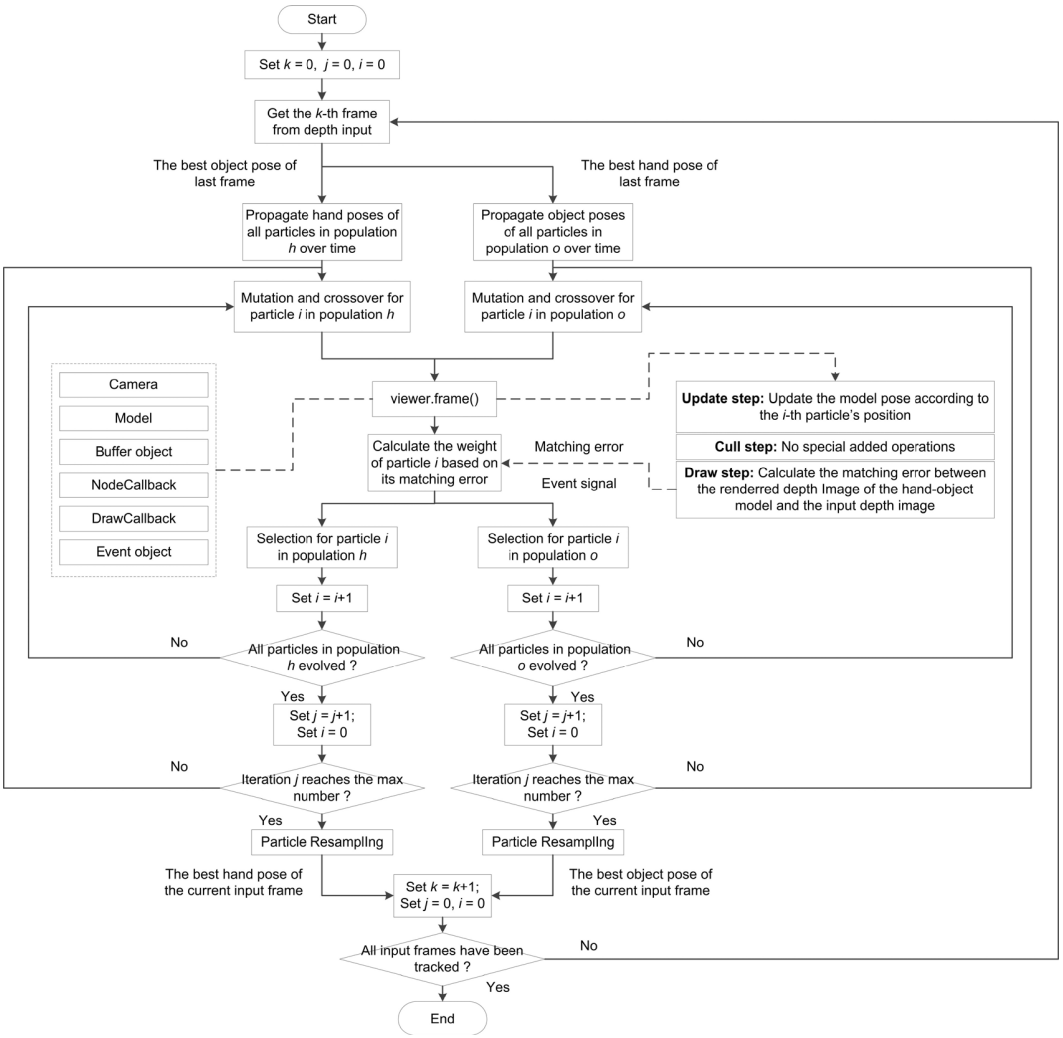

FIGURE 3. Flowchart of the prototype system.

matching error of the new candidate. The rendering of OSG frames is conducted by a multithread mode as the default. In the multithread mode, a thread is assigned to each camera and each graphics context. The cull and draw traversals are conducted in the threads of the cameras and graphics contexts, respectively. Before the current frame finishes drawing in the graphics context threads, the update traversal and cull traversal of the next frame will be started. To avoid data conflicts among different threads, our system uses the Win32 SetEvent() and WaitForSingleObject() functions for synchronization and communication among threads. When the matching error has been calculated, a signal is sent to the main thread by an event object. When this event signal is received, the system calculates the weight of the new candidate particle based on its matching error in the main thread. Then, a selection operation is conducted to decide whether the old individual or the new candidate will be retained. After a fixed number of iterations, the system combines the best hand and object poses attained respectively by the two populations as the solution.

VI. EXPERIMENTS

The effectiveness of the tracking method is verified by experiments on real sequences and synthetic sequences. The tracking is initialized manually by putting the real hand and object in their initial positions at the first input frame. In all experiments, the proposed CoDEF algorithm applies 32 particles for the hand tracker and 8 particles for the object tracker. For each input frame of the two trackers, the DE algorithm conducts 60 generations of iterative optimization. In this paper, the experiments are carried out on a PC with a quad-core Core i5 2.9 GHz CPU, 8.0 GBs of memory, and an Nvidia GTX 950M GPU. Tracking one input frame costs 5 s on average.

A. EXPERIMENTS ON REAL IMAGES

We use depth images, which are captured from a Kinect 1.0 sensor with the Microsoft Kinect 1.0 Beta2 SDK, as the observation input. The image resolution and frame rate are 640 $\times$ 480 and 30 fps, respectively. Two depth image sequences have been acquired. The first one shows a hand grasping and manipulating a sphere, whereas the second

TABLE 1. The tracking errors of CoDEF on the synthetic sequences.

Tracking error	Sphere		Cylinder	
	Mean value	Stdev.	Mean value	Stdev.
Hand positional error	1.5748 mm	0.7875 mm	2.0947 mm	2.2316 mm
Object positional error	1.2360 mm	1.5180 mm	1.6224 mm	2.6831 mm
Hand pose error	4.4059 °	2.1763 °	3.4758 °	1.9869 °

TABLE 2. The errors of the estimated parameters on the synthetic hand-sphere sequence.

Estimated parameter	Mean error	Stdev.
Palm y-axis rotation	0.4381 °	0.4352 °
Ring finger PIP flexion	3.0811 °	4.1654 °
Pinky finger MCP flexion	1.3276 °	2.4450 °
Thumb finger MCP flexion	2.0286 °	3.1980 °
Thumb finger TM flexion	0.7049 °	0.7481 °
Object x-axis translation	0.2271 mm	0.2842 mm

TABLE 3. The errors of the estimated parameters on the synthetic hand-cylinder sequence.

Estimated parameter	Mean error	Stdev.
Palm y-axis rotation	0.7891 °	1.0162 °
Middle finger PIP flexion	3.2982 °	5.9761 °
Ring finger MCP flexion	1.5973 °	2.2799 °
Pinky finger MCP flexion	1.6143 °	2.7798 °
Thumb finger TM flexion	1.4310 °	2.1650 °
Object x-axis translation	0.3606 mm	0.5085 mm

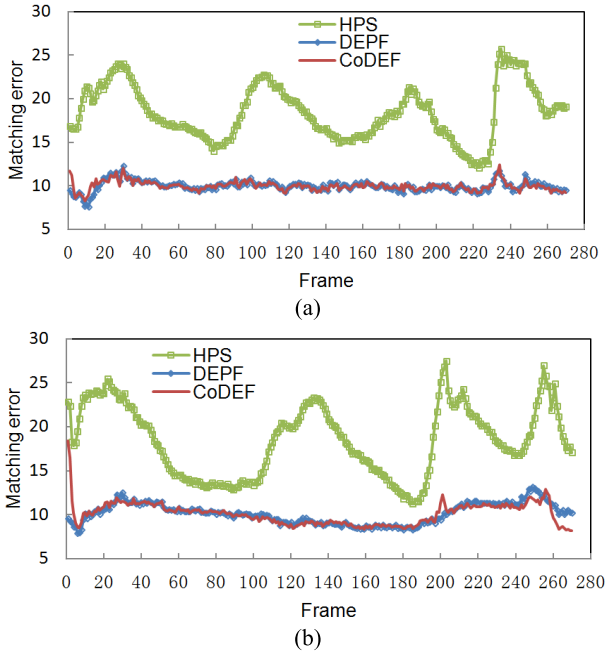

FIGURE 4. Matching errors of HPS, DEPF, and CoDEF on the real sequences. (a) Sphere, (b) Cylinder.

one shows a hand grasping and manipulating a cylinder. Both of the sequences consist of 270 frames. Experiments are conducted on the two real sequences to evaluate the proposed CoDEF algorithm. We compare CoDEF with two algorithms: another improved PF algorithm with DE operators (DEPF) [46] and a hybrid particle-based search (HPS) algorithm that derives from PSO and DE [33]. Both DEPF and HPS track in the hand-object joint pose space. In all experiments, both DEPF and HPS apply 40 particles and run for 60 generations for each input frame. The configuration of these parameters allows for a fair comparison among the three algorithms, since for each input frame, the three algorithms calculate the same numbers of matching errors.

The matching error values attained by the three tracking algorithms are plotted in Fig. 4. It can be seen from Fig. 4 that both CoDEF and DEPF outperform HPS on the two real sequences. CoDEF and DEPF have nearly equal performance in terms of matching errors. Then, we compare the validity of the estimates attained by CoDEF and DEPF by reconstructing the hand-object poses with the estimates. Fig. 5 and Fig. 6 show the 3D reconstruction of the results of CoDEF and DEPF on the real sequences. For Fig. 5 and Fig. 6, the first and second rows show the RGB images and depth images, respectively, which are acquired by the Kinect sensor. Here, the depth images have been segmented by a simple depth threshold. The third row shows the tracking results of DEPF. The fourth row shows the tracking results of the proposed CoDEF algorithm. Athough CoDEF and DEPF have nearly equal performance in terms of matching errors, the 3D reconstruction of the tracking results shows that CoDEF actually performs better than DEPF. Especially for the real hand-cylinder sequence, when severe occlusions happen, DEPF can not achieve accurate tracking, whereas CoDEF still tracks hand-object movement correctly.

B. EXPERIMENTS ON SYNTHETIC DATA

We conduct a quantitative evaluation of the proposed CoDEF algorithm based on synthetic depth images, since ground truth pose data are hard to acquire from real images. The synthetic images are rendered using the 3D hand-object models. In addition, the movement of the hand-object models is defined by the tracking results of CoDEF on the two real sequences. Therefore, for these synthetic sequences, the CoDEF tracking results on the real sequences are actually the ground truth values. The resulting two synthetic sequences both consist of 270 frames. By using synthetic data as the observation, experiments are carried out to evaluate the CoDEF tracking algorithm. Table 1 shows the tracking

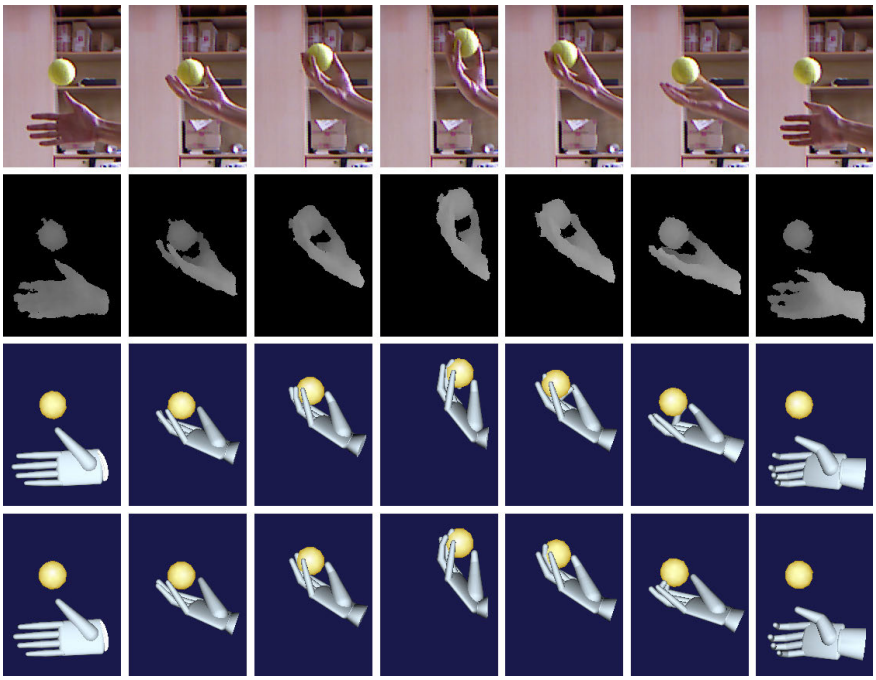

FIGURE 5. Sample results of CoDEF on the real hand-sphere sequence compared with DEPF. The results of frames 0, 30, 90, 120, 170, 230, 269 are shown. From top to bottom: RGB images, depth images, the results of DEPF and results of CoDEF.

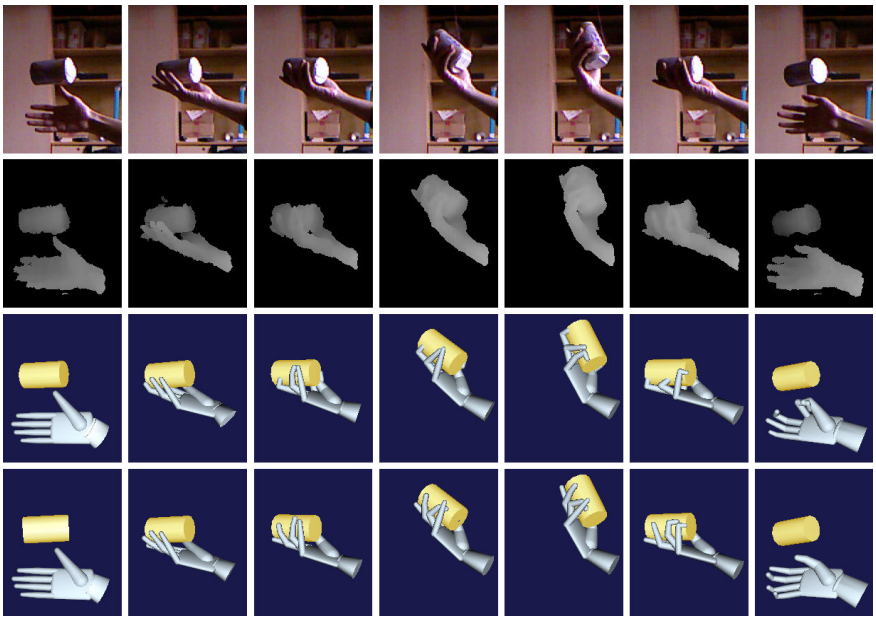

FIGURE 6. Sample results of CoDEF on the real hand-cylinder sequence compared with DEPF. The results of frames 0, 30, 60, 120, 170, 230, 269 are shown. From top to bottom: RGB images, depth images, the results of DEPF and results of CoDEF.

errors of CoDEF that are averaged over the entire sequence, including the positional errors of the hand and object, and the pose error of the hand. The positional error of the hand is the Euclidean distance between the estimated and ground truth positions of the palm center, whereas the positional error of the object is the Euclidean distance between the estimated and ground truth object positions. The pose error of the hand is the average angle error of the 25 rotation DOFs of the hand, including 3 DOFs for global hand rotation, 20 DOFs for local finger motion, and 2 DOFs for wrist rotation.

Comparisons between the estimates of the proposed CoDEF algorithm and the corresponding ground truth values on some parameters are shown in Fig. 7 and Fig. 8. Table 2 and Table 3 show the mean errors of the estimated parameters

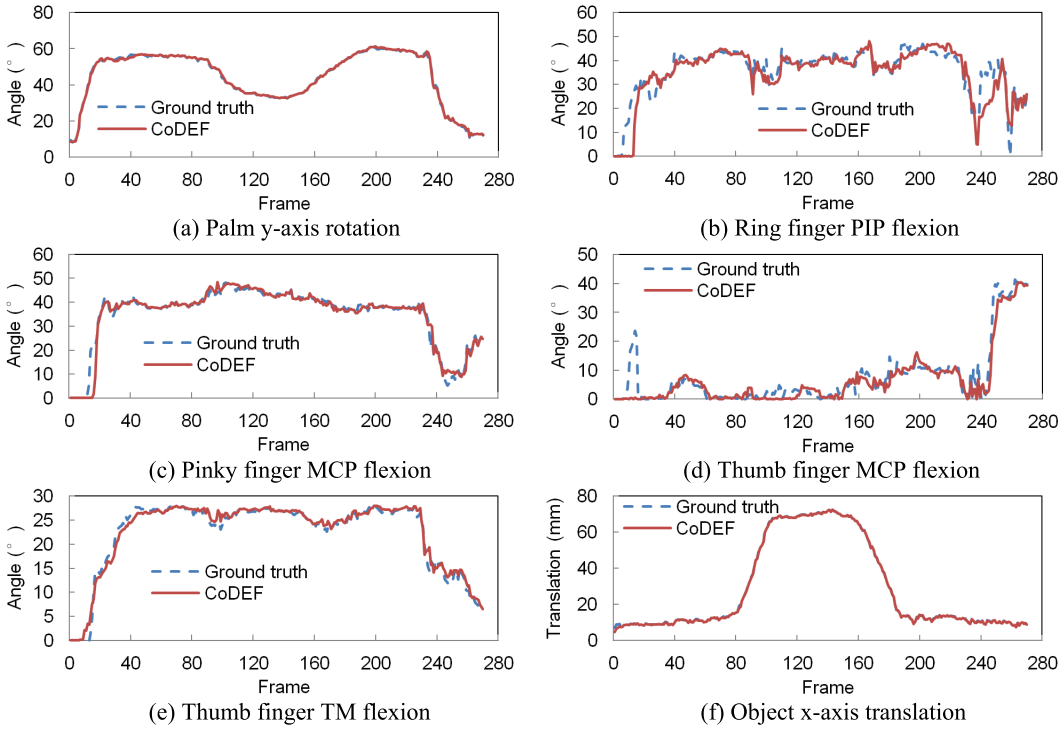

FIGURE 7. CoDEF tracking results vs. ground truth on the synthetic hand-sphere sequence.

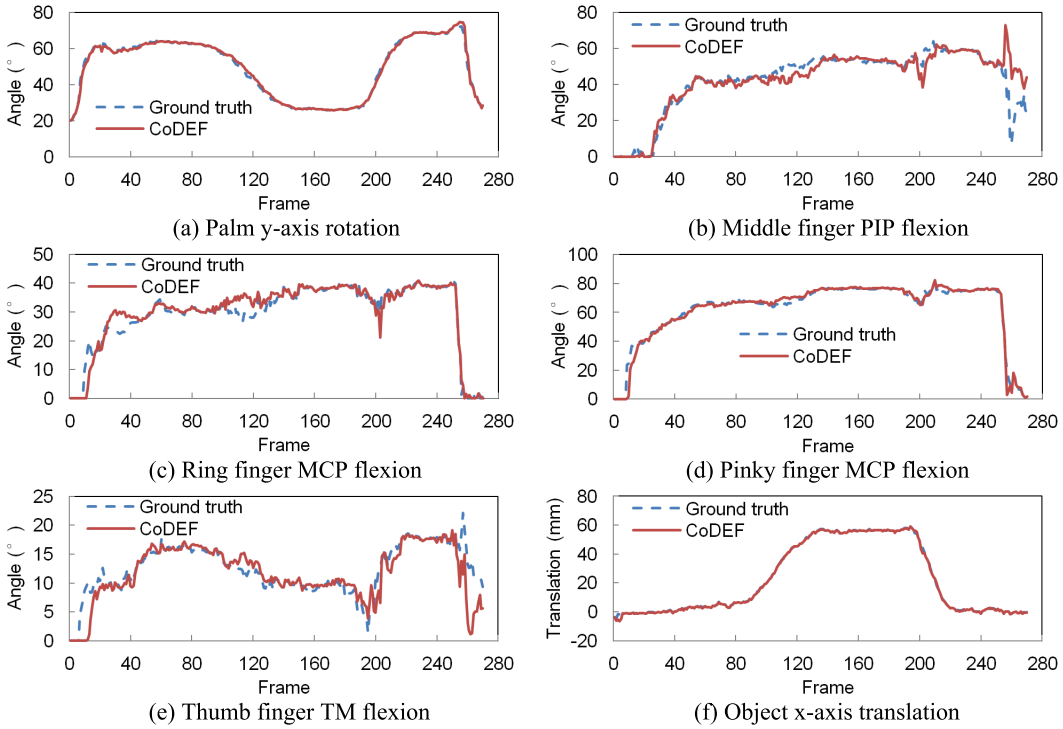

FIGURE 8. CoDEF tracking results vs. ground truth on the synthetic hand-cylinder sequence.

on the sequence and the corresponding standard deviations. The results show that the parameters estimated by CoDEF can follow the changes of the ground truth values along the sequence.

VII. CONCLUSION

In this paper, we propose an improved PF algorithm — CoDEF — to track hand-object interactions. We construct hand-object models with geometric primitives and establish

an observation model with depth observation. The proposed CoDEF algorithm integrates the DE algorithm into the PF framework. By optimizing the matching error with DE under the current observation, the PF sampling process is improved and the particles are moved towards the areas with a high probability. In addition, CoDEF tracks the movement of the hand and object by using two collaborative trackers. In this way, the hand-object space is decomposed and the complexity of optimum searching is decreased. We develop a prototype system using the proposed CoDEF algorithm with 3D graphic techniques. Experiments demonstrate that the proposed algorithm can achieve robust tracking of hand-object movement using fewer particles.

Since the proposed method is model-based, the tracking needs to be initialized, which is performed manually by putting the real hand and object in their initial positions at the first input frame. To make the method able to initialize automatically and enhance its capability to recover from tracking failures, our future research will combine some kind of learning-based method with model fitting for tracking hand-object interactions. We will use the learning-based method to predict a distribution for the hand-object poses. Then, using the hand and object hypotheses sampled from the distribution for initializing, the model-based tracking will be performed to estimate the hand and object poses. In this paper, CoDEF cannot track hand-object movement in real time. According to the parallel computing characteristics of the proposed CoDEF algorithm and matching error calculation, in the future, we will speed up the system by using CUDA programming.

REFERENCES

[1] H. Kjellström, J. Romero, D. Martinez, and D. Kragic, "Simultaneous visual recognition of manipulation actions and manipulated objects," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Marseille, France, Oct. 2008, pp. 336–349.

[2] J. Romero, H. Kjellstrom, and D. Kragic, "Monocular real-time 3D articulated hand pose estimation," in *Proc. 9th IEEE-RAS Int. Conf. Humanoid Robots*, Paris, France, Dec. 2009, pp. 87–92.

[3] J. Romero, H. Kjellstroám, and D. Kragic, "Hands in action: Real-time 3D reconstruction of hands in interaction with objects," in *Proc. IEEE Int. Conf. Robot. Autom.*, Anchorage, AK, USA, May 2010, pp. 458–463.

[4] A. Gupta, A. Kembhavi, and L. S. Davis, "Observing human-object interactions: Using spatial and functional compatibility for recognition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 10, pp. 1775–1789, Oct. 2009.

[5] B. Yao and L. Fei-Fei, "Modeling mutual context of object and human pose in human-object interaction activities," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Francisco, CA, USA, Jun. 2010, pp. 17–24.

[6] B. Yao and L. Fei-Fei, "Recognizing human-object interactions in still images by modeling the mutual context of objects and human poses," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 9, pp. 1691–1703, Sep. 2012.

[7] J. Shin and C. M. Kim, "Non-touch character input system based on hand tapping gestures using kinect sensor," *IEEE Access*, vol. 5, pp. 10496–10505, 2017.

[8] C. Keskin, F. Kiraç, Y. E. Kara, and L. Akarun, "Hand pose estimation and hand shape classification using multi-layered randomized decision forests," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Florence, Italy, Oct. 2012, pp. 852–863.

[9] C. Keskin, F. Kiraç, Y. E. Kara, and L. Akarun, "Real time hand pose estimation using depth sensors," in *Consumer Depth Cameras for Computer Vision*. London, U.K.: Springer, 2013, pp. 119–137.

[10] V. A. Prisacariu and I. Reid, "3D hand tracking for human computer interaction," *Image Vis. Comput.*, vol. 30, no. 3, pp. 236–250, Mar. 2012.

[11] D. Liu, S. Arai, J. Miao, J. Kinugawa, Z. Wang, and K. Kosuge, "Point pair feature-based pose estimation with multiple edge appearance models (PPF-MEAM) for robotic bin picking," *Sensors*, vol. 18, no. 8, 2719, pp. 1–20, Aug. 2018.

[12] J. Tompson, M. Stein, Y. Lecun, and K. Perlin, "Real-time continuous pose recovery of human hands using convolutional networks," *ACM Trans. Graph.*, vol. 33, no. 5, pp. 1–10, Sep. 2014.

[13] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Real-time 3D hand pose estimation with 3D convolutional neural networks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 41, no. 4, pp. 956–970, Apr. 2019.

[14] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "Robust 3D hand pose estimation in single depth images: From single-view CNN to multi-view CNNs," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Las Vegas, NV, USA Jun. 2016, pp. 3593–3601.

[15] L. Ge, H. Liang, J. Yuan, and D. Thalmann, "3D convolutional neural networks for efficient and robust hand pose estimation from single depth images," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Honolulu, HI, USA Jul. 2017, pp. 1991–2000.

[16] L. Ge, Y. Cai, J. Weng, and J. Yuan, "Hand PointNet: 3D hand pose estimation using point sets," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, Utah, USA Jun. 2018, pp. 8417–8426.

[17] L. Ge, Z. Ren, and J. Yuan, "Point-to-point regression pointnet for 3D hand pose estimation," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 475–491.

[18] F. Mueller, D. Mehta, O. Sotnychenko, S. Sridhar, D. Casas, and C. Theobalt, "Real-time hand tracking under occlusion from an egocentric RGB-D sensor," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 1284–1293.

[19] Y. Kim and B. Toomajian, "Hand gesture recognition using micro-Doppler signatures with convolutional neural network," *IEEE Access*, vol. 4, pp. 7125–7130, 2016.

[20] Y. Cai, L. Ge, J. Cai, and J. Yuan, "Weakly-supervised 3D hand pose estimation from monocular RGB images," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep.2018, pp. 666–682.

[21] J. Malik, A. Elhayek, F. Nunnari, K. Varanasi, K. Tamaddon, A. Heloir, and D. Stricker, "DeepHPS: End-to-end estimation of 3D hand pose and shape by learning from synthetic depth," in *Proc. Int. Conf. 3D Vis. (3DV)*, Verona, Italy, Sep. 2018, pp. 110–119.

[22] A. Kanazawa, M. J. Black, D. W. Jacobs, and J. Malik, "End-to-end recovery of human shape and pose," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, Utah, USA, Jun. 2018, pp. 7122–7131.

[23] F. Mueller, F. Bernard, O. Sotnychenko, D. Mehta, S. Sridhar, D. Casas, and C. Theobalt, "GANerated hands for real-time 3D hand tracking from monocular RGB," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, Utah, USA, Jun. 2018, pp. 49–59.

[24] U. Iqbal, P. Molchanov, T. Breuel, J. Gall, and J. Kautz, "Hand pose estimation via latent 2.5D heatmap regression," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Munich, Germany, Sep. 2018, pp. 118–134.

[25] C. Zimmermann and T. Brox, "Learning to estimate 3D hand pose from single RGB images," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Venice, Italy, Oct. 2017, pp. 4903–4911.

[26] C. Wan, T. Probst, L. V. Gool, and A. Yao, "Dense 3D regression for hand pose estimation," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit.*, Salt Lake City, Utah, USA, Jun. 2018, pp. 5147–5156.

[27] A. Boukhayma, R. de Bem, and P. H. S. Torr, "3D hand shape and pose from images in the wild," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Long Beach, CA, USA, Jun. 2019, pp. 10843–10852.

[28] C. Chen, T. Wang, D. Li, and J. Hong, "Repetitive assembly action recognition based on object detection and pose estimation," *J. Manuf. Syst.*, vol. 55, pp. 325–333, Apr. 2020.

[29] A. Erol, G. Bebis, M. Nicolescu, R. D. Boyle, and X. Twombly, "A review on vision-based full DOF hand motion estimation," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recognit.*, San Diego, CA, USA, Sep. 2005, pp. 75–82.

[30] H. Hamer, K. Schindler, E. Koller-Meier, and L. Van Gool, "Tracking a hand manipulating an object," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Kyoto, Japan, Sep. 2009, pp. 1475–1482.

[31] I. Oikonomidis, N. Kyriazis, and A. A. Argyros, "Full DOF tracking of a hand interacting with an object by modeling occlusions and physical constraints," in *Proc. Int. Conf. Comput. Vis.*, Barcelona, Spain, Nov. 2011, pp. 2088–2095.

[32] N. Kyriazis and A. Argyros, "Physically plausible 3D scene tracking: The single actor hypothesis," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Portland, OR, USA, Jun. 2013, pp. 9–16.
[33] Z. Zhang and H. S. Seah, "Skeleton body pose tracking from efficient three-dimensional motion estimation and volumetric reconstruction," *Appl. Opt.*, vol. 51, no. 23, pp. 5686–5697, Aug. 2012.
[34] I. Oikonomidis, N. Kyriazis, and A. Argyros, "Efficient model-based 3D tracking of hand articulations using kinect," in *Proc. Procedings Brit. Mach. Vis. Conf.*, Univ. Dundee, U.K., 2011, p. 101.
[35] N. Kyriazis and A. Argyros, "Scalable 3D tracking of multiple interacting objects," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Columbus, OH, USA, Jun. 2014, pp. 3430–3437.
[36] P. Doliotis, V. Athitsos, D. Kosmopoulos, and S. Perantonis, "Hand shape and 3D pose estimation using depth data from a single cluttered frame," in *Proc. ISVC*, Rethymnon, Greece, Jul. 2012, pp. 148–158.
[37] Y. Wang, J. Min, J. Zhang, Y. Liu, F. Xu, Q. Dai, and J. Chai, "Video-based hand manipulation capture through composite motion control," *ACM Trans. Graphic.*, vol. 32, no. 4, pp. 1–14, Jul. 2013.
[38] W. Zhao, J. Chai, and Y. Q. Xu, "Combining marker-based mocap and RGB-D camera for acquiring high-fidelity hand motion data," in *Proc. ACM SIGGRAPH/Eurograph. Symp. Comput. Animat. (SCA)*, Lausanne, Switzerland, Jul. 2012, pp. 33–42.
[39] H. Liang, J. Yuan, D. Thalmann, and Z. Zhang, "Model-based hand pose estimation via spatial-temporal hand parsing and 3D fingertip localization," *Vis. Comput.*, vol. 29, nos. 6–8, pp. 837–848, Jun. 2013.
[40] T. Sharp, Y. Wei, D. Freedman, P. Kohli, E. Krupka, A. Fitzgibbon, S. Izadi, C. Keskin, D. Robertson, J. Taylor, J. Shotton, D. Kim, C. Rhemann, I. Leichter, and A. Vinnikov, "Accurate, robust, and flexible real-time hand tracking," in *Proc. 33rd Annu. ACM Conf. Hum. Factors Comput. Syst.*, Seoul, South Korea, Apr. 2015, pp. 3633–3642.
[41] S. Sridhar, F. Mueller, M. Zollhöfer, D. Casas, A. Oulasvirta, and C. Theobalt, "Real-time joint tracking of a hand manipulating an object from RGB-D input," in *Proc. Eur. Conf. Comput. Vis. (ECCV)*, Amsterdam, The Netherlands, Oct. 2016, pp. 294–310.
[42] J. Taylor, L. Bordeaux, T. Cashman, B. Corish, C. Keskin, T. Sharp, E. Soto, D. Sweeney, J. Valentin, B. Luff, A. Topalian, E. Wood, S. Khamis, P. Kohli, S. Izadi, R. Banks, A. Fitzgibbon, and J. Shotton, "Efficient and precise interactive hand tracking through joint, continuous optimization of pose and correspondences," *ACM Trans. Graph.*, vol. 35, no. 4, pp. 1–12, Jul. 2016.
[43] M. Bray, E. Koller-Meier, and L. Van Gool, "Smart particle filtering for 3D hand tracking," in *Proc. 6th IEEE Int. Conf. Autom. Face Gesture Recognit.*, Seoul, South Korea, May 2004, pp. 675–680.
[44] J. Cui and Z. Sun, "Visual hand motion capture for guiding a dexterous hand," in *Proc. 6th IEEE Int. Conf. Autom. Face Gesture Recognit.*, Seoul, South Korea, May 2004, pp. 729–734.
[45] D. Li and C. Chen, "Tracking a hand in interaction with an object based on single depth images," *Multimedia Tools Appl.*, vol. 78, no. 6, pp. 6745–6762, Mar. 2019.
[46] D. Li and Y. Zhou, "Combining differential evolution with particle filtering for articulated hand tracking from single depth images," *Int. J. Signal Process., Image Process. Pattern Recognit.*, vol. 8, no. 4, pp. 237–248, Apr. 2015.
[47] J. Deutscher, A. Blake, and I. Reid, "Articulated body motion capture by annealed particle filtering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Hilton Head Island, SC, USA, Jun 2000, pp. 126–133.
[48] Z. Zhang, H. S. Seah, C. K. Quah, and J. Sun, "GPU-accelerated real-time tracking of full-body motion with multi-layer search," *IEEE Trans. Multimedia*, vol. 15, no. 1, pp. 106–119, Jan. 2013.
[49] J. Lee and T. Kunii, "Constraint-based hand animation," in *Models and Techniques in Computer Animation*, N. M. Thalmann D. Thalmann, Eds. Tokyo, Japan: Springer, 1993, pp. 110–127.
[50] A. Doucet, S. Godsill, and C. Andrieu, "On sequential Monte Carlo sampling methods for Bayesian filtering," *Statist. Comput.*, vol. 10, no. 3, pp. 197–208, Jul. 2000.
[51] R. Storn and K. Price, "Differential evolution—A simple and efficient heuristic for global optimization over continuous spaces," *J. Global Optim.*, vol. 11, no. 4, pp. 341–359, 1997.
[52] D. Zaharie, "Critical values for the control parameters of differential evolution algorithms," in *Proc. 8th Int. Conf. Soft Comput.*, Brno, Czech Republic, Jun. 2002, pp. 62–67.

DONGNIAN LI received the B.Eng. and Ph.D. degrees in mechatronics from the School of Mechanical Engineering, Shandong University, China, in 2009 and 2015, respectively. He is currently a Lecturer with the School of Mechanical and Automotive Engineering, Qingdao University of Technology, China. His major research interests include computer vision, graphics, and virtual reality.

YANG GUO received the B.Sc. degree from the School of Mathematics and Information Science, Langfang Teachers College, in 2008, the M.Sc. degree in computer science from Shijiazhuang Tiedao University, in 2011, and the Ph.D. degree from the School of Mechanical Engineering, Shandong University, China, in 2015. He is currently an Associate Professor with the School of Mechanical and Automotive Engineering, Qingdao University of Technology, China. His current research interests include virtual reality, complex networks, and software engineering.

CHENGJUN CHEN (Member, IEEE) received the B.Eng. and Ph.D. degrees from the School of Mechanical Engineering, Shandong University, China, in 2003 and 2008, respectively. He is currently a Professor with the School of Mechanical and Automotive Engineering, Qingdao University of Technology, China. His major research interests include virtual reality and augmented reality in industrial applications.

ZHENGXU ZHAO (Senior Member, IEEE) received the B.Sc. and M.Sc. degrees from the School of Mechanical Engineering, Shandong University, China, in 1977 and 1982, respectively, and the Ph.D. degree in applied computing from Staffordshire University, U.K., in 1992. He was a Professor and the Chair in computer integrated manufacturing systems with the School of Computing, University of Derby, U.K., in 1998. From 2008 to 2019, he was a Professor with the Institute of Information Science and Technology, Shijiazhuang Tiedao University. He is currently a Professor with the School of Mechanical and Automotive Engineering, Qingdao University of Technology, China. He has contributed more than 200 research-level papers to national and international journals and conferences. His current research interests include computer graphics, virtual environment, software engineering, and knowledge retention.