2023 IEEE International Conference on Robotics and Automation (ICRA 2023)
May 29 - June 2, 2023. London, UK

Discriminative 3D Shape Modeling for Few-Shot Instance Segmentation

Anoop Cherian Siddarth Jain Tim K. Marks Alan Sullivan

***Abstract*— In this paper, we present a simple and efficient scheme for segmenting approximately convex 3D object instances in depth images in a few-shot setting via discriminatively modeling the 3D shape of the object using a neural network. Our key idea is to select pairs of 3D points on the depth image between which we compute surface geodesics. As the number of such geodesics is quadratic in the number of image pixels, we can create a large training set of geodesics using only very limited ground truth instance annotations. These annotations are used to create a binary label for each geodesic, which indicates whether or not that geodesic belongs entirely to one instance segment. A neural network is then trained to classify the geodesics using these labels. During inference, we create geodesics from selected seed points in the test depth image, then produce a convex hull of the points that are classified by the neural network as belonging to the same instance, thereby achieving instance segmentation. We present experiments applying our method to segmenting instances of food items in real-world depth images. Our results demonstrate promising performances compared to prior methods in accuracy and computational efficiency.**

I. Introduction

Segmenting nearly identical object instances in an image is a problem that is ubiquitous in a variety of robotic bin-picking applications. Some examples include: (i) a robotic arm grasping products moving on a conveyor belt in a manufacturing setting, (ii) a supermarket robot picking and placing fruits from a bin, or (iii) a library-assistant robot taking books from a box and handing them to a human. Standard deep learning solutions used for instance segmentation tasks, such as Mask-RCNN [1] and recent variants [2], [3], [4], typically require large datasets for training the neural networks, which would demand significant annotation efforts that may not be practical in many situations; e.g., in a factory where there could be a multitude of bins for a robotic arm to pick instances from, and each bin containing a different object class. The instance segmentation task could also be addressed using region clustering and grouping methods such as K-Means, spectral clustering, or superpixels [5], [6], [7]; however such algorithms usually make assumptions on the object shape or would involve hyperparameters that may need to be tuned for each image. There are also recent approaches such as [8], [9], [10] for unsupervised instance segmentation. However, they require large unlabelled training sets with diversity in the instance arrangements, which may be difficult to obtain in many real-world situations.

In this work, we consider the problem of instance segmentation of approximately convex object instances in depth images in a few-shot setting. We assume access to a minimal set of annotated depth images, each having a few instances annotated with their ground truth segments. Our key idea is to create surface trajectories or geodesics on the 3D surface of the depth image, to train a neural network to classify each trajectory as being either contained *within* a single ground-truth instance or *across* two or more ground-truth instances. Thus, the network discriminatively learns to use the features from the surface of the instances for classification of the trajectories, thereby potentially learning an implicit 3D model of a single object instance. For a depth image with n pixels, $\mathcal{O}(n^2)$ such geodesics are potentially possible, which, if carefully used, could provide a significantly large dataset to train the neural network. Leveraging this insight, we design an instance segmentation pipeline to select seed points from which to compute such geodesics efficiently; these geodesics are then discretized to produce fixed-dimensional feature vectors to train a neural network for classification. At test time, we use the same pipeline and compute a convex hull of all of the points whose geodesics to a seed point are classified as within the same segment as the seed point. This yields a segmentation of the instance containing that seed point. After removing each segmented instance from the depth image, we repeat the process to segment other instances.

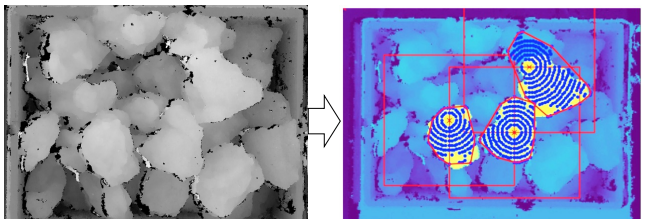


Fig. 1. Given a depth image (left) consisting of multiple instances of an object (e.g., chicken nuggets), our goal is to segment a few prominent instances. On the right, we show three instances segmented by our scheme. We highlight regions selected for segmentation (red boxes), the points classified as belonging to an instance (blue points), and the ground truth segmentation for each respective instance (yellow).

To validate our approach, we present experiments on a new Food-Items Instance Segmentation dataset, consisting of real-world depth images, each with multiple instances of one of four different food items, in a bin-picking setting. Unlike prior datasets [4], [11], [12], that mainly use simulated objects, we use images of real food items, and thus have significant inter-instance diversity (e.g., see chicken nuggets in Figure 1) with variations in their shapes, sizes, and count in the bin. Our experimental results show that in addition to being computationally very cheap, our approach produces segmentations that are more accurate than prior methods.

Mitsubishi Electric Research Laboratories (MERL), Cambridge, MA. {cherian, sjain, tmarks}@merl.com

II. Related Works

The topic of segmenting object instances for robotic bin-picking has been widely studied. Over the years, many different algorithms have been developed; see [13] for a survey. Below, we briefly review the key trends.

Classic methods and variants: A popular direction is to extend image morphological and region-growing operations for segmentation; e.g., the Watershed algorithm [14], [15], [16]; however such methods are notorious for oversegmenting the images [17], [18], [19], [20]. In [21], a deep variant of the Watershed transform for semantic instance segmentation is proposed; however, it requires a large data set for training a deep neural network. In [16], over-segmentation in the Watershed transforms is avoided using geodesic erosion to produce segmentation proposals, however, needs post-processing heuristics to select useful segments from the proposals. Watershed transforms on local distance maps is proposed in [17], and with geodesic distance transforms in [16], [22], however do not use any learning. Similarly, classic methods such as active contours have been used for medical image segmentation [23]; however, they do not consider implicitly modeling shapes using a classifier.

Clustering-based methods. One could also consider clustering algorithms to be adapted for instance segmentation tasks. For example, classical superpixeling methods such as [24], [5] and their modern variants such as SLIC [25] and C2NO [26] could be used for instance segmentation. However, such methods make strong assumptions on the segments, such as constant point cloud size in the clusters in [26], which may be difficult to conform to in segmenting non-rigid objects such as the food-items dataset we use. We also note that general approaches for few-shot instance segmentation such as [27], [28] and variants operate in a setting different from ours and cannot be directly used. Compared with super-pixel representations in 2D space [25], super-voxel representation can work better to cluster instances where the instance boundaries in 3D space are more accessible to segment based on the geometric continuity or local convexity properties [29]. However, these methods require parameter tuning based on the voxel size resolution and can be difficult to generalize.

Deep Learning Approaches: As alluded to above, recent deep learning methods for instance segmentation in a bin-picking setting is mainly built over the Mask-RCNN pipeline, and usually needs large training sets. Works such as [30], [4], [31] belong to this category and often assume rigid CAD objects or synthetic data [32], [33], [34], which is a setting different from our goal. Few-shot instance segmentation has been an important topic in mainstream computer vision [27], [28], [35], however their focus is in semantically segmenting general scenes.

To the best of our knowledge, the use of geodesic curves for discriminatively modeling the object shapes within a neural network setup has not been explored previously.

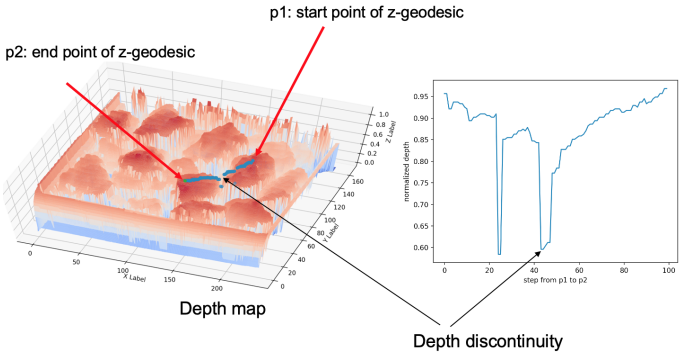


Fig. 2. An illustration of the depth geodesic between two points p_1 and p_2 on a depth map.

III. Proposed Method

Suppose we are given a set of annotated depth images $\mathscr{D} = \{D_1, D_2, \cdots, D_N\}$ where each $D \in \mathbb{R}_+^{H \times W}$ defines an image grid of width W and height H pixels, such that D_{xy} holds a non-negative value corresponding to the depth of the scene at location $(x,y) \in [H] \times [W]$, for $[Z]$ denoting the index set of integers $\{0,1,\cdots,Z-1\}$. For a pixel (x,y) on the image grid, we assume it is annotated with an instance label $\ell_{xy} \in [L_D] \cup \{L_B\}$, where L_D is the number of instances in the depth image D, and L_B corresponds to a background label. To introduce our method, we will need some background notation, which we describe next.

A. Surface Geodesics and Assumptions

For two distinct points (x_1,y_1) and (x_2,y_2) on the image grid, suppose $\gamma(t)$ (for $t \in [0,1]$) be the directed surface curve starting at $D_{x_1y_1}$ and ending at $D_{x_2y_2}$. That is, $\gamma(0) = D_{x_1y_1}$, and $\gamma(1) = D_{x_2y_2}$, with all its points $\gamma(t) \in D, \forall t \in [0,1]$. We define $L(\gamma) = \int_{x_1y_1}^{x_2y_2} \sqrt{D_\gamma(t)(\dot{\gamma}(t), \dot{\gamma}(t))}dt$ as the length of this curve γ, and a geodesic g is a curve with the minimal length connecting the two points [36], i.e., $g \in \inf_\gamma L(\gamma)$. To derive our method, we make the following assumptions on our problem setting.

Assumption 1 (Surface Convexity): We assume the objects used in our setup are convex and the depth patch associated with the instances form a convex smooth surface. By convex object surfaces, we mean that all the one-dimensional curves $\gamma(t)$ on the surface are convex with respect to t, i.e., $\gamma(t) \leq (1-t)\gamma(0) + t\gamma(1), \forall t \in [0,1]$. Suppose $D_\ell^S \subset D$ is a patch from the depth image D where all the elements in D_ℓ^S have the same instance label ℓ. Then, for two distinct points $(x_1,y_1),(x_2,y_2)$ on the image grid where both $D_{x_1y_1}, D_{x_2y_2} \in D_\ell^S$, if $g_{x_1y_1}^{x_2y_2}(t)$ is a geodesic starting at $g_{x_1y_1}^{x_2y_2}(0) = (x_1,y_1)$ and ending at $g_{x_1y_1}^{x_2y_2}(1) = (x_2,y_2)$, and if label denotes the instance label of the point $g_{x_1y_1}^{x_2y_2}(t)$ on the geodesic, then we have the following proposition that is straightforward to prove using the basic properties of convexity.

Proposition 1: If D_ℓ^S is a convex depth patch from a depth map D, and if $g(t)$ is a geodesic from $g(0) = (x_1,y_1) \in D_\ell^S$ to $g(1) = (x_2,y_2) \in D_\ell^S$, then $\text{label}(g(t)) = \ell, \forall t \in [0,1]$.

Assumption 2 (Orthogonal Projection): The camera projection plane is located suitably far from the instances, such that the image XY-plane is approximately orthogonal to the velocity $\dot{\gamma}(t)$ of any trajectory on the depth surface.

This assumption allows us to parameterize the geodesic $g^{x_2y_2}_{x_1y_1}(t)$ connecting 3D points $p_1 = (x_1, y_1, g^{x_2y_2}_{x_1y_1}(0))$ and $p_2 = (x_2, y_2, g^{x_2y_2}_{x_1y_1}(1))$ by the straight line $e^{x_2y_2}_{x_1y_1}(t)$ for $xy(t) = (1-t)(x_1, y_1) + t(x_2, y_2)$ for $t \in [0,1]$. We will use $e(xy(t))$ to denote $e^{x_2y_2}_{x_1y_1}(t)$ for simplicity, and with this parameterization, we use points on the straight line $e(xy(t))$ to index the geodesic.

B. Discriminative Shape Modeling

In Fig. 2, we show a depth image containing multiple instances of an object, and the surface curve between two arbitrary points p_1 and p_2. On the right in Fig. 2, we plot this curve as a one-dimensional curve $g(t)$ for varying t. Our key insight to develop our approach is that if the two ends of this curve belong to different instances, then this curve will be non-convex or non-smooth at the points where the instances overlap. Our approach attempts to leverage this insight into learning the object shapes implicitly in the parameters of a discriminative neural network. To set the stage for our discussions, we will start with explaining a few ingredients in our algorithm that are essential to derive of our pipeline.

Geodesic Discretization: From a practical sense, directly using the geodesics for instance segmentation is problematic as one would need their implicit paramtrization as continuous curves, which may not have any analytical form (e.g., a surface geodesic on a chicken nugget?). Instead, to keep things computationally cheap, we discretize the curves using a fixed number of steps. Specifically, for a geodesic $g(t)$, we represent it as an m-dimensional vector $v \in \mathbb{R}^m_+$ where the k-th dimension $v_k = g((k-1)/m)$. Such discretized geodesics can be computed very cheaply using Assumption 2 of orthogonal projection of the camera plane, as in that case, one just needs to first split the Euclidean geodesic approximation $e^{x_2y_2}_{x_1y_1}(t)$ to m parts, i.e., $xy(k) = e^{x_2y_2}_{x_1y_1}((k-1)/m)$ to obtain the (x,y) 2D image grid location, which can then be used to directly index the depth map to get $v_k = D_{xy(k)}$.

Instance Supervision: If the discontinuities of the surface geodesics are sufficient to find the instance boundaries, then why would one need instance annotations? This is because, the above discretization step may skip discontinuities in the curve; e.g., if the two instances are very close or when hole-filling [37] is applied to the depth images. To circumvent these issues, we assume to have access to ground truth instance labels, training using which will make the network to use other discrinimative features in the geodesic for classification, thereby implicitly learning the object shape.

C. Instance Segmentation Training Pipeline

In Figure 3, we illustrate our pipeline for training a neural network to learn the implicit shape model of a single instance of the object using surface geodesics. For a given training depth image D, the first step in our pipeline is to select a random set of M points on the 2D image grid. Let us call this set $\mathscr{P} = \{(x_1, y_1), (x_2, y_2), \cdots, (x_M, y_M)\}$. Next, for every pair of such points $(x_i, y_i), (x_j, y_j) \in \mathscr{P}$, we compute the Euclidean (geodesic) straight lines $e^{x_jy_j}_{x_iy_i}(t)$ (one such point and its straight lines to a couple of other points are only shown in Figure 3). This step is followed by computing the depth geodesics $g^{x_jy_j}_{x_iy_i}(t)$ on the depth image by projecting these Euclidean geodesics on the depth map. Each depth geodesic is then discretized into m bins forming the set $\mathscr{V} = \{v_1, v_2, \cdots, v_M\}$ of M vectors as described in the above section, each v corresponding to a discretized depth geodesic. Suppose $v := v^{x_jy_j}_{x_iy_i} \in \mathscr{V}$ is such a discretized vector corresponding to a depth geodesic from point (x_i, y_i) to (x_j, y_j), then we assign a label label_g to v: $\text{label}_\text{g}(v^{x_jy_j}_{x_iy_i}) = 1$, if $\ell_{x_iy_i} = \ell_{x_jy_j}$ and 0 otherwise, where ℓ_{xy} is the instance label associated with the image point (x,y).

Our final step in the training pipeline is to use the set $\mathscr{V}$ and its corresponding binary labels to train a neural network model $f_\theta : \mathscr{V} \to \{0,1\}$, parametrized by θ. Specifically, the neural network is a series of multi-layer perceptrons (MLP), and takes as input a batch of samples from $\mathscr{V}$ and predicts the label of the respective sample. This prediction is then matched with the ground truth binary label using the softmax-crossentropy loss, which is then used to derive a gradient to train the network parameters.

D. Instance Segmentation Inference Pipeline

At test time, given a test depth image D, our goal is to repeat the process during the training phase for instance segmentation. As our goal is finally to produce a segmentation for an instance in the bin that is perhaps most useful for a robotic arm to grasp and pick, we propose to segment instances that are at the top of the bin (i.e., those instances closest to the camera), with the goal of generating its instance segmentation mask first. We call such an instance as a *pickable* instance.

The inference pipeline in our setup is illustrated in Figure 4. First, we select a seed point in the test depth image, which corresponds to the tallest point on the pickable instance. Let us call this point H. Next, we use an approximate region around H where the instance could be within. We call the radius of this region the *pick radius* r, where r is decided based on the size of the annotated instances. The region is a square box centered at H, with a size $2r$. Next, we sample m' points $\{(x_1, y_1), (x_2, y_2), \cdots, (x_{m'}, y_{m'})\}$ uniformly around H, and create Euclidean geodesics $\{e^{x_iy_i}_H\}^{m'}_{i=1}$. These geodesics are then mapped to discretized depth geodesics v and classified using pre-trained f_θ to signify the other endpoint of v (corresponding to a point (x_i, y_i) around H) is within an instance segment or not. The points that are classified as within a segment are then fed to a robust convex hull computation algorithm [38] to produce a segmentation of the instance. Note that the convexity of the object is thus important for this step to work correctly.

To create a segmentation for a different instance, we select another tall point H' from the depth image such that the pick radius r around H' will not overlap with the pick radius around H. That is, we search for instances whose depth

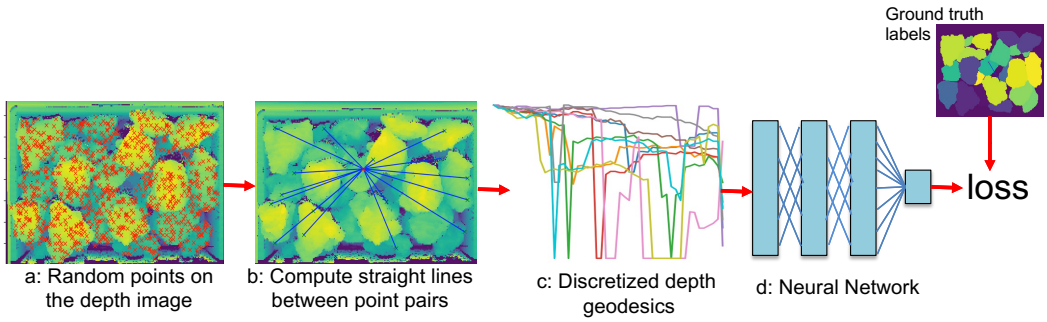


Fig. 3. Training Pipeline for learning the discriminative shape models.

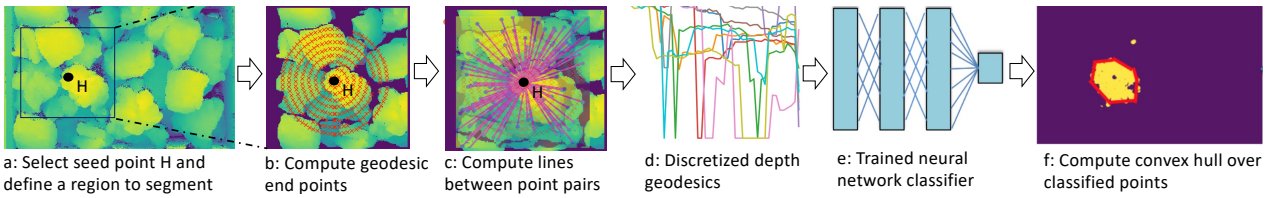


Fig. 4. A depiction of the steps involved during test time inference using a trained neural network.

geodesics will not overlap with the instance that we already segmented. Once we find a point H', we apply the procedure described above. We do this process sequentially, generating one instance segment at a time.

E. Systematic Sampling of the Test Geodesics

In the basic inference algorithm described above, we randomly sampled the test points around the seed point. However, a more efficient approach would be to select the points systematically. To this end, we propose to use the pick radius r to define a circular region around the pick point H; this region is then divided into equal sectors, by dividing r into β equal parts, and dividing the circle into ζ equal angles. This leads to $\beta\zeta$ points to consider for generating the surface geodesics, where these parameters can be adjusted depending on underlying shape of the segment we ought to learn. Figure 4 (a,b)-steps illustrates this idea.

IV. Experiments and Results

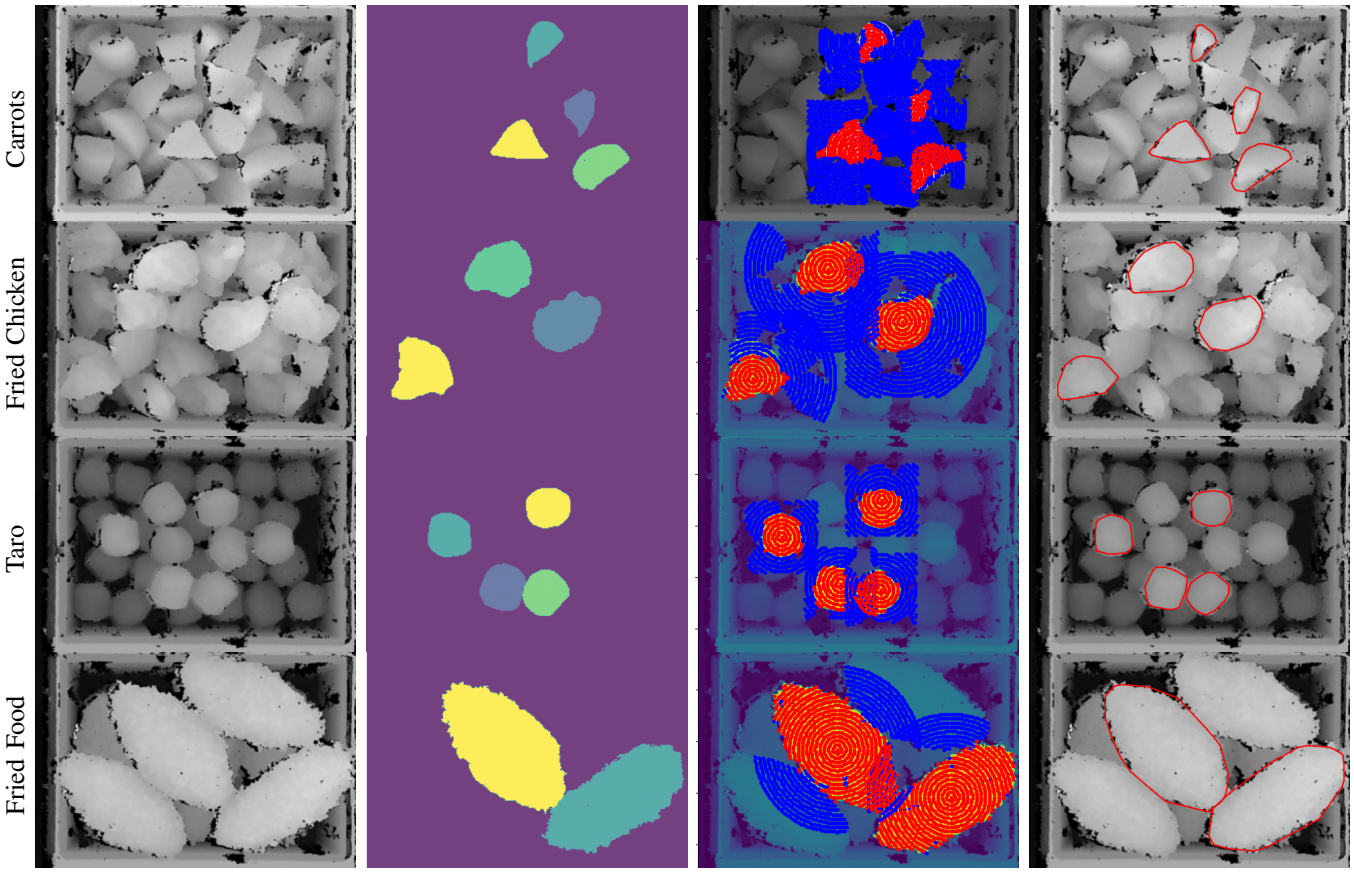
In this section, we provide experiments demonstrating the empirical performances of our method.

Food-Items Instance Segmentation Dataset: One key challenge to conducting experiments using our setup is the lack of a real-world dataset that conforms to our algorithmic requirements. For example, recent papers such as [4], [12], [11] use synthetic datasets for instance segmentation evaluation – these images may not incorporate real-world artifacts, e.g., sensor noise, intra-instance differences, etc. To this end, we created a new instance segmentation dataset consisting of four classes of food items, namely: (i) fried chicken, (ii) fried food, (iii) cut carrots, and (iv) taros. Example grayscale images from these classes are shown in Figure 5. We used an Ensenso depth camera to capture these images in a real bin. There are 20 annotated depth images for each class, of which three images are used for training our method, one for validation, and the rest for testing. Our training set is minimal; thus, training deep models is infeasible, motivating the need for a few-shot approach like ours. The number of instances in the bin for each object class varies from 2–30.

Learning Setup: We use a 5-layer fully-connected neural network with an architecture given by: $[m, 5m, m, m/2, 2]$, where m is the dimensionality of our geodesic feature vector after the discretization step. We used ReLU activations between the network layers. For training the network, we used Adam for the optimization using a learning rate of 0.001 and other default settings. The network is trained for a maximum of 500 epochs.

Hyperparameters: We sample 200 random points from each training image for all the classes in our dataset and compute pairwise geodesics. At test time, we select a maximum of 5 tallest points in the depth point cloud to seed our segmentations. We use 1000 endpoints around each seed and create geodesics from the seed to these endpoints and classify them using our trained neural network.

Evaluation: As alluded to above, our method produces instance segmentations iteratively via selecting seed points in the depth images. For evaluation, we compute the mean intersection-over-union of instance masks produced by a method against the ground truth masks for the respective seeds. We compare our method to classic baselines such as: (i) KMeans, (ii) spectral clustering with bandwidth selected automatically, (iii) Gaussian mixture models (GMM) with diagonal (diag) and full covariances, and (v) the Watershed algorithm [14]. We also compare to more recent and popular baselines such as: (i) LCCP [29], (ii) SLIC [25], and C2NO [26], as well as to Mask-RCNN [1], pretrained on MSCOCO and finetuned on each of our datasets. As we found that Mask-RCNN do not produce work at all when using three annotated images (as in our other comparisons), we used ten images and their ground truth for its training.



(a) Depth Images (b) Ground Truth (c) End-points/Classification (d) Predicted Segments

Fig. 5. Rows: Example images from our Food-Items Instance Segmentation dataset. Columns: (a) Input depth images, (b) A few of the annotated (ground truth) segments, (c) End-points of geodesics used at test time; points classified by the neural network as going outside an instance are blue and points classified as within an instance are red. We use the red points for computing the convex hulls and the instance segmentations depicted in (d).

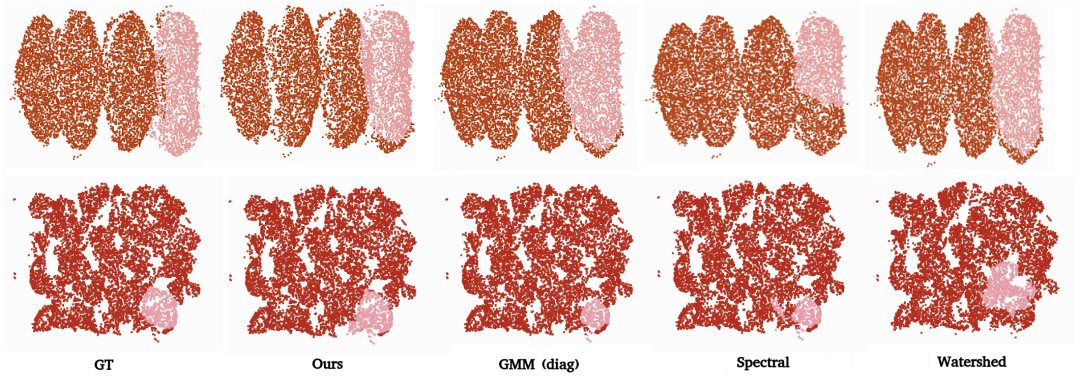
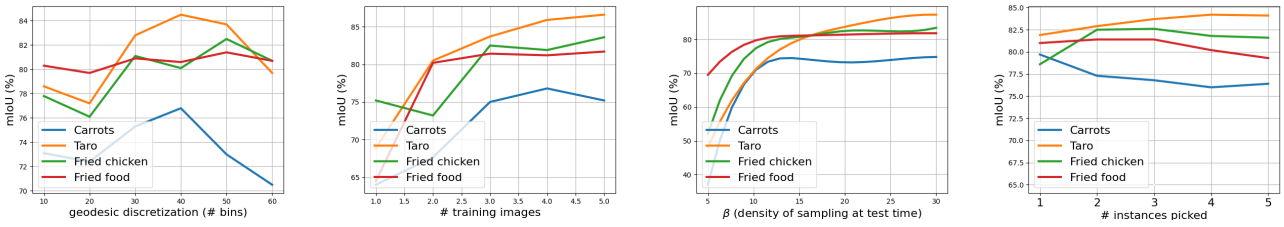


Fig. 6. Qualitative comparisons of segmentations between different approaches on Fried Food and Carrot classes. The produced segments are highlighted.



(a) (b) (c) (d)

Fig. 7. Analysis of the performance of our method against variations in the choices of the hyperparameters.

In our iterative scheme, we remove already segmented instances from the image to avoid segmenting the same instance again. Thus, we select a subsequent seed point outside a box of given dimensions around a selected seed. As a result, the segmentations produced by our method may not be aligned with the segments produced by other methods. To make the comparisons fair, we compare all methods only on the ground truth segments to which our seed points belong.

Experimental Results: In Table I, we present results comparing our approach against other methods on our Food-Items Dataset. We find that our simple approach leads to a significant improvement in performance. Specifically, we find that our scheme outperforms classical methods by nearly 20-30% (e.g., on the Carrot dataset), while very recent methods such as C2NO [26] was seen to underperform on our dataset, perhaps because its constant size cluster assumption may not hold when object shapes or sizes are changed. We find older methods, such as Watershed transform [14], give a promising performance, perhaps because of strong edges; however, our performance is significantly better. Further, we also find that Mask-RCNN (MRCNN) results are also inferior to ours (even when it used more data for training). When there are many annotated segments in the training set (e.g., carrots, taro, and chicken) the MRCNN results are reasonable, however when the number of instances are low (e.g., FriedFood), the results are very poor.

In Figures 5 and 6, we provide several qualitative results from our approach, as well as show intermediate outputs. In Figure 8, we show two qualitative results using Mask-RCNN on the carrots and taro datasets. As is clear, we find that Mask-RCNN either hallucinates instances (Figure 8, left) or oversegments the instances (Figure 8, right).

Ablation Studies

How many bins in Geodesic Discretization? In Figure 7(a), we plot the segmentation performance against increasing discretization granularity on all the four data classes. As we anticipate, we find that a higher number of bins steadily decrease the performance as it may capture fine-grained nuances on the instances affecting the models' generalizability.

How many images to use for training? In Figure 7(b), we explore the influence of the number of training images, changing from 1–5. We use 200 point pairs per image, and thus more images lead to more training geodesic trajectories. The plot shows that more training data does lead to performance improvements, however interestingly, the performances seem to saturate after 3 images; suggesting that our method needs only limited training data, while generalizing well to arbitrary number of instances.

How dense should we sample the test instances? A key parameter during inference is the density β at which end points are sampled around the seed (as described in Sec. III-E) – a higher number may slow down the inference. In Figure 7(c), we plot the performance against increasing β, i.e., higher β implies denser end point sampling. Our results suggest that beyond $\beta = 15$, the performance do not increase.

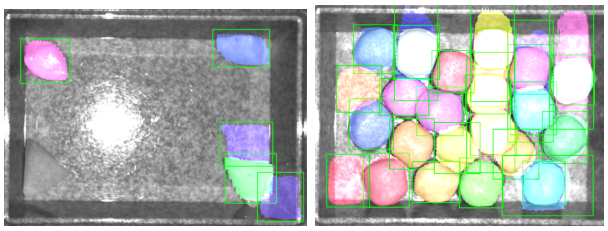


Fig. 8. Mask-RCNN finetuned on carrots and taro classes.

We found a similar trend for the angular sampling parameter ζ (and thus not shown), and used $\zeta = 2$.

Generalization to number of instances? We answer this question in Figure 7(d), where we find that our performance does not change much as the number of instances being picked is increased. Beyond 5 instances, the performances might not be comparable since we remove previously detected instances within a box, there may not be sufficient number of instances left to be detected.

Computational Time? Training our neural network takes less than 5 minutes on a 4-core Intel 3GHz CPU. In Table II, we report the average time taken for segmenting an instance (on CPU). Note that our implementation is currently in Python, while other methods use e.g., C++ backends, and our method is computationally similar to standard approaches. We also compare against the time taken by Mask-RCNN when using an Nvidia Titan RTX GPU and using the above CPU configuration. While, the GPU setting performs slightly better than ours, the CPU setting is very slow.

Method	Carrot	Taro	Chicken	Fried Food
Ours	**0.807**	**0.844**	**0.851**	**0.944**
KMeans	0.510	0.461	0.480	0.435
GMM (full)	0.518	0.409	0.439	0.442
GMM (diag)	0.459	0.446	0.466	0.578
Spectral [39]	0.487	0.434	0.477	0.572
Watershed [14]	0.687	0.339	0.585	0.862
LCCP [29]	0.486	0.437	0.440	0.501
SLIC [25]	0.420	0.357	0.370	0.429
C2NO [26]	0.261	0.232	0.280	0.444
Mask-RCNN	0.659	0.712	0.591	0.262

TABLE I

MEAN IOU COMPARISONS AGAINST PRIOR METHODS.

Method	KMeans	Spectral	LCCP	MRCNN	Ours
time (s)	0.082	0.324	0.059	1.59 (0.135)	0.170

TABLE II

AVERAGE TIME TAKEN (IN SECONDS) FOR SEGMENTATION. FOR MRCNN, WE SHOW TIME TAKEN USING A GPU IN BRACKETS.

V. CONCLUSIONS

We presented a simple, efficient, and few-shot approach to instance segmentation of objects (with non-consistent geometry) on depth images for a real-world robotic bin picking. As a robot needs to pick only a single instance at a time, we would only need to segment a few object instances for the robot to pick from; this insight led to the derivation of our iterative approach for classifying instance surface geodesics using a neural network, allowing for learning in a few-shot regime. Experimental results demonstrate empirical benefits of our approach, including accuracy, speed, generalizability, and scalability against prior approaches.

REFERENCES

[1] Kaiming He, Georgia Gkioxari, Piotr Dollár, and Ross Girshick, "Mask R-CNN," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961–2969.

[2] Shu Liu, Lu Qi, Haifang Qin, Jianping Shi, and Jiaya Jia, "Path aggregation network for instance segmentation," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[3] Bernardino Romera-Paredes and Philip Hilaire Sean Torr, "Recurrent instance segmentation," in *European conference on computer vision*. Springer, 2016, pp. 312–329.

[4] Yidan Feng, Biqi Yang, Xianzhi Li, Chi-Wing Fu, Rui Cao, Kai Chen, Qi Dou, Mingqiang Wei, Yun-Hui Liu, and Pheng-Ann Heng, "Towards robust part-aware instance segmentation for industrial bin picking," *arXiv preprint arXiv:2203.02767*, 2022.

[5] Pedro F Felzenszwalb and Daniel P Huttenlocher, "Efficient graph-based image segmentation," *International journal of computer vision*, vol. 59, no. 2, pp. 167–181, 2004.

[6] Tahir Rabbani, Frank Van Den Heuvel, and George Vosselmann, "Segmentation of point clouds using smoothness constraint," *International archives of photogrammetry, remote sensing and spatial information sciences*, vol. 36, no. 5, pp. 248–253, 2006.

[7] Anh-Vu Vo, Linh Truong-Hong, Debra F Laefer, and Michela Bertolotto, "Octree-based region growing for point cloud segmentation," *ISPRS Journal of Photogrammetry and Remote Sensing*, vol. 104, pp. 88–100, 2015.

[8] Anoop Cherian, Gonçalo Dias Pais, Siddarth Jain, Tim K Marks, and Alan Sullivan, "InSeGAN: A generative approach to segmenting identical instances in depth images," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021, pp. 10023–10032.

[9] Francesco Locatello, Dirk Weissenborn, Thomas Unterthiner, Aravindh Mahendran, Georg Heigold, Jakob Uszkoreit, Alexey Dosovitskiy, and Thomas Kipf, "Object-centric learning with slot attention," *Advances in Neural Information Processing Systems*, vol. 33, pp. 11525–11538, 2020.

[10] Christopher P Burgess, Loic Matthey, Nicholas Watters, Rishabh Kabra, Irina Higgins, Matt Botvinick, and Alexander Lerchner, "Monet: Unsupervised scene decomposition and representation," *arXiv preprint arXiv:1901.11390*, 2019.

[11] Kilian Kleeberger, Christian Landgraf, and Marco F Huber, "Large-scale 6d object pose estimation dataset for industrial bin-picking," in *2019 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. IEEE, 2019, pp. 2573–2578.

[12] Romain Brégier, Frédéric Devernay, Laetitia Leyrit, and James L. Crowley, "Symmetry aware evaluation of 3d object detection and pose estimation in scenes of many parts in bulk," in *The IEEE International Conference on Computer Vision (ICCV)*, Oct 2017.

[13] Masahiro Fujita, Yukiyasu Domae, Akio Noda, GA Garcia Ricardez, Tatsuya Nagatani, Andy Zeng, Shuran Song, Alberto Rodriguez, A Causo, I-Ming Chen, et al., "What are the important technologies for bin picking? technology analysis of robots in competitions based on a set of performance metrics," *Advanced Robotics*, vol. 34, no. 7-8, pp. 560–574, 2020.

[14] Luc Vincent and Pierre Soille, "Watersheds in digital spaces: an efficient algorithm based on immersion simulations," *IEEE Transactions on Pattern Analysis & Machine Intelligence*, vol. 13, no. 06, pp. 583–598, 1991.

[15] Serge Beucher and Fernand Meyer, "The morphological approach to segmentation: the watershed transformation," *Mathematical morphology in image processing*, vol. 34, pp. 433–481, 1993.

[16] Laurent Najman and Michel Schmitt, "Geodesic saliency of watershed contours and hierarchical segmentation," *IEEE Transactions on pattern analysis and machine intelligence*, vol. 18, no. 12, pp. 1163–1173, 1996.

[17] San-Mook Lee, A Lynn Abbott, and Philip A Araman, "Segmentation on statistical manifold with watershed transform," in *2008 15th IEEE International Conference on Image Processing*. IEEE, 2008, pp. 625–628.

[18] Ilya Levner and Hong Zhang, "Classification-driven watershed segmentation," *IEEE Transactions on Image Processing*, vol. 16, no. 5, pp. 1437–1445, 2007.

[19] Myung-Cheol Roh, Tae-Yong Kim, Jihun Park, and Seong-Whan Lee, "Accurate object contour tracking based on boundary edge selection," *Pattern Recognition*, vol. 40, no. 3, pp. 931–943, 2007.

[20] Philipp Krähenbühl and Vladlen Koltun, "Geodesic object proposals," in *European conference on computer vision*. Springer, 2014, pp. 725–739.

[21] Min Bai and Raquel Urtasun, "Deep watershed transform for instance segmentation," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 5221–5229.

[22] Michael Roberts, Ke Chen, and Klaus L Irion, "A convex geodesic selective model for image segmentation," *Journal of Mathematical Imaging and Vision*, vol. 61, no. 4, pp. 482–503, 2019.

[23] R Hemalatha, T Thamizhvani, A Josephin Arockia Dhivya, Josline Elsa Joseph, Bincy Babu, and R Chandrasekaran, "Active contour based segmentation techniques for medical image analysis," *Medical and Biological Image Analysis*, vol. 4, no. 17, pp. 2, 2018.

[24] Xing Wei, Qingxiong Yang, Yihong Gong, Narendra Ahuja, and Ming-Hsuan Yang, "Superpixel hierarchy," *IEEE Transactions on Image Processing*, vol. 27, no. 10, pp. 4838–4849, 2018.

[25] Radhakrishna Achanta, Appu Shaji, Kevin Smith, Aurelien Lucchi, Pascal Fua, and Sabine Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.

[26] André FR Guarda, Nuno MM Rodrigues, and Fernando Pereira, "Constant size point cloud clustering: A compact, non-overlapping solution," *IEEE Transactions on Multimedia*, vol. 23, pp. 77–91, 2020.

[27] Dan Andrei Ganea, Bas Boom, and Ronald Poppe, "Incremental few-shot instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2021, pp. 1185–1194.

[28] Wei-Yu Chen, Yen-Cheng Liu, Zsolt Kira, Yu-Chiang Frank Wang, and Jia-Bin Huang, "A closer look at few-shot classification," *arXiv preprint arXiv:1904.04232*, 2019.

[29] Simon Christoph Stein, Markus Schoeler, Jeremie Papon, and Florentin Worgotter, "Object partitioning using local convexity," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[30] Michael Danielczuk, Matthew Matl, Saurabh Gupta, Andrew Li, Andrew Lee, Jeffrey Mahler, and Ken Goldberg, "Segmenting unknown 3d objects from real depth images using mask r-cnn trained on synthetic data," in *2019 International Conference on Robotics and Automation (ICRA)*. IEEE, 2019, pp. 7283–7290.

[31] Chungang Zhuang, Zhe Wang, Heng Zhao, and Han Ding, "Semantic part segmentation method based 3d object pose estimation with rgb-d images for bin-picking," *Robotics and Computer-Integrated Manufacturing*, vol. 68, pp. 102086, 2021.

[32] Muhammad Usman Khalid, Janik M Hager, Werner Kraus, Marco F Huber, and Marc Toussaint, "Deep workpiece region segmentation for bin picking," in *2019 IEEE 15th International Conference on Automation Science and Engineering (CASE)*. IEEE, 2019, pp. 1138–1144.

[33] Xianzhi Li, Rui Cao, Yidan Feng, Kai Chen, Biqi Yang, Chi-Wing Fu, Yichuan Li, Qi Dou, Yun-Hui Liu, and Pheng-Ann Heng, "A sim-to-real object recognition and localization framework for industrial robotic bin picking," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 3961–3968, 2022.

[34] Ping Jiang, Yoshiyuki Ishihara, Nobukatsu Sugiyama, Junji Oaki, Seiji Tokura, Atsushi Sugahara, and Akihito Ogawa, "Depth image–based deep learning of grasp planning for textureless planar-faced objects in vision-guided robotic bin-picking," *Sensors*, vol. 20, no. 3, pp. 706, 2020.

[35] Khoi Nguyen and Sinisa Todorovic, "iFS-RCNN: An incremental few-shot instance segmenter," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2022, pp. 7010–7019.

[36] Ron Kimmel, Arnon Amir, and Alfred M. Bruckstein, "Finding shortest paths on surfaces using level sets propagation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 17, no. 6, pp. 635–640, 1995.

[37] Ji-Min Cho, Soon-Yong Park, and Sung-Il Chien, "Hole-filling of realsense depth images using a color edge map," *IEEE Access*, vol. 8, pp. 53901–53914, 2020.

[38] C Bradford Barber, David P Dobkin, and Hannu Huhdanpaa, "The quickhull algorithm for convex hulls," *ACM Transactions on Mathematical Software (TOMS)*, vol. 22, no. 4, pp. 469–483, 1996.

[39] Andrew Ng, Michael Jordan, and Yair Weiss, "On spectral clustering: Analysis and an algorithm," *Advances in neural information processing systems*, vol. 14, 2001.