Model-based 3D Hand Tracking with on-line Hand Shape Adaptation

Alexandros Makris
amakris@ics.forth.gr
Antonis Argyros
argyros@ics.forth.gr

Institute of Computer Science,
FORTH, Heraklion, Greece
Computer Science Department,
University of Crete,
Heraklion, Greece

Abstract

One of the shortcomings of the existing model-based 3D hand tracking methods is the fact that they consider a fixed hand model, i.e. one with fixed shape parameters. In this work we propose an online model-based method that tackles jointly the hand pose tracking and the hand shape estimation problems. The hand pose is estimated using a hierarchical particle filter. The hand shape is estimated by fitting the shape model parameters over the observations in a frame history. The candidate shapes required by the fitting framework are obtained by optimizing the shape parameters independently in each frame. Extensive experiments demonstrate that the proposed method tracks the pose of the hand and estimates its shape parameters accurately, even under heavy noise and inaccurate shape initialization.

1 Introduction

Tracking articulated objects in 3D is a challenging task with many applications in several fields such as health care, telepresence, surveillance, and entertainment. 3D hand tracking is an interesting sub-problem with high complexity due to the high dimensionality of the human hand and its frequent and often severe self occlusions. Lately, this problem has received a lot of attention and several works that took advantage of RGB-D sensors have advanced the state of the art in terms of tracking accuracy and processing time. However, most of the methods, track a human hand under the assumption that its parameters (e.g., finger lengths, palm dimensions, e.t.c.) are already known. This is a quite restrictive assumption that limits the applicability of tracking methods. Hand model inaccuracies will translate to hand tracking inaccuracies or even failure.

In this work we jointly solve the hand pose tracking and hand shape estimation problems with a model based approach. The hand shape parameters are unknown but constant throughout a tracking sequence while the hand pose and articulation parameters constantly change. The different nature of these two parameter sets call for distinct approaches for estimating them. The hand pose parameters are estimated using the Hierarchical Model Fusion framework (HMF) [10, 11]. At each frame a shape estimate is obtained by an evolutionary optimization algorithm (PSO) [7]. The best of these estimates is then selected by a robust fitting framework that evaluates them over a frame history. We found this optimization/fitting strategy for the shape estimation to be a good compromise between speed and accuracy. A joint optimization of the shape parameters over the whole history would probably lead to more accurate results but is much slower. On the other hand, performing only per-frame shape optimization is faster but leads to inaccurate and unstable hand shape estimates.

1.1 Related work

Bottom up vs top down approaches: There are mainly two families of methods that are used for hand tracking i.e. model based (top-down) approaches and appearance based (bottom-up) approaches. The appearance based approaches rely on image features and a classifier to find the corresponding pose within a set of predefined hand poses [3, 5, 8, 9, 12, 24, 25]. Model based approaches on the other hand [13, 15, 17, 18, 22] define a parametric model of the hand and search for the optimal solution in the model's continuous parameter space. The optimal solution comes either by local optimization around the estimate for the previous frame or by filtering. Bottom-up approaches require big training datasets to capture the large appearance variation that arises due to the highly articulated structure of the hand. To cope with this issue typically a coarse pose quantization is performed. The advantage of these data-driven approaches is that no hand initialization is required. In contrast, model based approaches usually perform local optimization and thus require initialization of the hand pose and shape in the first frame of the sequence. Finally, lately several hybrid approaches appeared that try to combine the advantages of both categories [16, 20, 21, 26].
Model Based Articulated Pose Tracking: Model based approaches can be categorized as holistic and part-based. Holistic approaches use a single hand model whereas part-based approaches decompose the model into parts and track them separately. Then constraints in the relative position of the parts are typically enforced. In [22], a part-based approach that uses non-parametric belief propagation to approximate the posterior is presented. In [6], the authors propose a part-based method for tracking a hand that interacts with an object. Each part has 6-DOF (position and orientation) for a total of 96-DOF for the whole hand. For each part, the objective function is defined locally and uses depth information after a skin color based hand segmentation. Several holistic model based approaches that perform local optimization around the previous estimate have been recently proposed. In [4] an approach that captures the motion of two hands interacting with an object is proposed. An iterative optimization scheme that uses the Levenberg-Marquard algorithm is employed. In [13, 15], Particle Swarm Optimization (PSO) is applied for single and two-hands tracking while a custom evolutionary optimization method is proposed in [14] for these two problems. The above methods assume a known hand shape. Recently, a hierarchical particle filter for hand tracking has been proposed [11]. In this work we employ this method to track the hand pose and we extend it to also estimate the user's hand shape which leads to better tracking accuracy.
Hand Shape Estimation: Recently, a few approaches that attempt to solve the hand shape estimation problem have been proposed.In [23], the hand shape is learned from a depth sequence. However, the problem is treated disjointly from tracking, and shape estimation requires a calibration sequence. Very recently, an approach that jointly estimates the hand pose and shape has been proposed [19]. The difference with our approach is that they don't split the state into shape and pose parameters but instead estimate, per-frame, the joint pose-shape state using a PSO optimizer. We claim that the inherent difference between the constant shape parameters and the varying pose parameters favors their splitting into two groups and the different treatment for each group. We experimentally validate this claim and show

that the latter strategy leads to superior performance.

1.2 Contribution

In this work we present an on-line method that solves simultaneously the hand tracking and hand shape estimation problems. The method performs in real time ($25-30$fps). For the pose estimation the Bayesian Hierarchical Model Framework (HMF) is employed [□, □]. The framework uses six *auxiliary models* that lie in lower dimensional spaces as proposals for the 26-DOF *main model* of the hand. A fast and robust metric measures the discrepancy between the rendered model and the RGB-D observations. The shape parameters are estimated using a novel approach. In a first step, shape parameters are optimized per frame using the PSO algorithm. A history of such per-frame estimates are then fed to a robust fitting framework that estimates the best hand shape parameters within this history of frames.

The method is tested qualitatively using real challenging sequences and quantitatively using simulated data. As a baseline for the comparison we use the HMF [□] that performs tracking without shape estimation. The experiments show clear benefit of the proposed approach in the case where the initial model shape differs from the tracked hand shape. The proposed method manages to converge early within the sequence to good and stable hand shape parameter estimates. In summary, the main contributions of this paper are:

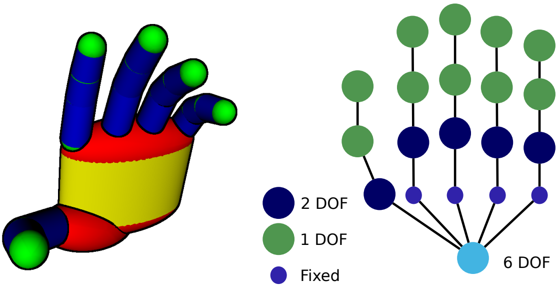
- An online and real time method that (a) tracks the 3D position, orientation and full articulation of the human hand and (b) estimates the hand shape parameters.
- The development of an optimization/fitting method that takes into account a frame history to estimate the shape parameters of a human hand.

2 Pose & Shape Tracking

Let $\mathbf{x}_t$ and $\mathbf{y}_t$ be the pose and shape state at time step t respectively. The method employs the HMF particle filter variant [□] to track the pose parameters given the hand shape. The shape estimate at each time step is provided by per-frame shape parameters optimization, followed by a robust fitting framework. The per-frame optimizer generates possible shape proposals $\mathbf{y}_t^{pso}$ by optimizing the shape parameters at each frame given fixed (i.e., already estimated) pose parameters. Since the actual shape parameters are constant, the robust fitting cross-validates the shape proposals over a frame history. The output of the fitting is the best estimate given the considered history of the shape parameters $\bar{\mathbf{y}}_t$ that is used in the subsequent frame by the pose tracker. The steps of the algorithm are shown in Alg. 1. In the following sections we describe in detail the employed model and the stages of this algorithm.

Algorithm 1 Pose/Shape Tracking

```
Initialize at $t = 0$ with the approximate pose and shape $\mathbf{x}_0, \mathbf{y}_0$.
for each frame $t$ do
    Update(HMF) the pose given the previous shape estimate $\bar{\mathbf{y}}_{t-1}$ (Sec. 2.2).
    Optimize(PSO) the shape to obtain the per-frame shape estimate $\mathbf{y}_t^{pso}$ (Sec. 2.3).
    Fit the shape params over the shape history to obtain $\bar{\mathbf{y}}_t$ (Sec. 2.4).
end for
```



(a) (b)

Figure 1: The employed 3D hand model: (a) hand geometry, (b) hand kinematics.

2.1 Hand model

The hand model (Fig. 1) is built using two basic geometric primitives: a sphere and a cylinder. The shape of the hand $\mathbf{y}_t$ is parametrized by an 11D vector controlling sphere radii and cylinder radii and heights. The vector includes the finger lengths (5D), the radii of the four spheres located on each finger (3 joints and tip resulting in 4D representation since we consider fixed radii ratios for the corresponding sphere of each finger that take into account the relative finger widths), and finally the width and height of the palm (2D). This shape parametrization was chosen since in practice it can model a broad spectrum of hand shape variations. However, the proposed method does not depend on the particular parametrization and, thus, it can be straightforwardly replaced by a different one either designed or learned. The pose of the hand $\mathbf{x}_t$ is parametrized by a 27D vector. The kinematics of each finger are modeled using four parameters, two for the base angles and two for the remaining joints. Bounds on the values of these parameters are set based on anatomical studies [1]. The global position of the hand is represented by a fixed point on the palm and the global orientation.

2.2 HMF Pose Tracking

In this section we outline the application of the HMF tracking framework [10] to the hand tracking problem first presented in [11]. The framework updates at each frame t the pose parameters $\mathbf{x}_t$ given the estimate of the shape parameters $\bar{\mathbf{y}}_{t-1}$. The HMF uses several auxiliary models that are able to provide information for the state of the main model which is to be estimated. Each of the auxiliary models tracks a distinct part of the hand; we use one for the palm with 6-DOF for its 3D position and orientation and one for each finger with 4-DOF for the joint angles (Fig. 1(b)). We define the full state $\mathbf{x}_t$ at a time step t as the concatenation of the sub-states that correspond to the M auxiliary models and the main model $\mathbf{x}_{[0:M]t}$. By $\mathbf{z}_t$ we denote the measurements at time step, t. For notational simplicity in the following we drop the dependence of measurements on the shape parameters since the latter are held constant throughout the update.

The framework follows the Bayesian approach for tracking [2]. By $\mathbf{x}_{0:t}$ we denote the state sequence $\{\mathbf{x}_0 ... \mathbf{x}_t\}$ and accordingly by $\mathbf{z}_{1:t}$ the set of all measurements $\{\mathbf{z}_1 ... \mathbf{z}_t\}$ from time step 1 to t. The tracking consists of calculating the posterior $p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t})$ at every step, given the measurements up to that step and a prior, $p(\mathbf{x}_0)$. Using the state decomposition the solution is expressed as:

$$p(\mathbf{x}_{0:t}|\mathbf{z}_{1:t}) \propto p(\mathbf{x}_{0:t-1}|\mathbf{z}_{1:t-1}) \prod_i p(\mathbf{z}_t|\mathbf{x}_{[i]t})p(\mathbf{x}_{[i]t}|Pa(\mathbf{x}_{[i]t})), \tag{1}$$

where by $Pa(\mathbf{x}_{[i]t})$ we denote the parent nodes of $\mathbf{x}_{[i]t}$. The parent nodes of the main model are the auxiliary models of the same time step. The parent node of an auxiliary model is the main model of the previous step. In (1) we make the approximation that the observation likelihood is proportional to the product of individual model likelihoods $p(\mathbf{z}_t|\mathbf{x}_{[i]t})$. To efficiently approximate the posterior given the above state decomposition we use a particle filter that sequentially updates the sub-states. The algorithm approximates this posterior by propagating a set of particles. The particles for each node are drawn from the proposal distribution $q(\mathbf{x}_{[i]t}|Pa(\mathbf{x}_{[i]t}))$ which models the sub-state of a node given its parents. Using these factorizations, the algorithm sequentially updates the sub-states by sampling from the factor of the proposal that corresponds to the i-th sub-state and subsequently updating the weights with the i-th factor of the likelihood. The state estimate for each frame $\bar{\mathbf{x}}_t$ is given by the weighted average of the main model particles.

The observation likelihood measures the degree of matching between a model pose and the observations as in [11]. The input of the method is an RGB-D image and a model pose. The model pose is rendered and is compared with the image which results in a distance D that takes into account the silhouette and depth match. The likelihood $L(\mathbf{x},\mathbf{z})$ is calculated as an exponential function of that distance.

2.3 Per-Frame Shape Estimation

At each time step t the particle filter described above maintains a set of N weighted particles for the main model: $\{\mathbf{x}_t^{(n)}, \bar{\mathbf{y}}_{t-1}, w_t^{(n)}\}_{n=1}^N$. An optimization of the shape parameters using the PSO algorithm is performed independently for the $N^{pso} << N$ (in this work we used values for the N^{pso} in the range of $5-10$) particles with the higher weights resulting in N^{pso} updated estimates for the shape parameters paired with the corresponding pose parameters: $\{\mathbf{x}_t^{(n)}, \mathbf{y}_t^{(n)}\}_{n=1}^{N^{pso}}$. The likelihood of these pairs is calculated and the shape parameters with the max-likelihood $\mathbf{y}_t^{pso}$ are retained as the current shape estimate. The algorithm steps are shown in Alg. 2.

Algorithm 2 Per-Frame Shape Optimization

```
Input: $\{\mathbf{x}_t^{(n)}, \bar{\mathbf{y}}_{t-1}, w_t^{(n)}\}_{n=1}^N$.
Sort the particles in descending order according to their weights.
for each particle $n = 1$ to $N^{pso}$ do
    Optimize(PSO) the shape to obtain: $\mathbf{y}_t^{(n)}$.
end for
Select the best shape: $\mathbf{y}_t^{pso} = \arg\max_{\mathbf{y}} L([\mathbf{x},\mathbf{y}],\mathbf{z}_t)$, $\mathbf{x}$, $\mathbf{y} \in \{\mathbf{x}_t^n, \mathbf{y}_t^n\}_{n=1}^{N^{pso}}$
Output: $\mathbf{y}_t^{pso}$.
```

2.4 Robust Shape Fitting

The per-frame shape estimates up to the current frame are processed by a robust fitting framework. The framework stores a history of N_f frames along with their corresponding poses $H_F = \{\mathbf{z}_f, \bar{\mathbf{x}}_f\}_{f=1}^{N_f}$, and a history of N_s shape parameters $H_S = \{\mathbf{y}_s^{pso}\}_{s=1}^{N_s}$. Every shape $\mathbf{y}_s$ in history H_S is paired with every pose $\bar{\mathbf{x}}_f$ in history H_f. The likelihood $L([\bar{\mathbf{x}}_f, \mathbf{y}_s^{pso}], \mathbf{z}_f)$ of each pair is evaluated and the shape parameters are ranked according to that likelihood. The per-frame ranks $R_f(\mathbf{x}_f, \mathbf{y}_s)$ of each shape parameter set $\mathbf{y}_s^{pso}$ are then averaged to obtain the global rank for the set $R(\mathbf{y}_s)$. The new estimate for the shape parameters is selected by choosing the estimate with the best average rank among the history frames. The steps of the algorithm are displayed in Alg. 3. In order to keep the computational cost reasonable we impose upper bounds in the number of history frames N_f and shape estimates N_s. The values for these bounds are discussed in the experiments section.

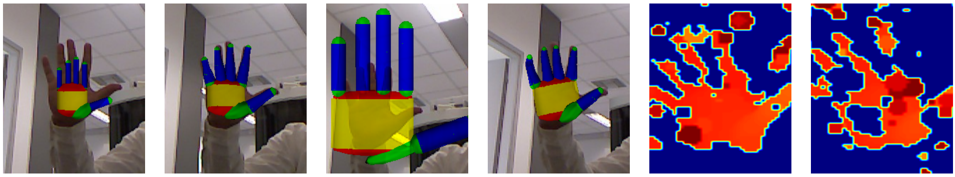
Algorithm 3 Shape Fitting

```
Initialize $H_F = \emptyset$, $H_S = \emptyset$.
for each frame $t$ do
    Add current frame/pose estimate: $H_F = H_F \cup \{\mathbf{z}_t, \bar{\mathbf{x}}_t\}$.
    Add current PSO shape estimate: $H_S = H_S \cup \mathbf{y}_t^{pso}$.
    for each frame/pose $\mathbf{z}_f$, $\mathbf{x}_f$ in $H_F$ do
        for each shape estimate $\mathbf{y}_s$ in $H_S$ do
            Calculate the data likelihood: $L([\mathbf{x}_f, \mathbf{y}_s], \mathbf{z}_f)$.
        end for
        Calculate the frame ranking of each shape estimate: $R_f(\mathbf{x}_f, \mathbf{y}_s)$
    end for
    Calculate the average rank of each shape estimate: $R(\mathbf{y}_s) = \langle \{R_f(\mathbf{x}_f, \mathbf{y}_s)\}_{f=1}^{N_f} \rangle$
    Output: $\bar{\mathbf{y}}_t = \arg\max_{\mathbf{y}} R(\mathbf{y}_s)$
end for
```

3 Experiments

We performed extensive experiments to assess the performance of the proposed approach. We used real data obtained by RGB-D sensors to qualitatively evaluate the methods. For quantitative evaluations we used synthetic data since real world annotated data are difficult to obtain. The methods that have been included in our comparative evaluation are: (i) **HMF**: The method of [□] that tracks a hand without estimating its shape. (ii) **SOP**: Tracking the hand through HMF and perform only per-frame shape optimization. (iii) **SFT**: The full proposed method. For all the methods we used the same likelihood to link the model with the observations. The methods were tested on a computer equipped with an Intel i7-4770 CPU (quad-core 3.4GHz), and an Nvidia GTX 780 GPU.

The synthetic dataset that we used for the evaluation consists of 700 frames of free hand movement mirrored to obtain a 1400 frames circular sequence. The reason why we did this mirroring is to permit the evaluation of each method from different starting points within the sequence. More specifically, the pose initialization of the methods is performed using the ground truth position on 14 different frames. For the shape initialization we test different parameter sets that are scaled with respect to the groundtruth shape by a ratio R_s. We test



(a) (b) (c) (d) (e) (f)

Figure 2: **SFT** Tracked sequences examples. Two sets of two frames of the same sequence tracked with two different shape initializations. Figures (a) and (c) show the initialization while (b) and (d) the pose/shape estimation several frames later. The example demonstrates that the method can handle shape initializations that are very far from the real shape. **(a,b):** $R_s = 0.75$. **(c,d):** $R_s = 2$. **(e,f):** Example depth maps with noise ratio of 0.4.

values for R_s from 0.5 to 2 that correspond to hand shapes initializations that are smaller or larger from the ground truth shape by the same ratios.

The pose error E_p measures the average distance between corresponding phalanx endpoints over a sequence similarly to [6]. The shape error E_s is the Euclidean distance between the 11D tracked and ground truth shape parameter vectors. For each experiment and for each method we perform 42 runs and we report the mean error in all runs.

We are also interested in assessing the performance of the proposed method with respect to noise in the observations. We simulate the imperfect data that come from a real depth sensor, the imperfect foreground detection and the possible occlusions from unknown objects. To do so, we randomly generated disk shaped regions in the first frame. Inside each such region we either: (i) completely remove the depth and foreground labeling; (ii) label the area as foreground and give it a depth value that is modeled by a Gaussian around the real average depth. These "noise disks" then perform random walk in the input sequence which renders the noise more persistent than the type of noise that has been used in [11] where at each frame the noise disks appear in random locations. Fig. 2(e),2(f) shows some example frames from a sequence with noise. This type of noise has the double effect of removing useful data while introducing observations with depth values that are close to the actual depth of the target. Thus, such artifacts cannot be discarded by some type of pre-processing.

History Size As already discussed, the proposed robust fitting method stores a history of N_f frames/pose estimates and N_s shape parameter estimates. Since these parameters affect the accuracy and the computational cost of the algorithm the first experiment that we performed measures the error metrics for different values of these parameters. We found that the value of N_s had little effect on the accuracy with any value greater that 3-5 performing equally well. However, the number of history frames affects the accuracy of the method, especially in the presence of noise. The results for a sequence with high noise ratio (0.3) are presented in Fig. 4(a),4(b). With short frame histories the method performs poorly. However, the accuracy increases fast with the N_f. For the rest of the experiments we chose the values $N_f = 50$ and $N_s = 5$ as a good compromise between speed and accuracy.

Computational Cost The most computationally demanding part of all evaluated methods is the likelihood evaluation. Therefore, a reasonable criterion for quantifying the computational cost is the total number of required such evaluations. In practice, even for two methods

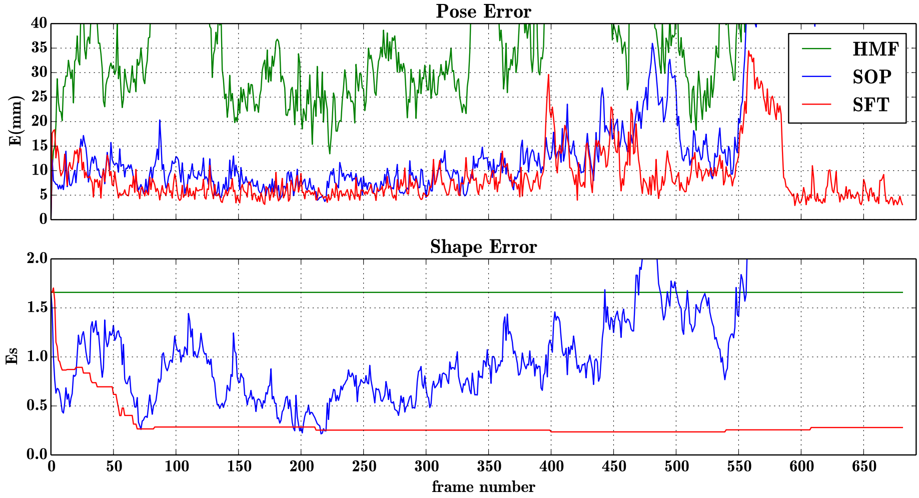
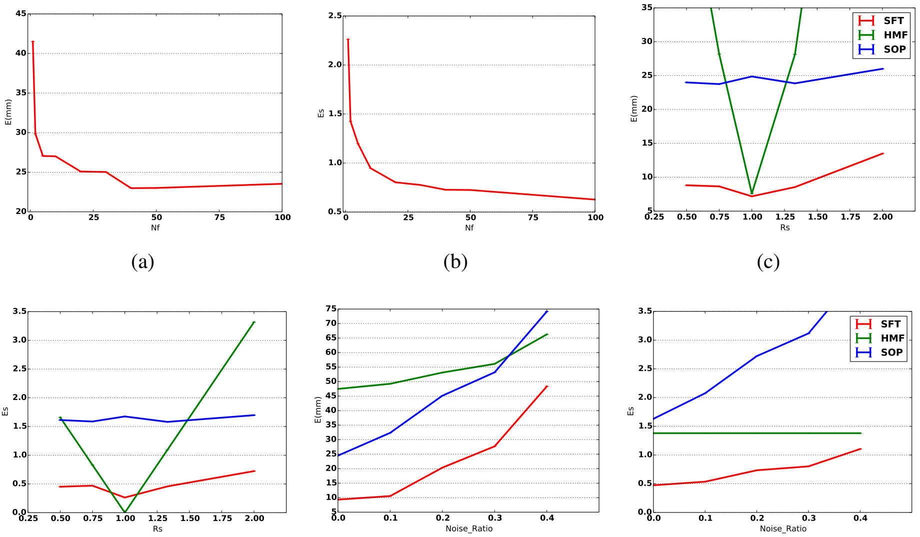


Figure 3: Pose (E_p) and shape (E_s) error evolution over a simulated noisy sequence. The **HMF** method does not alter the shape parameters and consequently E_s remains constant. **SOP** performs per-frame optimization of the shape parameters and provides unstable shape estimates which is evident from the evolution of E_s. **SFT** converges fast to a low E_s value.

that perform the same number of evaluations, the actual speed varies depending on their degree of parallelization. For the **HMF** tracker we chose a typical parameter setting with 100 particles that lead to around 800 likelihood evaluations and a frame rate of $60fps$. The per-frame optimization uses a PSO algorithm with 12 generations and 20 particles per generation for a total of 240 likelihood evaluations, around $80fps$. The meta-fitting with the aforementioned parameter settings requires the much lower number of 55 likelihood evaluations. The **SFT** method that makes use of all of the above modules runs at approximately $25-30fps$, thus in real-time.

Shape Estimation Convergence An important property that we verified was that the proposed **SFT** method's shape estimation converges. Fig. 3 displays the evolution of E_p and E_s throughout the sequence during a typical run. **HMF** method's shape error E_s does not change since it does not perform shape estimation. By only performing per-frame optimization (**SOP**) we get some low E_s values in several frames but the shape estimation remains unstable. On the other hand, **SFT** converges fast to a low E_s value that corresponds to a good shape estimation. From the pose error E_p we observe that the pose estimation is evidently influenced by the shape estimation and therefore **SFT** performs best while the two other methods completely lose track somewhere around frame 550.

Inaccurate Shape Initialization We investigated the effect of the shape initialization ratio R_s on the pose and shape estimation and we present the results in Fig. 4(c),4(d). The first thing to note here is that for the R_s value of 1, that is, for perfect hand shape initialization, the proposed **SFT** method has the same pose accuracy with **HMF** method. For R_s values that range from 0.5 to 2 **SFT** has a much lower E_p and E_s than the other two methods. From a practical point of view, this range ensures that with a single average shape initialization, we can cover the vast majority of the possible subject hands that we might encounter in practice, including hands of children.



(a) (b) (c)

(d) (e) (f)

Figure 4: **Quantitative Experiments (a, b)**:Pose and shape error for various maximum history frame values N_f. **(c, d)**:Pose and shape error for various shape initializations. The initialization ratios (x-axis) express the ratio between the shape parameters values that were used for initialization and the ground truth shape parameters. **(d, f):**Pose and shape error for various sequence noise levels.

Effects of Noise. In another experiment, we assessed the performance of the methods in the presence of noisy observations. We vary the ratio of the image area that is contaminated by noise and we plot the results in Fig. 4(e),4(f). As it can be verified, the proposed **SFT** method outperforms the other two. We also note that the **SOP** method's accuracy drops dramatically with increasing noise levels. This is explained by the fact that the method overfits the model to noise, as it performs only per-frame optimization.

Real World Data Experiments We performed extensive experiments with real world RGB-D data. The proposed method behaves well even when the shape initialization is very different from the actual hand shape. Fig. 2(a),2(c) shows indicative results with two different shape initializations. Test runs on real data are provided in: `https://youtu.be/4dgwoKkDSn8`.

4 Conclusions

We presented a method that performs online, real time 3D tracking of a human hand with simultaneous estimation of the hand model parameters. We experimentally demonstrated the efficacy of the method using real RGB-D and simulated data. The method converges even with initializations that are far from the correct shape. In practice, this means that a single, “average hand" model can set the starting point for hand adaptation and tracking of any possible hand. The personalization of the hand model can be of interest on its own right. At a minimum, the resulting model can be used by other tracking methods that do not perform hand model adaptation. In this work, we used a specific set of shape parameters. As future work, we consider learning the optimal hand shape parametrization from data, where optimality is understood in terms of being able to explaining the maximum actual hand shape variability with the minimum possible number of tunable parameters.

Acknowledgments

This work was supported by the EU IST-FP7-IP-288533 project RoboHow.Cog.

References

[1] Irene Albrecht, Jorg Haber, and Hans-Peter Seidel. Construction and animation of anatomically based human hand models. In *Proceedings of the 2003 ACM SIGGRAPH/Eurographics Symposium on Computer Animation*, SCA ’03, pages 98–109, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association. URL http://dl.acm.org/citation.cfm?id=846276.846290.

[2] M.S. Arulampalam, S. Maskell, N. Gordon, and T. Clapp. A tutorial on particle filters for online nonlinear/non-gaussian bayesian tracking. *IEEE Transactions on Signal Processing*, 50(2):174–188, 2002. ISSN 1053-587X. doi: 10.1109/78.978374.

[3] V. Athitsos and S. Sclaroff. Estimating 3d hand pose from a cluttered image. In *2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2003. Proceedings*, volume 2, pages II–432–9 vol.2, 2003. doi: 10.1109/CVPR.2003.1211500.

[4] Luca Ballan, Aparna Taneja, Jurgen Gall, Luc Van Gool, and Marc Pollefeys. Motion capture of hands in action using discriminative salient points. In *European Conference on Computer Vision ECCV*. Springer Berlin Heidelberg, 2012. URL http://link.springer.com/chapter/10.1007/978-3-642-33783-3_46.

[5] T.E. de Campos and D.W. Murray. Regression-based hand pose estimation from multiple cameras. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 1, pages 782–789, 2006. doi: 10.1109/CVPR.2006.252.

[6] H. Hamer, K. Schindler, E. Koller-Meier, and L. Van Gool. Tracking a hand manipulating an object. In *2009 IEEE 12th International Conference on Computer Vision*, pages 1475–1482, 2009. doi: 10.1109/ICCV.2009.5459282.

[7] J. Kennedy and R. Eberhart. Particle swarm optimization. In *, IEEE International Conference on Neural Networks, 1995. Proceedings*, volume 4, pages 1942–1948 vol.4, 1995. doi: 10.1109/ICNN.1995.488968.

[8] C. Keskin, F. Kirac, Y.E. Kara, and L. Akarun. Real time hand pose estimation using depth sensors. In *2011 IEEE International Conference on Computer Vision Workshops (ICCV Workshops)*, pages 1228–1234, 2011. doi: 10.1109/ICCVW.2011.6130391.

[9] Cem Keskin, Furkan Kirac, Yunus Emre Kara, and Lale Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *European Conference on Computer Vision (ECCV)*, Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012. URL http://link.springer.com/chapter/10.1007/978-3-642-33783-3_61.

[10] Alexandros Makris, Dimitrios Kosmopoulos, Stavros Perantonis, and Sergios Theodoridis. A hierarchical feature fusion framework for adaptive visual tracking. *Image and Vision Computing*, 29(9):594–606, 2011. ISSN 0262-8856. doi: 10.1016/j.imavis.2011.07.001. URL http://www.sciencedirect.com/science/article/pii/S0262885611000503.

[11] Alexandros Makris, Nikolaos Kyriazis, and Antonis Argyros. Hierarchical Particle Filtering for 3d Hand Tracking. In *IEEE Conference on Computer Vision and Pattern Recognition: IEEE Computer Society Workshop on Observing and understanding hands in action (HANDS)*, 2015.

[12] S. Malassiotis and M. G. Strintzis. Real-time hand posture recognition using range data. *Image and Vision Computing*, 26(7):1027–1037, 2008. ISSN 0262-8856. doi: 10.1016/j.imavis.2007.11.007. URL http://www.sciencedirect.com/science/article/pii/S0262885607002090.

[13] I Oikonomidis, N. Kyriazis, and AA Argyros. Tracking the articulated motion of two strongly interacting hands. In *2012 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1862–1869, 2012. doi: 10.1109/CVPR.2012.6247885.

[14] I. Oikonomidis, M.I.A. Lourakis, and A.A. Argyros. Evolutionary quasi-random search for hand articulations tracking. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3422–3429, 2014. doi: 10.1109/CVPR.2014.437.

[15] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis Argyros. Efficient model-based 3d tracking of hand articulations using kinect. pages 101.1–101.11. British Machine Vision Association, 2011. doi: 10.5244/C.25.101. URL http://www.bmva.org/bmvc/2011/proceedings/paper101/index.html.

[16] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. Realtime and robust hand tracking from depth. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1106–1113, 2014. doi: 10.1109/CVPR.2014.145.

[17] James M. Rehg and Takeo Kanade. Visual tracking of high DOF articulated structures: An application to human hand tracking. In *Proceedings of the Third European Conference-Volume II on Computer Vision - Volume II*, ECCV '94, pages 35–46, London, UK, UK, 1994. Springer-Verlag. URL http://dl.acm.org/citation.cfm?id=645308.649148.

[18] G. Ros, J.M. del Rincon, and G.G. Mateos. Articulated particle filter for hand tracking. In *2012 21st International Conference on Pattern Recognition (ICPR)*, pages 3581–3585, 2012.

[19] Toby Sharp, Cem Keskin, Duncan Robertson, Jonathan Taylor, Jamie Shotton, David Kim, Christoph Rhemann, Ido Leichter, Alon Vinnikov, Yichen Wei, Daniel Freedman, Pushmeet Kohli, Eyal Krupka, Andrew Fitzgibbon, and Shahram Izadi. Accurate, Robust, and Flexible Real-time Hand Tracking. In *Proceedings of the 33rd Annual ACM Conference on Human Factors in Computing Systems*, CHI '15, pages 3633–3642, New York, NY, USA, 2015. ACM. doi: 10.1145/2702123.2702179. URL http://doi.acm.org/10.1145/2702123.2702179.

[20] S. Sridhar, A. Oulasvirta, and C. Theobalt. Interactive markerless articulated hand motion tracking using RGB and depth data. In *2013 IEEE International Conference on Computer Vision (ICCV)*, pages 2456–2463, 2013. doi: 10.1109/ICCV.2013.305.

[21] Srinath Sridhar, Helge Rhodin, Hans-Peter Seidel, Antti Oulasvirta, and Christian Theobalt. Real-time hand tracking using a sum of anisotropic gaussians model. 2014. URL http://handtracker.mpi-inf.mpg.de/projects/ellipsoidtracker_3dv2014/.

[22] E.B. Sudderth, M.I Mandel, W.T. Freeman, and AS. Willsky. Visual hand tracking using nonparametric belief propagation. In *Conference on Computer Vision and Pattern Recognition Workshop, 2004. CVPRW '04*, pages 189–189, 2004. doi: 10.1109/CVPR.2004.200.

[23] J. Taylor, R. Stebbing, V. Ramakrishna, C. Keskin, J. Shotton, S. Izadi, A. Hertzmann, and A. Fitzgibbon. User-specific hand modeling from monocular depth sequences. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 644–651, 2014. doi: 10.1109/CVPR.2014.88.

[24] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Trans. Graph.*, 33(5):169:1–169:10, 2014. ISSN 0730-0301. doi: 10.1145/2629500. URL http://doi.acm.org/10.1145/2629500.

[25] Robert Wang, Sylvain Paris, and Jovan Popovic. 6d hands: Markerless hand-tracking for computer aided design. In *Proceedings of the 24th Annual ACM Symposium on User Interface Software and Technology*, UIST '11, pages 549–558, New York, NY, USA, 2011. ACM. doi: 10.1145/2047196.2047269. URL http://doi.acm.org/10.1145/2047196.2047269.

[26] Chi Xu and Li Cheng. Efficient hand pose estimation from a single depth image. In *2013 IEEE International Conference on Computer Vision (ICCV)*, pages 3456–3462, 2013. doi: 10.1109/ICCV.2013.429.