Convolution surfaces model for hand tracking

I. Introduction

Hand tracking is a process of accurately reconstructing shape and articulation of human hands. It is a crucial component of natural human-computer interfaces and animation of humanoid avatars. A number of hand tracking algorithms has been recently proposed Keskin et. al. [3], Melax et. al. [5], Tang et. al. [16], Oikonomidis et. al. [6], Schroder et. al. [10], Tompson et. al. [18], Qian et. al. [7], Tagliasacchi et. al. [15], Sridhar et. al. [12], Sun et. al. [14] and Sharp et. al. [11]. However, in most consumer applications hand tracking is just a single components of a bigger pipeline (Figure 1). Before tracking, a suitable hand model is obtained. Once the hand pose parameters are found, the tracking result is displayed by skinning the model. Both modeling and skinning tasks are not trivial.

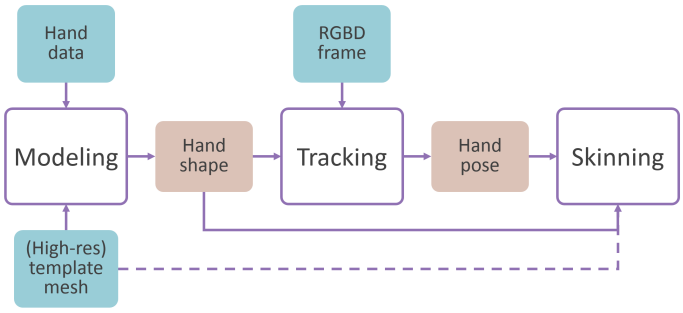

Fig. 1. Generic pipeline for hand tracking. The input data that does not depend on the internal hand model representation is shown in blue, the representation-dependent components are shown in beige, and are listed in Table I for different representations.

A. Why customizing the hand model?

Hand model should be able to accurately represent the observed data. The discrepancy between the optimal model pose given the data and the true hand pose can be significant, especially if the hand model does not reflect all the degrees of freedom of a hand (Figure 2, left).

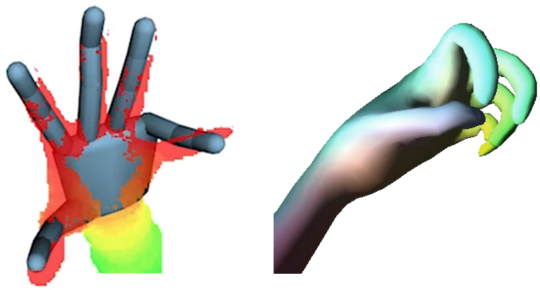

Fig. 2. Left - coarse hand model from [15]; right - hand model animated with linear blend skinning from [11].

B. Why realistically animating the hand model?

The hand skinning quality is obviously important for digital avatars applications. In AR and VR applications a 3D hand model can properly interact with 3D objects, establish a realistic contact and disappear behind them. Given that, the degree of immersion into virtual reality depends on whether a user sees own realistic hands (find a study mentioned by Leap Motion). The simple skinning approaches like linear blend skinning may generate implausible results (Figure 2, right).

C. Alternative hand model representations

Each stage of the pipeline requires a hand model. There is several different hand model representations suggested by previous authors (see Figure 3). Each representation is well suited for one of the stages, since it was used for the task on the first place. We argue that each representation also has weaknesses, which is why there exists a set of alternatives.
We suggest to use convolution surfaces representation of the hand model.
Convolution surface is an implicit surface which is described by a control skeleton. The skeleton may consist of points, edges or polygons [2]. In each vertex of the skeleton we define a radius. The radius in intermediate points is a linear combination of the radii at the neighboring vertices. Given the topology of the underlying skeleton, the model can be represented with convolution surface up to high precision (find some theoretical estimates). Next we present the arguments why convolution surfaces representation is suitable for all the stages of the pipeline.

	Hand pose	Hand shape
Triangular mesh with embedded skeleton, [17]	Vertices and bones positions	Vertices and bones positions
Cylinder model, [15]	Cylinders size and transformations	Cylinders transformations
Convolution surfaces model	Positions and radii of control points	Positions of control points

TABLE I
Comparison of different hand model representations

D. Convolution surfaces for model fitting

The spheres and mixed cylinders/spheres hand model representations (Figure 3 a, b) are ubiquitous in hand tracking, because they are well suited for tracking tack per se (see next) and can be quickly to created manually. If a small number of building blocks is used, the precision of the model is low, especially in the palm region. A higher precision can be obtained by increasing the number of primitives, which defeats the purpose of model simplicity. Convolution surfaces representation gives higher precision for the same number of building blocks. Add

experimental or theoretical support for convolution surfaces. The triangles mesh representation can approximate the hand to a high precision. However, there is obvious way to specify the model parts that are rigid and should be kept the same shape between the poses. This results in overfitting to local skin deformations.

Fig. 3. Hand model representations from works of (a) Qian et. al. [7], (b) Oikonomidis et. al. [6], (c) Melax et. al. [5] and (d) Sharp et. al [11].

E. Convolution surfaces for hand tracking

For model based tracking the main operation is to find the closest point on the model for a given data point. This operation is can be done in closed for each rigid segment with spheres/cylinder and convolution surfaces model representation. For a triangular mesh this operation has complexity linear in number of triangles. Moreover, the triangular mesh has (much) more degrees of freedom than the underlying problem. Without additional regularization, rigid parts of the hand model can deform to fit the data and the individual vertices can shift to fit the sensor noise.

F. Convolution surfaces for hand skinning

The Linear Blend Skinning approach used to pose the triangular mesh model in previous works ([11], [9]) creates artifacts, the fingers looks like made from rubber. The spheres/cylinders model is not suitable for realistic animation, therefore a retargeting step to a template mesh is required. Retargeting does not only demand additional effort, but also brings additional imprecision. The state of the art approaches in hand skinning are implicit surfaces-based ([19], [20]). A convolution surfaces model serves a ready to use input for such an approach.

G. Contributions

- Developed an approach for approximation a model with convolution surface, given a skeleton topology.
- Formulated position-based inverse kinematics algorithm for hand tracking. The position based approach does not require walking through the hierarchical chain of joint, thus saving the computational power. Also, position-based inverse kinematics does not involve linear approximations or rotations, thus making the optimization more stable.
- Suggested an automatic approach for field functions construction in implicit skinning. Replaced newton iteration for vertex projection by a closed form solution.

II. Related Literature

- Albrecht et al. [1] developed an approach for creating an anatomically realistic hand model that includes bones and muscles structure. Their approach requires several prerequisites including plaster cast of a human hand and laser scanner for manually creating a physically realistic hand template. Given user-defined correspondences between 3D feature points and the hand image, a specific hand model is created by deforming a generic hand model.
- Rhee et. al. [8] use a single image of a hand at rest pose to infer joint hand joint locations from skin creases. Given the skeleton obtained at the previous step and the hand contour from the image, they deform a template hand mesh to fit this data.
- Straka et al. [13] also fit the template mesh with attached skeleton to 3D data. The model is deformed to explain the data while keeping the vertices attached to their corresponding bones. It is on clear whether the approach whether be able to handle a hand motion sequence, since the results are demonstrated on a full body model.
- Taylor et. al. [17] generate a user-specific hand model from an RGBD video sequence. The model is represented as a triangular mesh with an embedded skeleton. In each frame the hand pose is initialized using an appearance-based tracking algorithm. The hand model parameters are found by solving a single optimization problem formulated for the entire video sequence which also finds hand pose in each frame.
- Khamis et. al. [4] fit a hand model for a specific user by finding its shape coordinates in the basis of mesh matrices and bones locations. As in the approach by Taylor et. al, they optimize simultaneously for pose and shape parameters in all the frames of an RGBD sequence across all the subjects. Requires large number of subjects as a regularization for excessive degrees of freedom.

  The results generated by the approaches listed above could be used as an input for our system to create a hand model representation adapted for efficient tracking and animation.

III. Results

A. Modeling

B. Tracking

In this paper instead of a standard inverse kinematics approach for aligning the model with data we use *"As Rigid As Possible"* approach. The hand model is parametrized with the locations of the vertices of the hand skeleton $c = c_1, c_2, ...c_N$.

$$\min_c E_{ICP} + E_{ARAP} \tag{1}$$

The first energy E_{ICP} models a 3D geometric registration in the spirit of ICP as

$$E_{ICP} = \omega_1 \sum_{p \in P} \|p - \Pi(p, c)\|_2 \tag{2}$$

where P is the set of data points.

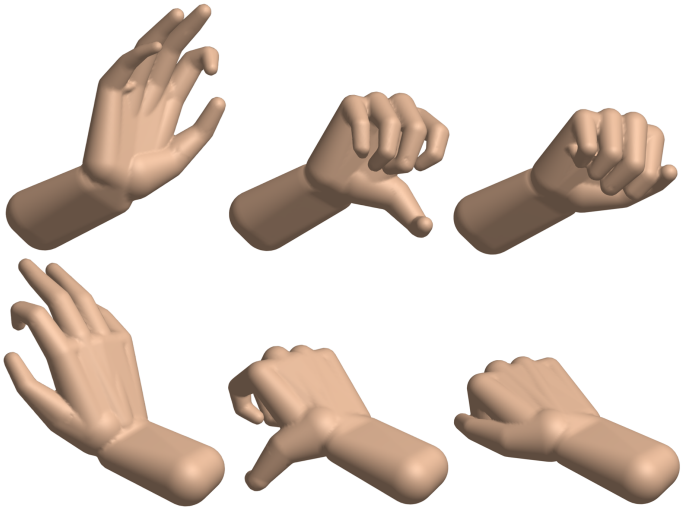

Fig. 4. Fitting convolution surfaces hand model to several poses of the same hand.

The second energy E_{ARAP} is needed for shape preservation. Denote locations of hand skeleton vertices at rest pose as c^0 and in iteration t as c^t. Denote the set of all edges of hand skeleton as E.

$$E_{ARAP} = \omega_2 \sum_{e \in E} \|e(c^t) - R_e e(c^0)\|_2^2, \tag{3}$$

where R_e is the optimal rotation to bring the rest pose edge e^0 to the current position e^t. Unless we what some set of edges to rotate as a solid body, the rotation R_e can be expressed in the closed form, such that $R_e e(c^0)$ is collinear to $e(c^t)$. Each optimization iteration consists of two alternating steps: first we find the projections of the data points to the model surface $\Pi(p, c)$ and the optimal rotations R_e. Then we make one step of Levenberg-Marquardt iteration for energy (1).

1) Trying to make several optimization steps while keeping the same correspondences: I am not sure how to implement what you suggested yesterday. If the data-model correspondences are kept fixed, after the first update of the parameters the model points are not in the model surface anymore. So, the optimization does not make much sense (see Figure 5). If I do such optimization, the model just floats away from the data. I also tried doing several iterations of E_{ARAP} only, The initial length of the segment are, of course, restored. However, the model does not take the data into account during these iterations, so the optimization takes more time to converge.

C. Skinning

APPENDIX

A. Convolution segment

A convolution segment (Figure 6) is defined by two spheres $S_1 = \{c_1, r_1\}$ and $S_2 = \{c_2, r_2\}$. Given the data points $P = \{p_i\}_{i=1}^N$ and their projections on the model $Q = \{q_i(c_1, c_2, r_1, r_2)\}_{i=1}^N$ we construct a vector-function $\mathbf{f}$

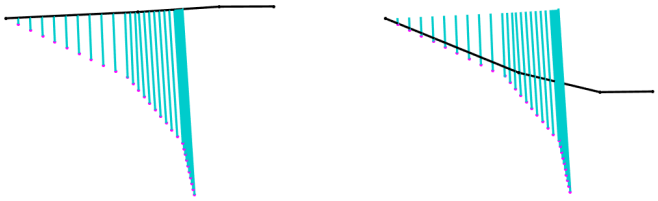

Fig. 5. Updating the centers locations while keeping the data-model correspondences fixed. The model is black, the data is pink, the corresponding points are connected in blue.

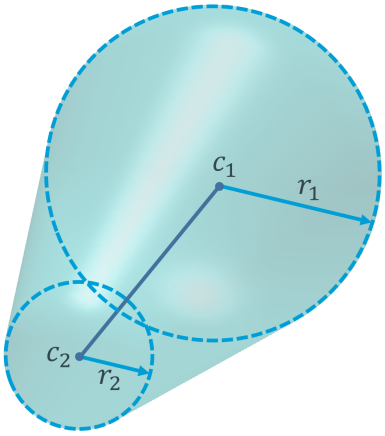

Fig. 6. Convolution segment.

$$\mathbf{f} = \begin{bmatrix} \vdots \\ p_i^x - q_i^x \\ p_i^y - q_i^y \\ p_i^z - q_i^z \\ \vdots \end{bmatrix}. \tag{4}$$

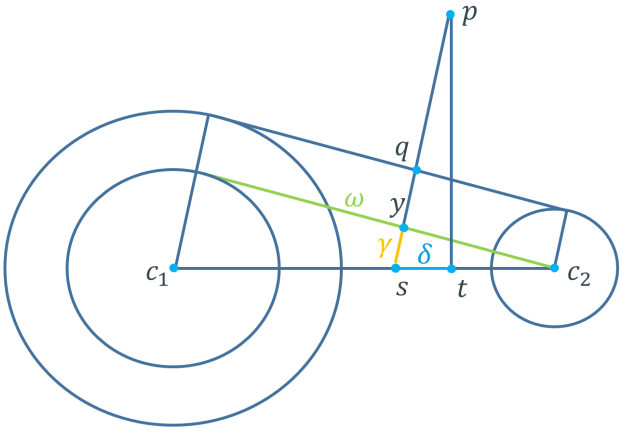

Fig. 7. Convolution segment.

To fit the model to the data, we iteratively minimize $\mathbf{f}$ using Levenberg-Marquardt iteration. In order to compute the Jacobian, the projections q_i should be expressed as functions of model parameters. Let us first find the projection q of a point

p on the segment $\{c_1, c_2\}$. Assume that $c_1 > c_2$. If the point on the segment t closest to p lies at the end of the segment, say, c_1, than $q = c_1 + \frac{r_1(p-c_1)}{\|p-c_1\|}$. Otherwise, the projection q is computed as

$$
\begin{aligned}
u &= c_2 - c_1, \\
v &= p - c_1, \\
\omega &= \sqrt{u^T u - (r_1 - r_2)^2}, \\
\delta &= \frac{(r_1 - r_2)\|p - t\|}{\omega}, \\
s &= t - \delta \frac{c_2 - c_1}{\|c_2 - c_1\|}, \\
\gamma &= \frac{(r_1 - r_2)\|c_2 - t + \delta \frac{u}{\|u\|}\|}{\|u\|}, \\
q &= s + \frac{(p - s)(\gamma + r_2)}{\|p - s\|}.
\end{aligned}
$$

(See Figure 7).

Given the projection $q_i = q_i(c_1, c_2, r_1, r_2)$, the objective function $\mathbf{f} = [\cdots, f_i^T, \cdots]^T$ has the Jacobian $\mathbf{J}$ with $J_{ij} = \frac{\partial f_i}{\partial x_j}$, where $\mathbf{x} = [c_1, c_2, r_1, r_2]$. At each iteration of Levenberg-Marquardt , we first compute the data-model correspondences. In case of convolution segments this amounts to determining whether the closest point is located at one of the end points of the segment or lies in between. Given this correspondence, the Jacobian is computed using chain rule, by composition of derivatives of algebraic operations required to find $q(c_1, c_2, r_1, r_2)$.

B. Convolution triangle

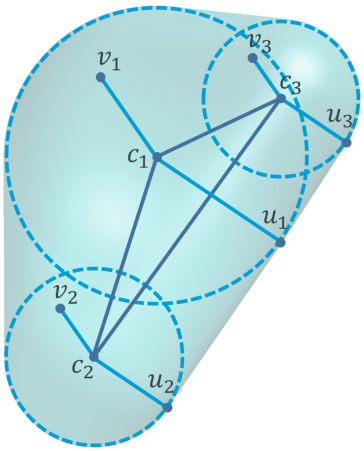

Fig. 8. Convolution triangle.

A convolution triangle (Figure 8) is defined by three spheres $S_1 = \{c_1, r_1\}$, $S_2 = \{c_2, r_2\}$ and $S_3 = \{c_3, r_3\}$. To express the projection q_i as a function of the model parameters, first, let us find the outer tangent planes for the spheres. There exists two outer tangent planes if none of the spheres lies entirely inside of the cone tangent to the other two spheres.

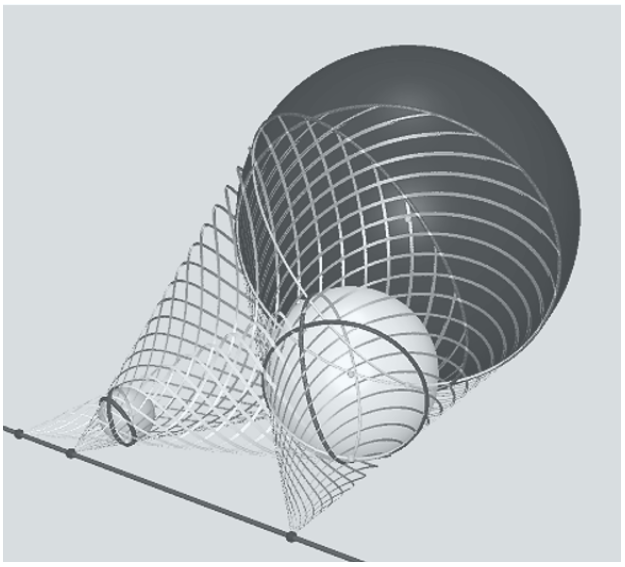

Fig. 9. The three apices of the cones tangent to each pair of spheres lie on a straight line.

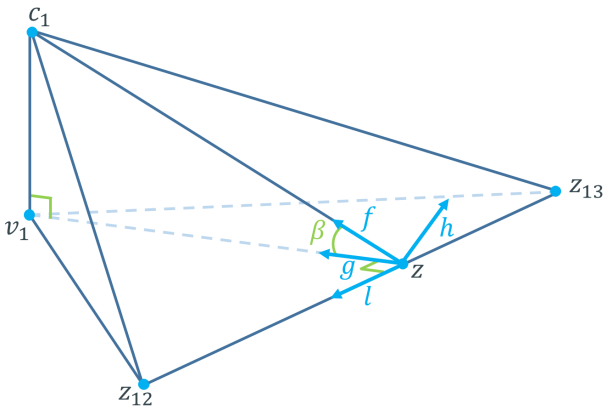

Fig. 10. Computing the tangent plane for three spheres.

C. Computing the tangent plane

In Figure 10, the point z_{12} is an apex of a cone tangent to the spheres S_1 and S_2. The position of z_{12} can be found from similarity of triangles:

$$z_{12} = c_1 + \frac{r_1(c_2 - c_1)}{r_1 - r_2}, \tag{5}$$

In the same way,

$$z_{13} = c_1 + \frac{r_1(c_3 - c_1)}{r_1 - r_3}. \tag{6}$$

The direction vector l of the line, that contains the apices of the tangent cones is found as

$$l = \frac{z_{12} - z_{13}}{\|z_{12} - z_{13}\|} \tag{7}$$

The intersection point of the plane orthogonal to l and going through the point c_1 with the line going through z_{12} and z_{13} is found as

$$z = z_{12} + ((c_1 - z_{12})^T l)l. \quad (8)$$

The sine and cosine of the angle β in triangle $\{c_1, z, v_1\}$ are given by

$$\sin(\beta) = \frac{r_1}{\eta}, \quad (9)$$

$$\cos(\beta) = \frac{\nu}{\eta}, \quad (10)$$

where $\eta = \|c_1 - z\|$ and $\nu = \sqrt{\eta^2 - r_1^2}$.
Denote the tangent point of the sphere S_1 as v_1. The direction vector g of the line $\{c_1, v_1\}$ can be found by rotating the direction vector f of the line $\{c_1, z\}$ by angle β around the axis l.

$$h = \frac{l \times f}{\|l \times f\|}, \quad (11)$$

$$g = \sin(\beta)h + \cos(\beta)f. \quad (12)$$

The tangent point v_1 is found as

$$v_1 = z + \nu g \quad (13)$$

Having the normal vector of the tangent plane $n = \frac{v_1 - c_1}{\|v_1 - c_1\|}$, we can find the tangent points of the spheres S_2 and S_3:

$$v_2 = c_2 + r_2 n \quad (14)$$

$$v_3 = c_3 + r_3 n \quad (15)$$

The second tangent plane $\{u_1, u_2, u_3\}$ is found by rotating the vector f around the axis l by an angle $-\beta$.

D. Computing the projection

If the point, closest to p on triangle $\{v_1, v_2, v_3\}$ or on triangle $\{u_1, u_2, u_3\}$ lies inside of the triangle, than it is the projection q. Otherwise q belongs to the surface of one of the convolution segments $\{c_1, c_2\}$, $\{c_1, c_3\}$, or $\{c_1, c_3\}$. Thus, the candidates for q are the projections of p on the segments q_{12}, q_{13} or q_{23}. The projection q is determined taking into account the distance to p and whether q_{12}, q_{13} and q_{23} are located inside or outside of the segments $\{c_1, c_2\}$, $\{c_1, c_3\}$, and $\{c_1, c_3\}$.
Consider the case when several convolution segments and triangles are used to create a more complex model. To find the projection q, we compute a projection q_j on each building block. The projection q is determined taking into account the distances between q_j and p and whether q_j are located inside or outside of the building blocks.

REFERENCES

[1] Irene Albrecht, Jörg 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*, pages 98–109. Eurographics Association, 2003.

[2] Jules Bloomenthal and Ken Shoemake. Convolution surfaces. In *ACM SIGGRAPH Computer Graphics*, volume 25, pages 251–256. ACM, 1991.

[3] Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. In *Computer Vision–ECCV 2012*, pages 852–863. Springer, 2012.

[4] Sameh Khamis12, Jonathan Taylor, Jamie Shotton, Cem Keskin, Shahram Izadi, and Andrew Fitzgibbon. Learning an efficient model of hand shape variation from depth images.

[5] Stan Melax, Leonid Keselman, and Sterling Orsten. Dynamics based 3d skeletal hand tracking. In *Proceedings of Graphics Interface 2013*, pages 63–70. Canadian Information Processing Society, 2013.

[6] Iason Oikonomidis, Manolis Lourakis, Antonis Argyros, et al. Evolutionary quasi-random search for hand articulations tracking. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 3422–3429. IEEE, 2014.

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

[8] Taehyun Rhee, Ulrich Neumann, and John P Lewis. Human hand modeling from surface anatomy. In *Proceedings of the 2006 symposium on Interactive 3D graphics and games*, pages 27–34. ACM, 2006.

[9] Matthias Schröder, Jonathan Maycock, Helge Ritter, and Mario Botsch. Analysis of hand synergies for inverse kinematics hand tracking. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2013.

[10] Michael Schroder, Jonathan Maycock, Helge Ritter, and Mario Botsch. Real-time hand tracking using synergistic inverse kinematics. In *Robotics and Automation (ICRA), 2014 IEEE International Conference on*, pages 5447–5454. IEEE, 2014.

[11] 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 *Proc. CHI*, volume 8, 2015.

[12] Srinath Sridhar, Franziska Mueller, Antti Oulasvirta, and Christian Theobalt. Fast and robust hand tracking using detection-guided optimization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3221, 2015.

[13] Matthias Straka, Stefan Hauswiesner, Matthias Rüther, and Horst Bischof. Simultaneous shape and pose adaption of articulated models using linear optimization. In *Computer Vision–ECCV 2012*, pages 724–737. Springer, 2012.

[14] Xiao Sun, Yichen Wei, Shuang Liang, Xiaoou Tang, and Jian Sun. Cascaded hand pose regression. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 824–832, 2015.

[15] Andrea Tagliasacchi, Matthias Schroeder, Anastasia Tkach, Sofien Bouaziz, Mario Botsch, and Mark Pauly. Robust articulated-icp for real-time hand tracking. Technical report, 2015.

[16] Danhang Tang, Tsz-Ho Yu, and Tae-Kyun Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3224–3231. IEEE, 2013.

[17] James Taylor, Richard Stebbing, Varun Ramakrishna, Cem Keskin, Jamie Shotton, Shahram Izadi, Aaron Hertzmann, and Andrew Fitzgibbon. User-specific hand modeling from monocular depth sequences. In *Computer Vision and Pattern Recognition (CVPR), 2014 IEEE Conference on*, pages 644–651. IEEE, 2014.

[18] Jonathan Tompson, Murphy Stein, Yann Lecun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics (TOG)*, 33(5):169, 2014.

[19] Rodolphe Vaillant, Loïc Barthe, Gaël Guennebaud, Marie-Paule Cani, Damien Rohmer, Brian Wyvill, Olivier Gourmel, and Mathias Paulin. Implicit skinning: Real-time skin deformation with contact modeling. *ACM Transactions on Graphics (TOG)*, 32(4):125, 2013.

[20] Rodolphe Vaillant, Gäel Guennebaud, Loïc Barthe, Brian Wyvill, and Marie-Paule Cani. Robust iso-surface tracking for interactive character skinning. *ACM Transactions on Graphics (TOG)*, 33(6):189, 2014.