Recovering 3D Models of Manipulated Objects through 3D Tracking of Hand-Object Interactions

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

We are interested in the vision-based 3D tracking of scenes where a human manipulates objects. Existing methods that track hand-object manipulations [3] require accurate $3D$ models of the manipulated objects. This is a limiting assumption because the acquisition of such $3D$ models can be a time-consuming process that often involves specialized equipment and accurate calibration. In this work we propose a method that solves the problem when the model of the manipulated object is unknown.

A human observing the motions of two hands manipulating an unknown object acquires rich information about the object itself. Previous techniques on in-hand scanning and recostruction [11, 13] where discarding or ignoring this information. Our work, draws inspiration from [5] and uses the hands as tools that facilitate the reconstruction. It leverages on the observed hand configurations and the induced hand-object occlusions. This enables simultaneous tracking and reconstruction of a previously unseen object. The output is the full ($26D$) articulation and $3D$ position of the hands and object in each frame as well as a textured $3D$ model of the manipulated object.

An overview of the proposed method is illustrated in Fig.1. 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 [7] 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 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).

The work flow of the proposed approach consists of five main components linked together as shown in Fig. 2.

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.

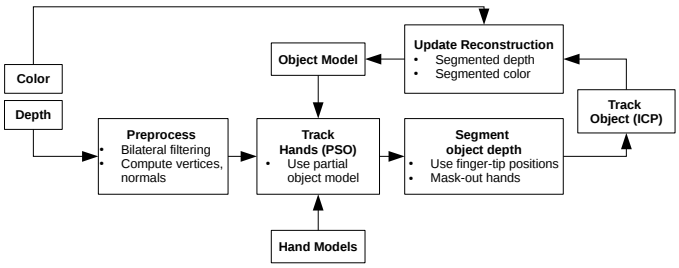

Figure 2: Work flow of the proposed method.

Preprocessing: For each depth frame k we apply bilateral filtering [12] in order to reduce sensor noise while preserving depth discontinuities [6]. 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., on a table). We detect the supporting plane by employing plane fitting through RANSAC. This results in a new vertex and normals map (V'_k and N'_k, respectively) just for 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 [7]. 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 are placed on the color of the tracked objects (for example, in [8] 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 [3]. 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^M_{k-1}.

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 show experimentally that it is enough to allow for the accurate tracking of the hands and for high quality object reconstruction.

Object Segmentation: We use the centers of the spheres located in the fingertips of the employed hand models as an estimation of the 3D positions for the user's fingertips. Each fingertip position $X^i_f = (X,Y,Z,1)^\top$ is re-projected using the camera projection matrix $x^i_f = PX^i_f$. Since at least some of the fingertips 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.

Experiment	Proposed mean/median error	[3], GT model mean/median error	[3], 3D model from proposed method 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 [3], although the latter assumes that the object model is known a priori.

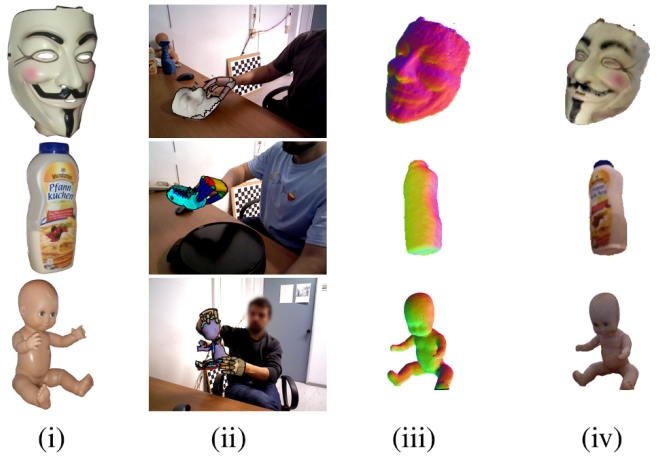

Figure 3: Multiple objects of various sizes reconstructed while being manipulated by a single or two hands. (i) Image of the object. (ii) A frame from the tracking sequence. Hand models and partial object models are superimposed on the image. (iii) 3D scan normals. (iv) Views of textured versions of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

Object Tracking: We perform object tracking using multi-scale ICP. The approach is detailed in Newcombe et al [6] and was first demonstrated in a 3D modeling system by Rusinkiewicz et al [11]. 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 [1]. The method has been used in many implementations that employ active depth sensors both for small [11, 13] and large [2] scale reconstruction. In the proposed method we used the truncated signed distance function (TSDF) explained in detail in [6] to perform the integration of the segmented object parts into a 3D volume representing the manipulated object.

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.) to exclude these voxels of the TSDF volume from being updated with invalid information.

At the end of the integration process, ray-casting [10] 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 [4]. 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. .

The proposed method was tested quantitatively and qualitatively in sequences where a person manipulates objects of different sizes. Table 1 shows the mean and median hand tracking error over a sequence with known ground truth. The first column shows that for the proposed method (unknown object model). The second shows that for [3] (ground truth object model). The third column shows the error obtained with [3] using the reconstructed object model created with our method. The results demonstrate that the hand tracking accuracy of our method is comparable to that of [3], although for our method the object model is unknown. Furthermore, the reconstructed model resulting from our method can be used to track the sequence with similar accuracy to the ground truth model, which demonstrates the high quality of the reconstruction. Qualitative results along with the resulting textured 3D models are shown in Fig. 3. Tracking a single hand in interaction with an object can be achieved at a frame rate of 10 fps, a performance that drops to 4 fps in the case of tracking bimanual manipulation scenarios. Further optimizations are expected to result in real time performance. A more detailed presentation of this work is available in [9]. Full run of the experiments is available with the supplementary material of this work at http://youtu.be/9r43PtJ0Fwg.

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

[1] Brian Curless and Marc Levoy. A volumetric method for building complex models from range images. In *Proc., 23rd annual conference on computer graphics and interactive techniques*, pages 303–312. ACM, 1996.

[2] 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.

[3] Nikolaos Kyriazis and Antonis Argyros. Scalable 3d tracking of multiple interacting objects. In *CVPR*, pages 3430–3437, 2014.

[4] Zoltan Csaba Marton, Radu Bogdan Rusu, and Michael Beetz. On Fast Surface Reconstruction Methods for Large and Noisy Datasets. In *ICRA*, Kobe, Japan, May 12-17 2009.

[5] Damien Michel, Xenophon Zabulis, and Antonis A. Argyros. Shape from interaction. *Mach. Vision Appl.*, 25(4), May 2014.

[6] 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 *ISMAR*, pages 127–136. IEEE, 2011.

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

[8] 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.

[9] P. Panteleris, N. Kyriazis, and A.A. Argyros. 3d tracking of human hands in interaction with unknown objects. In *BMVC*, Sep 2015.

[10] Steven Parker, Peter Shirley, Yarden Livnat, Charles Hansen, and Peter-Pike Sloan. Interactive ray tracing for isosurface rendering. In *Proc., Visualization'98*, pages 233–238. IEEE, 1998.

[11] 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.

[12] Carlo Tomasi and Roberto Manduchi. Bilateral filtering for gray and color images. In *ICCV*, pages 839–846. IEEE, 1998.

[13] Thibaut Weise, Thomas Wismer, Bastian Leibe, and Luc Van Gool. In-hand scanning with online loop closure. In *ICCV Workshops*, pages 1630–1637. IEEE, 2009.