V-PRISM: Probabilistic Mapping of Unknown Tabletop Scenes

Herbert Wright[1] Weiming Zhi[2] Matthew Johnson-Roberson[2] Tucker Hermans[1,3]

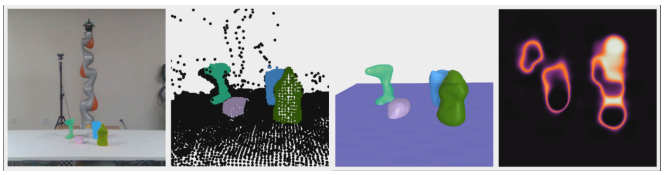

Figure 1: Our method takes a segmented (**left**) point cloud observation (**middle left**) and builds a continuous probabilistic map. This map can be used to reconstruct the scene (**middle right**) or measure uncertainty about the scene (**right**). The heat map shows uncertainty in a 2D slice parallel with the table plane.

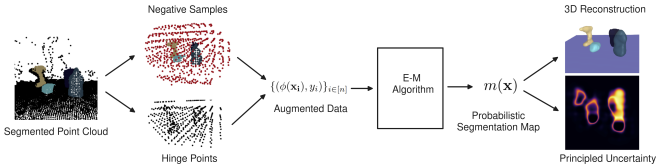

Figure 2: Overview of our method, V-PRISM. We take a segmented point cloud and output a probabilistic segmentation map over 3D space that can be used for both object reconstruction and principled uncertainty. Our method first generates negative samples and hinge points, then uses these to create an augmented dataset. Then the probabilistic map is constructed by running an EM algorithm.

I. Introduction

As robots continue to be deployed in the world, there is an ongoing need for methods that allow them to safely and robustly operate in unknown, noisy scenes. The planning techniques for tasks in such scenes often require an accurate 3D map of the objects within the scene. While some work reconstructs scenes where objects belong to known classes [1], we focus on the harder problem of unknown objects.

The safe operation of robots necessitates not only accuracy but also introspection and uncertainty-awareness. These notions of uncertainty can be incorporated into downstream motion planning solvers for robustness and safety. However, many algorithms typically used to reconstruct unkown objects utilize neural networks to predict geometry [2]–[8]. Such neural networks lack the ability to reason about uncertainty and confidently predict incorrect labels [9], [10]. Here, we outline a Bayesian approach to capture uncertainty in a principled manner.

We propose V-PRISM: Volumetric, Probabilistic, and Robust Instance Segmentation Maps*. V-PRISM is a framework for building differentiable segmentation and occupancy maps of tabletop scenes that contain multiple unseen objects. The produced maps have a principled uncertainty metric.

Section II gives an overview of Sigmoid Bayesian Hilbert Maps. An overview of our method is given in Section III. The algorithmic details are then explained in Section IV and Section V. We evaluate our method in Section VI.

II. Sigmoid Bayesian Hilbert Maps

Hilbert Maps. Introduced in [11], Hilbert Maps are a method for continuous occupancy mapping of a robotic environment. A map $m : \mathbb{R}^d \to [0, 1]$ is built from a feature transform $\phi(\mathbf{x})$ by first observing a point cloud with a depth sensor, then negative sampling along depth rays to create an augmented dataset. Gradient descent is used to find the optimal weights for a map of the form $m(\mathbf{x}) = \sigma(\mathbf{w}^\top \phi(\mathbf{x}))$ where $\sigma : \mathbb{R} \to (0, 1)$ is the sigmoid function. This is equivalent to performing logistic regression over the transformed points $\{(\phi(\mathbf{x}_i), y_i)\}_{i\in[n]}$.

Usually, the feature transform ϕ is constructed from a kernel function k and a set of hinge points $\mathbf{h}_1, ..., \mathbf{h}_m \in \mathbb{R}^3$. Usually, these hinge points are chosen to be an evenly spaced 3D grid of points. The feature transform is then given by:

$$\phi(x) = [k(\mathbf{x}, \mathbf{h}_1), k(\mathbf{x}, \mathbf{h}_2), \ldots, k(\mathbf{x}, \mathbf{h}_m), 1]^\top. \quad (1)$$

Bayesian Extension. Hilbert Maps were extended to the Bayesian setting in [12]. Instead of an individual weight vector, the weight is treated as a normally distributed random variable, $\mathbf{w} \sim P(\mathbf{w})$. Variational Bayesian logistic regression as described in [13] is then performed over the augmented data to obtain the approximate posterior distribution.

Once the posterior weight distribution is obtained, the map m is defined by taking the expectation over the $\mathbf{w}$ distribution. Because there is no analytic solution for this expectation, approximations are used. The most common is:

$$\mathbb{E}_{\mathbf{w}}[\sigma(\mathbf{w}^\top \phi(\mathbf{x}))] \approx \sigma\left(\frac{\mathbb{E}_{\mathbf{w}}[\mathbf{w}^\top \phi(\mathbf{x})]}{\sqrt{1 + \frac{\pi}{8}\mathrm{Var}(\mathbf{w}^\top \phi(\mathbf{x}))}}\right). \quad (2)$$

III. Method Overview

We want to construct a multiclass map $m(\mathbf{x})$ that outputs a distribution over c classes. Our method builds such a map from segmented camera depth observations of a multi-object scene through two main steps. A high level overview is displayed in Figure 2. First, negative sampling is performed as described in Section V, where additional points are added to the observed ones in order to form a new labelled point cloud. We then generate a set of hinge points that are used to construct a feature transform according to Equation (1). This creates a set of augmented data.

Then, we perform Bayesian multi-class regression over the transformed data with an expectation maximization (EM) algorithm. The EM algorithm and model are explored in Section IV, along with evaluating $m(\mathbf{x})$ for new $\mathbf{x}$ values.

[1] University of Utah Robotics Center and Kahlert School of Computing, University of Utah, Salt Lake City, UT, USA
[2] Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA
[3] NVIDIA Corporation, Santa Clara, CA, USA
*Project website: https://herb-wright.github.io/v-prism/

Algorithm 1 V-PRISM

Input:

Data $\{(\phi(\mathbf{x}_i), y_i)\}_{i\in[n]}$ and Priors $\{(\bar{\mu}_k, \bar{\Sigma}_k)\}_{k\in[c]}$

$\alpha \leftarrow 0$, $\xi \leftarrow 1$

for p iterations **do**

  $\hat{\Sigma}^{-1} \leftarrow \bar{\Sigma}^{-1} + 2\sum_i |\lambda(\xi_i)|\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^\top$

  $\hat{\mu}_k \leftarrow \hat{\Sigma}\left(\bar{\Sigma}^{-1}\bar{\mu} + \sum_i (y_i - \frac{1}{2} + 2\alpha_i\lambda(\xi_{i,k}))\phi(\mathbf{x}_i)\right)$

  $\alpha_i \leftarrow$ UPDATEALPHA$(\xi_i, \mathbf{x}_i, \hat{\mu}, \hat{\Sigma})$ with Equation (3)

  $\xi_{i,k} \leftarrow$ UPDATEXI$(\alpha_i, \mathbf{x}_i, \hat{\mu}, \hat{\Sigma})$ with Equation (4)

end for

return $\hat{\mu}, \hat{\Sigma}$

Once we have our map, we can use it to evaluate how likely different points are to be occupied by different objects. We can also reconstruct the meshes of each object by running the marching cubes algorithm [14].

IV. Softmax EM Algorithm

To create a Bayesian multi-class map, we consider using a weight matrix $\mathbf{W} \in \mathbb{R}^{c\times m}$ where each row is normally distributed, with the following likelihood function:

$$P(y = k|\mathbf{W}, \mathbf{x}) = \text{softmax}(\mathbf{W}\phi(\mathbf{x}))_k.$$

Because a conjugate prior for the softmax likelihood doesn't exist, we must use variational inference to find a posterior Gaussian distribution. In our case, we will approximate our posterior by a lower bound on the likelihood, $Q(y = k|\mathbf{W}, \mathbf{x}; \alpha, \xi) \leq P(y = k|\mathbf{W}, \mathbf{x})$. We can maximize this lower bound and use it as an approximation to the true likelihood by solving the following:

$$\arg\max_{\alpha,\xi} \mathbb{E}_{\mathbf{W}}\left[\ln Q(y = y_i|\mathbf{x}_i, \mathbf{W}; \alpha, \xi)\right].$$

This can be analytically solved for $\mathbf{W}_k \sim \mathcal{N}(\mu_k, \Sigma_k)$, yielding the following optimal values found in [15]:

$$\alpha_i = \frac{\frac{1}{2}(\frac{c}{2} - 1) + \sum_{k=1}^{c} \lambda(\xi_k)\mu_k^\top\phi(\mathbf{x}_i)}{\sum_{k=1}^{c}\lambda(\xi_k)}, \tag{3}$$

$$\xi_{i,k}^2 = \phi(\mathbf{x}_i)^\top\Sigma_k\phi(\mathbf{x}_i) + (\mu_k^\top\phi(\mathbf{x}_i))^2 + \alpha_i^2 - 2\alpha_i\mu_k^\top\phi(\mathbf{x}_i). \tag{4}$$

This implies a posterior distribution defined by $\hat{\mu}, \hat{\Sigma}$, where:

$$\hat{\Sigma}_k^{-1} = \bar{\Sigma}^{-1} + 2\sum_{i=1}^{n}\lambda(\xi_{i,k})\phi(\mathbf{x}_i)\phi(\mathbf{x}_i)^\top \tag{5}$$

$$\hat{\mu}_k = \hat{\Sigma}_k\left[\bar{\Sigma}_k^{-1}\bar{\mu}_k + \sum_{i=1}^{n}\left(y_{i,k} - \frac{1}{2} + 2\alpha_i\lambda(\xi_{i,k})\right)\phi(\mathbf{x}_i)\right] \tag{6}$$

are the updates for a prior parameterized by $\bar{\mu}, \bar{\Sigma}$. These equations create the EM algorithm shown in Algorithm 1, which performs an update to our $\mathbf{W}$ distribution.

In order to make predictions about new points we need to evaluate the following expectation:

$$\hat{P}(y = k|\mathbf{x}) = \mathbb{E}_{\mathbf{W}}\left[\text{softmax}(\mathbf{W}\phi(\mathbf{x}))\right]_k. \tag{7}$$

There is no closed-form solution, so this requires approximation. Instead of sampling, we use a more computationally efficient approximation described in [16]:

$$\mathbb{E}_{\mathbf{W}}\left[\text{softmax}(\mathbf{W}\phi(\mathbf{x}))\right]_k \approx \frac{1}{2 - c + \sum_{i\neq k}\mathbb{E}[\sigma(\tilde{\mathbf{z}}_i)]^{-1}},$$

with $\tilde{\mathbf{z}}_i = [\mathbf{W}\phi(\mathbf{x})]_k - [\mathbf{W}\phi(\mathbf{x})]_i$. When combined with the sigmoidal approximation in Equation (2), this becomes an easily computable approximation to Equation (7).

V. Object-Centric Negative Sampling

Similar to many mapping methods, V-PRISM requires negatively sampling points along depth camera rays. The traditional negative sampling used, mentioned in Section II performs poorly when the goal is to map an object resting on a tabletop or other surface. To fully utilize the tabletop structure within the environment, we propose a new negative sampling method for object-centric mapping based on two realizations: (1) negative samples are most useful when near known objects; (2) points below a surface plane cannot be occupied by objects resting entirely on or above that surface.

Our sampling method begins with a segmented point cloud of the scene. We perform stratified uniform sampling along each depth ray, only keeping points that are within r_{obj} distance from at least one object center. Kept points are labeled as unoccupied and added to the augmented dataset. Next, we run RANSAC [17] on the observed point cloud to recover the table plane. Then, we uniformly randomly sample points within r_{obj} from each object center and keep any such points that fall below the plane. Again, kept points are stored and labelled as unoccupied. Lastly, we perform grid subsampling as described in [18] in order to reduce the number of points. We use different resolutions to subsample empty points and points on object surfaces.

The resulting points are then transformed according to Equation (1) to construct our set of augmented data. We choose a set of hinge points consisting of a fixed grid around the scene as well as a fixed number of random points sampled from the surface points of each object.

VI. Experiments

A. Baseline and Metrics

Baseline: Our baseline is a learning-based approach. We use the PointSDF architecture from [5] with a sigmoid final activation. We train this model on a dataset of scenes similar to those discussed in Section VI-B composed of objects from a subset of the ShapeNet [19] dataset. We refer to this baseline as **PointSDF**.

Metrics: We use two main metrics for comparison: **intersection over union (IoU)** and **Chamfer distance (CD)**. IoU is calculated by evaluating points in a fixed grid around each object. Chamfer distance is calculated by first running the marching cubes algorithm [14] on a level set of the prediction function, then sampling surface points.

B. Generated Scenes from Benchmark Object Datasets

In this section, we evaluate our method against the PointSDF baseline and ablate our sampling method on procedurally generated scenes. We generate a scene by randomly placing meshes drawn from the ShapeNet [19], YCB [20],

Method	ShapeNet Scenes IoU	ShapeNet Scenes CD	YCB Scenes IoU	YCB Scenes CD	Objaverse Scenes IoU	Objaverse Scenes CD
PointSDF	**0.360**	**0.010**	0.460	0.015	0.347	0.025
V-PRISM	0.309	0.011	**0.500**	**0.012**	**0.464**	**0.018**

Table I: Quantitative experiments comparing our method to two baseline methods on procedurally generated scenes.

Method	ShapeNet	YCB	Objaverse
w/ BHM Sampling	0.156	0.313	0.326
V-PRISM (ours)	**0.309**	**0.500**	**0.464**
w/o Under the Table	0.291	**0.500**	0.439
w/o Stratified Sampling	0.145	0.294	0.291

Table II: IoU Ablation experiments on our negative sampling method on procedurally generated datasets.

and Objaverse [21] datasets. We generate 100 scenes of up to 10 objects for each mesh dataset.

Our first experiment on simulated scenes compares our method with the PointSDF baseline. We report the IoU and Chamfer distance in Table I. PointSDF outperforms our method on the ShapeNet scenes, where the meshes are drawn from the same mesh dataset that it was trained on. On all other datasets, our method outperforms PointSDF. The performance of our method relative to our baseline indicate that our method results in accurate reconstructions.

Our second experiment on simulated scenes ablates our negative sampling method. We observe the effect of removing key components of our technique. To remove the stratified sampling with discrete, we use fixed steps along each ray. We also compare against the original BHM sampling method explained in [12], labeled as **BHM Sampling**. The IoU for each generated dataset is reported in Table II. Our negative sampling method outperforms the others on each dataset and metric. This implies that our proposed sampling method improves reconstruction quality compared to alternatives. .

C. Real World Scenes

We evaluate our method by qualitatively comparing reconstructions on real world scenes. In order to get accurate segmentations of the scene, we use the model from [22]. We compute on five scenes consisting of multiple objects. We compare our method to PointSDF. The qualitative comparison can be seen in Figure 3. PointSDF struggles to coherently reconstruct the scene. In contrast, our method is capable of producing quality reconstructions even with very noisy input point clouds. This suggests that our method is capable is robust to unknown, noisy scenes.

D. Principled Uncertainty

We perform qualitative experiments on the uncertainty metric of our maps. We measure uncertainty with entropy:

$$H_m(\mathbf{x}) = -\sum_{k=1}^{c} \hat{P}(y=c|\mathbf{x}) \ln \hat{P}(y=c|\mathbf{x}).$$

We compare our method with an alternate version of our method, where we train a single weight vector with stochastic gradient descent (SGD) instead of the EM algorithm.

We calculate this uncertainty over a 2D slice for each of our 5 real world scenes, which can be seen in Figure 4. Qualitatively, we can see that our method obtains high uncertainty values in occluded sections of the scene. The heat maps suggest V-PRISM captures principled uncertainty.

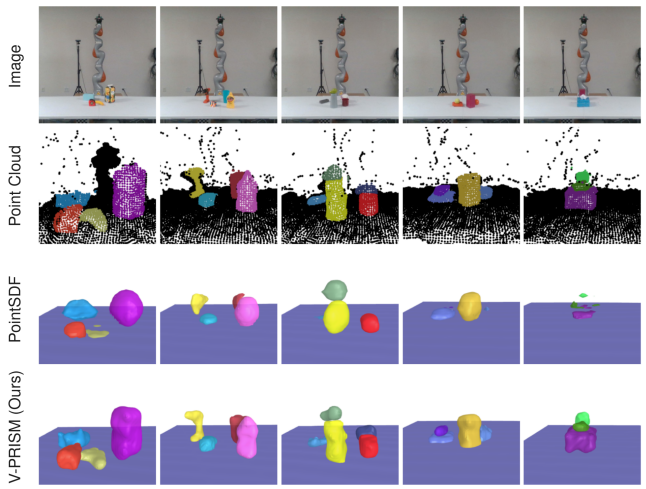

Figure 3: Qualitative comparisons with PointSDF reconstructions. **First row:** RGB images. **Second row:** the segmented point cloud used as input. **Third row:** PointSDF reconstructions. **Last row:** V-PRISM's (our method) reconstructions. V-PRISM results in quality reconstructions on noisy scenes.

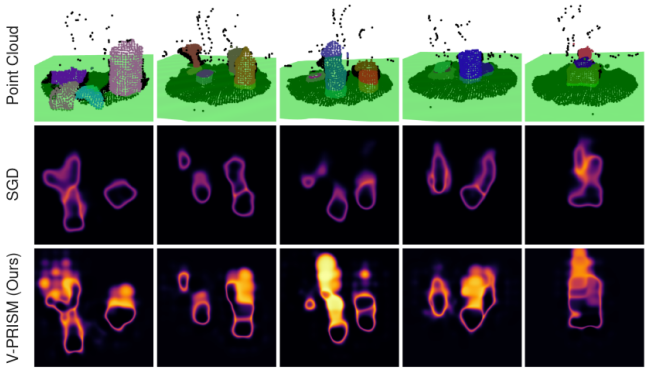

Figure 4: Qualitative comparison of uncertainty. **Top row:** the observed point cloud with a green plane corresponding to the 2D slice of the heat maps. We compare a non-probabilistic alternative of V-PRISM (**middle row**) and our method (**bottom row**). In the heat maps, the bottom is closer to the camera and the top is farther. Lighter areas correspond to more uncertainty. Our method predicts high uncertainty in occluded areas.

VII. Conclusion

Principled uncertainty is necessary for the safety of many robotics tasks. We proposed a framework for robustly constructing multi-class 3D maps of tabletop scenes named V-PRISM. Our method works by iterating an EM algorithm on augmented data to produce a volumetric Bayesian segmentation map. To fully incorporate information from depth measurements, we proposed a novel negative sampling technique. Our maps were shown to have desirable properties including robustness, quality reconstructions, and accurate uncertainty measures through both quantitative and qualitative experiments.

REFERENCES

[1] N. Gothoskar, M. Cusumano-Towner, B. Zinberg, M. Ghavamizadeh, F. Pollok, A. Garrett, J. Tenenbaum, D. Gutfreund, and V. Mansinghka, "3dp3: 3d scene perception via probabilistic programming," *Advances in Neural Information Processing Systems*, vol. 34, pp. 9600–9612, 2021.

[2] C. B. Choy, D. Xu, J. Gwak, K. Chen, and S. Savarese, "3d-r2n2: A unified approach for single and multi-view 3d object reconstruction," in *Computer Vision–ECCV 2016: 14th European Conference, Amsterdam, The Netherlands, October 11-14, 2016, Proceedings, Part VIII 14*, pp. 628–644, Springer, 2016.

[3] S. Tulsiani, S. Gupta, D. F. Fouhey, A. A. Efros, and J. Malik, "Factoring shape, pose, and layout from the 2d image of a 3d scene," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 302–310, 2018.

[4] J. J. Park, P. Florence, J. Straub, R. Newcombe, and S. Lovegrove, "Deepsdf: Learning continuous signed distance functions for shape representation," in *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 165–174, 2019.

[5] M. Van der Merwe, Q. Lu, B. Sundaralingam, M. Matak, and T. Hermans, "Learning continuous 3d reconstructions for geometrically aware grasping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pp. 11516–11522, IEEE, 2020.

[6] X. Zhang, Z. Zhang, C. Zhang, J. Tenenbaum, B. Freeman, and J. Wu, "Learning to reconstruct shapes from unseen classes," *Advances in neural information processing systems*, vol. 31, 2018.

[7] L. Li, S. Khan, and N. Barnes, "Silhouette-assisted 3d object instance reconstruction from a cluttered scene," in *Proceedings of the IEEE/CVF International Conference on Computer Vision Workshops*, pp. 0–0, 2019.

[8] W. Agnew, C. Xie, A. Walsman, O. Murad, Y. Wang, P. Domingos, and S. Srinivasa, "Amodal 3d reconstruction for robotic manipulation via stability and connectivity," in *Conference on Robot Learning*, pp. 1498–1508, PMLR, 2021.

[9] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," in *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings* (Y. Bengio and Y. LeCun, eds.), 2015.

[10] A. Nguyen, J. Yosinski, and J. Clune, "Deep neural networks are easily fooled: High confidence predictions for unrecognizable images," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, pp. 427–436, 2015.

[11] F. Ramos and L. Ott, "Hilbert maps: Scalable continuous occupancy mapping with stochastic gradient descent," *The International Journal of Robotics Research*, vol. 35, no. 14, pp. 1717–1730, 2016.

[12] R. Senanayake and F. Ramos, "Bayesian hilbert maps for dynamic continuous occupancy mapping," in *Conference on Robot Learning*, pp. 458–471, PMLR, 2017.

[13] T. S. Jaakkola and M. I. Jordan, "A variational approach to bayesian logistic regression models and their extensions," in *Sixth International Workshop on Artificial Intelligence and Statistics*, pp. 283–294, PMLR, 1997.

[14] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," in *Proceedings of the 14th Annual Conference on Computer Graphics and Interactive Techniques*, SIGGRAPH '87, (New York, NY, USA), p. 163–169, Association for Computing Machinery, 1987.

[15] G. Bouchard, "Efficient bounds for the softmax function and applications to approximate inference in hybrid models," in *NIPS 2007 workshop for approximate Bayesian inference in continuous/hybrid systems*, vol. 6, 2007.

[16] J. Daunizeau, "Semi-analytical approximations to statistical moments of sigmoid and softmax mappings of normal variables," *arXiv preprint arXiv:1703.00091*, 2017.

[17] M. A. Fischler and R. C. Bolles, "Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography," *Communications of the ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[18] H. Thomas, *Learning new representations for 3D point cloud semantic segmentation*. PhD thesis, Université Paris sciences et lettres, 2019.

[19] A. X. Chang, T. Funkhouser, L. Guibas, P. Hanrahan, Q. Huang, Z. Li, S. Savarese, M. Savva, S. Song, H. Su, J. Xiao, L. Yi, and F. Yu, "ShapeNet: An Information-Rich 3D Model Repository," Tech. Rep. arXiv:1512.03012 [cs.GR], Stanford University — Princeton University — Toyota Technological Institute at Chicago, 2015.

[20] B. Calli, A. Singh, A. Walsman, S. Srinivasa, P. Abbeel, and A. M. Dollar, "The ycb object and model set: Towards common benchmarks for manipulation research," in *2015 international conference on advanced robotics (ICAR)*, pp. 510–517, IEEE, 2015.

[21] M. Deitke, D. Schwenk, J. Salvador, L. Weihs, O. Michel, E. VanderBilt, L. Schmidt, K. Ehsani, A. Kembhavi, and A. Farhadi, "Objaverse: A universe of annotated 3d objects," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 13142–13153, 2023.

[22] A. Kirillov, E. Mintun, N. Ravi, H. Mao, C. Rolland, L. Gustafson, T. Xiao, S. Whitehead, A. C. Berg, W.-Y. Lo, P. Dollár, and R. Girshick, "Segment anything," *arXiv:2304.02643*, 2023.