2013 IEEE/RSJ International Conference on
Intelligent Robots and Systems (IROS)
November 3-7, 2013. Tokyo, Japan

# Point Cloud Video Object Segmentation using a Persistent Supervoxel World-Model

Jeremie Papon, Tomas Kulvicius, Eren Erdal Aksoy, Florentin Wörgötter
Bernstein Center for Computational Neuroscience (BCCN)
III. Physikalisches Institut - Biophysik, Georg-August University of Göttingen
{jpapon,tomas,eaksoye,worgott}@physik3.gwdg.de

*Abstract*— **Robust visual tracking is an essential precursor to understanding and replicating human actions in robotic systems. In order to accurately evaluate the semantic meaning of a sequence of video frames, or to replicate an action contained therein, one must be able to coherently track and segment all observed agents and objects. This work proposes a novel online point cloud based algorithm which simultaneously tracks 6DoF pose and determines spatial extent of all entities in indoor scenarios. This is accomplished using a persistent supervoxel world-model which is updated, rather than replaced, as new frames of data arrive. Maintenance of a world model enables general object permanence, permitting successful tracking through full occlusions. Object models are tracked using a bank of independent adaptive particle filters which use a supervoxel observation model to give rough estimates of object state. These are united using a novel multi-model RANSAC-like approach, which seeks to minimize a global energy function associating world-model supervoxels to predicted states. We present results on a standard robotic assembly benchmark for two application scenarios - human trajectory imitation and semantic action understanding - demonstrating the usefulness of the tracking in intelligent robotic systems.**

## I. INTRODUCTION

Multi-target visual tracking (MTVT) and 6DoF pose estimation are crucial challenges for many applications such as visual surveillance, action recognition, and robotic imitation learning. In many such functions, visual tracking serves as the precursor to all further high-level inference, making robust tracking fundamental to the success of a large variety of intelligent systems. Related to the problem of visual tracking is segmentation, the task of grouping observations according to the entities which they contain. Video object segmentation (VOS) attempts to cluster pixels of video frames into segments which are both spatially and temporally coherent. While generally similar to MTVT, VOS goes a step beyond localizing tracked objects, in that it makes an association decision for each observed pixel; in addition to estimating overall state, it must re-estimate spatial extent every frame. In both VOS and MTVT there are two chief challenges that must be addressed: first, the data association problem, whereby noisy observations must be associated with the proper targets, and secondly, the occlusion problem, in which targets may become partially or fully obscured for a number of observations.

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 269959, IntellAct.

While MTVT remains an unsolved problem, single target visual tracking (STVT) is a fairly well-studied problem, with many mature approaches [1], [2]. Additionally, recent work has progressed in estimating pose (in addition to tracks) for single targets, for example [3] uses a particle filter to track 6-DoF pose of arbitrary objects in point clouds. Recent work in MTVT [4] successfully tracks multiple objects using a segmentation and association approach and adaptive 3D appearance models, but is limited by the need to align model point clouds to the observed data every frame. This precludes it from handling occlusions, as once a target is no longer observed, its track must be terminated.

Multiple hypothesis video segmentation (MHVS) from superpixel flows [5] provides dense online unsupervised video segmentations, but is only able to handle partial occlusions for a few frames, and does not consider full occlusions. There also has been much recent work in VOS specifically addressing the problem of segmenting foreground from background [6], [7]. While these works have been to shown to perform very well in their task, they only solve the single target case, as they do not need to resolve the multiple association problem.

In [8] Papadakis and Bugeau use a dynamical model to guide successive segmentations, along with an energy function minimized using graph cuts to solve the label association problem. They formally model visible and occluded regions of tracked objects, tracking them as distinct parts. While they do consider occlusions, they do not maintain a world model, and as such their methodology must fail under complete occlusions. Although it does not address segmentation or tracking, we should mention Isack and Boykov [9], as its use of a global energy function and RANSAC model sampling to solve a geometric multi-model fitting is similar to the approach taken in this work.

While MTVT and VOS are clearly related, they traditionally have been considered separate areas of research. In this work, we unify them by taking a mature tracking approach, particle filtering, and apply it to tracking supervoxels (3d segments) from a recent 3d segmentation technique [10], Voxel Cloud Connectivity Segmentation (VCCS). To make this possible, we extend the concept of VCCS to dynamic scenes by maintaining a world-octree supervoxel model

Fig. 1. Supervoxels found using VCCS. From left to right: original RGB image, supervoxels with $R_{seed} = 0.08\,\text{m}$, and supervoxels with $R_{seed} = 0.03\,\text{m}$.

which lets objects persist indefinitely through occlusions. Additionally, we use a novel global energy function to associate observations to predictions, and thereby extract accurate object segmentations (even for fully occluded objects) from tracker predictions.

The paper is organized as follows: Section II first presents the VCCS framework, then extends it to incremental updating using sequential frames. Next, Section III discusses the particle filters used for predictive tracking, and describes the joint energy minimization used to associate predictions with observed supervoxels. Finally, Section IV consists of application scenarios where the pose and tracks of tracked objects are used as the basis for robot imitation and generation of semantic summaries of human actions, and Section V describes current limitations of the algorithm, discusses future work, and concludes.

## II. SEQUENTIALLY UPDATED SUPERVOXELS

### A. Voxel Cloud Connectivity Segmentation

VCCS[10] is a recent method which generates volumetric over-segmentations of 3D point cloud data, known as supervoxels. Supervoxels adhere to object boundaries better than state-of-the-art 2D methods, while remaining efficient enough to use in online applications. VCCS uses a region growing variant of k-means clustering for generating its labeling of points directly within a voxel octree structure. Supervoxels have two important properties; they are evenly distributed across the 3D space, and they cannot cross boundaries unless the underlying voxels are spatially connected. The former is accomplished by seeding supervoxels directly in the cloud, rather than the projected plane, while the latter uses an octree structure which maintains adjacency information of leaves.

Supervoxels maintain adjacency relations in voxelized 3D space; specifically, 26-adjacency- that is neighboring voxels are those that share a face, edge, or vertex. The adjacency graph of supervoxels (and the underlying voxels) is maintained efficiently within the octree by searching for neighboring leaves in the voxel grid, where $R_{voxel}$ specifies the octree leaf resolution. This adjacency graph is used extensively for both the region growing used to generate the supervoxels as well as determining adjacency of the resulting supervoxels themselves.

VCCS is a region growing method which incrementally expand supervoxels from a set of seed points distributed evenly in space on a grid with resolution $R_{seed}$. Fig. 1 shows how $R_{seed}$ effects the resulting supervoxels. Expansion from the seed points is governed by a distance measure calculated in a feature space consisting of spatial extent, color, and normals. The spatial distance $D_s$ is normalized by the seeding resolution, color distance $D_c$ is the euclidean distance in normalized RGB space, and normal distance measures the angle between surface normal vectors: $D_n = 1 - p_k \cdot p_j$.

$$D = \sqrt{D_c^2 + \frac{D_s^2}{3R_{seed}^2} + D_n^2}, \tag{1}$$

Supervoxels are grown iteratively, using a local k-means clustering which considers connectivity and flow. The general process is as follows. Beginning at the voxel nearest the cluster center, we flow outward to adjacent voxels and compute the distance from each of these to the supervoxel center using (1). If the distance is the smallest this voxel has seen, its label is set, and using the adjacency graph, we add its neighbors which are further from the center to our search queue for this label. We then proceed to the next supervoxel, so that each level outwards from the center is considered at the same time for all supervoxels. We proceed iteratively outwards until we have reached the edge of the search volume for each supervoxel (or have no more neighbors to check).

Since VCCS operates within an octree structure, additional point clouds can be added directly into the model, meaning it can be used to segment clouds coming from many sensor observations - either using multiple calibrated cameras [11] or by accumulating clouds from one [12]. While previously this was limited to clouds from the same time step or of a static scene, in this work we extend the supervoxel framework to handle changing scenes using a novel algorithm which updates the supervoxels by considering the dynamics of the octree leaves.

### B. Adding Sequential Clouds to an Octree

Adding newly observed points to an existing supervoxel octree is accomplished through a three stage process. First, we must insert the points into the octree, and initialize new leaves for them if they did not exist previously. This

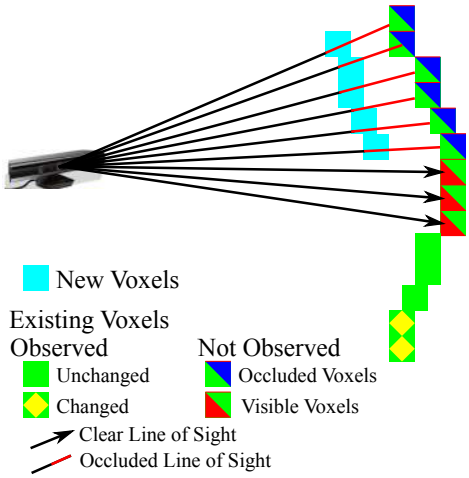


Fig. 2. Categorization of voxels based on new frame of data. Voxels fall into three categories, they are either new, observed or not observed in the frame. Furthermore, observed voxels can either have changed or remained the same, while voxels not observed in the frame are either occluded or no longer exist (in which case they should be deleted).

results in an octree where leaves fall into three possible categories (illustrated in Fig. 2; they are either new, observed, or unobserved in the most recent observation. Handling of new leaves is straightforward; we simply calculate adjacency relations to existing leaves and flag them as unlabeled.

To determine whether a leaf which existed previously has changed, we test the distance between the centroid of the points falling within its voxel (from the new frame) and its previous centroid. This is done in the same feature space used for growing the supervoxels, that is, we test whether the normal, color, and spatial location have varied more than a threshold value. This threshold is set to a relatively low constant value so that it favors false-positives (finding change when there was none), as they do not impact the tracking performance of the algorithm, but only have a slight effect on its run-time. If a leaf is found to have changed, we remove its previous labeling. We also perform a global check to see if more than half of a supervoxels support has changed; if so, we completely remove the supervoxels label from all of its constituent voxels.

Finally, we must consider how to handle leaves which were not observed in the inserted point cloud. Rather than simply prune them, we first check if it was possible to observe them from the viewpoint of the sensor which generated the input cloud. This occlusion check can be accomplished efficiently using the octree by determining if any voxels exist between unobserved leaves and the sensor viewpoint. If a clear line of sight exists from the leaf to the camera, it can safely be deleted. Conversely, if the path is obstructed, we "freeze" the leaf, meaning that it will remain constant until it is either observed or passes the line of sight test in a future frame (in which case, it can be safely deleted). This occlusion testing means that tracking of occluded objects is trivial, as occluded voxels remain in the observations which are used for tracking.

Once the octree voxels have been updated, we then proceed to update the supervoxels as before. That is, first we generate new seeds in regions of large unlabeled voxels, and then conduct the iterative region growing. This results in new supervoxels in regions which are new or changing, while leaving supervoxels in static and occluded regions unchanged. This reduces the tracking and segmentation problem to finding the best joint association of these new supervoxels with those from the prior time-step.

## III. TRACKING SUPERVOXELS

### A. Initialization of Object Models

While ideally one could directly track supervoxels themselves, this is generally not reliable due to the aperture problem seen in neural visual fields [13]; local motion can only be estimated perpendicular to a contour that extends beyond its field of view [14]. This means that in order to properly estimate motion of supervoxels, we must extend our considered field of view significantly beyond the size of the supervoxel itself; in fact, our aperture must contain the borders of the object, otherwise pairwise association of supervoxels is indeterminate.

As such, we first merge supervoxels into contiguous higher level object groupings. For this work, we use a plane fitting and removal algorithm to remove supporting surfaces, followed by a euclidean clustering of the remaining supervoxels as in [15]. It should be stressed that the overall tracking itself is independent of the segmentation used to initialize objects; one could easily use a model-based segmentation, or even a 2D classifier scheme on the original RGB image. Regardless of the segmentation used, the supervoxel clusters found are used to initialize the models which will be tracked.

### B. Tracking with Parallel Particle Filters

Tracking of the segmented models is accomplished using a bank of independent parallel particle filters. The models consist of clouds of supervoxels, and observations are the supervoxels produced using the persistent scheme discussed in Section II. The observation model measures distance in a feature space of spatial distance, normals, color (in HSV space), and labels. Weights of predicted states ( $x, y, z$, roll, pitch, yaw) are measured by associating observed supervoxels with nearest supervoxels from the transformed models, and then measuring total distance in the feature space as (2). That is, the weight $w_i^k$ of particle $i$ belonging to object $k$ (of size $N_k$) with state $x_i$ is the sum of the products of the coherences $W$,

$$w_i^k \sum_{p,q \in x_i} W_d W_{HSV} W_n W_l \quad . \tag{2}$$

Coherences are calculated for each correspondence pair between model supervoxel $p$ and observed supervoxel $q$,

$$\begin{aligned} W_d &= \frac{1}{1+\frac{\|p-q\|^2}{3R_{seed}^2}} \\ W_{HSV} &= \frac{1}{1+\|p_{HS}-q_{HS}\|^2} \\ W_n &= \frac{1}{1+|1-n_p \cdot n_q|} \\ W_l &= \begin{cases} 1, & L_p = L_q \\ \frac{N_k-1}{N_k}, & L_p \neq L_q \end{cases} \end{aligned} , \quad (3)$$

where supervoxel $q$ has label $L_q$, normal $n_q$, and hue & saturation $p_{HS}$.

KLD sampling [16] is used to dynamically adapt the number of particles to the certainty of predictions. As matching supervoxel labels gives a high certainty of a correct prediction, objects which are not moving, and therefore have static supervoxel labels, need very few particles for accurate tracking. Details of the particle filters themselves are beyond the scope of this work, but we refer the reader to [16] for an in-depth description of their operation. For this work, it is sufficient to understand that the particle filters yield independent predictions of 6DoF object state, allowing a transformation of the model to the current time-step - roughly aligning it with the currently observed supervoxels.

### C. Joint Energy Minimization to Associate Supervoxels

The final step in the tracking process is to associate the observed supervoxels to the predictions coming from the particle filters, that is, we need to solve the multiple target data association problem. This is accomplished using an energy minimization which seeks to find an optimal global association of supervoxels to predictions. To do this, we first create a list of all observed supervoxels which lie within a radius $R_{seed}$ of each predicted supervoxel coming from the particle filters (see Fig. 3). Then we determine all supervoxels which could only be associated with one possible object, associate them, and remove them from further consideration.

To associate the remaining observed supervoxels, we determine which objects are competing for them, and then find the predicted supervoxel from each object which lies closest to them in the feature space (using spatial location, normals, and color as in (1)). We adopt a RANSAC-like approach, similar to [9], to sample from the set of possible associations and determine a global association which best aligns the predictions to the observed supervoxels. Additionally, we use a weighted sampling strategy where the likelihood of assigning object $k$ as the label $L$ of supervoxel $q$ falls off with increasing distance from the object centroid $C_k$

$$\mathscr{L}(L_q = k|C_k) = \frac{1}{C_k}. \quad (4)$$

To score a set of assignments, we compute a global energy, given in (5). Each global label association $\mathscr{A}$ consists of local associations $a$ which assign an object label $k$ to each observed supervoxel $q$. The first summation term, $\sum_p \|p_k - q\|$, measures error in feature space between the observed supervoxel and the closest supervoxel in its associated predicted object $p_k$.

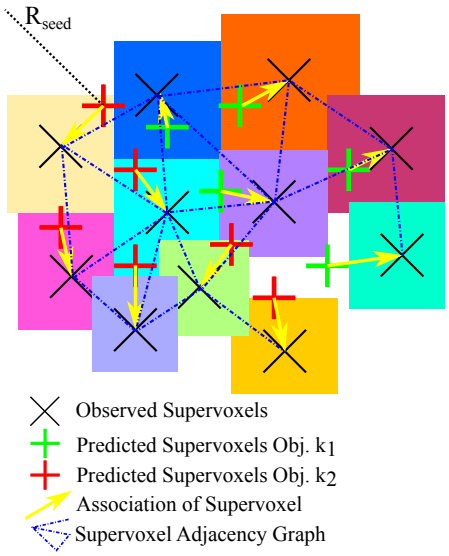


Fig. 3. Association of observed supervoxels with predicted model supervoxels using global energy.

$$E_{\mathscr{A}} = \prod_{a \in \mathscr{A}} \Delta_k \left( \sum_p \|p_k - q\| + \lambda \sum_{(q,q') \in \mathscr{N}} \delta(L_q \neq L_{q'}) \right) \quad (5)$$

The second summation is a smoothing prior which considers the adjacency graph of observed supervoxels. For every observed supervoxel, we compare its assigned label $L_q$ to the label of all supervoxels $q'$ which lie within its adjacency neighborhood $\mathscr{N}$. We adopt the Potts model as in [17], where $\delta()$ is 1 if the specified condition holds, and 0 otherwise, and $\lambda$ is a weighting coefficient which controls the importance given to spatial continuity of labels.

Finally, the multiplicative term $\prod_{a \in \mathscr{A}} \Delta_k$ controls for the expansion or contraction of object volumes through the number of observed supervoxels associated with them. $\Delta_k$ penalizes for changes in volume by increasing the energy for deviations from unity in the ratio of observed supervoxels assigned to an object $\hat{N}_k$ with the number in the object model itself $\hat{N}_k$, that is

$$\Delta_k = \begin{cases} \hat{N}_k / N_k & \text{if } \hat{N}_k \geq N_k \\ 2 - \hat{N}_k / N_k & \text{if } \hat{N}_k < N_k \end{cases}. \quad (6)$$

Once the energy arrives at a stable minimum, we extract the resulting association of observed supervoxels to predicted results, and use them to update the tracked models.

### D. Alignment and Update of Models

The joint energy minimization results in a global association $\mathscr{A}$ which assigns observed supervoxels to tracked objects. In order to use this to update the object models, we determine a transform which aligns it to the internal representation stored by the particle filter. As an initial guess, we use the inverse of the predicted state, and then use an iterative closest point [18] procedure to refine the transform

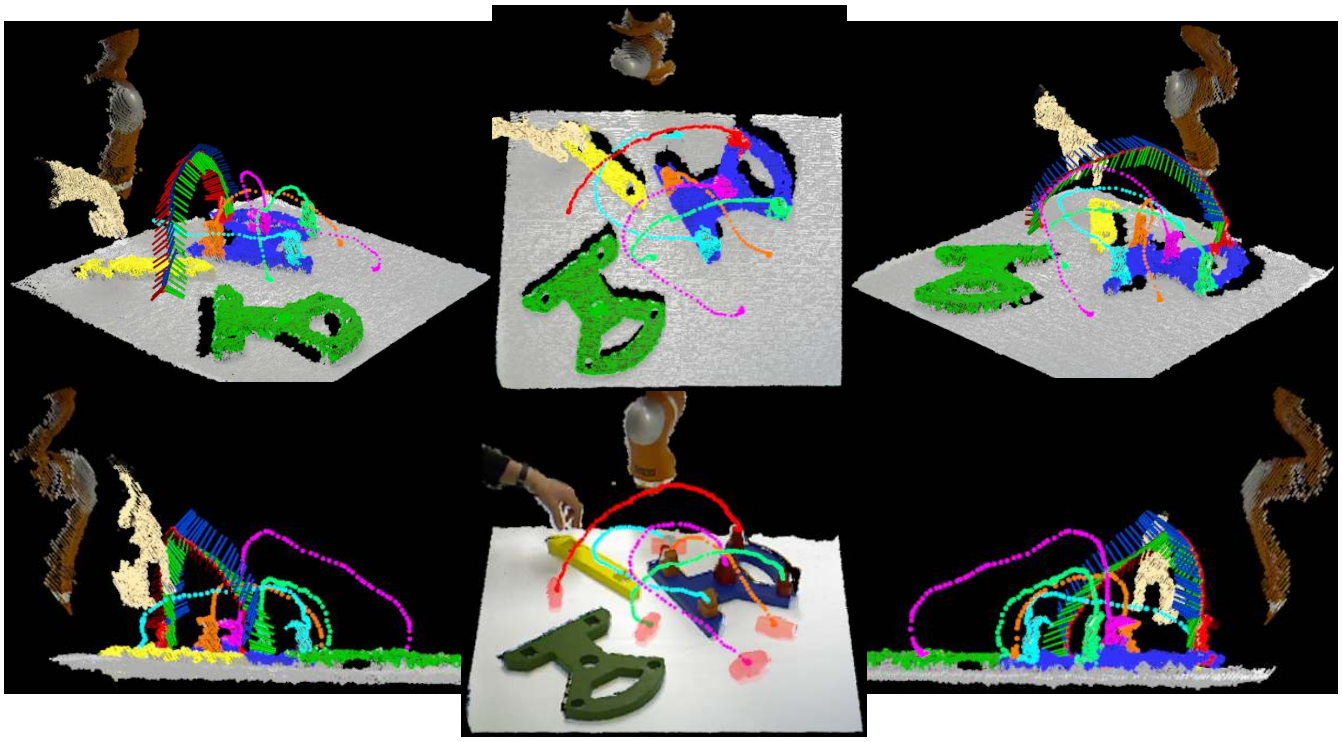

Fig. 4. Result of tracking and segmentation on Cranfield scenario from different views. Here the tracks are shown as dots of the color of the tracked label for each timestep. Initial locations of the pegs are shown in the middle bottom frame as semi-transparent masks. Calculated orientation is shown for the red peg with a set of axes every second time-step; these axes show pose in a frame relative to the start.

such that the set of observed supervoxels best aligns with the model prior. We then replace the model prior with the new observed supervoxels.

As a final step, we use the refined transform to update the states of the particles. To do this, we shift each particle $x_i$ towards the refined state $\hat{x}$, weighting the importance given to the refined state by a constant factor $\varepsilon$

$$x'_{i\in L} = (1-\varepsilon)x_i + \varepsilon\hat{x} \ . \tag{7}$$

For this work, we found that an $\varepsilon$ of 0.5 effectively removes noise (jitter) introduced by the replacement of the tracked model. Additionally, we correct the internal motion model of the particle filters to correspond to the new updated state.

## IV. RESULTS

In order to demonstrate the usefulness of the proposed method, in this Section we provide results from two successful applications. Both applications use the Cranfield scenario [19], a benchmark developed for assessing performance of assembly robot systems. Fig. 4 and the supplementary material[1] show the results of tracking and segmentation (only the pegs are shown in Fig. 4 to avoid clutter) using our Cranfield pieces. It can be seen that the algorithm is able to successfully track 6DoF states through the whole assembly task, even maintaining proper tracks for the pieces when they are fully occluded.

[1]See also http://www.youtube.com/watch?v=0dVzWgW6Bs8 and https://www.youtube.com/watch?v=GjmUhm2JitU for longer versions.

### A. Imitation of Trajectories for Robot Manipulation

The standard way of teaching robots to perform human-like actions is imitation learning, also called programming by demonstration [20], [21]. There are several ways to demonstrate movements: 1) recording movements in joint-space (joint angles) or target-space (Cartesian space) by ways of a motion capture device (requires putting markers on human body), 2) using kinaesthetic guidance (guiding a robot's movements by a human hand), or 3) via teleoperation (controlling a robot via joystick). The only way to obtain motion trajectories from human observation in a "non-invasive" procedure is by using stereo vision [22], however, usually it is model based. The tracking algorithm we have presented here can be used as an alternative method to obtain motion trajectories (in Cartesian space) in a model-free way.

To demonstrate this, we applied our tracking algorithm to obtain human motion trajectories in Cartesian space including orientation of manipulated object (in total six DoFs). We tested it using a recording of the Cranfield scenario where, first, we let a human demonstrate the action and then reproduced it using a KUKA Light Weight Robot (LWR) arm [23]. Specifically, here we imitate a human putting the separator block on the pegs. To generate trajectories for the robot from human demonstrations, we used a modified version of Dynamic Movement Primitives [24], [25] (DMP) and learning method as described in [26]. We used Cartesian impedance control and, thus, generated six DMPs (three for motion of the end-effector in Cartesian space and three for orientation of the hand) based on trajectories obtained from

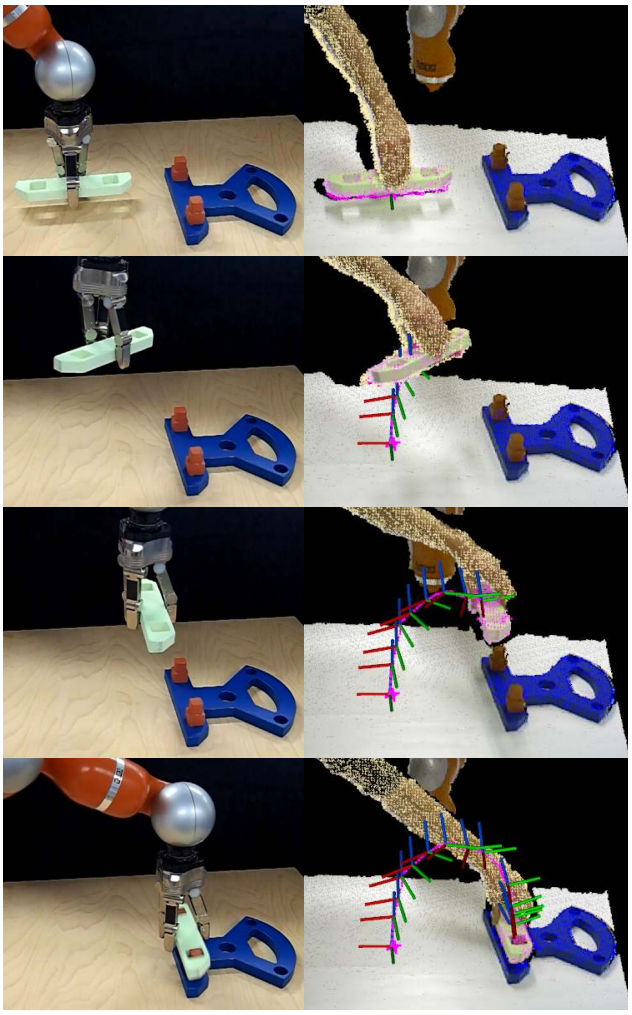

Fig. 5. Kuka LWR arm imitating trajectory and pose learned from tracked human demonstration.

the tracking algorithm. Here we used 100 equally spaced kernels with width $\sigma = 0.05$ for each dimension (for more details please refer to [26]). As demonstrated in Fig. 5 and the supplementary video, trajectories obtained by the proposed tracking algorithm are sufficiently accurate to allow reproduction of the human motion.

### B. Semantic Summaries of Actions

A fundamental task for intelligent autonomous robots is the problem of encoding long chain manipulations in a generic way, for use in tasks such as learning and recognition. As a demonstration of the usefulness of the proposed tracking framework, we use a recently introduced novel Semantic Event Chain (SEC) approach [27] which converts each segmented scene to a graph: nodes represent segment (i.e. object) centers and edges indicate whether two objects touch each other or not. By using an exact graph matching technique the SEC framework discretizes the entire graph sequence into decisive main graphs. A new main graph is identified whenever a new node or edge is formed or an existing edge or node is deleted. Thus, each main graph represents a key frame in the manipulation sequence. Figure 6 shows a few detected sample key frames from the long Cranfield action. While the complete action has in total 1453 frames, the SEC representation reduces it to just 35 key frames, each of which represents a topological change in the scene.

## V. CONCLUSIONS

Robust tracking is a fundamental piece of any intelligent system which seeks to use vision to interact with the world. In order to make a useful link between observations and high-level semantic knowledge, one must be able to coherently track and segment all observed agents and objects. In this work we have presented a method which moves the state of the art towards this goal by ensuring consistent tracking of multiple objects through full occlusions. Additionally, we have shown that associating supervoxels with tracked models in every scene can be done with a global energy minimization, permitting a full segmentation which is consistent with tracked results. This has the advantage of allowing straightforward update of models as well as precise determination of the spatial extent of objects.

We have made source code for the complete tracking and segmentation framework freely available as part of the Point Cloud Library (PCL)[2] so that the community can easily take advantage of it and integrate it in their intelligent systems.

## VI. ACKNOWLEDGEMENTS

Authors would like to thank Mohamad Javad Aein for help with the robot experiment, and Simon Stein for the helpful discussions. The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 (Specific Programme Cooperation, Theme 3, Information and Communication Technologies) under grant agreement no. 270273, Xperience and grant agreement no. 269959, Intellact.

## REFERENCES

[1] B. Babenko, M.-H. Yang, and S. Belongie, "Robust object tracking with online multiple instance learning," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 33, no. 8, pp. 1619–1632, Aug. 2011.

[2] T. Bai, Y. Li, and Y. Tang, "Structured sparse representation appearance model for robust visual tracking," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May, pp. 4399–4404.

[3] P. Azad, D. Munch, T. Asfour, and R. Dillmann, "6-dof model-based tracking of arbitrarily shaped 3d objects," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May, pp. 5204–5209.

[4] D. Mitzel and B. Leibe, "Taking mobile multi-object tracking to the next level: People, unknown objects, and carried items," in *Computer Vision ECCV 2012*, ser. Lecture Notes in Computer Science. Springer Berlin Heidelberg, 2012, vol. 7576, pp. 566–579.

[5] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller, "Multiple hypothesis video segmentation from superpixel flows," in *European Conference on Computer Vision (ECCV)*, ser. Lecture Notes in Computer Science, K. Daniilidis, P. Maragos, and N. Paragios, Eds. Springer Berlin / Heidelberg, 2010, vol. 6315, pp. 268–281.

[6] T. Ma and L. Latecki, "Maximum weight cliques with mutex constraints for video object segmentation," in *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, June, pp. 670–677.

[7] S. Kudo, H. Koga, T. Yokoyama, and T. Watanabe, "Robust automatic video object segmentation with graphcut assisted by surf features," in *Image Processing (ICIP), 2012 19th IEEE International Conference on*, 30 2012-Oct. 3, pp. 297–300.

[2]http://www.pointclouds.org/

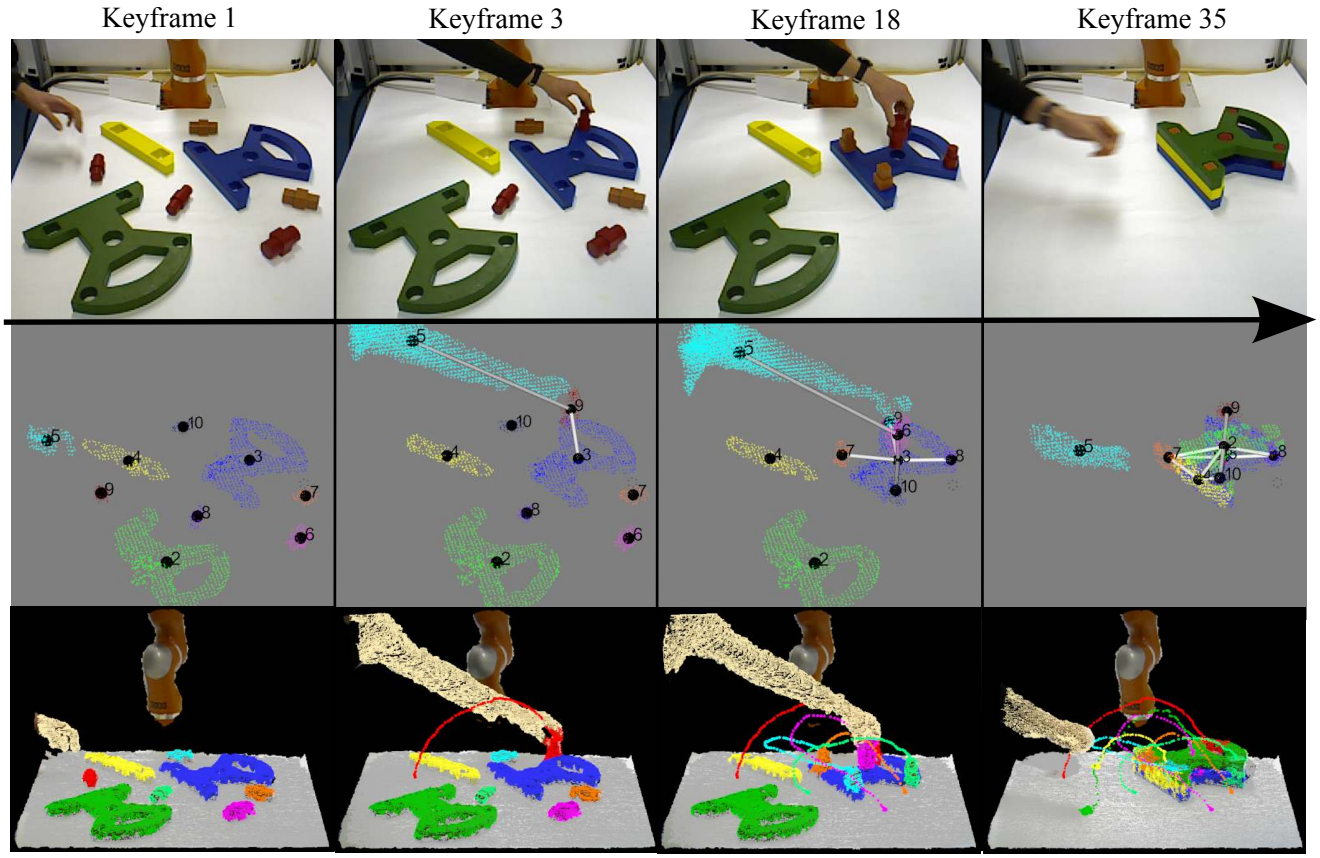


Fig. 6. A few example key frames extracted from the long Cranfield action. Numbered nodes represent interacting objects, while edges show touching relations between objects. Each keyframe represents a topological change in the scene - here we show 4 of the 35 keyframes.


[8] N. Papadakis and A. Bugeau, "Tracking with occlusions via graph cuts," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 33, no. 1, pp. 144 –157, Jan. 2011.

[9] H. Isack and Y. Boykov, "Energy-based geometric multi-model fitting," *International Journal of Computer Vision*, vol. 97, pp. 123–147, 2012.

[10] J. Papon, A. Abramov, M. Schoeler, and F. Wörgötter, "Voxel cloud connectivity segmentation - supervoxels for point clouds," in *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, June 2013.

[11] D. A. Butler, S. Izadi, O. Hilliges, D. Molyneaux, S. Hodges, and D. Kim, "Shake'n'sense: reducing interference for overlapping structured light depth cameras," in *Proceedings of the 2012 ACM annual conference on Human Factors in Computing Systems*, ser. CHI '12. New York, NY, USA: ACM, 2012, pp. 1933–1936.

[12] K. Lai, L. Bo, X. Ren, and D. Fox, "A large-scale hierarchical multi-view rgb-d object dataset," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, may 2011, pp. 1817 –1824.

[13] D. Marr and S. Ullman, "Directional selectivity and its use in early visual processing," *Proceedings of the Royal Society of London. Series B. Biological Sciences*, vol. 211, no. 1183, pp. 151–180, 1981.

[14] S. Shimojo, G. H. Silverman, and K. Nakayama, "Occlusion and the solution to the aperture problem for motion," *Vision research*, vol. 29, no. 5, pp. 619–626, 1989.

[15] R. Rusu and S. Cousins, "3d is here: Point cloud library (pcl)," in *Robotics and Automation (ICRA), 2011 IEEE International Conference on*, May, pp. 1–4.

[16] D. Fox, "Adapting the sample size in particle filters through kld-sampling," *The International Journal of Robotics Research*, vol. 22, no. 12, pp. 985–1003, 2003. [Online]. Available: http://ijr.sagepub.com/content/22/12/985.abstract

[17] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 23, no. 11, pp. 1222–1239, Nov 2001.

[18] D. Chetverikov, D. Stepanov, and P. Krsek, "Robust euclidean alignment of 3d point sets: the trimmed iterative closest point algorithm," *Image and Vision Computing*, vol. 23, no. 3, pp. 299 – 309, 2005.

[19] K. Collins, A. Palmer, and K. Rathmill, "The development of a european benchmark for the comparison of assembly robot programming systems," in *Proceedings of the 1st Robotics Europe Conference, Brussels*, 1984, pp. 27–28.

[20] A. Billard, S. Calinon, R. Dillmann, and S. Schaal, *Survey: Robot Programming by Demonstration*. MIT Press, 2008.

[21] B. Argall, S. Chernova, M. M. Veloso, and B. Browning, "A survey of robot learning from demonstration," *Robots and Auton. Sys.*, vol. 57, no. 5, pp. 469–483, 2009.

[22] F. Hecht, P. Azad, T. Asfour, and R. Dillmann, "Markerless human motion tracking with a flexible model and appearance learning," in *Robotics and Automation (ICRA), 2009 IEEE International Conference on*, 2009.

[23] Kuka Robot Systems. [Online]. Available: http://www.kuka-robotics.com

[24] J. A. Ijspeert, J. Nakanishi, and S. Schaal, "Movement imitation with nonlinear dynamical systems in humanoid robots," in *Robotics and Automation (ICRA), 2002 IEEE International Conference on*, 2002, pp. 1398–1403.

[25] A. Ijspeert, J. Nakanishi, P. Pastor, H. Hoffmann, and S. Schaal, "Dynamical movement primitives: learning attractor models formotor behaviors," *Neural Comput.*, no. 25, pp. 328–373, 2013.

[26] T. Kulvicius, K. J. Ning, M. Tamosiunaite, and F. Wörgötter, "Joining movement sequences: Modified dynamic movement primitives for robotics applications exemplified on handwriting," *IEEE Trans. Robot. Automat.*, vol. 28, no. 1, pp. 145–157, 2012.

[27] E. E. Aksoy, A. Abramov, J. Dörr, K. Ning, B. Dellen, and F. Wörgötter, "Learning the semantics of object-action relations by observation," *The International Journal of Robotics Research*, vol. 30, no. 10, pp. 1229–1249, 2011.