Proactive Sensing for Improving Hand Pose Estimation

Dun-Yu Hsiao
University of Washington
dyhsiao@u.washington.edu

Min Sun
National Tsing Hua Unversity
sunmin@ee.nthu.edu.tw

Christy Ballweber
University of Washington
c.a.ballweber@gmail.com

Seth Cooper
Northeastern University
scooper@ccs.neu.edu

Zoran Popović
University of Washington
zoran@cs.washington.edu

ABSTRACT

We propose a novel sensing technique called *proactive sensing*. Proactive sensing continually repositions a camera-based sensor as a way to improve hand pose estimation. Our core contribution is a scheme that effectively learns how to move the sensor to improve pose estimation confidence while requiring no ground truth hand poses. We demonstrate this concept using a low-cost rapid swing arm system built around the state-of-the-art commercial sensing system Leap Motion. The results from our user study show that proactive sensing helps estimate users' hand poses with higher confidence compared to both *static* and *random* sensing. We further present an online model update to improve performance for each user.

Author Keywords

machine adaptation; hand pose estimation; learning; active

ACM Classification Keywords

H.5.2 Information interfaces and presentation: User interfaces

INTRODUCTION

Using the hands as a natural user interface has gained attention in both academia and industry. In particular, real-time, fine-grained 3D hand pose estimation is the key enabler for a wide range of applications, such as immersive virtual reality, assistive technologies, robotics, home automation, and gaming.

However, real-time 3D hand pose estimation is extremely challenging. Hands have numerous degrees of freedom due to their large number of joints, and hands come with different shapes, sizes, and covering materials (e.g., gloves). Successful early systems that augmented the user's hand with gloves or markers were cumbersome and inaccurate. More recent work focuses on camera-based systems that relax the requirement of augmenting a user's hand, thus allowing a more natural user interaction. Nevertheless, most modern systems still struggle with estimation failure due to the ambiguity of fingers under certain gestures and self-occlusion among different parts of the hand. Such types of failures are also common in commercial systems on the market. Currently, many approaches address these issues by constraining the setup. For instance, they mostly support only front-facing camera scenarios or require multiple cameras to deal with occlusions.

CHI '16, May 07–12, 2016, San Jose, CA, USA
ACM 978-1-4503-3362-7/16/05…$15.00
DOI: http://dx.doi.org/10.1145/2858036.2858587

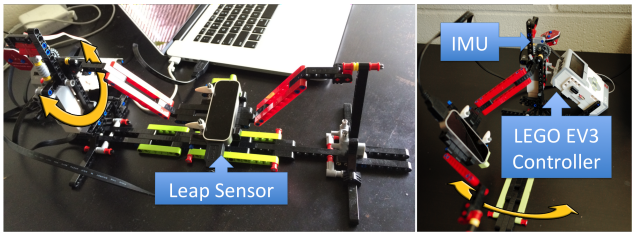


Figure 1. The swing arm setup for proactive sensing. Left: at 30°. Right: at −60°. The arm can swing around one axis, as shown in yellow arrows.

In this work, we present a novel system, called *proactive sensing* (Fig. 1), which addresses these issues by allowing a camera-based sensor with a single *viewpoint* to move and find a better sensing position. There may be multiple cameras in the sensor, but they share a similar, single viewpoint (as with Leap Motion or Kinect systems) and are thus susceptible to similar occlusions. This approach allows users to move their hand freely during interactions and does not require a cumbersome setting with multiple viewpoints. Our approach was inspired by the observation that different hand poses can be robustly estimated under different viewpoints. However, instead of setting up a multiple viewpoint system to capture all viewpoints at all times, we propose to learn a user's operating habits and dynamically predict which viewpoint is the best for estimating the user's current hand pose.

Our prototype system consists of a circular moving swing arm that allows the sensor to freely move to different viewpoints. We show that an existing commercial sensor can benefit from our moving sensor system. In particular, we evaluated 17 users on the task of playing the protein folding game *Foldit* [3]. Our proposed system consistently improved 3D hand pose estimation confidence across different users compared to both a state-of-the-art static sensor solution and a random moving sensor solution. Our system also has the ability to adapt to the habits of each specific user, so that the more they use it, the more robust it becomes.

RELATED WORK

Many real-time hand pose estimation methods have recently been proposed. We summarize the different approaches below.

RGB image. Hand pose estimation using monocular RGB images has been a challenge (see Erol *et al.* [5] for a summary). Much early work (e.g., Wu *et al.* [23] and de La Gorce *et al.* [4]) operated offline by processing recorded sequences. The work of Heap and Hogg [8] using a deformable model is an exception which estimates hand poses at ~10 Hz.

However, the method struggles with complex poses, changing backgrounds, and occlusions. Recently, more real-time systems have been proposed. Song *et al.* [18] propose an efficient multi-stage random forest based hand gesture recognition system on mobile devices. Song *et al.* [17] further propose a system to directly map 2D color images to 3D hand positions and gestures.

Depth image. With the development of consumer depth cameras such as Kinect, a number of new methods have been proposed for reliably estimating hand poses in real-time. Oikonomidis *et al.* [14] present a generative method for hand pose estimation at 15 Hz on a GPU. However, it requires an initial hand pose and cannot recover from estimation failures. Qian *et al.* [15] remove these limitations but requires fingertips to be clearly visible to start. Keskin *et al.* [9] propose a novel framework for real-time hand pose estimation on a CPU. Sharp *et al.* [16] propose a system with an enhanced reinitializer to more robustly handle estimation failure, and used temporal information to achieve a smooth and accurate result. Sridhar *et al.* [19] further achieves 50 Hz using a CPU only implementation.

Multiple viewpoints. To obtain reliable and high quality hand pose estimation results, many methods have relied on data captured by multiple-camera rigs. Most of the work (e.g., Wang *et al.* [21], Zhao *et al.* [24], and Ballan *et al.* [1]) operates offline. One of the exceptions is Sridhar *et al.* [20], which uses a rig with five RGB cameras and a time-of-flight sensor to estimate a user's hand at ~10 Hz. Neverthless, the setup is expensive and complicated, and thus less applicable for general use scenarios.

Wearable solutions. To mitigate failures due to occlusion, Kim *et al.* [10] propose that the user wears a low power depth camera on the wrist. Similarly, Colaço *et al.* [2] explore a low-power, head mounted 3D gesture sensing solution. Harrison *et al.* [7] also propose wearing a RGBD hand gesture recognition system, but on the shoulder. However, these methods have assumptions—such as not holding objects or restricting hand positions—that prevent users from using their hands freely.

In this work, we allow a single-viewpoint sensor to proactively search for a better sensing position according to the user's behavior. Essentially, our low-cost single viewpoint solution shares the advantages an expensive multiple-viewpoint solution.

SYSTEM OVERVIEW

We describe our proactive sensing platform and the underlying algorithm used to automatically control the sensor's movement. We demonstrate that our proactive sensing system can improve hand pose estimation performance of a state-of-the-art Leap Motion sensor [11]. As a proof of concept, we built a sensor platform with one degree of freedom.

Swing Arm

We used LEGO bricks and a MINDSTORMS EV3 Intelligent Brick to construct a swing arm, adding one degree of freedom to the sensor (between $-90°$ to $90°$ with top speed of $25°$ per second, shown in Fig. 1). An inertial measurement unit (IMU) was attached to the sensor for measuring the position of the swing arm. The EV3 Intelligent Brick was used as an interface to drive the motor and read the data from the attached IMU.

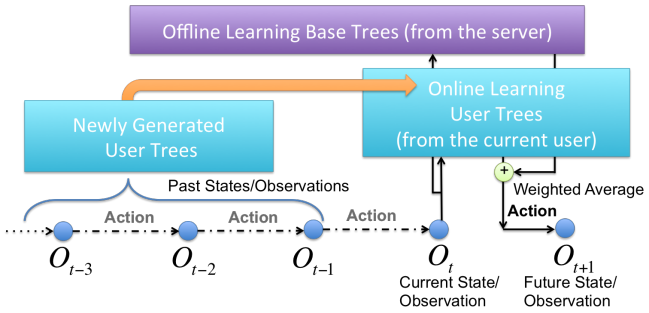


Figure 2. Online action learning and prediction.

When the sensor moved, the position of the hand in the sensor coordinate system changes. We converted from sensor coordinates to world coordinates. We used the IMU mounted on the swing arm to calculate the initial transformation between sensor and world coordinates. To mitigate sensor error and drift, we updated the transformation using Iterative Closest Point (ICP) to align the current hand pose to the initial hand pose. We found the IMU and ICP worked well in our case, though more sophisticated methods such as those of those of Newcombe *et al.* [13] could be applied.

Action Learning

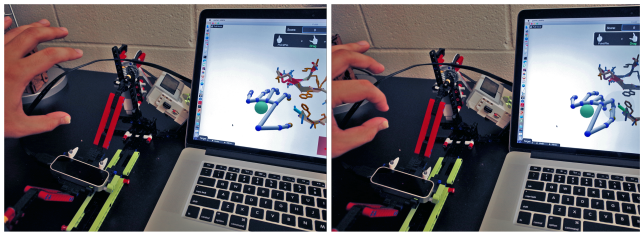
Our proactive sensing system chooses an action in order to improve its performance in estimating a user's hand pose. We take a learning approach to learn a function $f : \mathbb{O} \rightarrow \mathbb{A}$, where $\mathbb{O}$ is the set of possible observations and $\mathbb{A}$ is the set of possible actions. Our system learns to improve estimation *confidence*. Confidence is provided by Leap Motion at each time-stamp, as a single score from 0 (worst) to 1 (best) indicating confidence in the current pose estimate. We chose confidence as it does not require ground truth and is highly correlated with accuracy (discussed below).

Approximate kNN Classification

k-Nearest Neighbor (kNN) is a widely used algorithm with desirable properties. Firstly, kNN can be significantly sped up during testing by incorporating Approximate Nearest Neighbor (ANN) search [12]. Secondly, training a KD-tree for ANN search is also efficient. We train multiple KD-trees in an online fashion utilizing data collected on individual behavior.

We take a supervised approach to train the action function f. Our training data includes:

- Observations $\mathbb{O}$ containing features from the hand pose estimation system. We start with 84 features: hand type (left/right), swing arm angle, confidence, palm position (3), velocity (3), and orientation (3), and positions of the pose (24×3). This is reduced to 10 features as described below.
- The best action $a \in \mathbb{A}$ where $\mathbb{A} = \{left, right, still\}$.



(a) Open (b) Closed

Figure 3. Task overview. Subjects can modify the structure freely using any hand poses they want to pinch.

Offline Training

We first generated training data in an offline stage and trained a number of KD-trees, called *base trees*, to predict actions for all users when first using our system.

The training data set was built by sampling 12 sets of gestures, both static (7 different hand postures from fully open to fully closed) and dynamic (grab in the air using 5 different hand postures). Each gesture was performed while the swing arm swung through its whole range 10-20 times.

To precompute the best action for each training entry, first we used kNN to find neighbors of the entry (within 10 swing degrees). Each neighbor voted for the action toward higher weighted confidence (weighted by confidence and distance). At runtime, if the current confidence is above 0.93, the action *still* is used; otherwise, the same kNN lookup is used, using the pre-computed actions for each neighbor, to compute the action taken. For classification, we search among all available trees to get the three closest training samples for each tree, and accumulate the votes for each action as the prediction score.

Online Training

Since the confidence of the hand pose estimation is continuously observed, we propose to collect training data while each user interacts with the system, and train a set of new KD-trees, called *user trees*, in an online fashion (every 2,000 samples). The scores of the predicted actions from both base and user trees are fused by weighted averaging of the scores into *combined trees*.

The block diagram showing the online training procedure and action prediction fusion is shown in Fig. 2.

Dimensionality Reduction for Speed-up

By collecting training data at 60 fps and each observation having 84 dimensions, our data and KD-trees took considerable amounts of memory (10 MB per 2,000 samples). We used dimensionality reduction techniques to reduce the feature dimensions from 84 to 10. First, we used feature selection techniques to select the top few features (i.e., $1 \sim 4$ features). We simply exhaustively tested different combinations using cross validation and found the three best features, which were the hand type (left/right), sensor angle, and the tracking confidence from the sensor. For the remaining 81 features, we applied PCA to reduce dimensions to 7, such that the final total dimensions were $3+7=10$. After dimension reduction, it took 10KB per 2,000 samples, and the precomputation time was reduced from about 30 minutes to around 20 seconds for a small dataset (~6.7MB). On a 2.3 GHz Intel Core i7 laptop, a new classification tree can usually be trained in 30 seconds, and the query among 500 trees can be done faster than 30 Hz.

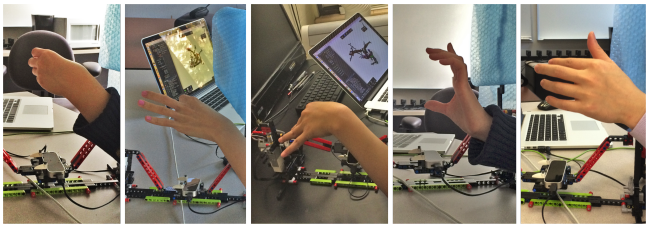


Figure 4. Pinch gesture variations among test subjects.

USER STUDY

We carried out a user study to examine the effect of different types of sensing on pose estimation confidence. We conducted a within-subjects experiment with 17 subjects recruited randomly from the college, most of whom were novices at using the Leap Motion sensor.

We used three different *sensing types* to utilize the degrees of freedom of the swing arm:

- `PROACTIVE`: The swing arm used our proposed approach to move the sensor, selecting the best action given current the observation.
- `STATIC` (baseline 1): The swing arm was always static, which is the most common sensing type adopted in many state-of-the-art systems (e.g., [16, 11]).
- `RANDOM` (baseline 2): The swing arm moved to a random angle from time to time. This was to confirm that moving the sensor meaningfully is important (rather than randomly moving the sensor, which might occasionally avoid occlusion to achieve better confidence).

The task given to subjects was to use a pinch gesture to do 3D object manipulation (Fig. 3). Subjects manipulated a highly deformable 3D object's shape by dragging its parts to different locations. Subjects were asked to continuously manipulate the 3D structure with no provided goal. Note that any pinch gesture which could be recognized was allowed. Hence, we observed a large variation of pinch gestures (Fig. 4). All subjects were asked to use their left hand in the entire experiment. In order to eliminate users' attention from the swing arm, we blocked the subjects from seeing their acting hand and the sensor. We also had the subjects listen to white noise to ensure they could not hear the sound from the motor.

For each subject, the experiment consisted of three *tree types*: base, user, and combined. Within each tree type, there were three *task iterations*. Each task iteration took 2 minutes. During each task iteration, we switched between the three *sensing types* without any interruption (i.e., 40 seconds per sensing type); the swing arm rapidly reset to 0° between each sensing type. We counterbalanced for any learning effects. We randomized the tree type order; however, if user or combined trees were to be first, subjects did an additional base trees first (to get user data to construct the other tree types), and we

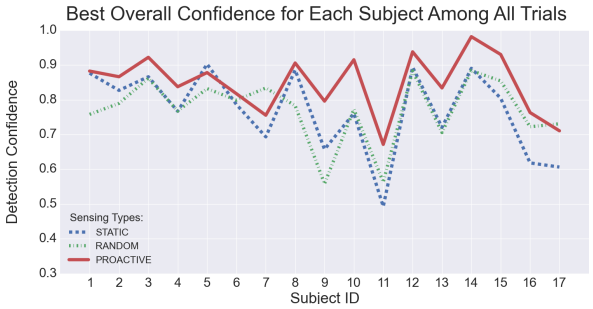


Figure 5. Best overall detection confidence for each subject among all task iterations.

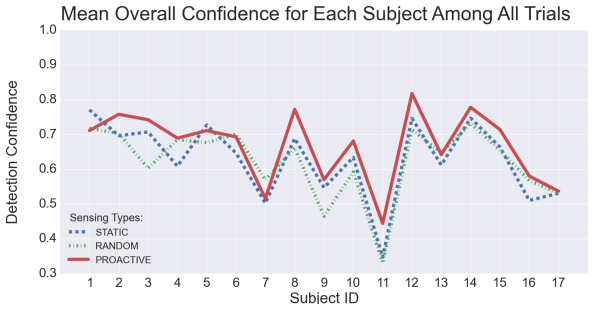


Figure 6. Mean overall detection confidence for each subject among all task iterations.

compared the better of the two base trees uses. We randomized the sensing type order in each task iteration for each subject. At the beginning of each tree type, subjects were given up to 5 minutes to practice the task to make sure they understood it fully and were capable of completing it, although most subjects did not use the full time.

Aggregate Confidence

We examined two outcomes of aggregate confidence by subject. We took the overall (mean) confidence, collected at 60 Hz, within each 40 second use of a sensing type. Then, for each sensing type, we took the *mean* across all uses as the *mean overall confidence* and the *best* across all uses as the *best overall confidence*. Since our outcome variables were not all normally distributed, we used the non-parametric Friedman Test for repeated measures, along with subsequent Wilcoxon Rank Sum Tests and the Bonferroni correction, to identify statistically significant 2-way comparisons.

Sensing Types

For best overall confidence, there was a statistically significant difference, $\chi^2(2,N=17)=17.294$, $p<.001$, among the three sensing types. Post-hoc tests further found that there was no statistically significant difference between `STATIC` (M = .767, SD = .121) and `RANDOM` (M = .770, SD= .095), Z=-.118, p=.906. There was a statistically significant difference between `STATIC` and `PROACTIVE` (M = .847, SD = .085), Z=-3.479, $p < .001$. `RANDOM` was also statistically significantly different from `PROACTIVE`, Z=-3.053, $p = .002$. Fig. 5 shows comparison between three sensing types across all subjects.

For mean overall confidence, we also found a statistically significant difference among the three sensing types, $\chi^2(2, N = 17) = 10.706, p = .005$. Post-hoc tests revealed that although there was no statistically significant difference between `STATIC` (M = .629, SD = .112) and `RANDOM` (M = .621, SD= .105), Z=-.544, p=.586, there was a statistically significant difference between `STATIC` and `PROACTIVE` (M = .668, SD = .104), Z=-2.959, $p = .003$, and `RANDOM` was statistically significantly different from `PROACTIVE`, Z=-2.675, $p = .007$. Fig. 6 shows comparison between three sensing types across all subjects.

Tree Types

During our experiment, we captured confidence of the different tree types and averaged across all subjects. User trees obtained a 2.7% higher average, and 3.8% higher best, relative confidence; however, results were not statistically significant.

Confidence	Base	User	Combined
Mean overall	0.658	0.676	0.672
Best overall	0.737	0.766	0.763

Confidence and Accuracy Correlation

We assume confidence is highly correlated to accuracy, so that improving confidence improves accuracy. To confirm, we gathered data using a technique similar to existing work that has reported findings of Leap Motion sensor accuracy [6, 22]. We used an artificial hand to perform known poses while the sensor moved. In the global coordinate system, we measured accuracy for each pose and compared to confidence at each time-stamp. We tested six sets of static hand poses (fully open, fully closed, and partially open, each with 0 and 45 degree palm facings). The cross-correlations (with no shifting) between confidence and accuracy for each pose were 0.939, 0.971, 0.955, 0.931, 0.924, and 0.890.

CONCLUSION AND FUTURE WORK

In this work, we demonstrated the potential for proactive sensing to improve sensing performance by actively positioning the sensor. Our core contribution is an effective learning scheme that requires no ground truth hand poses. We also presented an online model update to improve performance for each user.

Currently, our method predicts the best action to take for the current hand pose, while ignoring that the hand pose can change. By acting at a high frame rate, our sensor can still move to a better position to catch up with the movement of the hand. We believe a model explicitly predicting the movement of a hand can further improve the performance of our system. Additionally, our proactive sensing algorithm could be applied to other movable platforms with higher degrees of freedom. For instance, we imagine that a personal drone could serve as the ultimate movable platform. Our algorithm can help drones to learn their best sensing position to understand human behaviors, including hand and body poses.

ACKNOWLEDGMENTS

This work was supported by the MOST grant 104-2220-E-007-016, the Office of Naval Research grant N00014-12-C-0158, the Bill and Melinda Gates Foundation grant OPP1031488, the Hewlett Foundation grant 2012-8161, Adobe, and Microsoft.

REFERENCES

1. Luca Ballan, Aparna Taneja, Jürgen Gall, Luc Van Gool, and Marc Pollefeys. 2012. Motion capture of hands in action using discriminative salient points. In *ECCV*. Springer.
2. Andrea Colaço, Ahmed Kirmani, Hye Soo Yang, Nan-Wei Gong, Chris Schmandt, and Vivek K. Goyal. 2013. Mime: Compact, Low Power 3D Gesture Sensing for Interaction with Head Mounted Displays. In *UIST*.
3. Seth Cooper, Firas Khatib, Adrien Treuille, Janos Barbero, Jeehyung Lee, Michael Beenen, Andrew Leaver-Fay, David Baker, Zoran Popović, and Foldit Players. 2010. Predicting protein structures with a multiplayer online game. *Nature* 466, 7307 (2010), 756–760.
4. Martin de La Gorce, David J. Fleet, and Nikos Paragios. 2011. Model-Based 3D Hand Pose Estimation from Monocular Video. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 33, 9 (Sept 2011), 1793–1805.
5. Ali Erol, George Bebis, Mircea Nicolescu, Richard D. Boyle, and Xander Twombly. 2007. Vision-based Hand Pose Estimation: A Review. *Computer Vision and Image Understanding* 108, 1-2 (Oct. 2007), 52–73.
6. Jože Guna, Grega Jakus, Matevž Pogačnik, Sašo Tomažič, and Jaka Sodnik. 2014. An Analysis of the Precision and Reliability of the Leap Motion Sensor and Its Suitability for Static and Dynamic Tracking. *Sensors* 14, 2 (2014), 3702.
7. Chris Harrison, Hrvoje Benko, and Andrew D. Wilson. 2011. OmniTouch: Wearable Multitouch Interaction Everywhere. In *UIST*.
8. Tony Heap and David Hogg. 1996. Towards 3D hand tracking using a deformable model. In *Automatic Face and Gesture Recognition, 1996., Proceedings of the Second International Conference on*. 140–145.
9. Cem Keskin, Furkan Kıraç, Yunus Emre Kara, and Lale Akarun. 2012. Hand Pose Estimation and Hand Shape Classification Using Multi-layered Randomized Decision Forests. In *ECCV*.
10. David Kim, Otmar Hilliges, Shahram Izadi, Alex D Butler, Jiawen Chen, Iason Oikonomidis, and Patrick Olivier. 2012. Digits: freehand 3D interactions anywhere using a wrist-worn gloveless sensor. In *Proceedings of the 25th annual ACM symposium on User interface software and technology*. ACM, 167–176.
11. Leap Motion, Inc. 2014. (2014). `http://leapmotion.com/`
12. Marius Muja and David G. Lowe. 2014. Scalable Nearest Neighbor Algorithms for High Dimensional Data. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 36 (2014).
13. Richard A. Newcombe, Dieter Fox, and Steven M. Seitz. 2015. DynamicFusion: Reconstruction and Tracking of Non-Rigid Scenes in Real-Time. *The IEEE Conference on Computer Vision and Pattern Recognition (CVPR)* (June 2015).
14. Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A Argyros. 2011. Efficient model-based 3D tracking of hand articulations using Kinect.. In *BMVC*.
15. Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. 2014. Realtime and robust hand tracking from depth. In *CVPR*.
16. 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. 2015. Accurate, Robust, and Flexible Real-time Hand Tracking. CHI.
17. Jie Song, Fabrizio Pece, Gábor Sörös, Marion Koelle, and Otmar Hilliges. 2015. Joint Estimation of 3D Hand Position and Gestures from Monocular Video for Mobile Interaction. In *CHI*.
18. Jie Song, Gábor Sörös, Fabrizio Pece, Sean Ryan Fanello, Shahram Izadi, Cem Keskin, and Otmar Hilliges. 2014. In-air Gestures Around Unmodified Mobile Devices. In *UIST*.
19. Srinath Sridhar, Franziska Mueller, Antti Oulasvirta, and Christian Theobalt. 2015. Fast and Robust Hand Tracking Using Detection-Guided Optimization. In *CVPR*.
20. Srinath Sridhar, Antti Oulasvirta, and Christian Theobalt. 2013. Interactive Markerless Articulated Hand Motion Tracking Using RGB and Depth Data. In *ICCV*.
21. Yangang Wang, Jianyuan Min, Jianjie Zhang, Yebin Liu, Feng Xu, Qionghai Dai, and Jinxiang Chai. 2013. Video-based hand manipulation capture through composite motion control. *ACM Transactions on Graphics (TOG)* 32, 4 (2013), 43.
22. Frank Weichert, Daniel Bachmann, Bartholomäus Rudak, and Denis Fisseler. 2013. Analysis of the Accuracy and Robustness of the Leap Motion Controller. *Sensors (Basel, Switzerland)* 13, 5 (05 2013), 6380–6393.
23. Ying Wu, John Y Lin, and Thomas S Huang. 2001. Capturing natural hand articulation. In *ICCV*, Vol. 2.
24. Wenping Zhao, Jinxiang Chai, and Ying-Qing Xu. 2012. Combining marker-based mocap and RGB-D camera for acquiring high-fidelity hand motion data. In *Proceedings of the ACM SIGGRAPH/eurographics symposium on computer animation*. Eurographics Association, 33–42.