Fusing RGBD Tracking and Segmentation Tree Sampling for Multi-Hypothesis Volumetric Segmentation

Andrew Price* Kun Huang* Dmitry Berenson

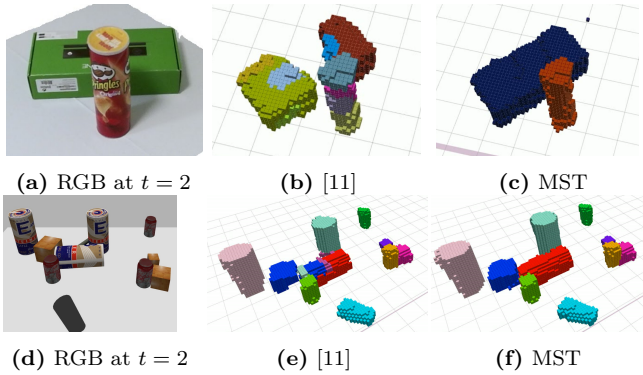
***Abstract*— Despite rapid progress in scene segmentation in recent years, 3D segmentation methods are still limited when there is severe occlusion. The key challenge is estimating the segment boundaries of (partially) occluded objects, which are inherently ambiguous when considering only a single frame. In this work, we propose Multihypothesis Segmentation Tracking (MST), a novel method for volumetric segmentation in changing scenes, which allows scene ambiguity to be tracked and our estimates to be adjusted over time as we interact with the scene. Two main innovations allow us to tackle this difficult problem: 1) A novel way to sample possible segmentations from a segmentation tree; and 2) A novel approach to fusing tracking results with multiple segmentation estimates. These methods allow MST to track the segmentation state over time and incorporate new information, such as new objects being revealed. We evaluate our method on several cluttered tabletop environments in simulation and reality. Our results show that MST outperforms baselines in all tested scenes.**

I. Introduction

Instance segmentation of 3D environments is a crucial problem for robotic manipulation, particularly in tabletop and household scenarios. Fortunately, instance segmentation of images and videos has made rapid progress in recent years, driven by advances in computational capacity, dataset size, and learning algorithm innovations. Even so, current state-of-the-art algorithms [1], [2] will struggle with moderately-complex scenes that are common in manipulation-in-clutter tasks. Similarly, modern semi-supervised video segmentation [3], [4] will be limited by the quality of the initial masks, placing increasing importance on the quality of a single-frame segmentation. Instance segmentation in 3D is even more challenging, requiring either multiview mapping or volumetric shape completion, which are still active areas of research [5], [6], [7], [8], [9].

3D segmentation in cluttered scenes is challenging for several reasons. First, occlusions from objects in the environment and the robot itself can be severe, even entirely occluding some objects. Second, human environments can be highly dynamic and greatly varied, meaning that most scenes will be novel to some degree, precluding the use of model-registration techniques [10].

*Andrew Price and Kun Huang contributed equally to this work. This work was supported in part by NSF grant IIS-1750489 and by Toyota Research Institute (TRI). This article solely reflects the opinions of its authors and not TRI or any other Toyota entity. The authors are with the University of Michigan, Ann Arbor, MI, USA.{huangkun, pricear, dmitryb}@umich.edu



(a) RGB at $t = 2$ **(b)** [11] **(c)** MST

(d) RGB at $t = 2$ **(e)** [11] **(f)** MST

Fig. 1: Qualitative comparison of the volumetric segmentation results obtained directly using SceneCut [1] + [11] and our method (MST) on both real-world and simulation experiments.

While using appropriate priors, high-fidelity sensing and memory, and a physics-based constraints can be helpful, an effective approach also requires a way to explicitly reason about uncertainty in segmentation estimates.

Multi-hypothesis state representations are commonly used in robot navigation to deal with uncertain states and measurements, but they are seldom employed in segmentation. Maintaining a set of hypotheses about a scene can be useful for many manipulation applications, e.g. planning actions which are robust to uncertainty or for active perception. Most importantly, maintaining a set of hypotheses enables an algorithm to consider possibilities which are not currently the most probable, but may be a better basis for estimates when future data is received.

In this work, we present, Multihypothesis Segmentation Tracking (MST), a novel fusion of sampling and tracking methods to perform multi-hypothesis volumetric instance segmentation of cluttered scenes. Specifically, our contributions are 1) A novel Markov Chain Monte Carlo (MCMC) technique for sampling plausible segmentations from a segmentation tree; and 2) A novel approach to fusing tracking and segmentation measurements for multiple segmentation hypotheses. These methods allow us to maintain multiple uncertain but plausible segmentations across time and to incorporate new information, such as a new object being revealed.

Our experiments[1] show that MST outperforms previous work employing single-instant scene segmentation [11] and using video object tracking [4] in generating 3D segmentations. Our code is available open-source[2].

[1] https://youtu.be/kottSLebgBA

[2] https://github.com/UM-ARM-Lab/multihypothesis_segmentation_tracking

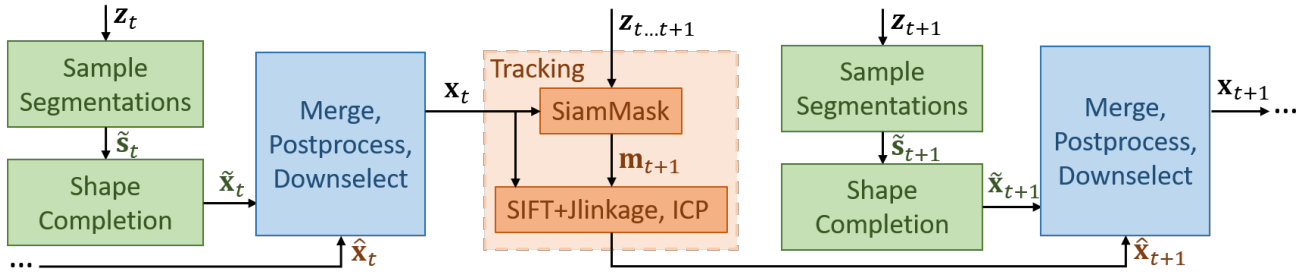


Fig. 2: An overview of the main components of our method. Green: segmentation sampling; Blue: Merging and processing estimates; Orange: Tracking. $\hat{\mathbf{x}}_0 = \emptyset$.

II. Related Work

Instance segmentation of manipulation scenes is a well-studied and rich field [12].However, some limitations occur regularly, including framing the segmentation problem in 2D or 2.5D space [13] (i.e. not estimating full volumetric occupancy), relying on pre-specified [14] or simple geometric models [15] in the scene, or restricting the belief about the scene to a unimodal representation, even if tracking is performed in a multimodal fashion [16]. Working in 3D, as opposed to 2D or 2.5D, is particularly important, as it allows us to construct and retain object geometry estimates in the presence of occlusion, which is frequent in cluttered scenes. In this work, we aim to embrace the challenges of manipulating unknown objects in real environments by addressing the problem in its native 3D occupancy space, acknowledging that our segmentation and tracking tools are imperfect by considering multiple hypotheses of the segmentation state.

Scene segmentation has been addressed in a variety of ways, and though we do not attempt a complete taxonomy here, several broad classes of techniques can be found in the literature. With the development of deep learning object detection, some approach the instance segmentation problem by treating it as object detection [17], [18]. Another approach is to localize known objects within the scene, providing 3D knowledge from prior geometric information [19], [20], [21]. While very powerful in certain contexts, these two approaches scale poorly to handling general household manipulation scenes with novel object categories.

With the growth in deep networks over the past decade, there has been significant work in producing an occupancy grid from a single depth image or a stitched 3D scene [22], [23], [24], [25], [26], [27], [28]. While there are some similarities between this problem and ours, there are also significant differences: these methods operate on sequences of images of static scenes (as opposed to the dynamic ones we consider), and while the outputs may have an associated probability, they do not provide multiple segmentation hypotheses, as we aim to do here.

One application domain that has seen significant work on voxel-based segmentation and classification is medical imaging, since techniques like MRI or CT offer true 3D imaging instead of the 2.5D of RGBD sensors. Uses include mapping airways [29] and brain tissue [30], [31], [32], with techniques like graph cuts, KNN, and deep networks drawn from the broader perception community, plus hand-tuned heuristic ones drawn from biology domain knowledge. In our work, we do not assume true 3D information is available and must rely on RGBD images.

Related work in sensor fusion via particle filter, which has been commonly used for non-linear state estimation, has seen various improvements on sampling sufficient valid states and avoiding degeneracy of the proposal distribution [33], [34]. Although these probabilistic methods have been used in manipulation [13], [14], they either only produce 2D estimates, or require prior knowledge of object models.

III. Problem Statement

Let $v \in \mathbb{N}^3$ denote the coordinate of one voxel. $K \in \mathbb{N}$ is the estimated number of objects in the region of interest. Let $k \in \mathcal{K} \doteq \{0, 1, 2, ..., K\}$ denote the object label of v where $k = 0$ means the voxel is in free space. Our segmentation state vector $\mathbf{x}$ is an assignment from a voxel coordinate to an object label. Similarly, an image segmentation $\mathbf{s}$ assigns a label to each pixel.

$$\mathbf{x}\colon \mathcal{V} \to \mathcal{K}, \qquad \mathbf{s}\colon \mathcal{P} \to \mathcal{K} \tag{1}$$

For a multi-hypothesis system evolving over discretized time, we will use $t \in \{1, ..., T\}$ to indicate the time index, $i \in \{1, ..., N\}$ to indicate the hypothesis index, and k to indicate an object index. So, a single object hypothesis at a point in time can be written as ${}^{k}o_t^i \doteq \{ v \in \mathcal{V} \mid {}^{v}\mathbf{x}_t^i = k \}$.

To compute the similarity of two segmentations, we can define a match quality $q\colon \mathbf{X} \times \mathbf{X} \to [0, 1]$ where $q = 1$ represents a perfect match between two segmentation states. For this work, use a symmetric version of the weighted coverage [35]:

$$q(\mathbf{x}^i, \mathbf{x}^j) \doteq \tfrac{1}{2} C(\mathbf{x}^i, \mathbf{x}^j) + \tfrac{1}{2} C(\mathbf{x}^j, \mathbf{x}^i), \tag{2}$$

$$C(\mathbf{x}^i, \mathbf{x}^j) \doteq \frac{1}{|\mathcal{V}|} \sum_{m \in \mathcal{K}^i} \left| {}^{m}o^i \right| \max_{\alpha} J({}^{m}o^i, {}^{\alpha}o^j) \tag{3}$$

where $J({}^{m}o, {}^{n}o) \doteq \frac{|{}^{m}o \cap {}^{n}o|}{|{}^{m}o \cup {}^{n}o|}$ represents the intersection over union (IOU, or Jaccard distance). The symmetric weighted coverage accounts for both false positives and false negatives in the volumetric segmentation. We do not

Fig. 3: Sampled tree cuts and their resulting image segmentations. The segmentation trees have been truncated for clarity; the full size is approximately 1000 nodes.

Algorithm 1 Metropolis-Hastings Segmentation Sampler

Input: RGBD image $\mathbf{z}$, number of samples N, autocorrelation steps a
Output: Weighted sample set $\{ \langle r^i, \tilde{\mathbf{s}}^i \rangle \}$

$\tau \leftarrow \text{COB}(\mathbf{z})$
$c_t \leftarrow c^* \leftarrow \textsc{SceneCut}(\tau, \mathbf{z})$
for $i = 1$ to N **do**
  for $j = 1$ to a **do**
    $k \leftarrow \textsc{Rand}\{1, \ldots, |c_t|\}$
    $c' \leftarrow \textsc{MoveNode}(c_t[k], \textsc{Rand}\{\textsc{Up}, \textsc{Down}\})$
    $\alpha \leftarrow \min(\frac{p(c')}{p(c_t)} \frac{g(c_t, c')}{g(c', c_t)}, 1)$
    if $\textsc{Rand}(0, 1) < \alpha$ **then**
      $c_t \leftarrow c'$
    end if
  end for
  $\langle r^i, \tilde{\mathbf{s}}^i \rangle \leftarrow \langle p(c_t), \textsc{Apply}(c_t, \tau) \rangle$
end for
return $\{ \langle r^i, \tilde{\mathbf{s}}^i \rangle \}$

use the Precision-Recall curve, which is a common metric for object detection and instance segmentation, because our goal is to segment the entire scene.

Given a sequence of RGBD images $\mathbf{z}_{0\ldots T}$ representing an object manipulation sequence, we wish to produce a set of N diverse 3D segmentations, which are consistent with $\mathbf{z}_{0\ldots T}$.

IV. Approach

A. Overview

MST follows this basic outline: observe the static scene and sample possible 3D segmentations, observe an interaction with the scene and estimate the rigid motions via tracking, then combine the tracked prior segmentations with new samples directly from the subsequent static scene. The combination of segmentation hypotheses is performed by computing which object hypotheses conflict with one another, sampling a set of merges and splits between these objects, and resampling this new population according to their current and historical fitness. The overall process is illustrated in Figure 2 and described in detail below.

B. Segmentation Sampling

Because a single image segmentation is unlikely to be exactly correct, we wish to generate a weighted collection of segmentation hypotheses from a sensor measurement. Let the sampling procedure be defined as $\textsc{Sample}: \mathbf{z}_t \mapsto \{ \langle r_t^i, \tilde{\mathbf{s}}_t^i \rangle \}$, with weight r_t^i representing the quality of the sample i and operator $\tilde{\cdot}$ indicating "sampled". We begin with the segmentation tree τ (called an Ultrametric Contour Map [UCM] [36]) generated by the Convolutional Oriented Boundaries [37] algorithm, which we then walk in a Metropolis-Hastings manner [38]. The segmentation tree has nodes representing contiguous regions of the image, with the root node containing the whole image, and the children of a node representing a partition of that image region, such that the leaf nodes represent atomic "superpixel" regions. A "cut" c of this tree is a collection of nodes separating the root from the leaves, and representing a possible partition of the full image. The SceneCut [1] algorithm assigns a value to a particular tree cut as $v(c)$, and computes the optimal cut $c^* = \max_c v(c)$.

A Metropolis-Hastings sampler is a stateful, random-walk approach to generating random samples from a distribution whose value can be computed for a given state x, but which can't be sampled from directly. M-H sampling is powerful in part because it handles its inputs in a black box fashion: the details of $v(c)$ are irrelevant to the sampler as long as we can provide a the two inputs to the algorithm: the proposal distribution $g(x_t, x')$ and the posterior distribution $p(x)$.

Our M-H sampler is described in Alg. 1. We define our M-H posterior as $p(c) \sim \exp(-(v(c^*) - v(c))^2/\sigma^2)$, and the M-H proposal distribution $g(c_t, c')$ by selecting a node in the cut c_t uniformly at random and moving the cut at that node up or down the tree to generate a proposed cut c'. The σ parameter controls how likely we are to consider lower-scoring segmentations. We calculate the acceptance ratio $\alpha = \frac{p(c')}{p(c_t)} \frac{g(c_t, c')}{g(c', c_t)}$, and accept the proposal with probability $\min(1, \alpha)$, in the usual fashion. The process repeats until n samples have been generated, with options to insert steps for burn-in and to reduce autocorrelation. Employing an MCMC approach means we do not need to estimate the probability of an individual

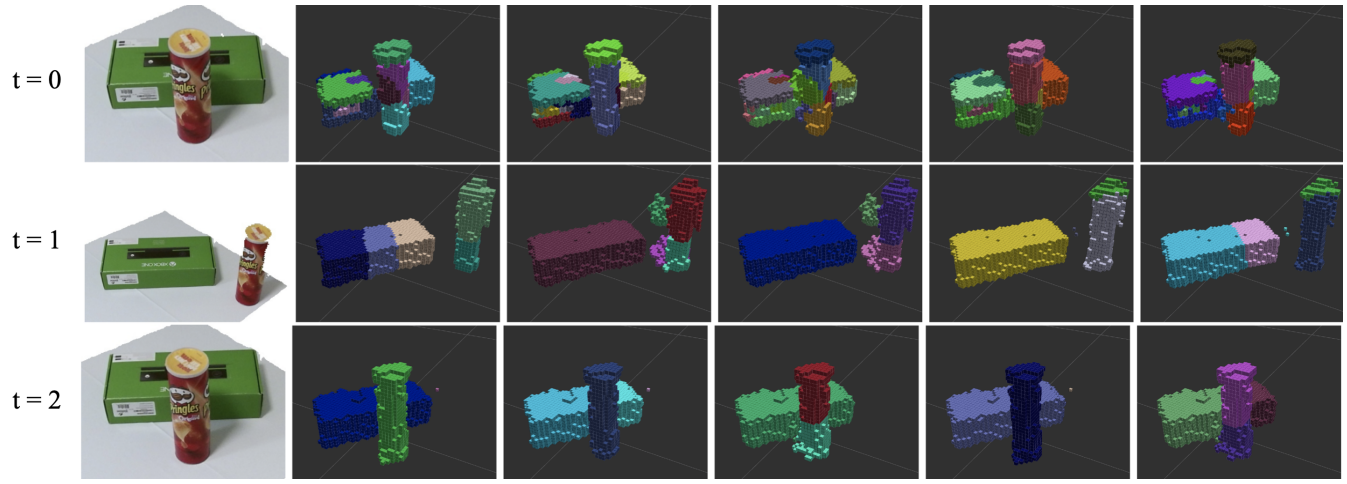


Fig. 4: Real-world experiment R1 with 5 hypotheses shows the convergence of hypotheses toward ground truth. Hypotheses at $t = 0$ are initialized with segmentation sampling and shape completion. Our final hypotheses $\mathbf{x}_t$ are shown for $t = 1$ and $t = 2$ in decreasing order of weight w_t^m.

node as in [39], [40], which is especially useful as the SceneCut results suggest that a Ultrametric Contour Map node's segmentation probability is better understood in the context of its neighbors than standalone. To our knowledge, this the first MCMC approach for generating segmentations from a segmentation tree, and the first segmentation sampler using the SceneCut quality metric. Some example segmentations sampled by our method are shown in Figure 3.

1) Shape Completion and Imaging: Given an image segmentation of the RGBD measurement, we can use shape completion COMPLETE [11] to estimate the occupancy of occluded voxels, which takes each 2.5D segment as the input and reconstructs the 3D shape using a deep neural network. The inverse operation PROJECT projects a 3D segmentation to a 2D one using ray-casting (given camera intrinsics and extrinsics), and is used when evaluating scenes for which we have no 3D ground truth.

$$\text{COMPLETE}\colon \tilde{\mathbf{s}}_t^i \mapsto \tilde{\mathbf{x}}_t^i, \qquad \text{PROJECT}\colon \mathbf{x}_t^i \mapsto \mathbf{s}_t^i \quad (4)$$

The completion function is run for every 2D segment in the sampled segmentations.

C. Tracking

Static scenes can contain ambiguities that can be challenging even for human observers, so we introduce motion into the scene to assist in differentiating object boundaries. We use video/object tracking to compute rigid body trajectories $\boldsymbol{\xi}\colon \mathcal{K} \to \mathbb{SE}(3)$ for each segment from an observation sequence $\mathbf{z}_{t\ldots t+1}$ (see Alg. 2, where $\odot$ denotes element-wise multiplication). The process consists of two steps: object mask tracking and transform estimation.

1) Video Object Tracking: We employ the state-of-art video object tracking algorithm SiamMask [4] to determine the correspondence of objects between the frames at t and $t+1$. The output mask video from SiamMask will be further utilized in rigid body transformation estimation of each object. SiamMask requires the bounding box of a target object as input, so we first project each volumetric representation $\mathbf{x}_t$ onto a 2-D segmentation image by PROJECT, then compute a bounding box for each unique label in the new segmentation image. For each object ${}^k o_t^i$, we apply SiamMask to the recorded video between t and $t+1$ and its corresponding bounding box, generating the single-frame mask ${}^k\mathbf{m}_{t+1}^i$ of the object ${}^k o_{t+1}^i$.

2) Rigid Body Transformation Estimation: In order to estimate the motion $\boldsymbol{\xi}$ of every object, we first compute the rigid body transformation by matching SIFT [41] key points in ${}^k\mathbf{m}_t^i$ to those in ${}^k\mathbf{m}_{t+1}^i$. However, there are sometimes erroneously-matched features because some objects have multiple similar SIFT key points. We thus use JLinkage [42], a clustering algorithm which is able to

Algorithm 2 RGBD Object Tracker

Input: RGBD video $\mathbf{z}_{t\ldots t+1}$, $\mathbf{x}_t$
Output: $\boldsymbol{\xi}_t\colon \mathcal{K} \to \mathbb{SE}(3)$

PROJECT : $\mathbf{x}_t \mapsto \mathbf{s}_t$
for $k = 0$ to n **do**
  ${}^k\mathbf{m}_{t+1} \leftarrow \text{SIAMMASK}({}^k\mathbf{m}_t, \mathbf{z}_{t\ldots t+1})$
  Source Point Cloud $\mathbf{P_{src}} \leftarrow \mathbf{z}_t \odot {}^k\mathbf{m}_t$
  Target Point Cloud $\mathbf{P_{tar}} \leftarrow \mathbf{z}_{t+1} \odot {}^k\mathbf{m}_{t+1}$
  Feature displacements $\{\mathbf{d}_j\} \leftarrow \text{SIFT}(\mathbf{P_{src}}, \mathbf{P_{tar}})$
  Rigid body transform $\mathbf{T} \leftarrow \text{JLINKAGE}(\{\mathbf{d}_j\})$
  if $\text{NUMBERINLIERS}(\{\mathbf{d}_j\}, \mathbf{T}) > \text{thres}_{SIFT}$ **then**
    ${}^k\boldsymbol{\xi}_t \leftarrow \mathbf{T}$
    continue
  end if
  $\mathbf{T} \leftarrow \text{ICP}(\mathbf{P_{src}}, \mathbf{P_{tar}})$
  if $\text{FITERROR}(\mathbf{P_{src}}, \mathbf{P_{tar}}, \mathbf{T}) < \text{thres}_{ICP}$ **then**
    ${}^k\boldsymbol{\xi}_t \leftarrow \mathbf{T}$
  else
    ${}^k\boldsymbol{\xi}_t \leftarrow \mathbf{I}$
  end if
end for
return $\boldsymbol{\xi}_t$

Fig. 5: Initial configurations of simulated experiments S1, S2, and S3, and real-world experiments R1, R2.

handle outliers (similar to RANSAC + Hough voting), to obtain a candidate rigid body transformation from the SIFT matches.

SIFT works well on objects with distinguishable feature points, but it works poorly on textureless objects. As a fallback, we also compute the transform between the parts of the point cloud corresponding to ${}^{k}\mathbf{m}_t^i$ and ${}^{k}\mathbf{m}_{t+1}^i$ using Iterative Closest Point (ICP). We use the best fit of JLinkage or ICP to estimate $\boldsymbol{\xi}$. If both methods fail, a transform of identity is assumed. This approach performs well when the tracked object is occluded by an object pushed in front of it, but struggles when the tracking failure is due to erroneous masking.

D. Merging in New Information

While tracking can be effective, some events cannot be represented by our transforms $\boldsymbol{\xi}$, such as previously-unseen objects being revealed between t and $t+1$, so we need another way to update our segmentation estimates to accord with new information. Related work in particle filtering has considered an analogous problem: Manifold particle filters [33] use the measurement step to perform a projection $\pi\colon \mathbf{X}^{\mathbf{c}} \to \partial\mathbf{X}$ carrying invalid (or highly improbable) states to a nearby valid one. Evolutionary particle filters [34] forego the default resampling process in favor of a genetic algorithm style crossover operation to update the particle population. We employ the first technique using *free space refinement*: voxels believed to be in free space based on $\mathbf{z}_t$ are cleared for all $\hat{\mathbf{x}}_t^i$. We also construct a quality function $f_r(\hat{\mathbf{x}}_t^i) = \left|\hat{\mathbf{x}}_t^i\right|_{after} / \left|\hat{\mathbf{x}}_t^i\right|_{before}$ describing the level of error corrected by refinement.

We propose a novel approach to accommodate situations where objects have appeared in the scene, inspired by [34]. We first generate a new set of samples $\tilde{\mathbf{x}}_t^i$ which we then combine with the $\hat{\mathbf{x}}_t^j$ generated by tracking to form the final estimate $\mathbf{x}_t$. The procedure is as follows: For a pair (i, j) of segmentations between the predicted and newly sampled populations, we generate a conflict graph $G(n, e)$ where nodes n represent objects in ${}^{k}\tilde{\mathbf{x}}_t^i$ or ${}^{\ell}\hat{\mathbf{x}}_t^j$, and edges e with weights w_e representing the voxel IOU between the objects. Thus, objects that are newly occluded or disoccluded will have no conflicts, and will be inserted directly into the resulting state.

For each connected component in G, we need to decide whether to merge the competing objects, or whether to separate them with only one object "winning" for a given voxel. Since it is unclear which hypothesis to favor when merging/separating, we sample over the possibilities to preserve diversity in our set of hypotheses: We induce a precedence order by applying a random edge

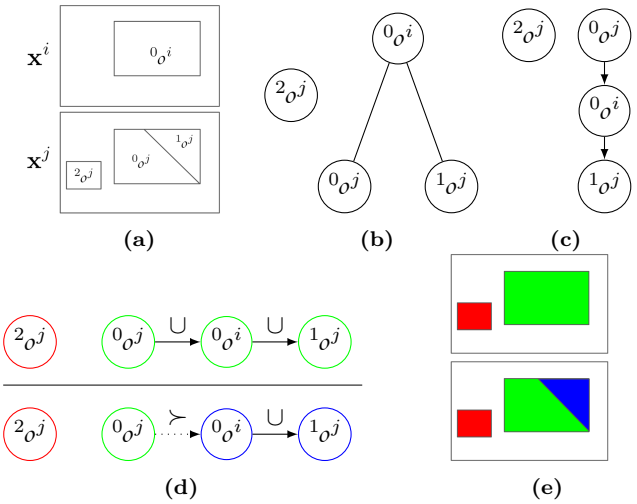


Fig. 6: An illustration of the merging process in 2D. (a) Two segmentations to be merged. (b) The conflict graph. (c) A randomly sampled topological order, in which parent nodes will overwrite child nodes. (d) Two different merge samples. Operator $A \succ B$ here means that segments A and B will not be merged, and labels from A will have the higher precedence according to the ordering. Nodes of the same color have been merged ($\cup$). (e) The two resulting segmentations.

orientation and topological ordering to the component, then sample merges from node n_1 to its parent node n_2 with probability $w_e|n_2|/|n_1|^2$, so that we can get similar probability for merging and separating. A consensus-thresholded mode filter, which replaces voxels with the most frequently occurring voxel value selected from a certain window size, is applied to postprocess the state, cleaning up cases where only the periphery of an object survived the merger (usually due to a small misalignment between the prediction and new sample). Applying η samples of this procedure to all pairs (i, j) produces a population much larger than the desired number of hypotheses, so a weight $w_{t+1}^m = w_t^j r_{t+1}^i f_r(\hat{\mathbf{x}}_{t+1}^j)^\lambda$ is used to downsample the population back down to the desired size, with selection probability for object $m \propto w_{t+1}^m$. The hyper-parameter λ represents the relative trust in tracking results.

V. Experiments and Results

To evaluate the performance of MST it would be ideal to compare to methods that perform volumetric segmentation in changing scenes. The closest approach we are aware of is MID-Fusion [5], a volumetric dynamic SLAM algorithm, but we are unaware of an open-source implementation. We test in three simulation and two real-world tabletop scenes. In simulation all approaches are evaluated by comparing to the volumetric ground

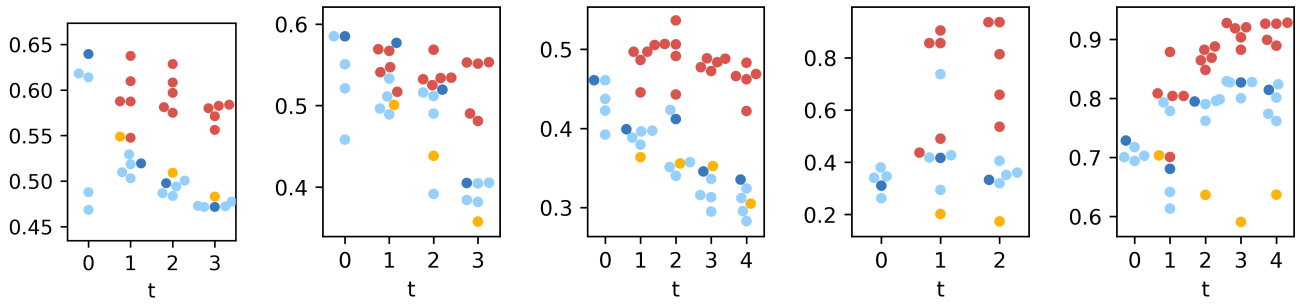


Fig. 7: Left to right: coverage quality q of hypotheses from various algorithms vs. ground truth for experiments S1, S2, S3, R1, and R2. Dark blue: single-frame segmentation [11] baseline; Light blue: segmentation samples $\tilde{\mathbf{x}}$; Orange: tracking-only baseline [4]; Red: set of hypotheses $\mathbf{x}$ produced by MST. Higher is better. (Note the recovery from the sampler's oversegmentation bias in R1-2.)

truth according to the q function above (Equation (2)), while the real-world results are evaluated with respect to a hand-labeled 2D image segmentation ground truth. Figure 5 shows the initial conditions of all experiments. We use voxels of size 1cm, and a workspace of 1m x 1m x 0.5m containing 2-20 objects. For algorithm parameters, we set sampling $\sigma = 0.25v(c^*)^2$, $\lambda = 3$, $N = 5$, $\text{thres}_{SIFT} = 5$ and $\text{thres}_{ICP} = 1\text{mm}$ in all experiments. See the accompanying video for visualizations of results.

A. Simulation Experiments

To evaluate the system with known 3D ground truth segmentation results, we created simulated scenes in Gazebo, then used a simulated robot to push and pull objects. The robot performed 3-4 manipulation actions, generated via the technique described in [11], pausing to observe the scene between each. Experiments S1, S2, and S3 demonstrate increasing degrees of clutter, with duplicated objects contacting and occluding one another.

Figure 7 shows the quality of segmentations over time, with the results of our method clearly outperforming baselines. Of particular note is the generally decreasing quality trends in S1 and S2. Due to the difficulty of simulating grasping in Gazebo and the robot's limited dextrous workspace, many of the sampled robot actions push the objects into tighter configurations with greater occlusion and increased segmentation difficulty. By retaining some of the information from earlier, less ambiguous scenes, our approach is able to maintain higher quality over time.

B. Real-world Experiments

There are significant sensing differences between simulation and physical environments, particularly involving sensor noise, lighting, and object texture, so we have also evaluated our system on real tabletop environments utilizing objects from the YCB dataset [43]. Experiment R1, detailed in Figure 4, is an illustrative example showing how performing the estimation in 3D can help recover from poor initial segmentations. In particular, R1 shows that the un-tuned segmentation is biased towards oversegmentation of the initial scene, but MST is able to recover from this by incorporating the object motion. Experiment R2 shows a more challenging and realistic scene in which objects are pushed from one cluster to another. Performance results are shown in Figure 7: MST strongly outperforms the baselines because of its ability to fuse hypotheses in 3D and retain them in the presence of occlusion.

Currently, the pipeline implementation has significant runtime, requiring ∼1 hour to process a sequence of images from one of the experiments described above. The primary bottleneck is currently serialized calls to the dense video tracking, which must be done for every segment in every hypothesis at every time step. The redundant calls can be reduced and they could be parallelized with additional engineering effort, and there are many opportunities for GPU implementations of the occupancy operations.

VI. Conclusions

In this work, we have shown how our method for multi-hypothesis volumetric estimation can outperform single-frame scene segmentation and tracking-propagated segmentation, particularly in cluttered manipulation scenarios. To do so, we have introduced novel techniques for sampling different segmentations and for combining them after perceived motions. Experiments in simulated and real environments show that this technique is promising for tabletop manipulation in challenging scenes. However, there remain significant opportunities to improve the system. As with any serialized data pipeline, individual improvements to the accuracy of any one stage will improve the overall performance, just as egregious errors will derail an otherwise reasonable segmentation hypothesis. Thus advances in image segmentation, shape completion, and 3D registration will boost the accuracy of the system overall. Second, there are significant engineering gains to be made by improving the parallelism at all levels of the system, from voxels to hypotheses. Third, improved use of physical understanding of the scene could allow for more accurate transition predictions, enhancing our ability to resolve voxel conflicts. Despite these avenues for future improvement, we feel these results provide a case in favor of explicitly modeling and propagating uncertainty in sequential instance segmentation.

References

[1] Trung T Pham, Thanh-Toan Do, Niko Sünderhauf, and Ian Reid. Scenecut: joint geometric and object segmentation for indoor scenes. In *ICRA*. IEEE, 2018.

[2] Christopher Xie, Yu Xiang, Arsalan Mousavian, and Dieter Fox. The best of both modes: Separately leveraging rgb and depth for unseen object instance segmentation. In *CoRL*, pages 1369–1378. PMLR, 2020.

[3] Sergi Caelles, Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Laura Leal-Taixé, Daniel Cremers, and Luc Van Gool. One-shot video object segmentation. In *CVPR*, 2017.

[4] Qiang Wang, Li Zhang, Luca Bertinetto, Weiming Hu, and Philip HS Torr. Fast online object tracking and segmentation: A unifying approach. In *CVPR*, 2019.

[5] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger. Mid-fusion: Octree-based object-level multi-instance dynamic slam. In *ICRA*, pages 5231–5237, 2019.

[6] Abhishek Venkataraman, Brent Griffin, and Jason J Corso. Kinematically-informed interactive perception: Robot-generated 3d models for classification. *arXiv preprint arXiv:1901.05580*, 2019.

[7] David Schiebener, Jun Morimoto, Tamim Asfour, and Aleš Ude. Integrating visual perception and manipulation for autonomous learning of object representations. *Adaptive Behavior*, 21(5):328–345, 2013.

[8] Shubham Tulsiani, Saurabh Gupta, David F. Fouhey, Alexei A. Efros, and Jitendra Malik. Factoring shape, pose, and layout from the 2d image of a 3d scene. In *CVPR*, June 2018.

[9] J. Lundell, F. Verdoja, and V. Kyrki. Robust grasp planning over uncertain shape completions. In *IROS*, pages 1526–1532, 2019.

[10] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *PAMI*, 14(2):239–256, 1992.

[11] Andrew Price, Linyi Jin, and Dmitry Berenson. Inferring occluded geometry improves performance when retrieving an object from dense clutter. In *ISRR*, 2019.

[12] Jeannette Bohg, Karol Hausman, Bharath Sankaran, Oliver Brock, Danica Kragic, Stefan Schaal, and Gaurav S Sukhatme. Interactive perception: Leveraging action in perception and perception in action. *IEEE TRO*, 33(6):1273–1291, 2017.

[13] Herke Van Hoof, Oliver Kroemer, and Jan Peters. Probabilistic segmentation and targeted exploration of objects in cluttered environments. *IEEE TRO*, 30(5):1198–1209, 2014.

[14] Zhiqiang Sui, Zheming Zhou, Zhen Zeng, and Odest Chadwicke Jenkins. Sum: Sequential scene understanding and manipulation. In *IROS*, pages 3281–3288. IEEE, 2017.

[15] Tonci Novkovic, Remi Pautrat, Fadri Furrer, Michel Breyer, Roland Siegwart, and Juan Nieto. Object finding in cluttered scenes using interactive perception. *arXiv preprint arXiv:1911.07482*, 2019.

[16] Karol Hausman, Ferenc Balint-Benczedi, Dejan Pangercic, Zoltan-Csaba Marton, Ryohei Ueda, Kei Okada, and Michael Beetz. Tracking-based interactive segmentation of textureless objects. In *ICRA*, pages 1122–1129. IEEE, 2013.

[17] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick. Mask r-cnn. In *ICCV*, pages 2961–2969, 2017.

[18] Xinlong Wang, Tao Kong, Chunhua Shen, Yuning Jiang, and Lei Li. Solo: Segmenting objects by locations. *arXiv preprint arXiv:1912.04488*, 2019.

[19] Yu Xiang, Tanner Schmidt, Venkatraman Narayanan, and Dieter Fox. Posecnn: A convolutional neural network for 6d object pose estimation in cluttered scenes. *arXiv preprint arXiv:1711.00199*, 2017.

[20] Andy Zeng, Kuan-Ting Yu, Shuran Song, Daniel Suo, Ed Walker, Alberto Rodriguez, and Jianxiong Xiao. Multi-view self-supervised deep learning for 6d pose estimation in the amazon picking challenge. In *ICRA*, pages 1386–1383. IEEE, 2017.

[21] Max Schwarz, Anton Milan, Arul Selvam Periyasamy, and Sven Behnke. Rgb-d object detection and semantic segmentation for autonomous manipulation in clutter. *IJRR*, 37(4-5):437–451, 2018.

[22] Ji Hou, Angela Dai, and Matthias Niessner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2019.

[23] Zhirong Wu, Shuran Song, Aditya Khosla, Fisher Yu, Linguang Zhang, Xiaoou Tang, and Jianxiong Xiao. 3d shapenets: A deep representation for volumetric shapes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2015.

[24] D. Maturana and S. Scherer. Voxnet: A 3d convolutional neural network for real-time object recognition. In *IROS*, pages 922–928, 2015.

[25] Charles R. Qi, Hao Su, Matthias Niessner, Angela Dai, Mengyuan Yan, and Leonidas J. Guibas. Volumetric and multi-view cnns for object classification on 3d data. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2016.

[26] Gernot Riegler, Ali Osman Ulusoy, and Andreas Geiger. Octnet: Learning deep 3d representations at high resolutions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, July 2017.

[27] Yida Wang, David Joseph Tan, Nassir Navab, and Federico Tombari. Forknet: Multi-branch volumetric semantic completion from a single depth image. In *ICCV*, October 2019.

[28] Hsien-Yu Meng, Lin Gao, Yu-Kun Lai, and Dinesh Manocha. Vv-net: Voxel vae net with group convolutions for point cloud segmentation. In *ICCV*, October 2019.

[29] Pechin Lo, Jon Sporring, Haseem Ashraf, Jesper J.H. Pedersen, and Marleen de Bruijne. Vessel-guided airway tree segmentation: A voxel classification approach. *Medical Image Analysis*, 14(4):527 – 538, 2010.

[30] Fedde van der Lijn, Tom den Heijer, Monique M.B. Breteler, and Wiro J. Niessen. Hippocampus segmentation in mr images using atlas registration, voxel classification, and graph cuts. *NeuroImage*, 43(4):708 – 720, 2008.

[31] Scott Quadrelli, Carolyn Mountford, and Saadallah Ramadan. Hitchhiker's guide to voxel segmentation for partial volume correction of in vivo magnetic resonance spectroscopy. *Magnetic Resonance Insights*, 9:MRI.S32903, 2016. PMID: 27147822.

[32] Tanya Nair, Doina Precup, Douglas L. Arnold, and Tal Arbel. Exploring uncertainty measures in deep networks for multiple sclerosis lesion detection and segmentation. *Medical Image Analysis*, 59:101557, 2020.

[33] Michael C Koval, Matthew Klingensmith, Siddhartha S Srinivasa, Nancy S Pollard, and Michael Kaess. The manifold particle filter for state estimation on high-dimensional implicit manifolds. In *ICRA*, pages 4673–4680. IEEE, 2017.

[34] N. M. Kwok, Gu Fang, and W. Zhou. Evolutionary particle filter: Re-sampling from the genetic algorithm perspective. In *IROS*, pages 2935–2940, 2005.

[35] Nathan Silberman, David Sontag, and Rob Fergus. Instance segmentation of indoor scenes using a coverage loss. In *ECCV*, pages 616–631. Springer, 2014.

[36] Pablo Arbelaez. Boundary extraction in natural images using ultrametric contour maps. In *CVPR Workshop*, pages 182–182. IEEE, 2006.

[37] Kevis-Kokitsi Maninis, Jordi Pont-Tuset, Pablo Arbeláez, and Luc Van Gool. Convolutional oriented boundaries: From image segmentation to high-level tasks. *PAMI*, 40(4):819–833, 2018.

[38] W. K. Hastings. Monte Carlo sampling methods using Markov chains and their applications. *Biometrika*, 57(1):97–109, 04 1970.

[39] Shell X. Hu, Christopher K. I. Williams, and Sinisa Todorovic. Tree-cut for probabilistic image segmentation, 2015.

[40] Jake Snell and Richard S. Zemel. Stochastic segmentation trees for multiple ground truths. In *UAI*, 2017.

[41] David G Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[42] Roberto Toldo and Andrea Fusiello. Robust multiple structures estimation with j-linkage. In *ECCV*, pages 537–547. Springer, 2008.

[43] Berk Calli, Aaron Walsman, Arjun Singh, Siddhartha Srinivasa, Pieter Abbeel, and Aaron M Dollar. Benchmarking in manipulation research: Using the yale-cmu-berkeley object and model set. *IEEE Robotics & Automation Magazine*, 22(3):36–52, 2015.