An Integrated Approach to Visual Perception of Articulated Objects

Roberto Martín-Martín Sebastian Höfer Oliver Brock

***Abstract*— We present an integrated approach for perception of unknown articulated objects. To robustly perceive objects and understand interactions, our method tightly integrates pose tracking, shape reconstruction, and the estimation of their kinematic structure. The key insight of our method is that these sub-problems complement each other: for example, tracking is greatly facilitated by knowing the shape of the object, whereas the shape and the kinematic structure can be more easily reconstructed if the motion of the object is known. Our combined method leverages these synergies to improve the performance of perception. We analyze the proposed method in average cases and difficult scenarios using a variety of rigid and articulated objects. The results show that our integrated solution achieves better results than solutions for the individual problems. This demonstrates the benefits of approaching robot perception problems in an integrated manner.**

I. Introduction

To successfully manipulate previously unseen objects, robots must be able to acquire information about their shape, pose, and kinematic relationship to other parts of the environment. The majority of existing approaches to robot perception, however, address these problems in isolation. In this paper, we present arguments why instead they should be seen as a single problem and propose a method that integrates these sub-problems into a single approach to robot perception. We show that this integrated solution achieves better results than solutions for the individual problems.

Shape reconstruction, pose tracking, and kinematic structure estimation naturally complement each other. To reconstruct the shape of an object from RGB-D sensors, it is necessary to integrate multiple views of the object under the assumption that the relative poses of the views are known. Most approaches therefore require knowledge of the pose [1]. On the other hand, to track the pose of an object, methods commonly rely on knowledge of the object's shape and its segmentation in the image [2, 3]. Similarly, the estimation of kinematic structure of an unknown object is facilitated by knowing the poses of its rigid parts—but knowing the kinematic structure can also improve pose estimation [4]. Since each of these problems requires input that is provided by the others, we propose to combine them in a synergistic manner so that each sub-problem provides helpful information to the others. Interestingly, in the combined problem, object segmentation serves as the connection between shape estimation and pose tracking: each of the two subcomponents passes information about its current object segmentation hypothesis to the other in order to improve the estimation (Fig. 2). We show that the tight coupling of sub-problems enables perception of previously unseen articulated objects, when partial solutions would not be applicable.

Fig. 1. Our robot perceiving an articulated object using our integrated approach; it interacts with the drawer and detects the moving body, tracks it and incrementally reconstructs its shape (yellow layer); the robot estimates and tracks the kinematic structure, including the joint axis (narrow green cylinder) and joint value (wider green cylinder), and an estimate of the uncertainty (transparent green cone)

Our method extends two recent methods that follow the same insight [5, 6]. Most importantly, it includes and exploits the estimation of kinematic structures for interacted articulated objects (Fig. 1). We also provide a thorough experimental evaluation to analyze the improvements afforded by an integrated solution. We analyze the contribution of each component to the final result. The evaluation includes difficult cases that are unsolvable when the sub-problems are not tightly integrated. Our experimental evaluation demonstrates the benefits of combining problems in robot perception and solving them in an integrated manner.

II. Related Work

We first review approaches that address pose estimation, shape reconstruction, and segmentation independently, also covering previous work on kinematic structure estimation. We then turn to combined approaches to these problems.

Visual pose estimation is the problem of inferring an object's pose from an image; the problem is called *visual tracking* if performed on a stream of images. We distinguish two main approaches to visual tracking: based on a *known shape* model of the object [2], and based on *sufficient texture* using point features [4] or dense optical flow [5, 7]. While tracking with a known model is more accurate, texture-based approaches are also applicable to unknown objects. We exploit the advantages of both approaches by bootstrapping the system with feature-based tracking, and subsequently combining it with shape-based tracking.

Based on the tracked poses one can infer the *kinematic structure* restricting their relative motion [8]. In our previous

We gratefully acknowledge the funding provided by the Alexander von Humboldt foundation and the Federal Ministry of Education and Research (BMBF), the European Commission (SOMA project, H2020-ICT-645599) and the German Research Foundation (DFG, Exploration Challenge, BR 2248/3-1).

All authors are with the Robotics and Biology Laboratory, Technische Universität Berlin, Germany

work we have shown that tracking can also greatly benefit from a known kinematic structure [4].

Shape reconstruction acquires a 3-D appearance model of an object by merging a set of partial object views into a coherent shape model using information about the relative object poses w.r.t. the camera. Partial object views can be obtained using image segmentation and pose information by controlling camera and object motion [1]. Our approach automatically generates both, partial object views and their pose information, and uses them to reconstruct the shape.

The *segmentation* problem consists of finding the region in the visual input occupied by the object. We distinguish between *single-image-based segmentation* and *motion-based segmentation* operating on image streams. For segmenting single images a wide variety of different approaches has been proposed, e.g. assuming *surface continuity* as exploited by conditional random fields [9] graph-cuts, [10], and supervoxel region growing [11], or exploiting *object location* as in active segmentation [12]. Motion-based segmentation exploits the notion of "object-ness" by assuming that all points on a rigid body move together. To detect which points moved due to object motion, image differencing [13, 14] can be applied. To reject changes caused by background motion it can be combined with information from a tracker to select only points that move consistently with the object [5, 7]. In our approach we apply motion-based segmentation to generate sparse segments and extend them using supervoxel region growing, and we use single-image segmentation by using the continuously updated reconstruction of the shape.

A prominent *combined approach* to tracking and shape reconstruction is visual SLAM [15]. However, it reconstructs an entire scene which it assumes to be static, and does not segment and reconstruct single objects.

This *object* perception problem has recently been addressed in a combined manner. Ren et al. present a method to simultaneously track and reconstruct 3-D objects [16] by refining an initial primitive shape model; in contrast to our method it can only reconstruct and track one moving object and the initial location of this object must be manually provided. Stückler et al. suggest a method which combines object tracking, segmentation, and reconstruction using an EM algorithm [5]. Their method differs from ours, as it relies on an initial oversegmentation and groups the segments using motion and surface clues, making it sensitive to a wrong initial segmentation. Other methods [6, 17] build on top of KinectFusion [15]. These methods build a model of the environment and consider as a new object any part that becomes inconsistent with this model. In contrast to these methods, our method is able to track articulated objects.

III. Method

We now describe our integrated method for pose tracking, object segmentation, shape reconstruction, and kinematic structure estimation (Fig. 2). Our feature and shape trackers (Sec. III-A) provide the motion information to segment RGB-D frames into objects (Sec. III-B). These object segments are used to reconstruct the shape of the object over time

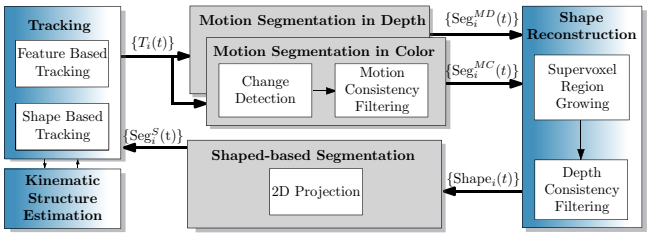

Fig. 2. Our tightly integrated shape, pose and kinematic structure estimation pipeline, using segmentation as an intermediate process; see text for details

(Sec. III-C). To close the loop, we use the result from shape reconstruction to find better object segments (Sec. III-D) and use them to refine tracking (Sec. III-A.2).

A. Sensing and Tracking

The input to our method is an RGB-D stream represented as a sequence of color images $I(t)$ and depth maps $D(t)$ for every timestep t. Some parts of our method directly operate on point clouds $P(t)$ which combine color and depth. We preprocess the raw depth images by applying a joint bilateral filter [18] in order to fill the depth-missing areas based on their surrounding depth and color information.

1) Feature-based Tracking and Kinematic Structure Estimation: To bootstrap our pipeline, we acquire information about object motion and location by using our previously developed combined approach to motion tracking and kinematic structure estimation *Online Interactive Perception (Online IP)* [4]. Online IP is composed of three interconnected processes: a *3-D point feature tracker* maintains a constant set of tracked point features. Their locations and velocities are then passed to a *rigid body tracker* which groups coherently moving features to rigid bodies. Finally, a *joint tracker* estimates the kinematic constraints between the rigid bodies. Similar to the approach presented here, the key of Online IP is the intercommunication between these three processes to mutually improve each other's results.

The output of Online IP is a set of 6-D object poses $\{T_i(t)\}_{i\in\{1,\ldots,n\}}$ of the currently tracked objects $\{o_1,\ldots,o_n\}$, their 6-D velocities, a sparse set of tracked 3-D point features on each object, and kinematic constraints, i.e. joints, between the objects. We use the output from Online IP (and later of the combined tracker) in our motion segmentation component (Sec. III-B) which generates inputs for the shape reconstruction.

2) Shape-based Tracking: Based on the result of the shape reconstruction (Sec. III-C), we compute a more complete segmentation (Sec. III-D) and use it as input to a shape-based tracker. The tracker exploits this segmentation to refine the pose estimate of Online IP using ICP [19]. We initialize ICP with the predicted object pose (based on its tracked velocity) to reduce computation time and to facilitate convergence to a favorable pose. This shape-based tracking overcomes the limitation of Online IP that can only track texturized objects, as shown in our experiments.

B. Motion Segmentation

To use the information from the pose tracker in shape reconstruction, we compute a motion segmentation of the objects. Together with the pose information these segments are used to reconstruct the entire object as detailed in Sec. III-D.

The general idea of motion segmentation is to first detect changes in the depth and color images of two consecutive time steps, and then use the tracked 6-D poses to identify areas that change consistently with the motion of the object. These areas are the desired motion segments. Using the tracking information is beneficial for rejecting false positives found by change detection, as in very dynamic scenes the latter is prone to wrongly attribute changes of the environment to the motion of the object.

Algorithm 1 Motion Segmentation in Depth

1: $\Delta D = D(t) - D(t-1)$
2: $M_{\mathrm{E}} := \Delta D < -\gamma_{\mathrm{motion}}$
3: $M_{\mathrm{L}} := \Delta D > \gamma_{\mathrm{motion}}$
4: $M_{\mathrm{AccE}} := M_{\mathrm{AccE}} \wedge M_{\mathrm{E}} \wedge \neg M_{\mathrm{L}}$
5: **for all** T_i **do**
6: $\quad \Delta T_i := T_i(t) \cdot T_i^{-1}(t-1)$
7: $\quad \mathrm{Seg}_i^{\mathrm{AccE}} := 1\mathrm{NN}(\Delta T_i \cdot P(t-1), P(t)|_{M_{\mathrm{AccE}}})$
8: $\quad \mathrm{Seg}_i^{\mathrm{L}} := 1\mathrm{NN}(\Delta T_i \cdot P(t-1)|_{M_{\mathrm{L}}}, P(t))$
9: $\quad \mathbf{Seg_i^{MD}} := \mathrm{Seg}^{AccE} \cup \mathrm{Seg}^{L}$

1) Motion Segmentation in Depth: We first detect changes in the scene by computing a difference image ΔD of depth maps from subsequent time steps (line 1 in Algorithm 1). Assuming for a moment that every change in ΔD has been caused by the motion of the body, we know that the body has left some part of the image and/or entered some other. In a particular region of the image, we can discriminate between these two cases by looking at the sign of ΔD and computing a binary motion mask for each case (lines 2-3): the *entering-motion mask* M_E contains pixels whose distance decreased (sign is negative), the *leaving-motion mask* contains pixels whose distance increased (wrt. noise threshold γ_{motion}). To handle small motions between subsequent time steps we accumulate point in M_E over time in the *accumulated entering-motion mask* M_{AccE}.

Next, we discriminate which of the detected changes are consistent with each object's tracked motion ΔT_i (line 6). The basic idea is to apply ΔT_i to the previous point cloud $P(t-1)$ and compare the result to the current $P(t)$ using nearest-neighbor search (1NN, line 7). By first filtering $P(t)$ with M_{AccE} we only take into account points that changed in depth. In a similar fashion we use M_L to find points in that belong to the object in $P(t-1)$ (line 8), and add the two point clouds together to obtain the final depth-based motion segment $\mathrm{Seg}_i^{\mathrm{MD}}$.

2) Motion Segmentation in Color: Our depth-based motion segmentation method is rather conservative as it does not add points to the segment that have not changed their depth value, even if they are consistent with ΔT_i. Although reducing the risk of adding false positives, the approach fails if no change in depth is present, e.g. a rotating globe. We therefore add a color-based motion segmentation which works similar to the depth-based version presented in Algorithm 1. The main differences are that we compute the image difference in HSV color space, using only hue (H) values for all pixels with sufficient saturation (S $>$ 90) and that we do not discriminate between image regions the body has left or entered (since depth information is required for this).

C. Shape Reconstruction

We can now exploit the information from the pose tracker (T_i) to reconstruct the shape based on the motion-based segments ($\mathrm{Seg}_i^{\mathrm{MD}}$ and $\mathrm{Seg}_i^{\mathrm{MC}}$). We represent the shape by a point cloud, which is resampled using a voxel-grid filter at every time step in order to keep constant the required memory and to address the inhomogeneous point distribution resulting from the depth measurements. To deal with regularly shaped objects with uniform color, which usually generate sparse motion segments, we first extend these segments by exploiting *surface continuity* and *known object location*, and finally filter out points that are inconsistent with the current view of the scene.

1) Supervoxel Region Growing: We extend the partial motion segments by using supervoxel segmentation followed by region growing. We first seed region growing with the supervoxels that coincide with the location of the mean feature given by Online IP (Sec. III-A). Region growing then extends to a neighboring supervoxel if most of the points it contains (i) demonstrated coherent motion (resulting from the segmentation based on motion III-B), (ii) are part of the reconstructed shape so far, or (iii) if the mean color and mean surface normal are very similar. We also extend the segment if the neighboring supervoxel contains features of the rigid body. The result is then merged into the shape model.

2) Depth Consistency Filtering: All previous steps are adding points to the shape model. However, sensor noise, errors in tracking and overly optimistic supervoxel extensions can lead to wrong points being added to the model. We can remove many of these points by verifying whether they are consistent with the current depth map $D(t)$ when projecting the model to the image plane. We therefore remove every point for which the projection calculates a lower depth value than observed in $D(t)$ – this means that we see a background point in the image where we expected a point of the object.

D. Shape-based Segmentation

Using the results from the previous shape reconstruction steps, shape segmentation becomes trivial. The shape segment at the current time step $\mathrm{Seg}_i^S(t)$ is the result of transforming the shape model using its tracked pose T_i and projecting the result into the image plane. The shape segment is then fed to the shape tracking component to improve tracking and kinematic structure estimation, and the procedure starts over in the next time step.

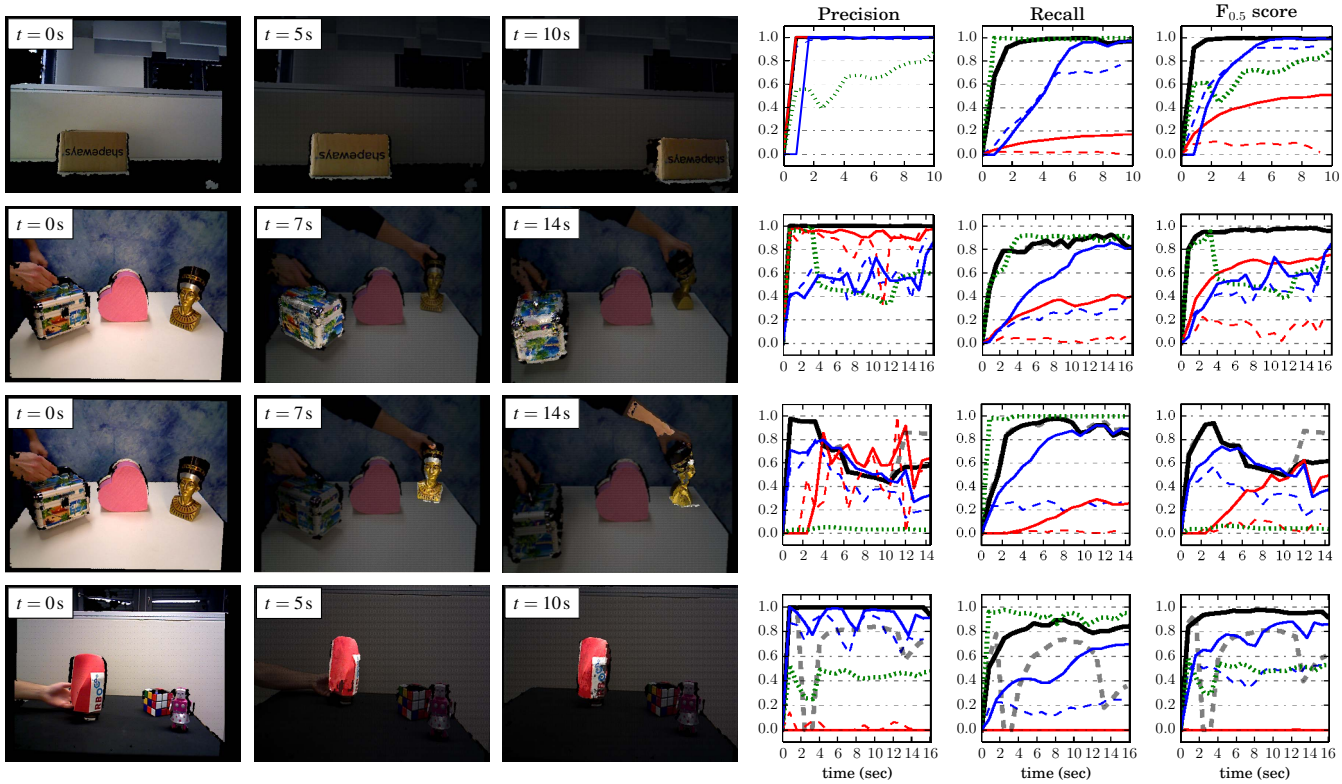

Fig. 3. Results of the segmentation (each row represents a different object); from left to right: full initial scene (RGB-D point cloud projected to image plane), result after first segmentation, and final segmentation result (solid color indicates the segment), precision, recall, F-Score; we compare our full pipeline to subparts of it and to the segmentation generated by [7]; the insets in the three images show the time t

IV. Experiments

We evaluate our approach in eight different experiments, each carefully selected to verify the contribution of each component of our method.[1] In each experiment, a human or a robot actuates one or more objects in the scene. During robot manipulation, we exploit additional information (forward kinematics and a known shape model of the robot arm) to infer the part of the image that corresponds to the robot and exclude it from tracking and segmentation. We recorded each experiment using a statically mounted Asus Xtion RGB-D sensor. We run our algorithm on an Intel Xeon E5520 CPU at 2.27 GHz, reaching 3 to 10 frames per second for the shape-based tracker, depending on the size and number of moving objects. Segmentation and shape reconstruction are running at a lower rate of 0.8 s, mostly due to the computationally demanding supervoxel segmentation. But since tracking is running at a high rate, slow reconstruction time does not affect the capability to track fast motions.

The experiments consist of three scenes containing only rigid objects and five scenes with articulated ones.

Box: A box with little texture moves parallel to the viewing-plane for about 50cm (duration: 9 s). We expect approaches that do not exploit surface continuity to require long time to reconstruct the shape.

Two Bodies: In this experiment two bodies move freely on a table-top at the same time (duration: 14 s). We want to verify that the method can cope with multi-body settings and to which extent the hand of the experimentator is added to the reconstructions.

Red Figure: A red figure is moved freely on a table-top (duration: 10 s). This experiment is designed to test how the pipeline performs when the quality of the features abruptly degrades. We therefore manually removed all features after 3 seconds, to verify the contribution of shape-based tracking.

Drawer: A robot opens a drawer (duration: 2 s). An easy articulated object case.

Globe: A globe rotating 360 degrees around its revolute axis (duration: 18 s). We expect that pure feature-based tracking is inaccurate due to the large uniformly colored areas, and we expect incomplete reconstructions if surface continuity is not used.

Head: The first author rotates his head left and right (duration: 8 s). This experiment evaluates to which extent semi-rigid objects pose a problem for our method.

Cabinet and Drawer: A cabinet moves freely on the floor (duration: 15 s). At some point a drawer is pulled out of the cabinet and pushed inside again. We evaluate the performance when objects partially getting out of the field

[1] Our datasets are publicly available under http://tinyurl.com/o3bu7pd.

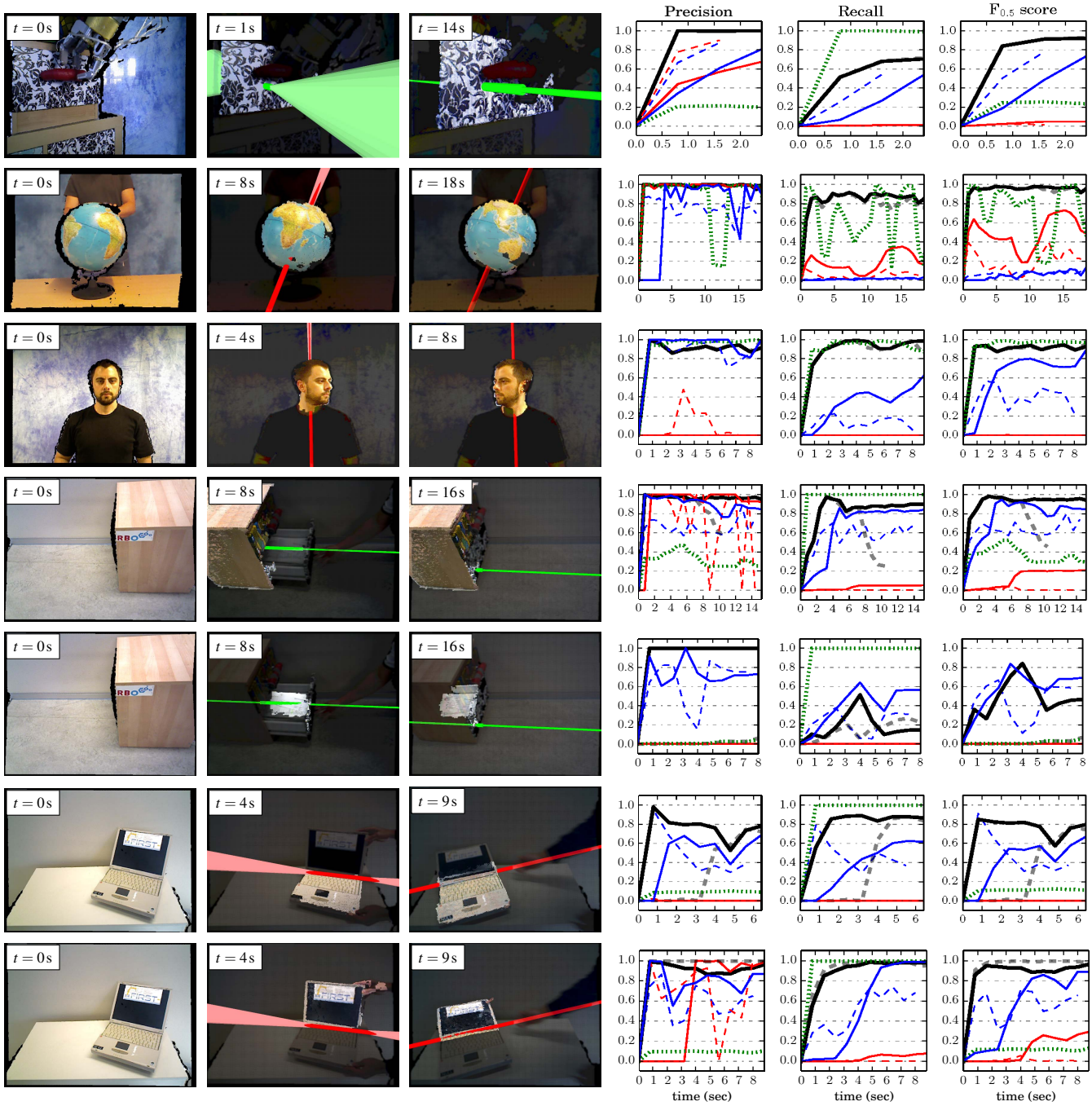

Fig. 4. Results of the segmentation of articulated objects (each row represents a different rigid body); from left to right: full initial scene, result after first segmentation, final segmentation result, precision, recall, F-Score; the insets in the three images show the time t; estimated prismatic joints are shown as solid green cylinders, revolute joints as solid red cylinders; transparent, narrow cones represent the uncertainty of the axes orientation

of view.

Laptop: A laptop is moved freely on a table-top and then being closed and opened (duration: 9 s). We evaluate the effect of purely rotational motion on the reconstruction.

Evaluation

We evaluate the contribution of each component our method using two criteria. First, we quantitatively assess the object segmentation results provided by each component. This gives an indirect means of comparing the impact of the different parts, as the accuracy of the pose tracker directly influences the quality of the motion-based segmentation, and the correctness of the shape reconstruction affects the shape-based segmentation. Secondly, we evaluate the quality of the reconstructed shape and the estimated kinematic structure

(using Online IP [4] with shape-based tracking) by visual inspection of the results.

To evaluate the segmentation results, we manually annotated each video sequence with the ground truth every 0.8 seconds. We compute precision, recall and $f_{0.5}$-score[2] for the *full pipeline* (i) and five additional variants of our algorithm: to assess how integrating tracking and shape reconstruction affects the result we evaluate the *full pipeline without feedback from shape tracking* (ii); to evaluate the contribution of the tracker to the pipeline we look at *depth-based motion segmentation* (iii) and *color-based motion segmentation* (iv); finally, we assess the contribution of shape-base tracking by evaluating *shape-based segmentation using only depth* (v) and *shape-based segmentation using only color information* (vi). We additionally compare our results to a baseline, a dense optical flow approach using RGB presented by Ochs et al. [7] (using the recommended standard parameters). We compare against this method as it is the only relevant one for which code was available at the time of writing.

V. Results

The segmentation results are depicted in Fig. 3 (rigid objects) and Fig. 4 (articulated objects). We observe that at the end of each experiment for all except two cases (two bodies: statue; cabinet: drawer) the full pipeline with tracking (solid black curve) outperforms all other variants, attaining $f_{0.5}$-scores above 0.8. Thus overall, our method detects most of the area occupied by each object (high recall) while adding few false points (high precision). Secondly, we observe that both full pipeline variants converge very fast to their final results. The reason is the effective combination of the different priors: whereas the pure motion segmentation variants (blue and red curves) usually require long time to obtain the full segment and reach high recall, extending the motion segment with the neighboring supervoxels allows to quickly obtain the entire segment. The baseline by Ochs et al. [7] often fails to find the correct segments and gives competitive results only in the head experiment.

The reconstruction and kinematic structure estimation results are shown in Fig. 5 and Fig. 6. The results are in line with the segmentation results since all objects except for the statue and the drawer are reconstructed correctly. We also see that joint estimation is much more accurate when including shape tracking which indicates that the combined tracker provides higher quality pose estimates.

We now turn to a detailed analysis of every scene.

Box: Most variants perform well in segmentation and reconstruction, but close to the object borders some variants add wrong points that belong to the background. The reason is inaccurate registration of depth and RGB pixels by the sensor, causing wrong depth measurements at the objects borders.

[2]The $f_{0.5}$-score is a standard variant of the f-score which weighs precision higher. We use this variant since points wrongly attributed to the object – which only affect precision – have a more significant negative impact on tracking and thus reconstruction performance than missed points.

Two Bodies – Metal case: The full pipeline succeeds in quickly segmenting the metal case and reconstructing all three visible sides of the object. The motion-based segmentation methods fail because they do not exploit knowledge about object location, and add spurious points on the arm.

Two Bodies – Statue: This is the only case where the full pipeline without shape tracking (dashed black curve) outperforms the other strategies. The reason is that early during the experiment the hand of the experimenter is added to the segment. When the hand starts retracting at $t = 9.8s$ till the end, the full pipeline wrongly biases the feature tracker to pay attention to the motion of the hand, whereas the variant without shape tracker removes the hand from the model. This effect is a limit of our motion segmentation approach: two bodies moving similarly during enough time are consider to be the same body.

Red Figure: The red figure is best segmented by the full pipeline. Without shape-based tracking the performance drops drastically when the features disappear. However, even without the shape tracker the pipeline can partially recover because enough changes in depth are visible.

Drawer: Almost all variants segment and reconstruct the front lid of the drawer (which is the only visible part) and detect the prismatic joint quickly. Color-based segmentation fails because the scene is very dark and we ignore points where the saturation in the HSV space is low.

Globe: Since there is no change in depth and little change in color, pure motion segmentation fails to segment large parts, indicated by the low recall values. The quality of the reconstruction and joint estimation depends on the tracker (Fig. 6). A large part of the globe does not exhibit sufficient texture to accurately track a large number of features and hence tracking is not very stable, resulting in a non-spherical shape reconstruction. By constantly incorporating the feedback from shape reconstruction, we obtain a much more spherical reconstruction.

Head: Similar to the globe, the head is only well segmented by the full pipeline (and the baseline). Some misclassifications of the neck and the hair lead to minor segmentation errors. The reconstruction is accurate but exhibits some abrupt color changes due to non-uniform lighting.

Cabinet – Frame: The cabinet frame is quickly segmented by the full pipeline which is also reflected in the reconstruction. Without shape tracking, the feature-based tracker loses the frame object at $t = 7$ seconds after the drawer has moved for 3 seconds. The reason is that there are more features on the drawer, so the tracker assumes that the remaining features on the cabinet are outliers and drops them. In contrast, by taking into account the shape the tracker correctly splits the cabinet and drawer into two rigid objects. In this case, the cabinet is correctly reconstructed and remains stable even when it leaves the scene.

Cabinet – Drawer: The drawer is only partially segmented by the full pipeline. The reason is that the motion segmentation provides two disconnected components: the inner part and the lateral part. The supervoxel growing is seeded from the inner part and due to the gap in depth

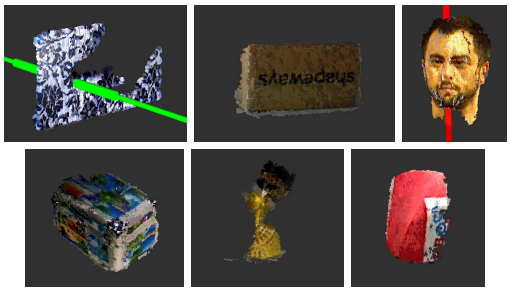

Fig. 5. Results of the shape reconstruction in combination with motion tracker; estimated prismatic joints are shown as solid green cylinders, revolute joints as solid red cylinders; joint value shown as wider cylinder

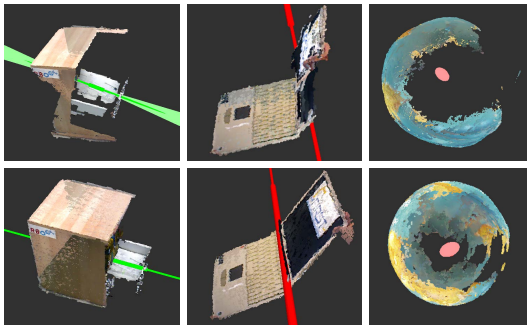

Fig. 6. Results of the shape reconstruction and kinematic structure estimation; each column represents a different articulated object; from top to bottom: results when shape-based motion tracker is not integrated, results when shape-based motion tracker is integrated; estimated prismatic joints are shown as solid green cylinders, revolute joints as solid red cylinders; joint value is shown as wider cylinder; uncertainty about the joint is represented as transparent cones

does not extend to the lateral part. The purely motion-based segmentation performs better because it does not use this location information (which however leads to degraded performance in the box experiment as mentioned earlier). The segmentation reduces the quality of the reconstruction. Still, using the partially reconstructed drawer for shape-based tracking largely improves reconstruction and the estimation of the prismatic joint, as shown in Fig. 6, first column. Moreover, our method correctly handles the occlusion of the drawer when it is closed and remembers its shape.

Laptop – Bottom and Lid: Both rigid bodies of the laptop are correctly segmented by the full pipeline, with only some points incorrectly added. These points are close to the revolute axis and present small errors when the body motion is applied to them. We foreseen this as a limitation of our current algorithm that could be solved by reconstructing jointly all parts of the articulated object.
Again, shape reconstruction and tracking is much more accurate when the shape-based tracker is used. Without shape-tracking the kinematic structure diverges by approx. 5°, as visible in Fig. 6, second column.

To conclude, in all experiments the full pipeline with tracking feedback achieves very good performance and outperforms the other variants with respect to segmentation and reconstruction in all but two experiments.

VI. Conclusion

We presented a combined approach for estimating pose, shape, and kinematic structure of articulated objects. The algorithm exploits the synergies between the sub-solutions and integrates them in an online manner. The sub-solutions provide information to each other, leading to performance improvement and eliminating the dependency on a priori knowledge about the world. We demonstrate the benefits of the combined approach by comparing its performance with that of the sub-components in several difficult cases. Our algorithm perceives and tracks the shape, pose and kinematic structure of multiple objects.

References

[1] M. Krainin, P. Henry, X. Ren, and D. Fox, "Manipulator and object tracking for in-hand 3d object modeling," *The International Journal of Robotics Research*, vol. 17, no. 11, pp. 1311–1327, Jul. 2011.

[2] M. Wüthrich, P. Pastor, M. Kalakrishnan, J. Bohg, and S. Schaal, "Probabilistic object tracking using a range camera," in *International Conference on Intelligent Robots and Systems*, 2013, pp. 3195–3202.

[3] C. Choi and H. I. Christensen, "RGB-D object tracking: A particle filter approach on GPU," in *International Conference on Intelligent Robots and Systems*, 2013, pp. 1084–1091.

[4] R. Martín-Martín and O. Brock, "Online Interactive Perception of Articulated Objects with Multi-Level Recursive Estimation Based on Task-Specific Priors," in *International Conference on Intelligent Robots and Systems*, 2014.

[5] J. Stückler and S. Behnke, "Efficient dense 3D rigid-body motion segmentation in RGB-D video," in *British Machine Vision Conf.*, 2013.

[6] L. Ma and G. Sibley, "Unsupervised dense object discovery, detection, tracking and reconstruction," in *Computer Vision*. Springer, 2014.

[7] P. Ochs, J. Malik, and T. Brox, "Segmentation of Moving Objects by Long Term Video Analysis," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 36, no. 6, pp. 1187–1200, 2014.

[8] J. Sturm, C. Stachniss, and W. Burgard, "A Probabilistic Framework for Learning Kinematic Models of Articulated Objects," *Journal of Artificial Intelligence Research*, vol. 41, no. 2, pp. 477–526, 2011.

[9] J. Lafferty, A. McCallum, and F. Pereira, "Conditional random fields," *International Conference on Machine Learning*, pp. 282–289, 2001.

[10] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient Graph-Based Image Segmentation," *International Journal of Computer Vision*, vol. 59, no. 2, pp. 167–181, Sep. 2004.

[11] J. Papon, A. Abramov, M. Schoeler, and F. Wörgötter, "Voxel Cloud Connectivity Segmentation - Supervoxels for Point Clouds," in *IEEE Conference on Computer Vision and Pattern Recognition*, 2013, pp. 2027–2034.

[12] A. Mishra, Y. Aloimonos, and C. Fermüller, "Active segmentation for robotics," in *International Conference on Intelligent Robots and Systems*. IEEE, 2009, pp. 3133–3139.

[13] S.-Y. Chien, S.-Y. Ma, and L.-G. Chen, "Efficient moving object segmentation algorithm using background registration technique," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 12, no. 7, pp. 577–586, Jul. 2002.

[14] J. Kenney, T. Buckley, and O. Brock, "Interactive segmentation for manipulation in unstructured environments," in *IEEE International Conference on Robotics and Automation*, 2009, pp. 1377–1382.

[15] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohi, J. Shotton, S. Hodges, and A. Fitzgibbon, "KinectFusion: Real-time dense surface mapping and tracking," in *IEEE Intl. Symp. on Mixed and Augmented Reality*, 2011, pp. 127–136.

[16] C. Y. Ren, V. Prisacariu, D. Murray, and I. Reid, "STAR3d: Simultaneous Tracking and Reconstruction of 3D Objects Using RGB-D Data," in *IEEE Intl. Conf. on Computer Vision*, 2013, pp. 1561–1568.

[17] E. Herbst, P. Henry, and D. Fox, "Toward Online 3-D Object Segmentation and Mapping," in *Proc. of the IEEE Int. Conf. on Robotics & Automation (ICRA)*, Hong Kong, China, 2014, pp. 3193 – 3200.

[18] A. Le, S.-W. Jung, and C. Won, "Directional Joint Bilateral Filter for Depth Images," *Sensors*, vol. 14, no. 7, pp. 11 362–11 378, Jun. 2014.

[19] F. Pomerleau, S. Magnenat, F. Colas, M. Liu, and R. Siegwart, "Tracking a depth camera: Parameter exploration for fast ICP," in *International Conference on Intelligent Robots and Systems*, 2011.