3D Tracking of Human Hands in Interaction with Unknown Objects

Paschalis Panteleris[1]
padeler@ics.forth.gr
Nikolaos Kyriazis[1]
kyriazis@ics.forth.gr
Antonis A. Argyros[21]
argyros@ics.forth.gr

[1] Insitute of Computer Science, FORTH,
N. Plastira 100, Vassilika Vouton,
GR70013, Heraklion, Crete, Greece

[2] Computer Science Department,
University of Crete,
Heraklion, Crete, Greece

Abstract

The analysis and the understanding of object manipulation scenarios based on computer vision techniques can be greatly facilitated if we can gain access to the full articulation of the manipulating hands and the 3D pose of the manipulated objects. Currently, there exist methods for tracking hands in interaction with objects whose 3D models are known. There are also methods that can reconstruct 3D models of objects that are partially observable in each frame of a sequence. However, to the best of our knowledge, no method can track hands in interaction with unknown objects. In this paper we propose such a method. Experimental results show that hand tracking can be achieved with an accuracy that is comparable to the one obtained by methods that assume knowledge of the object models. Additionally, as a by-product, the proposed method delivers accurate 3D models of the manipulated objects.

1 Introduction

Tracking the articulation of hand(s) in interaction with objects is an interesting and challenging computer vision problem. Existing approaches require accurate 3D models of the manipulated object and/or rely on strong assumptions regarding the expected observations. Creating the required 3D models can be a difficult and time consuming process as it often involves specialized equipment and accurate calibration. The recent release of several cheap RGB-D sensors inspired many researchers to develop methods that rely on these cameras to map the environment and track human actions.

In this paper we propose a novel approach that can track human hands in interaction with unknown objects, i.e., objects for which no a priori 3D model is given. As illustrated in Fig.1, the input to the method is a sequence of RGBD frames showing the interaction of one or two hands with an unknown object. Starting with the raw depth map (left) we perform a pre-processing step and compute the scene point cloud. We employ an appropriately modified model based hand tracker and temporal information to track the hand 3D positions and posture (middle bottom). In this process, a progressively built object model is also taken into account to cope with hand-object occlusions. We use the estimated fingertip positions

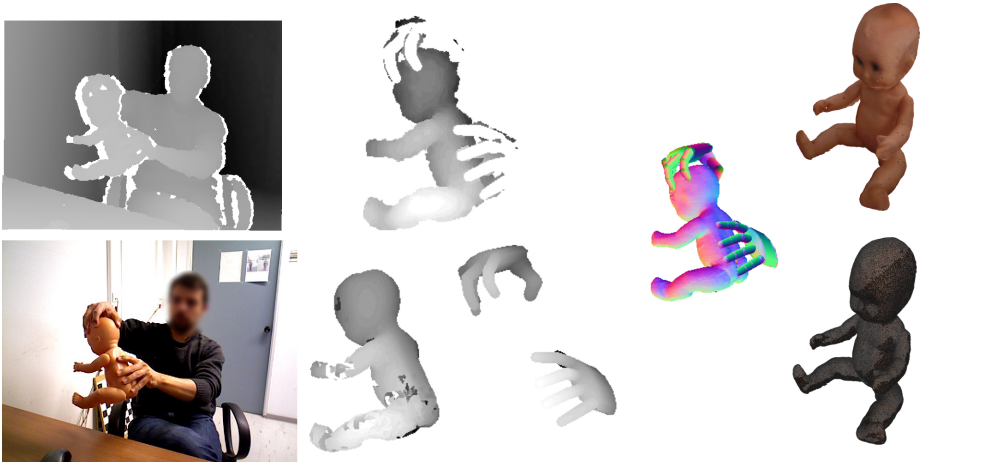

Figure 1: Method overview. Left: input depth and color frames. Middle: Object depth segmented using the fingertip 3D positions. Partially scanned object model and hand models. Right: 3D Rendering of the scene and final scanned model.

of the hand to segment the manipulated object from the rest of the scene (middle top). The segmented object points are used to update the object position and orientation in the current frame and are integrated into the object 3D representation (right). At the end, the 3D model of the object is reconstructed, provided that every part of the object was observed at at least one frame of the sequence. Besides the accurate tracking of the hands, the proposed method provides an accurate 3D model of the object in the form of texture-mapped 3D mesh.

The main contributions of the proposed method are two. First, we propose the first model-based 3D hand tracking approach that can track the interaction of hands with unknown objects. Second, by accounting for the hands in the scene, we enable the 3D reconstruction and tracking of the originally unknown object. Quantitative and qualitative experiments show that both 3D hand tracking and object 3D model acquisition can be performed at an accuracy that is comparable to that achieved under much more restrictive assumptions.

2 Related work

Several approaches have been proposed that track articulated objects like the human hands. Furthermore, over the recent years a lot of research has been carried out with respect to object 3D scanning and reconstruction. In this section we focus on methods that try to tackle the problem of hand-object manipulation and in-hand reconstruction.

Hand-Object tracking: Hand tracking methods can be classified into model-based (e.g., [6, 13, 15]) data-driven (e.g., [8, 23]) or hybrid (e.g., [1, 19, 22, 25]). Model based approaches solve an optimization problem whose goal is to come up with the parameters of a 3D hand model that best explains the set of available 3D observations. Data-driven approaches explore the mapping of visual observations to hand poses by employing learning techniques. Model based approaches are typically more accurate, easily generalizable to different scenarios and provide anatomically and physically plausible solutions at the cost

of being computationally demanding. Their high computational requirements are typically handled by exploiting the inherent parallelism of computations in GPGPU architectures. Data-driven approaches require training examples from the very high dimensional space of hand articulations. In that sense, they are less accurate and less easily generalizable compared to model-based approaches. Their solutions are synthesized in a bottom-up fashion, thus they may lack anatomical validity and physical plausibility. On the other hand, despite the computational requirements of the training/learning process, they are very efficient at run time. Finally, hybrid approaches have both data-driven and top-down components, in an effort to combine the best of both worlds.

Especially for the problem of tracking hand-object interactions, the model-based paradigm appears to be preferable. The human hand is modelled as an articulated object of 26 degrees of freedom (DoFs). At their offline learning stage, data-driven approaches need to sample this high dimensional space quite densely, which is already a very demanding task. As soon as we consider hands in interaction with objects, training would require to learn the appearance of a hand in interaction with any possible object, which, at least theoretically, is a task on unmanageable complexity. Hybrid methods contain data-driven components that require learning, so they also share this shortcoming. For this reason, for tracking hand(s) in interaction with unknown objects, we restrict our interest to model based approaches.

Oikonomidis et al [17], used a model based approach and multicamera input to track a single hand interacting with an object that was modelled as a parametric 3D shape (e.g., parallelepiped, cylinder, ellipsoid, etc). The actual parameters of the 3D shape were not a priori known but were estimated together with hand articulation. While the approach is inspiring, the assumption that the object can be represented as a parametric model is valid for only a small subset of interesting objects. Krainin et al [9] implemented a system that scans an object handled by a robotic arm. The method exploits RGB-D input fused with proprioceptive sensory data to track both objects. Our work is using only depth information thus allowing tracking of non-instrumented manipulators. Recently, Kyriazis et al [10, 11] demonstrated model based methods that accurately track human hands interacting with objects. The method requires the initialization of the scene with the exact models of all the manipulated objects.

In-hand reconstruction: Rusinkiewicz et al [21] presented an early work on the 3D reconstruction of in-hand objects using a structured light sensor. They focused on 3D object scanning and masked-out the hand from the observations. All occlusions were treated as missing information. The employed volumetric approach for object model integration was based on previous work by Curless and Levoy [4]. Similarly, Weise et al [26] implemented a 3D scanner using ICP [2] to perform the registration of the observed depth maps, and a surfel-based representation for the reconstructed 3D model. The model was deformed during integration to account for misalignments. A ToF camera was used by Cui et al [3] to scan 3D objects by applying super-resolution and global registration using ICP. Ren et al [20] used both RGB and depth to achieve in-hand object scanning. Their approach used a bag-of-pixels representation and back projection to perform tracking and a space carving approach to integrate the measurements into an object model. Newcombe et al [12] proposed a large scale mapping and tracking pipeline based on ICP and volumetric reconstruction. While this work was intended for room sized scenes, it was demonstrated to perform well even with smaller scale “human sized" objects. In our work, we extend the ideas of Kinect Fusion to work with hand-held small objects by accounting for the hand object occlusions.

In all these methods any hand-object interaction is ignored or accounted for as noise. Even more importantly, no information is provided for the articulation of the hand(s) that

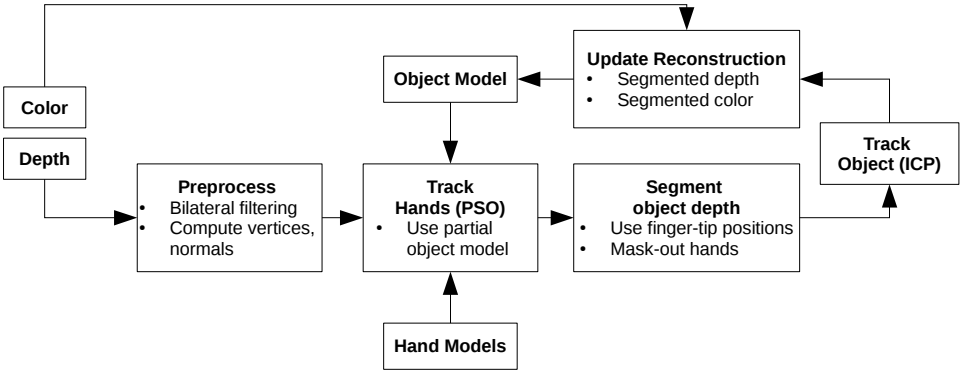

Figure 2: Work flow of the proposed method. The basic pipeline employs depth information, only. Color is used solely for producing textured versions of the computed object 3D model.

manipulate the object. In this work we model and track the interactions of the hand(s) with the object while, simultaneously, we improve both the 3D hand tracking and reconstruction results. The experimental results demonstrate that both the 3D hand tracking and the 3D scans produced as a by-product of our method are of comparable accuracy and quality to that of the current state of the art that, nevertheless, is based in considerably more limiting assumptions.

3 Tracking Hands Interacting with Unknown Objects

The work flow of the proposed approach consists of five main components linked together as shown in Fig. 2. The proposed method accepts RGBD input provided by a Kinect-like sensor. At a first, preprocessing stage, the raw depth information from the sensor is prepared to enter the pipeline. A point cloud is computed along with the normals for each vertex. Then, the user's hands are tracked in the scene. An articulated model for the left and right hands, with 26 degrees of freedom each, is fit to the pre-processed depth input. The current, possibly incomplete (or even empty, for the first frame) object model is incorporated to hand tracking to assist in handling hand/object occlusions. Using the computed 3D location of the user's hands as well as the last position of the (possibly incomplete) object model, the region of the object is segmented in the input depth map. The hands are masked-out from the observation, by comparing it to the rendered hand models. Object tracking is achieved using a mutli-scale ICP. The segmented object depth is used for a coarse to fine alignment with the (partially reconstructed) object model. Finally, the segmented and aligned depth data of the object with the current, partial 3D model are merged. The object's 3D model is maintained in a voxel grid with a Truncated Signed Distance Function (TSDF) [14] representation.

In all the above, we assume that a hand consists of a collection of 22 spheres and 15 cylinders that are appropriately transformed and coupled to represent its appearance and kinematic structure. The resulting model has 26 DoFs. At each frame we extract the fingertip 3D positions X_f^i with respect to the camera center. The unknown object is assumed to be rigid. Its position and orientation with respect to the camera for a frame at time k is expressed

with a rigid body transformation matrix:

$$T_k^M = \begin{bmatrix} R_k^M & t_k^M \\ 0^\top & 1 \end{bmatrix} \in \mathbb{SE}_3, \tag{1}$$

where $\mathbb{SE}_3 := \{R,t|R \in \mathbb{SO}_3, t \in \mathbb{R}^3\}$. The transformation in Eq.(1) maps points in the camera coordinate space to the object's coordinate space at frame k.

At any given frame k, we maintain the partially reconstructed triangle mesh M_k of the object along with the normals for each vertex. Thus, $M_k = \{V_k^M, N_k^M, F_k^M\}$, where V_k^M is the set of mesh vertices in homogeneous coordinates, N_k^M is the set of normals and F_k^M is the set of faces of the triangle mesh.

The hand tracking method does not assume a static camera. However, we assume that the intrinsic parameters of the camera do not change during tracking. The camera projection matrix P maps 3D homogeneous coordinates $X = \{X,Y,Z,1\}$ to a 2D point p on the image plane, i.e., $p = PX$.

In the following sections, we provide more details for each of the building blocks of the proposed approach.

Preprocessing: For each depth frame R_k we first perform bilateral filtering [24] in order to reduce noise while preserving depth discontinuities [14]. The new depth map D_k is used to compute the vertex map V_k. To compute the normal map N_k efficiently, we exploit the fact that points that are neighbours in 3D will project to neighbours in the input depth map.

In a typical hand-object interaction sequence, objects are expected to initially rest on a flat surface (i.e., a table). We detect the supporting plane by employing plane fitting through RANSAC [8]. This results in a new vertex and normals map (V_k' and N_k', respectively) for just the points that are above the supporting plane, as well as the corresponding segmented depth map D_k'. If the camera is static this plane detection step needs to be performed only once, in the first frame of the sequence.

Hand Tracking: Hand tracking is based on a variant of the work of Oikonomidis et al [16]. Instead of relying on skin color for hand segmentation, we consider a 3D volume around the 3D position of the hand in the previous frame. In our implementation this volume was empirically set to be $15cm$ around the palm center. By using only depth information, hand tracking is not affected by illumination changes. Additionally, no restrictions exist on the color of the tracked objects (for example, in [17] objects cannot have skin-colored parts).

In order to account for the presence of multiple interacting objects in the scene (i.e., two hands and one object), we formulate our objective function similarly to [11]. In each frame k, we generate hypotheses about the hand configurations H_k and test them against the observations V_k'. We extend the objective function to ignore the error generated by vertices in the observation that belong to the object and not to the hands. To do so, we employ the partially reconstructed model of the object from frame $k-1$, M_{k-1} and its last known orientation and position with respect to the camera T_{k-1}^M.

The object information from frame $k-1$ is incomplete, since it accounts neither for the motion in the current frame nor for the appearance of previously unobserved object parts. Despite that, we experimentally show it is enough to allow for the accurate tracking of the hands and for high quality object reconstruction.

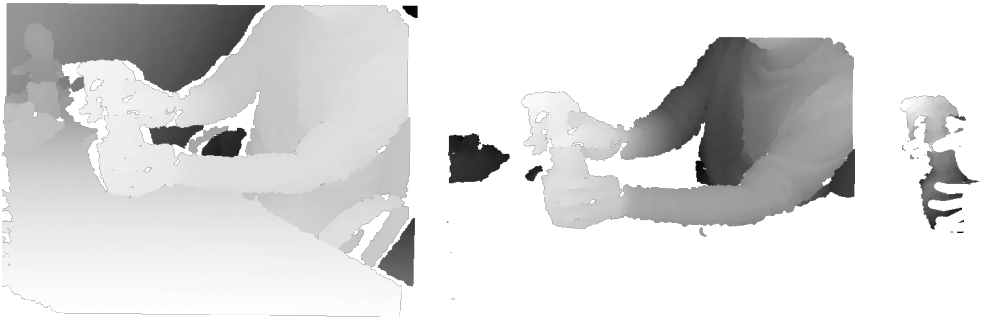

Figure 3: Object segmentation. Left to right: the raw depth map; removal of the supporting plane; results of flood-fill around finger tips and masking out the hands.

Object Segmentation: We use the centers of the spheres located in the fingertips of the hand models as an estimation of the 3D positions for the user's fingertips. Each fingertip position $X_f^i = (X,Y,Z,1)^\top$ is re-projected using the camera projection matrix $x_f^i = PX_f^i$. Since at least some of the finger tips are going to be in touch with the object, we use x_f as seeds to segment the hand and the object in D'_k. This is achieved with a connected components based segmentation that connects together all points of D'_k that are (recursively) connected to the estimated fingertips. It should be noted that because of masking out the supporting plane, D'_k does not include points of that plane.

The final step in object segmentation is to remove from D'_k points that correspond to hands. To do so, we use the hand pose estimated by the hand tracker and we render a synthetic depth map, D_h, of the user's hands. The object points D'_o are then chosen by masking out the points corresponding to the hand from the already segmented hand-object depth map D'_k. Using D'_o, we segment the vertex and normal maps V'_k and N'_k to obtain V'_o and N'_o. Figure 3 illustrates the steps of object segmentation starting from the raw depth.

Object Tracking: We perform object tracking using multi-scale ICP. The approach is detailed in Newcombe et al [14] and was first demonstrated in a 3D modeling system by Rusinkiewicz et al [21]. Our approach employs three layers that perform a coarse to fine object pose estimation. The pyramid layers are computed starting from the segmented input $S_k^1 = \{D'_k, V'_k, N'_k\}$ as the bottom layer and by sub-sampling by block averaging to obtain the next layers S_k^2 and S_k^3. Similarly, the depth, vertex and normal components of the model M_{k-1} are prepared, using the last known orientation and position with respect to the camera T_{k-1}^M.

The registration step results in a transformation T_f^M that maps the model from frame $k-1$ to k. The global transformation from the object coordinate space to the camera space is given by $T_k^M = T_f^M T_{k-1}^M$.

Updating the Object Model: Volumetric integration of range images using a signed distance function (SDF) was first introduced by Curless and Levoy [4]. The method has been used in many implementations that employ active depth sensors both for small [21, 26] and large [7] scale reconstruction. In the proposed method we used the truncated signed distance function (TSDF) explained in detail in [14] to perform the integration of the segmented object parts into a 3D volume representing the manipulated object.

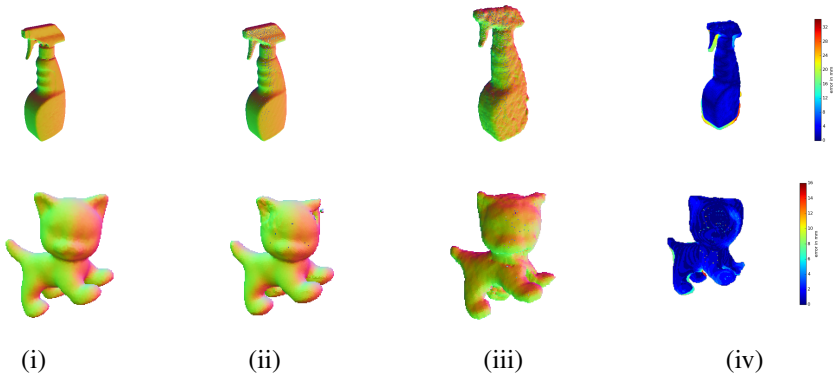

Figure 4: Object reconstruction accuracy of the spray (top) and cat (bottom) sequences. (i) The actual object models. (ii) The 3D scans produced from the synthetic sequences. (iii) The 3D scans produced from the real sequences. (iv) The error plot comparing the real model (i) with the 3D scan (iii).

While the TSDF volume can cope with the noise and possible inaccuracies in the integration data, it is sensitive to cases where parts of the volume are occluded or missing for a large number of frames. This might happen in our scenario when parts of the object that were visible at some frame, become invisible for a large number of frames later on, due to hand-object occlusions. To account for these cases we use our knowledge of the synthetic hands depth map D_h (see Sec. 3) to exclude these voxels of the TSDF volume from being updated with invalid information.

At the end of the integration process, ray-casting [18] is used to extract the model point cloud. The normals are computed using the nearest neighbours of each point and triangulated using greedy projection triangulation [17]. The new object model M_k is used in the next frame, to account for the object observations in the hand tracking step as explained in Sec. 3.

Initialization: We initialize the pipeline with no information about the unknown rigid object. In order to bootstrap the process the user places his hands at pre-set initialization positions and the hand tracking module initializes. At each frame k the method computes the location of the hand parts in 3D space and segments the depth map around the fingers as explained in Sec. 3. Since the foreground of the scene may contain any number of objects, we initialize the reconstruction process only when the number of points in the segmented object depth map D'_o becomes greater that a preset, empirically defined threshold.

4 Experimental Evaluation

The proposed method was tested quantitatively and qualitatively in sequences where a person manipulates objects of different sizes, with either one or two hands. The experiments demonstrate that the hand tracking accuracy is practically identical to the one obtained when the object model is known and fed into the tracking process. Moreover, the comparison of the reconstructed object models to the actual ones shows only minor 3D reconstruction errors.

Experiment	Proposed mean/median error	[□], GT model mean/median error	[□], Scanned model mean/median error
Single hand, cat	0.42 / 0.39	0.47 / 0.43	0.45 / 0.43
Single hand, spray	0.65 / 0.63	0.70 / 0.53	0.63 / 0.47
Two hands, cat	0.38 / 0.34	0.33 / 0.31	0.44 / 0.39
Two hands, spray	0.59 / 0.44	0.51 / 0.38	0.62 / 0.41

Table 1: Hand tracking accuracy (in cm) measured on the synthetic datasets. The accuracy of the method is close to that of [□], although the latter assumes that the object model is known a priori.

4.1 Quantitative Experiments

For the quantitative experiments, we tracked sequences using objects with known 3D models and our implementation of [□]. The objects used are shown on the first column of Fig. 4. The spray bottle model was acquired using a laser scanner, while the cat toy was 3D printed from a CAD model. Both models have sub-millimetre accuracy. The dimensions of the objects are approximately 8×5×25cm and 11×7×10cm (L×W×H), respectively.

For each object we created and tracked two sequences, one in which the object was manipulated with a single hand and another with two-hands. Subsequently, synthetic sequences were rendered (depth frames) using the tracking information and the hand and object models. These synthetic sequences were used as ground-truth-annotated input. Each synthetic sequence was fed to the proposed method that estimated the hand(s) articulation and the 3D model of the object which was not known to it. The same input was also fed to the method presented in [□] which was aware of the exact object models. We compare the proposed approach to [□] with respect to hand tracking accuracy. Additionally, we compute the 3D object reconstruction error resulting from our method.

Hand tracking accuracy: For each frame of a sequence, we measure hand tracking accuracy by averaging the distances of the estimated hand joints from their ground truth positions. Table 1 shows the mean and median hand tracking error over the whole sequence. The first column shows that for the proposed method (object model is not known). The second shows that for [□] (perfectly accurate object model - ground truth - a priori known). The obtained results demonstrate that our approach results in hand tracking accuracy that is comparable to that of [□], although our method is not aware of the object model.

The third column of Table 1 shows that the accuracy of [□] is basically unaffected if it is fed with the object model that our method computed. This is an indirect indication of the quality of the 3D model provided by our method.

Interestingly, [□] fails soon and completely in cases where the object model is not available. This is shown in Fig. 5. The yellow plot shows the hand tracking error of [□] in the case of the synthetic dataset (spray bottle with two hands) and assuming two hands and no object. As it can be verified, the hands are soon completely lost. The same method performs very accurately when a precise object of the model is known (green plot). The proposed approach has comparable performance (red plot), although it is not aware of the object manipulated by the two hands.

Object reconstruction accuracy: The second column of Fig. 4 illustrates the 3D models that were reconstructed by the proposed method when operating on the synthetic datasets (single hand experiment). The synthetic datasets do not suffer from sensor noise, so these

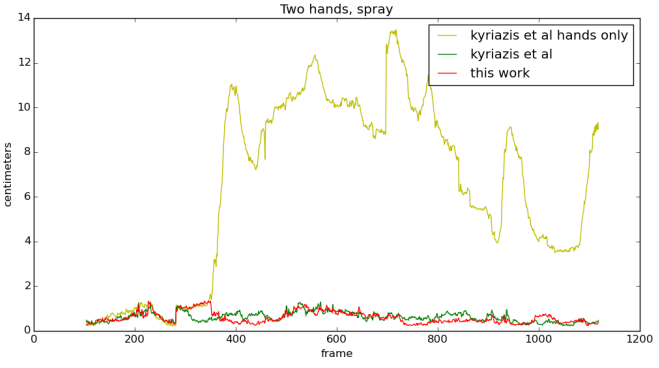

Figure 5: Model based methods that do not account for the object in the scene will fail and will not recover. Our method performs similarly with the full model based approach even though the object is initially unknown.

models are very accurate. The third column shows the models acquired from the real sequences. Finally, the fourth column visualizes the error plot in millimeters between columns (i) and (iii). The models produced using the volumetric reconstruction are thicker than the real objects and this is evident on the edges of the plots. On the main body of the object the error is in the order of 5 mm. Furthermore, while the trajectories of the scanned and the ground truth models are not directly comparable, the quality of the reconstruction is a measure of the quality of the object tracking.

4.2 Qualitative experiments

We performed extensive tests with different objects of unknown 3D models. The minimum object size for our method is limited mainly by the accuracy of the sensor. In the case of the Kinect and the Xtion RGBD cameras the minimum size was found to be close to the toy cat presented in Sec. 4.1. Figure 6 illustrates sample results of these tests. Our method performed well in manipulating objects such as a mask (first row) and in complex actions such as preparing pancakes (second row) and handing an object from one hand to the other (third row). A video showing qualitative experimental results is available at `https://youtu.be/9r43PtJ0Fwg`.

5 Conclusions

We proposed a method that enables the full 3D tracking of complex object manipulation activities without the need of prior scene knowledge. This has been achieved by modifying and integrating state of the art hand tracking and object modelling techniques. Experimental results demonstrated that hand tracking accuracy is, essentially, not compromised by the lack of the knowledge of the object models. Additionally, the proposed method delivers 3D object models that are fairly accurate. While real time performance was not the goal of this work, tracking a single hand in interaction with an object can be achieved at a frame rate of 10 fps

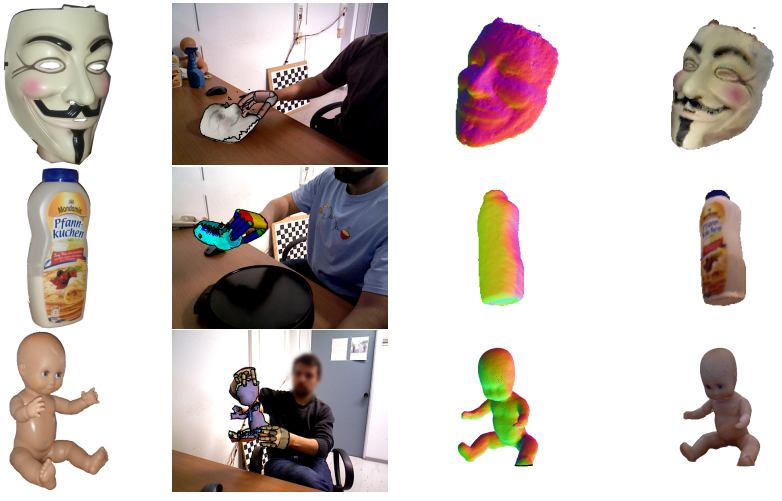

Figure 6: Qualitative results, multiple objects of various sizes reconstructed while being manipulated by a single or two hands. (i) Image of the object. (ii) A frame from the tracking sequence. Hand models and partial object models are superimposed on the image. (iii) 3D scan normals. (iv) Views of textured versions of the acquired 3D models.

(python and C++ implementation on an i7 processor equipped with an NVIDIA GTX970 GPU). This performance drops at $\sim$ 4fps in the case of tracking bimanual manipulation scenarios. Further optimizations on our implementation are expected to result in real time performance.

Acknowledgements

This work was partially supported by the EU FP7-ICT-288533 project ROBOHOW.COG.

References

[1] Luca Ballan, Aparna Taneja, Jurgen Gall, Luc Van Gool, and Marc Pollefeys. Motion capture of hands in action using discriminative salient points. In *Computer Vision âĂŞ ECCV 2012*. Springer Berlin Heidelberg, 2012.

[2] Paul J Besl and Neil D McKay. Method for registration of 3-d shapes. In *Robotics-DL tentative*, pages 586–606. International Society for Optics and Photonics, 1992.

[3] Yan Cui, Sebastian Schuon, Derek Chan, Sebastian Thrun, and Christian Theobalt. 3d shape scanning with a time-of-flight camera. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1173–1180. IEEE, 2010.

[4] Brian Curless and Marc Levoy. A volumetric method for building complex models from

range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996.

[5] Martin de La Gorce, David J. Fleet, and Nikos Paragios. Model-based 3d hand pose estimation from monocular video. 33(9):1793–1805, 2011.

[6] Martin A Fischler and Robert C Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 24(6):381–395, 1981.

[7] Shahram Izadi, David Kim, Otmar Hilliges, David Molyneaux, Richard Newcombe, Pushmeet Kohli, Jamie Shotton, Steve Hodges, Dustin Freeman, Andrew Davison, et al. Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera. In *Proceedings of the 24th annual ACM symposium on User interface software and technology*, pages 559–568. ACM, 2011.

[8] Cem Keskin, Furkan Kiraç, Yunus Emre Kara, and Lale Akarun. Hand pose estimation and hand shape classification using multi-layered randomized decision forests. 2012.

[9] Michael Krainin, Peter Henry, Xiaofeng Ren, and Dieter Fox. Manipulator and object tracking for in-hand 3d object modeling. *The International Journal of Robotics Research*, 30(11):1311–1327, 2011.

[10] Nikolaos Kyriazis and Antonis Argyros. Physically plausible 3d scene tracking: The single actor hypothesis. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 9–16. IEEE, 2013.

[11] Nikolaos Kyriazis and Antonis Argyros. Scalable 3d tracking of multiple interacting objects. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3430–3437, 2014.

[12] Zoltan Csaba Marton, Radu Bogdan Rusu, and Michael Beetz. On Fast Surface Reconstruction Methods for Large and Noisy Datasets. In *Proceedings of the IEEE International Conference on Robotics and Automation (ICRA)*, Kobe, Japan, May 12-17 2009.

[13] Stan Melax, Leonid Keselman, and Sterling Orsten. Dynamics based 3d skeletal hand tracking. In *Proc. Graphics Interface*, 2013.

[14] Richard A Newcombe, Shahram Izadi, Otmar Hilliges, David Molyneaux, David Kim, Andrew J Davison, Pushmeet Kohi, Jamie Shotton, Steve Hodges, and Andrew Fitzgibbon. Kinectfusion: Real-time dense surface mapping and tracking. In *Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on*, pages 127–136. IEEE, 2011.

[15] 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, June 2014. doi: 10.1109/CVPR.2014.437.

[16] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A. Argyros. Efficient model-based 3d tracking of hand articulations using kinect. In *BMVC*, Dundee, UK, Aug. 2011.

[17] Iason Oikonomidis, Nikolaos Kyriazis, and Antonis A. Argyros. Full dof tracking of a hand interacting with an object by modeling occlusions and physical constraints. In *ICCV*, pages 2088–2095. IEEE, Nov. 2011.

[18] Steven Parker, Peter Shirley, Yarden Livnat, Charles Hansen, and Peter-Pike Sloan. Interactive ray tracing for isosurface rendering. In *Proceedings of the conference on Visualization'98*, pages 233–238. IEEE Computer Society Press, 1998.

[19] Chen Qian, Xiao Sun, Yichen Wei, Xiaoou Tang, and Jian Sun. Realtime and robust hand tracking from depth. 2014.

[20] Carl Yuheng Ren, Victor Prisacariu, David Murray, and Ian Reid. Star3d: simultaneous tracking and reconstruction of 3d objects using rgb-d data. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 1561–1568. IEEE, 2013.

[21] Szymon Rusinkiewicz, Olaf Hall-Holt, and Marc Levoy. Real-time 3d model acquisition. In *ACM Transactions on Graphics (TOG)*, volume 21, pages 438–446. ACM, 2002.

[22] Danhang Tang, Tsz-Ho Yu, and Tae-Kyun Kim. Real-time articulated hand pose estimation using semi-supervised transductive regression forests. 2013.

[23] Danhang Tang, Hyung Jin Chang, A. Tejani, and Tae-Kyun Kim. Latent regression forest: Structured estimation of 3d articulated hand posture. In *2014 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 3786–3793, June 2014. doi: 10.1109/CVPR.2014.490.

[24] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *Computer Vision, 1998. Sixth International Conference on*, pages 839–846. IEEE, 1998.

[25] Jonathan Tompson, Murphy Stein, Yann LeCun, and Ken Perlin. Real-time continuous pose recovery of human hands using convolutional networks. *ACM Transactions on Graphics*, 33(5):169:1–169:10, 2014.

[26] Thibaut Weise, Thomas Wismer, Bastian Leibe, and Luc Van Gool. In-hand scanning with online loop closure. In *Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on*, pages 1630–1637. IEEE, 2009.