Efficient 3D Hand Tracking in Articulation Subspaces for the Manipulation of Virtual Objects

Gabyong Park
KAIST GSCT UVR Lab.
Daejeon, Republic of Korea
gypark@kaist.ac.kr

Antonis Argyros
Computational Vision and Robotics Lab., Institute of Computer Science, FORTH
University of Crete, Greece
argyros@ics.forth.gr

Woontack Woo
KAIST GSCT UVR Lab.
Daejeon, Republic of Korea
wwoo@kaist.ac.kr

ABSTRACT

We propose an efficient method for model-based 3D tracking of hand articulations observed from an egocentric viewpoint that aims at supporting the manipulation of virtual objects. Previous model-based approaches optimize non-convex objective functions defined in the 26 Degrees of Freedom (DoFs) space of possible hand articulations. In our work, we decompose this space into six articulation subspaces (6 DoFs for the palm and 4 DoFs for each finger). We also label each finger with a Gaussian model that is propagated between successive image frames. As confirmed by a number of experiments, this divide-and-conquer approach tracks hand articulations more accurately than existing model-based approaches. At the same time, real time performance is achieved without the need of GPGPU processing. Additional experiments show that the proposed approach is preferable for supporting the accurate manipulation of virtual objects in VR/AR scenarios.

CCS Concepts

•**Computer systems organization** → *Computer vision problems;* Tracking; •**Human-centered computing** → Human computer interaction (HCI);

Keywords

3D tracking; Hand articulations; Manipulation of virtual objects

1. INTRODUCTION

Hand tracking, an actively studied problem in the field of computer vision nowadays, is employed to manipulate virtual objects in Virtual Reality (VR) and Augmented Reality (AR) appications such as those presented in [1, 2]. Among many available means of interaction, the human hand is distinctive in that it functions as a natural and intuitive 3D interface through which various activities are performed. Therefore, tracking the articulations of the hand in 3D is crucial for the success of scenarios involving the manipulation of virtual objects.

CGI '16, June 28-July 01, 2016, Heraklion, Greece

DOI: http://dx.doi.org/10.1145/2949035.2949044

3D tracking of hand articulations is faced with challenging problems. Some computational difficulties are due to the high DoF of the human hand and its uniform color appearance. This problem becomes even more challenging in an egocentric camera placement (i.e., the hand is observed by a camera placed close to the eyes of the human whose hand is tracked), as self-occlusion of the human hand occurs more frequently and to a stronger degree.

There are largely two main classes of approaches to 3D hand tracking, the discriminative and the generative ones. Discriminative methods require training of a classifier, which is used mainly to estimate the hand pose in a single frame. Generative methods define an objective function that quantifies the discrepancy between visual observations from 3D image sensor and rendered 3D hand model hypotheses. Typically, the defined objective function is multimodal, non-differentiatable and with several local minima. Although GPGPU processing has resulted in real-time performance [6, 7], model-based methods are computationally intensive. This is because their objective function needs to be evaluated repeatedly at several points in the high (26D) dimensional space of hand configurations. Additionally, due to the uniform appearance of hand parts, there may be tracking failures because of mismatches between real fingers and their model counterparts.

In order to cope with these problems, in our work, we perform model-based optimization, which has been actively studied to alleviate the problems. More specifically, after the 6 DoFs for the position and orientation of the palm are estimated, the remaining 20 DoFs are processed in a row in 5 threads, on for the 4 DoFs of each finger. This divide-and-conquer approach leads to a set of much simpler optimization problems. A similar decomposition is proposed in [4]. However, also in that work, real time performance is achieved based on GPGPU processing. And, labeling each finger is also demonstrated in [9]. However, [9] depends on a color glove that the user wears, while in our work, we perform automatic labeling of the fingers of a bare hand.

2. RELATED WORK

In this section, we review briefly existing work on 3D hand pose estimation and tracking. There are three broad classes of solutions to these problems, the generative, the discriminative and the hybrid approaches.

Generative approaches first define a 3D hand model whose articulation is controled with a set of parameters. Then, they solve an optimization problem whose goal is to estimate the 26 parameters of the hand model that make it fit to the set of available observations. To do so, they define an objective function that quantifies the discrepancy between a hand model and the set of available observations [6, 7]. The optimization of this objective function involves

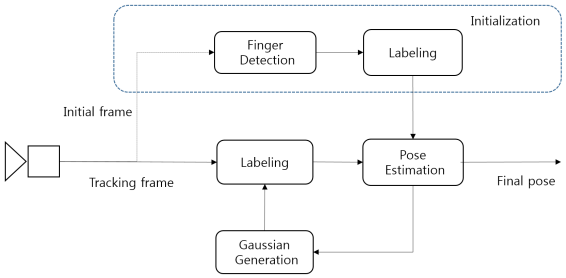


Figure 1. **The flowchart of the proposed hand articulations tracking method.**

(a) (b)

Figure 2. **(a) The result of labeling each finger (b) The result of model fitting.**

its evaluation at several points in the 26D hand configuration space. This raises considerably the computational requirements of generative methods.

The discriminative approaches estimate the articulation parameters through the use of pre-trained classifiers that learn the mapping between 3D hand configurations and actual observations. A benefit of all these discriminative methods is that they perform hand pose estimation in a single frame, i.e., they do not capitalize on tracking [11, 12]. However, a disadvantage is that some of these methods only recognize (rather than estimate) hand poses that are present in the dataset on which they were trained.

Hybrid approaches [8, 10] have recently emerged and aim at coupling the benefits of both the generative and the discriminant methods. The general idea is to use a discriminative method that provides a solution close to the actual one, combined with a generative component that fine tunes this solution by performing search in the continuous space of solutions around the solution of the discriminative method.

3. METHODOLOGY

3.1 Initialization

The primary goal of the initialization step is to estimate the initial pose of the hand before running the tracker. We assume that hand object is segmented well with simple threshold method in depth and HSV space, and wrist position X_w is detected initially from wirstband before running our algorithm. The hand contours of segmented binary image are extracted through edge detection from sobel operation. It is assumed that hand has a pose so that the boundaries of the fingers are close to vertical. Parts of the contour that correspond to individual fingers are identified and labeled by a clockwise scanning of the extracted contours and by grouping the edges with positive and negative responses to the sobel filter. Then, the corresponding 3D points (from the depth image) are associated to each finger. Finally, the hand model is fitted into these observations.

3.2 Hand model and hand 3D points from kinematics

We now present the employed 3D hand model and the method with which 3D points from kinematics are retrieved. Such points are used in generation of initial Gaussian models for labeling finger observations and in the formulation of the objective function to compare a given hand hypothesis against the available observations. Unlike [6, 7] using full rendering of a hand model to get 3D points in the model, we sample a number of points between hand joints based on forward kinematics.

More specifically, the rotation matrix and the translation vector of the i-th joint can be embedded into the 4×4 displacement matrix T_i with forward kinematic transformation Q. Thus, T_i is

$$T_i = Q(\Theta_1)Q(\Theta_2)...Q(\Theta_i), \tag{1}$$

where the motion of the hand is constrained by the kinematic parameters $\Theta_1,...,\Theta_i$. Therefore, the i-th joint position is obtained as the translation vector of T_i. With this formulation, we cut a finger object into 9 slices, and one slice evenly has 4 points. Finally, we get 36 points in the finger. In the palm model, 10 points are sampled between a proximal phalanx and wrist point. Finally, we get 50 points in the palm model.

3.3 Tracking

Labeling finger observations.

To label each finger, we use Gaussian distributions that model the 3D points of the fingers. A Gaussian model is useful to label corresponding finger in observation because it can be used to calculate Mahalanobis distance and generated easily from 3D points of the hand model. We assume that a Gaussian distribution $G_i(t)$=$((\mu_i(t),\Sigma_i(t))$, $1\leq i\leq 5$, models the 3D points of the i-th finger at frame t. Thus, a certain 3D point X_o in observation is assigned to the label c as defined in the following equation:

$$c = \arg\min_i \sqrt{(X_o-\mu_i(t))^T\Sigma_i(t)(X_o-\mu_i(t))}. \tag{2}$$

The assignment of 3D points to fingers at frame t (based on Eq. 2) is based on the Gaussian distributions of fingers as those were defined from the solution at the previous time step. However, the iterative update is performed for a Gaussian i at frame t if the displacement of a finger between $t-2$ and $t-1$ (that is, $|\mu_i(t-2)-\mu_i(t-1)|_2$) is larger than an empirically decided threshold value. To do so, we adopt an iterative Expectation-Maximization (EM) like update of the parameters of the Gaussians. The iterative adaptation of the Gaussian distributions is performed until either a maximum number of iterations (10) is reached or the mean vectors of all distributions have converged (i.e., they all remain unchanged between successive iterations). The result of labeling each finger is shown in Fig. 2 (a).

Objective functions for pose estimation.

The objective function E_{6D} for the optimization of the palm pose is defined as follows:

$$E_{6D} = \sum_{i=1}^{N_o}\|X_{bi}^o - X_{bi}^m\|/N_o + \sum_{j=1}^{N_m} K_p(X_{pj}^m)/N_m, \tag{3}$$

where N_o is the number of detected labels so that the maximum value is five and N_m is the number of points in the palm model. The first term of Eq.3 represents the discrepancy between the proximal phalanx points X^o_{bi} in the observation and the corresponding proximal phalanx points X^m_{bi} in the model. X^o_{bi} is decided as the center of labeled region within the predefined distance from wrist position which is initially detected by the wrist band. Furthermore, we add the additional term $K_p(X^m_{pj})$ to fit the palm model to the observation. The j-th 3D point X^m_{pj} sampled from the palm model are projected to the image plane. If these pixels project outside the hand image, the corresponding solution is penalized.

The objective function E_{4D} for the optimization of the each finger pose is defined as follows:

$$E_{4D} = \sum_{j=1}^{N_m} (min\|X^o_{Li} - X^m_{i,j}\| + K_p(X^m_{i,j}))/N_m, \quad (4)$$

where N_m is the number of points in finger model and $X^m_{i,j}$ is the j-th point in the i-th finger model. The first term in Eq. 4 is to fit the points in the finger model to the labeled 3D point clouds in the observation. The 48 points are sampled randomly among labeled 3D point clouds in the observation. The term $K_p(X^m_{i,j})$ operates similarly as in Eq. 3. However, in this case, it is calculated from the 3D points projected to the labeled finger image.

Optimization.

The objective functions defined in Eqs. 3 and 4 are optimized based on Particle Swarm Optimization [3]. It operates by defining particles (candidate problem solutions) that move in the solution space based on simple rules that emulate social interaction. Particles move for a number of generations. To exploit temporal continuity, the solution in the present frame is used to generate the initial particles in the next frame.

The updating of the velocity v of a particle in each generation is performed based on:

$$v_{k+1} = w(v_k + c_1 r_1 (p_k - x_k) + c_2 r_2 (g_k - x_k)). \quad (5)$$

In the above equations p_k is the local best position of the particle and g_k is the global best position of all the particles of the swarm. c_1 is the cognitive component, c_2 is the social component, and r_1, r_2 are random numbers in the range from 0 to 1. The values c_1=2.8, c_2=1.3, and $w = 2/|2 - \Psi - \sqrt{\Psi^2 - 4\Psi}|$ with $\Psi = c_1 + c_2$ are used as in [6].

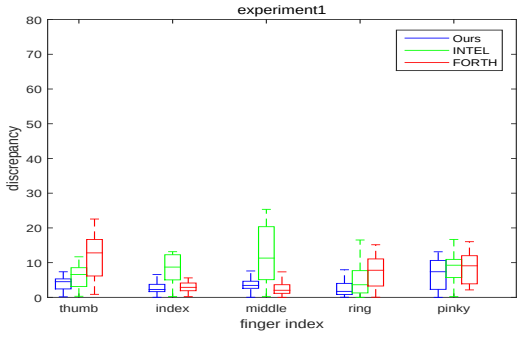
Given the above, the updating of the position x of a particle in each generation is performed based on:

$$x_{k+1} = x_k + v_{k+1}. \quad (6)$$

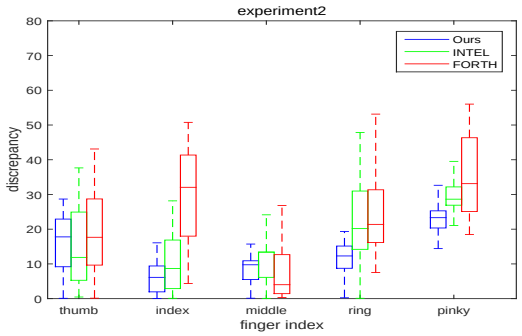
4. EXPERIMENTS

4.1 Quantitative results

We synthesized a touch-based interaction scenario in which a human hand has to interact with virtual objects. To achieve this, 25 particles in a 5×5 grid were arranged. The distance between the position of each fingertip and the corresponding particle was measured. We repeated the experiment 10 times and calculated the average achieved distance. We evaluated comparatively three methods: ours, the hand tracking method of [6] (called FORTH) and the method at [5] (called INTEL). The videos acquired from all the performed experiments are shown in the supplementary material accompanying this paper.



(a)



(b)

Figure 3: (a) Experimental result about (a) clearly visible palm and (b) not clearly visible palm.

4.1.1 *Interaction Experiment 1: clearly visible palm*

In the first experiment, all evaluated method showed overall good performance during the 10 trials (see Fig. 3(a)). The palm region of the hand was clearly visible and, thus, the 6 DoFs of the palm can be estimated easily. With respect to the INTEL and FORTH, the discrepancy between real fingers and virtual ones was relatively higher than ours for some trials, but they mostly succeeded to touch the virtual objects.

4.1.2 *Interaction Experiment 2: not clearly visible palm*

Unlike the first experiment, the palm was not so clearly visible in the second one. Only some fingers and partial region of the palm were visible. Furthermore, the some fingers were hidden by other fingers. When a large-scale occlusion occurred owing to wrist rotation, the estimation of the hand articulation was not accurate because of the mismatch between the real fingers and corresponding finger models.

In this case, 3D hand tracking in articulation subspaces was beneficial. Even though palm region is not clearly visible, some visible fingers are labeled so that correspondence between observation and model is more clear. Furthermore, optimization problem is more simple due to divided dimensions. Finally, compared to other methods, ours showed a lower level of discrepancy between result and the ground truth, and succeeded to touch virtual objects.

4.2 Qualitative results: virtual object manipulation

We built an application for virtual object manipulation that compares our method with other approaches such as INTEL [5] and FORTH [6] based on real (i.e., not synthetic) data. The task was

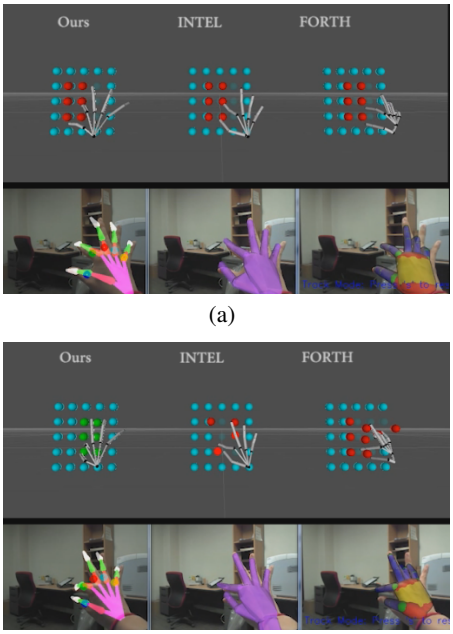


Figure 4. (a) Initial configuration for virtual object manipulation (relocation) and (b) manipulation result, compared with other algorithms. See text for more details.

to interact with a set of virtual (red) spheres and to relocate them in some target positions. The system for interaction was implemented in Unity3D. When the particle lies in the wrong position, the color remains red. If the particle is positioned correctly, the color changes to green, giving a visual feedback. Sample frames from relevant experiments are shown in Fig. 4.

A video acquired from this experiment are shown in the supplementary material accompanying this paper. As it can be verified, compared to the rest of the evaluated algorithms, the algorithm proposed in this paper achieves more accurate hand tracking and is shown to be able to better support hand object interaction in VR/AR scenarios.

5. CONCLUSION

The model-based approaches to 3D hand tracking provide fairly accurate solutions to the challenging problem. However, they may result in tracking failures as a result of high dimensional space and the inherent ambiguities resulting from the uniform color appearance of the fingers. In this work, we provide solutions to these problems (a) by decomposing the optimization problem into divided dimensional ones and (b) by propagating finger labels between successive frames. The proposed algorithm achieves a more accurate tracking result compared with other model-based approaches. This conclusion is supported by a series of quantitative and qualitative experiments in both synthetic and real datasets.

We focused on the tracking approach itself, rather than issues concerning single frame hand pose estimation. We believe that model based tracking approaches and single frame hand pose estimation methods can be combined to safeguard from tracking errors. Thus, further research on detection-guided tracking (i.e. hybrid methods) to improve the performance of virtual object manipulation is planned for the immediate future.

6. ACKNOWLEDGEMENT

This work was supported by the National Research Foundation of Korea(NRF) grant funded by the Korea government(MSIP) (No. 2014R1A2A2A01003005) and by the Global Frontier R&D Program on <Human-centered Interaction for Coexistence> funded by the National Research Foundation of Korea grant funded by the Korean Government(MSIP) (NRF-2015M3A6A3073746)

7. REFERENCES

[1] T. Ha, S. Feiner, and W. Woo. Wearhand: Head-worn, rgb-d camera-based, bare-hand user interface with visually enhanced depth perception. In *Mixed and Augmented Reality (ISMAR), 2014 IEEE International Symposium on*, pages 219–228, Sept 2014.

[2] Y. Jang, S.-T. Noh, H. J. Chang, T.-K. Kim, and W. Woo. 3D Finger CAPE: Clicking Action and Position Estimation under Self-Occlusions in Egocentric Viewpoint. *IEEE Transactions on Visualization and Computer Graphics (TVCG)*, 21(4), 2015.

[3] J. Kennedy and R. Eberhart. Particle Swarm Optimization. *International Conference on Neural Networks*, 4:1942–1948, 1995.

[4] A. Makris, N. Kyriazis, and A. Argyros. Hierarchical particle filtering for 3d hand tracking. In *IEEE Computer Vision and Pattern Recognition Workshops (HANDS 2015 - CVPRW 2015)*, pages 8–17, 2015.

[5] S. Melax, L. Keselman, and S. Orsten. Dynamics based 3d skeletal hand tracking. In *Proceedings of Graphics Interface 2013*, GI '13, pages 63–70, Toronto, Ont., Canada, Canada, 2013. Canadian Information Processing Society.

[6] I. Oikonomidis, N. Kyriazis, and A. Argyros. Efficient model-based 3D tracking of hand articulations using Kinect. *Procedings of the British Machine Vision Conference 2011*, pages 101.1–101.11, 2011.

[7] I. Oikonomidis, M. I. A. Lourakis, and A. Argyros. Evolutionary Quasi-random Search for Hand Articulations Tracking. *IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2014)*, pages 3422–3429, 2014.

[8] G. Poier, K. Roditakis, S. Schulter, D. Michel, H. Bischof, and A. Argyros. Hybrid one-shot 3d hand pose estimation by exploiting uncertainties. In *British Machine Vision Conference (BMVC 2015)*, pages 182–1, 2015.

[9] K. Roditakis and A. Argyros. Quantifying the effect of a colored glove in the 3d tracking of a human hand. In *Computer Vision Systems - 10th International Conference, ICVS 2015, Copenhagen, Denmark, July 6-9, 2015, Proceedings*, pages 404–414. Springer, 2015.

[10] S. Sridhar, F. Mueller, A. Oulasvirta, and C. Theobalt. Fast and robust hand tracking using detection-guided optimization. In *Proceedings of Computer Vision and Pattern Recognition (CVPR)*, 2015.

[11] D. Tang, H. J. Chang, A. Tejani, and T.-K. Kim. Latent regression forest: Structural estimation of 3d articulated hand posture. In *Computer Vision and Pattern Recognition*, CVPR '14, 2014.

[12] D. Tang, T.-H. Yu, and T.-K. Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *Proceedings of the 2013 IEEE International Conference on Computer Vision*, ICCV '13, pages 3224–3231, Washington, DC, USA, 2013. IEEE Computer Society.