International Journal of Signal Processing, Image Processing and Pattern Recognition
Vol. 8, No. 4 (2015), pp. 237-248
http://dx.doi.org/10.14257/ijsip.2015.8.4.21

Combining Differential Evolution with Particle Filtering for Articulated Hand Tracking from Single Depth Images

Dongnian Li[1,2] and Yiqi Zhou[1,2,*]

[1] Virtual Engineering Research Center, School of Mechanical Engineering,
Shandong University, Jinan 250061, China
[2] Key Laboratory of High Efficiency and Clean Mechanical Manufacture
(Shandong University), Ministry of Education, Jinan 250061, China
** corresponding author: yqzhou@sdu.edu.cn*

Abstract

Tracking articulated hand motion from visual observations is challenging mainly due to the high dimensionality of the state space. Dense sampling is difficult to be performed in such high-dimensional space, so the traditional particle filtering can't track articulated motion well. In this paper, we propose a new algorithm by combining differential evolution with a particle filter, to track the articulated motion of a hand from single depth images captured by a Kinect sensor. Through the optimization procedure of differential evolution, the particles are moved to the regions with a high likelihood. Only single depth information is used as the input, so our method is immune to illumination and background changes. The tracking system is developed with OpenSceneGraph (OSG). Experiments based on both synthetic and real image sequences demonstrate that the proposed method is capable of tracking articulated hand motion accurately and robustly.

Keywords: *articulated hand tracking, depth image, differential evolution, particle filter*

1. Introduction

Estimation and tracking of articulated hand motion from visual observations is an important technique that has a variety of applications, including, but not limited to, visual surveillance, computer animation, human-computer interaction, robot instruction. However, it is also a challenging problem in the area of computer vision, because of self-occlusions, the high-dimensional state space, and the time-varying dynamics of hand motions.

Various methods have been proposed to capture articulated hand motion. One kind of method is the appearance-based method [1-4], which estimates the hand poses directly from the images, by using a pre-learned mapping from the image features to the hand state space. These "bottom-up" methods are usually computationally efficient, but the accuracy of pose estimation depends on the training data collected for learning the mapping.

Another kind of method is the model-based method [5-12], which solves the problem in a "top-down" manner, by generating model hypotheses and then evaluating them on the visual observations. The task becomes a search for the state parameters that minimize the matching error between model features and observed image features. The block diagram for model-based tracking is shown in Figure 1. For these methods, the high dimensionality of the state space needs to be tackled specifically.

Model-based tracking is often addressed in a particle filter framework which can deal with multiple hypotheses [5-11]. However, in a high-dimensional space, the traditional particle filtering needs a large number of samples to represent the true

ISSN: 2005-4254 IJSIP

posterior, making the algorithm too slow. Therefore, some research focuses on reducing the dimensionality of the state space by using the learned strong motion prior models [5-6]. But these methods are restricted to specific activities and have problems dealing with general motions that are too different form the training set.

Other efforts have been devoted to providing a modified particle filter that works well with fewer samples by introducing some kind of optimization method [7-11]. During the optimization procedure, the particles are moved to the peaks of the posterior distribution. Gradient-based optimization [7-8], simulated annealing [9], swarm-based optimization [10-11] all have been used for this purpose. As the development of the computational power of computer hardware, swarm-based methods have attracted more attention. Cui, *et al.*, [10] introduce genetic algorithm into a particle filter to track articulated hand motion from RGB images. And Zhang, *et al.*, [11] integrate a niching particle swarm optimization (PSO) into a particle filter to capture full-body motion from volume data reconstructed with multiple RGB image views.

In this paper, we propose to combine differential evolution with a particle filter to track the articulated motion of a hand from single depth images captured by a Kinect sensor. Although, recently, Oikonomidis, *et al.*, [12] have proposed a model-based method for articulated hand tracking from the observations of a Kinect sensor, there are some differences between our work and theirs. [12] uses the RGB color plus depth observation as the input of the system, however, to make our system immune to illumination and background changes, we only use single depth information. And different from [12] using a PSO-based single hypothesis tracking method, we perform the tracking in a modified particle filter framework which incorporates an optimization procedure of differential evolution.

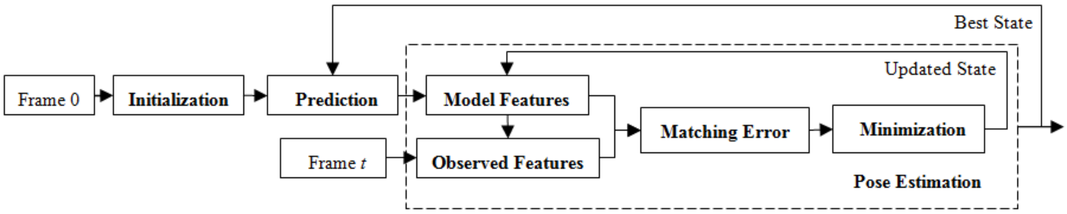
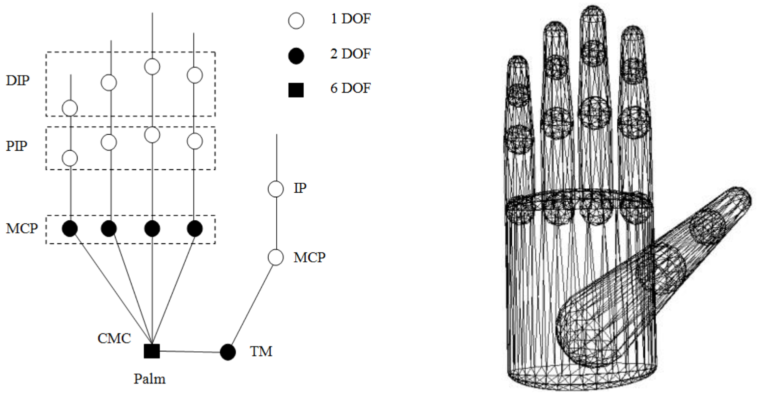


Figure 1. Block Diagram of the Model-based Tracking System

2. Hand Model

The hand model is built with basic geometric primitives in the 3D parametric modeling software Pro/Engineer. Though the Wavefront OBJ file format, it is then imported into Multigen Creator, a 3D modeling software for visualization. In Creator, we organize the meshes of the hand model into a hierarchical structure and add DOF nodes into the structure. The final resulting model is saved as an OpenFlight file and shown in Figure 2(b).

AS shown in Figure 2(a), the hand kinematics is modeled with 26 DOF, including 6 DOF for the global motion of the palm and 20 DOF for the local motion of the fingers. Assuming the CMC joints fixed, the palm is modeled as a rigid body with 6 DOF. All fingers are connected to the palm by five 2-DOF revolute joints (TM for the thumb and MCP for the other fingers), with 1 DOF for flexion-extension motion and 1 DOF for abduction-adduction motion. Each finger consists of three parts connected by two 1-DOF joints which are only capable of flexion-extension motion.



(a) Skeleton model (b) Mesh model

Figure 2. Hand Skeleton Model and Mesh Model

3. Observation Model

To measure the compatibility of a hand pose hypothesis $\boldsymbol{s}$ to the observation $\boldsymbol{o}$, we establish the observation model by comparing the feature maps extracted from the observation and the ones rendered from the hypothesis $\boldsymbol{s}$. Specifically, for a given input frame, the hand area is extracted from the depth observation by using simple depth segmentation, resulting a depth map o_d. And, for a given hand pose hypothesis $\boldsymbol{s}$, a depth map r_d is rendered by using the configurable 3D hand model. Then, two binary silhouette maps o_b and r_b are derived from o_d and r_d, respectively, by labeling foreground pixels to be 1. To measure the matching error between the observed features $\{o_d, o_b\}$ and the rendered features $\{r_d, r_b\}$, a function is defined as follows:

$$f(\boldsymbol{o},\boldsymbol{s}) = \lambda_d f_d(\boldsymbol{o},\boldsymbol{s}) + \lambda_b f_b(\boldsymbol{o},\boldsymbol{s}) + \lambda_m f_m(\boldsymbol{s}) \tag{1}$$

which consists of three terms: the depth term f_d, the silhouette term f_b and the smoothness term f_m. By experimental research, the weights λ_d, λ_b and λ_m are set to 1.0, 2.22 and 0.005, respectively, in this paper.

The depth term f_d measures the depth difference between the observed depth map o_d and the rendered depth map r_d

$$f_d(\boldsymbol{o},\boldsymbol{s}) = \frac{\sum \min\left(\left|o_d - r_d\right|, T_d\right)}{\sum \left(o_b \vee r_b\right)} \tag{2}$$

The calculation of the depth differences (in millimeters) is operated in a pixel-wise manner. To avoid some large differences impacting the performance of the search method, the absolute depth differences are clamped in the range $[0, T_d]$. In this paper, T_d is set to 40mm.

The silhouette term f_b measures the area of non-overlapping hand regions between the rendered silhouette map r_b and the observed silhouette map o_b

$$f_b(\boldsymbol{o},\boldsymbol{s}) = \frac{\sum o_b(1 - r_b)}{\sum o_b} + \frac{\sum r_b(1 - o_b)}{\sum r_b} \tag{3}$$

whose first term describes the non-overlapping hand regions in o_b and second term describes the non-overlapping hand regions in r_b. The introduction of the silhouette term f_b makes the objective function smoother and helps the search process converge to the true optimum.

To penalize the sudden changes of hand poses between two consecutive frames, a smoothness term is introduced

$$f_m(\boldsymbol{s}_t) = \| \boldsymbol{s}_t - \hat{\boldsymbol{s}}_{t-1} \| \tag{4}$$

where $\boldsymbol{s}_t$ is a hypothetical pose for the current frame and $\hat{\boldsymbol{s}}_{t-1}$ is the reconstructed pose for the previous frame.

Then, for a given pose $\boldsymbol{s}$, the likelihood of the observation $\boldsymbol{o}$ is defined as

$$p(\boldsymbol{o} \mid \boldsymbol{s}) \propto \exp\left(-\lambda_f \cdot f(\boldsymbol{o}, \boldsymbol{s})\right) \tag{5}$$

where λ_f is a normalization factor that is set according to the noise of the observation.

4. The Tracking Algorithm

4.1. Particle Filtering

In this paper, we address the problem of articulated hand tracking in a particle filter. Based on a stochastic sampling method, particle filtering implements a recursive Bayesian filter, which approximates the posterior distribution by a set of weighted particles $\{(\boldsymbol{s}^i, \pi^i)\}_{i=1}^N$, where $\boldsymbol{s}^i$ is a sample state and π^i is its corresponding weight. After particles $\{(\boldsymbol{s}_{t-1}^i, \pi_{t-1}^i)\}_{i=1}^N$ are sampled from the posterior distribution of time *t*–1, they are propagated to new positions according to the transition model $p(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1})$, and then based on the observation likelihood $p(\boldsymbol{o}_t \mid \boldsymbol{s}_t)$ their weights are updated to obtain a new particle set $\{(\boldsymbol{s}_t^i, \pi_t^i)\}_{i=1}^N$, which represents the posterior distribution $p(\boldsymbol{s}_t \mid \boldsymbol{o}_{1:t})$ of the state $\boldsymbol{s}_t$ conditioned on the observations $\boldsymbol{o}_{1:t} = \{\boldsymbol{o}_1, \ldots, \boldsymbol{o}_t\}$ up to time *t*.

Particle filtering provides a robust framework for motion tracking in cases of clutter and occlusion. By modeling uncertainty and keeping multiple hypotheses along time, it has the power to represent multimodal distributions. However, for easy implementation, the transition prior $p(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1})$, instead of the optimal distribution $p(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1}, \boldsymbol{o}_t)$ [13], is usually taken as the proposal distribution for importance sampling. This is inefficient when $p(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1})$ lies in the tail of the observation likelihood $p(\boldsymbol{o}_t \mid \boldsymbol{s}_t)$, which is a case that happens quite often. As articulated hand tracking is a high-dimensional problem, a very large number of samples are needed to maintain an effective representation of the true posterior, making the algorithm too slow. If the number of samples is not enough, the samples will be too diffused and the tracking could be lost. To provide solutions that work well with fewer samples, some kind of optimization method is often introduced into the framework of particle filtering.

4.2. Differential Evolution

In this paper, we use differential evolution for the minimization of the matching error function (see Equation (1)). Differential evolution (DE) [14] is a simple and efficient swarm-based optimization method for minimizing non-linear and non-differentiable objective functions. After initialized, DE evolves a set of *N* *D*-dimensional vectors $\{\boldsymbol{x}_g^i\}_{i=1}^N$ with the proceeding of generation *g* to search the global optimum over continuous space. The evolution is performed via three operations: mutation, crossover, and selection. Mutation and crossover are used to produce the trial vectors and selection is used to determine whether the new trial vector should survive into the next generation.

During mutation, for each vector index *i* in the population, DE randomly chooses three different vectors from the previous generation and combines them to create a mutant vector

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

where vector indexes r_1 , r_2 , r_3 are randomly chosen in the range [1,2,…,*N*], different from each other and different from the index *i*. The scale factor *F* for the difference vector $(\boldsymbol{x}_{g}^{r_2} - \boldsymbol{x}_{g}^{r_3})$ controls the convergence rate of the search. Originally, *F* is a constant. In this paper, to improve convergence, we use a jitter [15] factor $\sigma = 1.0$ to modulate *F* for each parameter. Then, we have $F = F_C \cdot N(0,1)$, where F_C is a constant and $N(0,1)$ is a Gaussian random number with mean 0 and variance 1.0. For the 6 global dimensions, we set F_C =0.5, and for the 20 finger joint dimensions, we set F_C =0.7.

The mutant vector $\boldsymbol{v}_{g+1}^{i}$ is then combined with the old vector $\boldsymbol{x}_{g}^{i}$ to form the trial vector $\boldsymbol{u}_{g+1}^{i} = \{u_{g+1}^{j,i}\}_{j=1}^{D}$ via 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 uniform random number in the range [0,1] for the *j*-th dimension. The crossover constant *CR* determines the probability for the trial parameter to be inherited from the mutant vector. In this paper, we set *CR*=0.9. A parameter index r_{g+1}^{i} is randomly chosen in the range [1,2,…,*D*] to make sure that the trial vector gets at least one parameter from the mutant vector.

After mutation and crossover, a greedy selection operation is performed

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

The trial vector $\boldsymbol{u}_{g+1}^{i}$ is compared with the old vector $\boldsymbol{x}_{g}^{i}$ to decide whether it should be passed to the next generation. If the trial vector gets an equal or better objective function value than the old vector, then it replaces the old vector in the next generation; otherwise, the old vector is retained for at least one more generation.

According to [14], DE has several variants and the one described above can be noted as DE/rand/1/bin where ‘rand’ means the base vector for mutation is a randomly chosen population vector, ‘1’ indicates only one difference vector is used, and ‘bin’ denotes binomial crossover. The original DE algorithm is a parallel search method which maintains two distinct vector sets for two consecutive generations respectively. However, to accelerate the convergence rate, we makes DE a serial one by mixing two generations into one vector set.

4.3. Combining Differential Evolution with Particle Filtering

In this paper, DE is integrated into a particle filter to create a new solution for 3D articulated hand tracking. After predicting new positions for the particles using the transition prior, we run DE to optimize the particles based on the newest observation $\boldsymbol{o}_t$. By this way, the particles are moved towards the promising areas in the state space where the observation likelihood has a larger value. The optimization of DE can be seen as a procedure of importance sampling, which results a new particle set to approximate the optimal proposal distribution $p(\boldsymbol{s}_t \mid \boldsymbol{s}_{t-1}, \boldsymbol{o}_t)$.

Algorithm 1: Combining differential evolution with particle filtering

```
For $t > 0$:
1) Resample: resample $\{(\boldsymbol{s}_{t-1}^{i}, \pi_{t-1}^{i})\}_{i=1}^{N}$ into $\{(\boldsymbol{s}_{t-1}^{i}, 1/N))\}_{i=1}^{N}$ based on weights
2) Predict: propagate the particles from $t$–1 to $t$ to give $\{(\boldsymbol{s}_{t}'^{i}, 1/N))\}_{i=1}^{N}$ using Equation (9)
3) Optimize: optimize $\{(\boldsymbol{s}_{t}'^{i}, 1/N))\}_{i=1}^{N}$ via DE
   ● initialize the population of DE
   for $i$=1 to $N$
      $\boldsymbol{x}_{0}^{i} \leftarrow \boldsymbol{s}_{t}'^{i}$
   ● iterate
   for $g$=1 to $G$
      for $i$=1 to $N$
         do the mutation operation to obtain $\boldsymbol{v}_{g}^{i}$ using Equation (6)
         do the crossover operation to obtain $\boldsymbol{u}_{g}^{i}$ using Equation (7)
         do the selection operation to obtain $\boldsymbol{x}_{g}^{i}$ using Equation (8)
   ● after optimization
   for $i$=1 to $N$
      $\boldsymbol{s}_{t}^{i} \leftarrow \boldsymbol{x}_{G}^{i}$
4) Weight: weight the particles as $\pi_{t}^{i} \propto p(\boldsymbol{o}_{t} \mid \boldsymbol{s}_{t}^{i})$ to give $\{(\boldsymbol{s}_{t}^{i}, \pi_{t}^{i})\}_{i=1}^{N}$, and normalize $\{\pi_{t}^{i}\}_{i=1}^{N}$ so that $\sum_{i=1}^{N} \pi_{t}^{i} = 1$
5) Estimate: use the particle with the biggest weight as the output
```

To propagate the particles along the sequence, a first-order motion model is defined for the transition prior

$$\boldsymbol{s}_{t}'^{i} = \boldsymbol{s}_{t-1}^{i} + \boldsymbol{w}_{t-1}^{i} \tag{9}$$

where $\{\boldsymbol{s}_{t-1}^{i}\}_{i=1}^{N}$ are the final positions of the particles converged after the optimization procedure of DE at time t–1. $\boldsymbol{w}_{t-1}^{i} \sim N(0, \boldsymbol{\Sigma})$ is a multivariate Gaussian noise with mean 0 and a diagonal covariance $\boldsymbol{\Sigma}$ whose diagonal elements are determined according to the maximum inter-frame angular or translational differences. The obtained new particle set $\{\boldsymbol{s}_{t}'^{i}\}_{i=1}^{N}$ is then used to initialize the population of DE for time t. Finally, the presented tracking algorithm is summed up in Algorithm 1.

5. Experiments

The tracking system is developed with OpenSceneGraph (OSG), an open source 3D graphics toolkit. In OSG, we use a framebuffer object to render each pose hypothesis into a depth image for the calculation of its observation likelihood. The system runs on a computer with a 2.0 GHz Intel dual-core CPU, 2 GB RAM and a GeForce 8400M GS GPU. Using 40 particles and 45 generations for the optimization process of DE, the proposed method takes about 5 seconds to track one frame.

To evaluate the proposed method, experiments are performed based on both synthetic data and real image sequences. We compare the proposed method with the standard particle filtering (PF) and the standard PSO [16] based single hypothesis tracking method. To make fair comparisons, standard PF uses 1800 particles, while standard PSO is run with 60 particles and 30 generations. Hence, for the three methods, the numbers of likelihood evaluations to track one frame are all 1800.

5.1. Experiments on Synthetic Sequences

It is difficult to obtain the ground truth of hand motions from a real image sequence. Therefore, we produce a synthetic sequence of 150 hand poses by linear interpolation among four predefined key poses shown in Figure 3. Rendering is used to synthesize the required depth image for each hand pose. By using the synthetic image sequence as the input of the system, the tracking results can be compared against the ground truth.

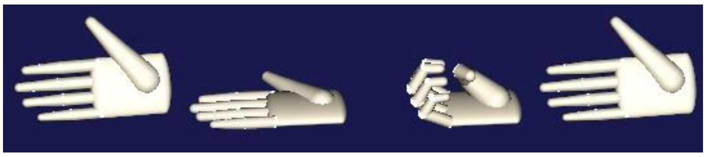
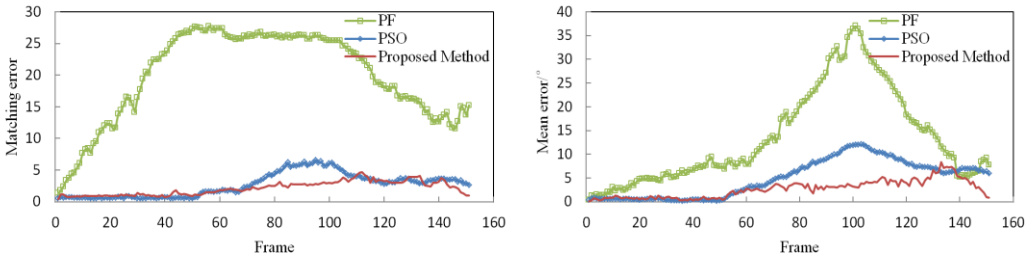


Figure 3. The Key Poses of the Synthetic Sequence

Figure 4(a) plots the matching errors (see Equation (1)) of the three methods on the synthetic sequence, while Figure 4(b) plots the mean errors of the pose angles (including the 20 finger joint angles and the 3 global angles which represent the orientation of the palm). The statistics of the mean angle errors along the sequence is shown in Table 1. The results show that the standard PF method meets the problem of serious error accumulation and thus can’t track articulated hand motion well. In Figure 4(b), it can be seen that the errors of standard PF even accumulate to a peak value larger than 35 degrees. The proposed method and the standard PSO method both clearly perform better than standard PF. However, the proposed method attains higher accuracy than the standard PSO method.



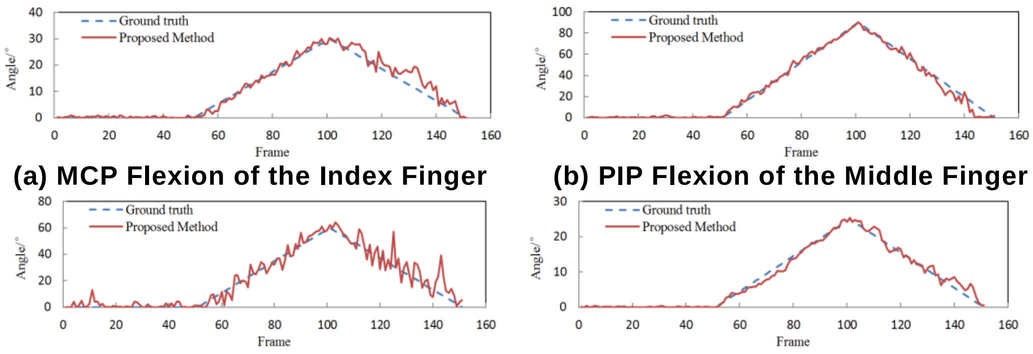
(a) Matching Errors **(b) Mean Angle Errors**

Figure 4. Tracking Errors on the Synthetic Sequence

Table 1. Statistics of Mean Angle Errors

Method	Total Average(°)	Standard Deviation(°)
PF	13.5243	9.8956
PSO	4.8908	3.9283
Proposed method	2.6748	1.8660

Some of the state parameters attained by the proposed method are shown in Figure 5. The results are plotted in solid curves while the ground truth data is plotted in dash curves. It can be seen that our results are in good agreement with the ground truth.



(a) MCP Flexion of the Index Finger

(b) PIP Flexion of the Middle Finger

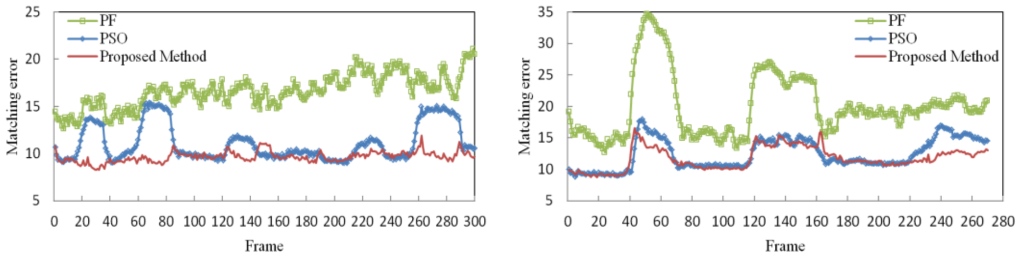
(c) DIP Flexion of the Middle Finger

(d) TM Flexion of the Thumb Finger

Figure 5. Comparison of our Estimates and the Ground Truth

5.2. Experiments on Real Sequences

A Kinect sensor is used to capture two real image sequences, the first one consisting of 300 frames and the second 270. By using the Microsoft Kinect 1.0 Beta2 SDK, the images are captured at a resolution of 640×480 pixels and a rate of 30 *fps*. As the ground truth can’t be obtained, we only compare the matching errors on the real sequences (see Figure 6). It can be seen that the proposed method and the standard PSO method still clearly perform better than standard PF. However, for some frames when the articulated motion of the hand becomes a little more complicated, such as frame 20~40, 60~80, 260~290 in the first sequence and frame 220~270 in the second sequence, the proposed method outperforms the standard PSO method in accuracy and robustness.



(a) The First Real Sequence

(b) The Second Real Sequence

Figure 6. Matching Errors on Real Sequences

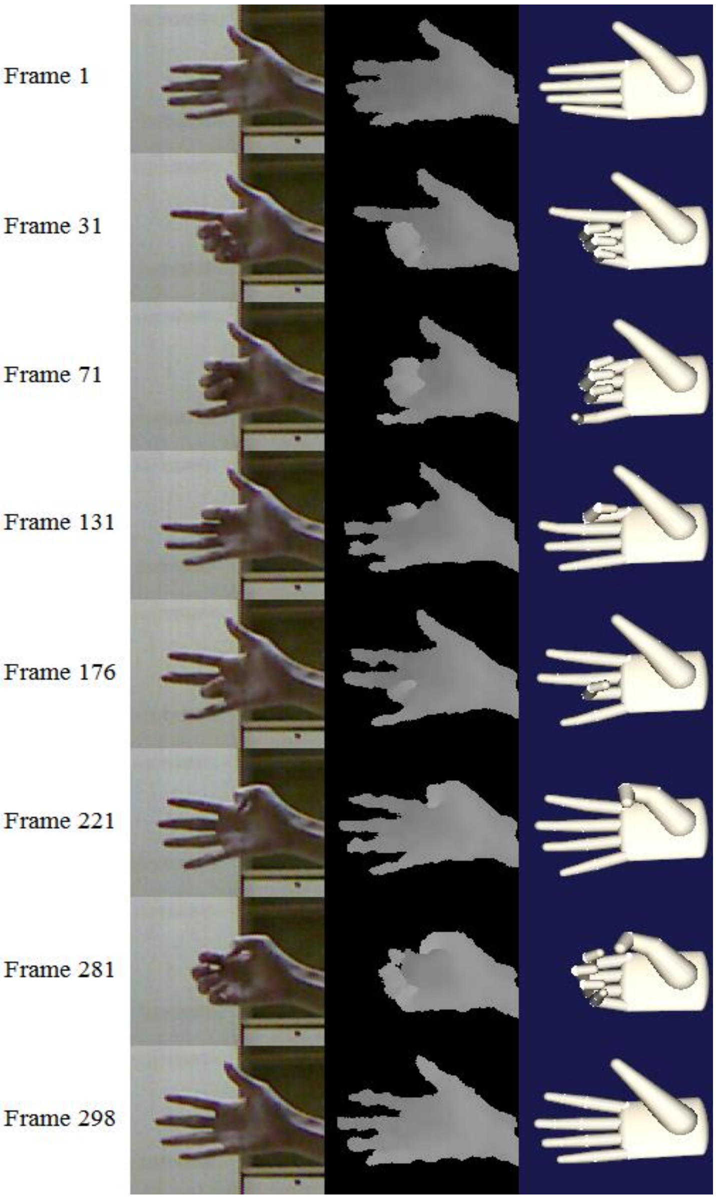


Figure 7. Some of Our Results on the First Real Sequence

Figure 7 and Figure 8 show some visual results of the proposed method on the two real sequences, respectively. The left column of each figure shows the corresponding synchronized RGB color images captured by the Kinect RGB camera. The middle column of each figure shows the depth images captured by the Kinect depth camera, which are used as the input of the tracking system. The results of the proposed method are presented in the right column of each Figure. It can be seen that the proposed method can track articulated hand motion accurately and robustly.

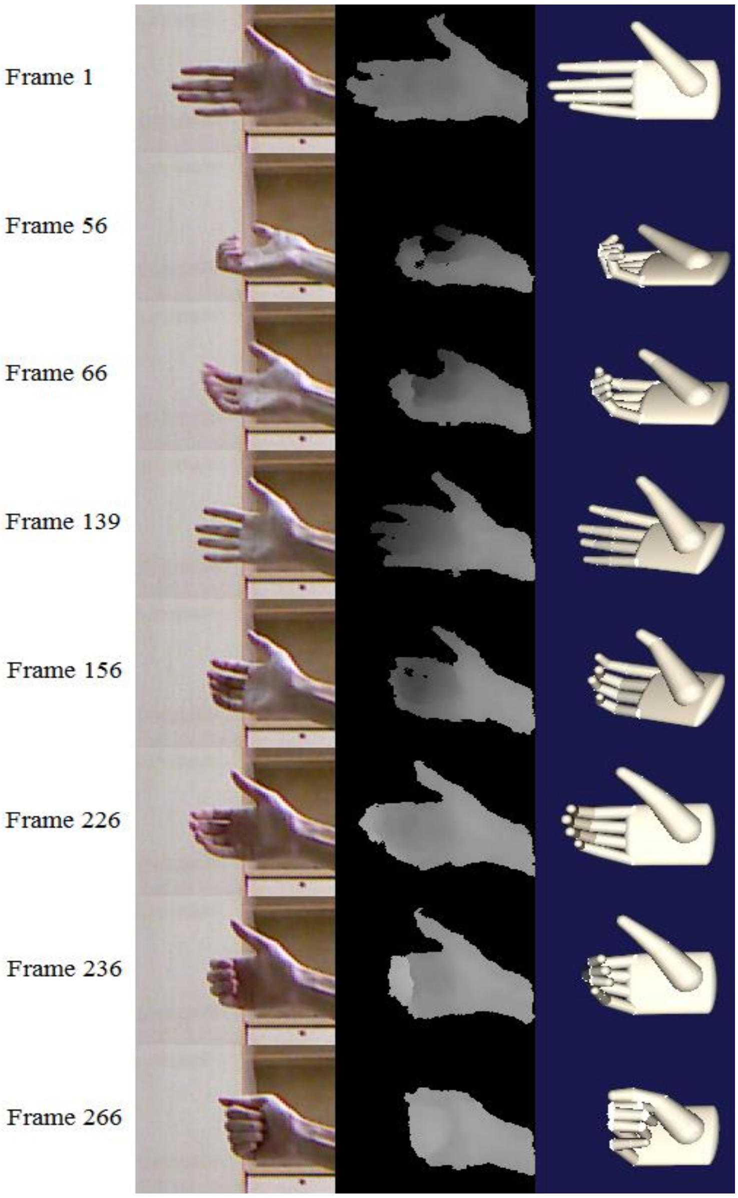


Figure 8. Some of Our Results on the Second Real Sequence

6. Conclusion

In this paper, we have presented a new algorithm by combing differential evolution and a particle filter, to track articulated motion in a high-dimensional state space. Through the optimization procedure of differential evolution, the particles are moved to the regions with a high likelihood. By using single depth images as the only input, our system is immune to illumination and background changes. Experiments based on both synthetic data and real image sequences have demonstrated that the proposed method is accurate and robust for articulated hand motion tracking.

The depth observation obtained from the Kinect sensor is rough and noisy, where the depth “holes”, which result from the missing depth information, happen all the time. That causes a big impact on the accuracy of our system. In the future, to make the system more robust to sensor noise, we will apply multiple depth cameras for tracking. The most time-consuming step of the tracking system is the calculation of the matching error function, which is easy to be parallelized and implemented on a GPU. For future work, we will accelerate the tracking system by using GPU implementations.

Acknowledgements

This work is supported by the Key Science and Technology Program of Shandong Province (Grant No. 2010G0020233), and the Natural Science Foundation of China (Grant No. 61403232). In addition, we are very grateful to the editor and the anonymous reviewers for their valuable comments.

References

[1] J. Romero, H. Kjellström and D. Kragic, “Monocular real-time 3D articulated hand pose estimation”, Proceedings of the 9th IEEE-RAS International Conference on Humanoid Robots, Paris, France, **(2009)** December 7-10.

[2] C. Keskin, F. kiraç, Y. E. Kara and L. Akarun, “Hand pose estimation and hand shape classification using multi-layered randomized decision forests”, Proceedings of the 12th European Conference on Computer Vision, Florence, Italy, **(2012)** October 7-13.

[3] V. A. Prisacariu and I. Reid, “3D hand tracking for human computer interaction”, Image and Vision Computing, vol. 30, no. 3, **(2012)**, pp. 236-250.

[4] H. Ali, J. Dargham, C. Ali and E. G. Moung, “Gait recognition using gait energy image”, International Journal of Signal Processing, Image Processing and Pattern Recognition, vol.4, no. 3, **(2011)**, pp. 141-152.

[5] Y. Wu, J. Y. Lin and T. S. Huang, “Capturing natural hand articulation”, Proceedings of the Eighth IEEE International Conference on Computer Vision, Vancouver, Canada, **(2001)** July 7-14.

[6] H. Sidenbladh, M. J. Black and D. J. Fleet, “Stochastic tracking of 3D human figures using 2D image motion”, Proceedings of the 6th European Conference on Computer Vision, Dublin, Ireland, **(2000)** June 26-July 1.

[7] C. Sminchisescu and B. Triggs, “Estimating articulated human motion with covariance scaled sampling”, International Journal of Robotics Research, vol. 22, no. 6, **(2003)**, pp. 371-393.

[8] M. Bray, E. Koller-Meier and L. Van Gool, “Smart particle filtering for high-dimensional tracking”, Computer Vision and Image Understanding, vol. 106, no. 1, **(2007)**, pp. 116-129.

[9] J. Deutscher and I. Reid, “Articulated body motion capture by stochastic search”, International Journal of Computer Vision, vol. 61, no. 2, **(2005)**, pp. 185–205.

[10] J. Cui and Z. Sun, “Visual hand motion capture for guiding a dexterous hand”, Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition, Seoul, South Korea, **(2004)** May 17-19.

[11] 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 Transactions on Multimedia, vol. 15, no. 1, **(2013)**, pp. 106-119.

[12] I. Oikonomidis, N. Kyriazis and A. A. Argyros, “Efficient model-based 3D tracking of hand articulations using Kinect”, Proceedings of the 22nd British Machine Vision Conference, Dundee, UK, **(2011)** August 29-September 2.

[13] A. Doucet, S. Godsill and C. Andrieu, “On sequential Monte Carlo sampling methods for Bayesian filtering”, Statistics and Computing, vol. 10, no. 3, **(2000)**, pp. 197-208.

[14] R. Storn and K. Price, “Differential evolution – A simple and efficient heuristic for global optimization over continuous spaces”, Journal of Global Optimization, vol. 11, no. 4, **(1997)**, pp. 341-359.

[15] D. Zaharie, “Critical values for the control parameters of differential evolution algorithms”, Proceedings of the 8th international conference on soft computing, Brno, Czech Republic, **(2002)** June 5-7.

[16] J. Kennedy and R. Eberhart, “Particle swarm optimization”, Proceedings of the IEEE International Conference on Neural Networks, Perth, Australia, **(1995)** November 27-December 1.

Authors

Dongnian Li is a Ph.D. candidate of School of Mechanical Engineering, Shandong University, China. He received his B.S. degree from Shandong University, in 2009. His major research interests include computer vision, graphics and virtual reality.

E-mail: dongnianli87@163.com.

Yiqi Zhou is a professor of School of Mechanical Engineering, Shandong University, China. He received his Ph.D. degree from Shandong University, in 2002. His major research interests include automatic control, virtual reality and virtual engineering.

E-mail: yqzhou@sdu.edu.cn.