Dense Spatial Segmentation from Sparse Semantic Information

Qiaojun Feng Yue Meng Nikolay Atanasov
Electrical and Computer Engineering Department
University of California, San Diego
La Jolla, California 92092
{qjfeng,yum107,natanasov}@ucsd.edu

***Abstract*—This paper develops an environment representation that affords reasoning about the occupancy of space, necessary for safe navigation, and about the identity of objects, necessary for complex task interpretation. The main challenge is to provide accurate dense classification of 3-D space, while observing the limitations of onboard, realtime inference and storage. Our approach constructs a graphical model of object geometry and semantics and extends this sparse graph structure into a tetrahedral decomposition of 3-D space. The resulting mesh map can be used to interpolate the sparse object properties into a dense spatial segmentation.**

I. Introduction

Technological advances and improved affordability of embedded sensing and computation create an opportunity for autonomous robot systems to accomplish increasingly complex tasks in unstructured environments. Robots need to reason about the occupancy of space to guarantee safe navigation and about the identity of objects to accommodate high-level task specifications. The limitations of onboard processing and the need for efficient inference and minimal latency of decision making place strict requirements on the size and complexity of artificial perceptual models. The objective of this paper is to design an environment representation that unifies geometry (occupancy of space) and semantic information (object classes) and enables dense, yet efficient spatial reasoning. Our technical approach is to detect objects in streaming image data and extract *semantic keypoints* corresponding to mid-level object parts [19] such as the windshield, doors, and wheels of a car. The semantic keypoints and camera poses over time are represented as the nodes of graphical model with edges capturing constraints among the keypoints (object structure), among the camera poses (sensor odometry) and across camera poses and keypoints (camera projective model). Our **main contribution** is to extend the graphical structure into a tetrahedral decomposition of space that allows us to extrapolate the sparse metric-semantic information at the nodes to a dense segmentation of space. The advantage of this representation is that it contains a sparse substructure allowing efficient inference and storage and yet affords dense classification of space where necessary.

Related work in robotics and computer vision is specialized only to narrow aspects of the problem. For example, visual-inertial odometry [2, 6, 8, 21] offers impressive tracking over long trajectories but relies only on sparse geometric points (ORB, SIFT, etc. features) and cannot account for the semantic content or dense occupancy structure of space. Neural network [12, 9, 25, 10] and structured object [4, 7] architectures have been extremely successful for object recognition, segmentation, and scene understanding but do not provide global positioning of the semantic content. Keypoints of objects can carry useful information such as structure, pose and scale of objects yet maintain data-efficiency. [16, 26, 19] discuss about how to detect and classify keypoints of specific categories. Category-agnostic keypoint detector [28] is also proposed for more general cases.

In SLAM application, map building is tightly combined with localization and affects each other. Some object-based map representation [24, 3] are introduced, mainly considering the data-efficiency when a long-term task is performed. Also object-based method can carry more semantic information which can serve for the following high-level tasks.

Finally, volumetric occupancy mapping techniques [17, 27] allow evaluating the safety and dynamic feasibility of planned motion trajectories but require significant processing and storage, do not capture semantic content, and do not allow for incremental smoothing of past estimates based on new information (e.g., loop closure). Efficient mesh-based mapping methods [23, 15, 20] can distinguish the free and occupied space by assigning occupancy status to each tetrahedron. However semantic information is seldom included.

II. Problem Formulation

Given a sequence of RGBD images $\{i_1, i_2, \dots, i_n\}$, a corresponding camera trajectory $\{x_1, x_2, \dots, x_n\} \subset SE(3)$, and an object detector for C different object classes, our objective is to build a representation $f : \mathbb{R}^3 \to \{0\} \cup \{1, \dots, C\}$, capable of classifying arbitrary 3-D points into unoccupied (0) or belonging to one of the detectable classes. Our goal is to have an efficient (based on few elements), yet accurate (capable of dense classification) representation.

III. Technical Approach

A. Semantic Keypoint Extraction

Drawing inspiration from [29] and [5], we first compress the image information into a collection of class-specific keypoints and associate them to the camera trajectory in a graphical model. Given an image i, we run object detection [22] to

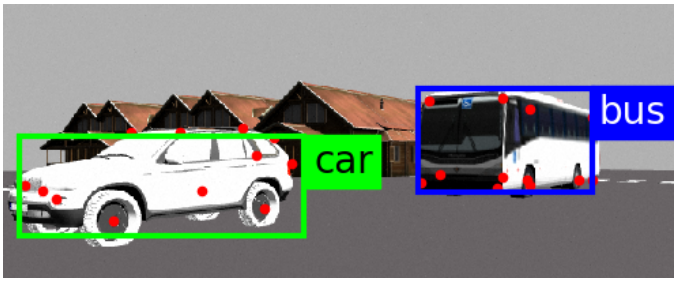

Fig. 1: RGB image from a simulated environment overlaid with the bounding boxes (green and blue) of detected objects (car and bus) and associated semantic keypoints (red).

obtain bounding boxes (see Fig. 1). We extract semantic keypoints corresponding to mid-level object parts (e.g., left front wheel of a bus) from each bounding box using the approach of [19]. In detail, we use a stacked hourglass neural network [18], composed of two sets of convolution and deconvolution layers with an intermediate supervision layer that ensures nonvanishing gradients [13]. The output of the network is a set of heatmaps, each of which indicates the confidence in the 2D location of a corresponding object-specific keypoint. In this preliminary work, we assume that the data association both among object detections and among object-specific keypoints over time is known. We also assume that depth information is available, simplifying the process of projecting the 2-D semantic keypoints to 3-D space.

B. From 2-D Information to 3-D Structure

Using the image depth and the known intrinsic and extrinsic camera parameters, we compute the 3-D positions of the semantic keypoints in the world frame. Since we assume known camera localization and data association, the main source of error in the keypoint position estimates is due to uncertainty in depth (due to occlusions or limited field of view) and in the 2-D keypoint localization. We develop a smoothing technique to obtain accurate global keypoint positions over time. At time step t, we store all valid keypoints associated with object k organized by keypoint ID as $\mathcal{S} := \{S_1, S_2, \ldots, S_{M_k}\}$. Here, each S_i is a set that is augmented with newly estimated positions for keypoint i over time. Given the keypoint sets $\mathcal{S}$ associated with object k and a prior model of the object's keypoints $\mathcal{L} := \{l_1, l_2, \ldots, l_{M_k}\}$, we seek to find a transformation and scaling of $\mathcal{L}$ to accurately match the estimates in $\mathcal{S}$. First, we measure the compactness of each cluster by calculating the mean distances from its keypoints to its centroid:

$$\frac{1}{|S_i|}\sum_{z\in S_i}\left\| z - \frac{1}{|S_i|}\sum_{q\in S_i} q \right\|_2 \tag{1}$$

The compactness measure is used to select the three most compact clusters $S_{p_1}, S_{p_2}, S_{p_3}$ and seek the three corresponding keypoints in the model $\mathcal{L}$, denoted as $l_{p_1}, l_{p_2}, l_{p_3}$. To determine these keypoints, we need to obtain a transformation (translation p, rotation R, scaling s) that minimizes the L2-norm between the clusters and the model keypoints:

$$\begin{aligned} &\arg\min_{s,R,p} \sum_{j=1}^{3}\sum_{z\in S_{p_j}} ||Rz + p - \frac{l_{p_j}}{s}||_2 \\ &s.t. \quad s>0,\ R\in SO(3) \end{aligned} \tag{2}$$

Using the cluster centroids $\bar{S}_{p_j} = \frac{1}{|S_{p_j}|}\sum_{q\in S_{p_j}} q$ and the corresponding keypoints in our model l_{p_j} for $j = 1, 2, 3$, we estimate the scale s from the ratio of two triangles:

$$s = \sqrt{\frac{\|(l_{p_1} - l_{p_2}) \times (l_{p_1} - l_{p_3})\|_2}{\|(\bar{S}_{p_1} - \bar{S}_{p_2}) \times (\bar{S}_{p_1} - \bar{S}_{p_3})\|_2}} \tag{3}$$

Then, denoting the size of the clusters by $N = \sum_{j=1}^{3} |S_{p_j}|$, we can obtain p and R in closed form by method in [11],

$$\begin{cases} R = Z \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & \mathbf{det}(ZL^T) \end{bmatrix} L^T \\ p = \frac{1}{sN}\sum_{j=1}^{3} |S_{p_j}| \cdot (l_{p_j} - sR\bar{S}_{p_j}) \end{cases} \tag{4}$$

where Z and L are obtained from the singular value decomposition $M = Z\Sigma L^T$ of

$$M = \sum_{j=1}^{3}(l_{p_j} - \frac{1}{N}\sum_{i=1}^{3}|S_{p_i}|\cdot l_{p_i}) \sum_{z\in S_{p_j}} (z - \frac{1}{N}\sum_{i=1}^{3}|S_{p_i}|\cdot \bar{S}_{p_i})^T$$

Finally, using s, p, and R, we can reconstruct the 3-D keypoint positions in the world frame. In future work, we intend to used pose graph optimization to improve this process and remove the dependence on depth. Also, deformable shape object models generated from annotated CAD models [19] can add additional constraints to this inference process.

C. From Sparse Structure to Dense Segmentation

Mesh-representation can combine both sparse structure and dense segmentation. Sparse structure only keeps a few points and edges. And we can interpolate a new query point by checking its neighboring saved points to build the dense segmentation.

Here, the environment is defined as the convex hull containing semantic keypoints of objects together with the camera points. Camera points represent the camera's locations along the trajectory and they are assigned to be free. Semantic keypoints are assigned to be occupied because they are points of objects. Delaunary triangulation in 3-D space subdivides the convex hull into tetrahedra and guarantees that no points are inside of any circumscribed sphere of the tetrahedra. Tetrahedra with both free and occupied vertices enable probabilistic interpolation of the occupancy status of the vertices to the tetrahedron interior.

Fig. 2 shows an example triangulation in 2-D for ease of explanation but our experiments are carried out in 3-D. In 2-D, the space is divided into triangles (rather than tetrahedra). We

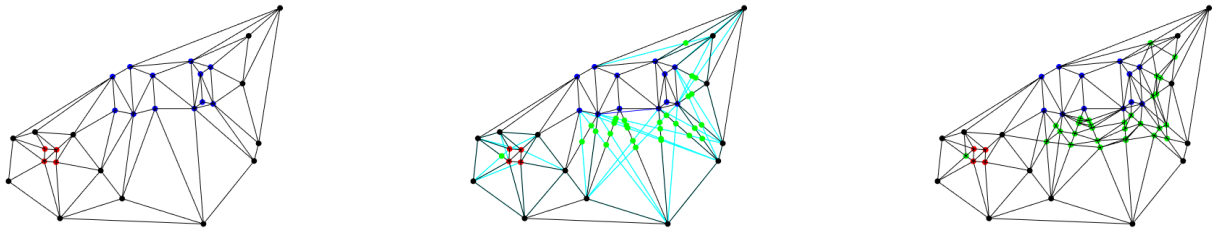

Fig. 2: Triangulation example in 2D case: the original triangulation (left), the addition of auxiliary points (middle), and the refined triangulation (right). The node colors indicate the camera (black), object (red and blue), and auxiliary (green) classes.

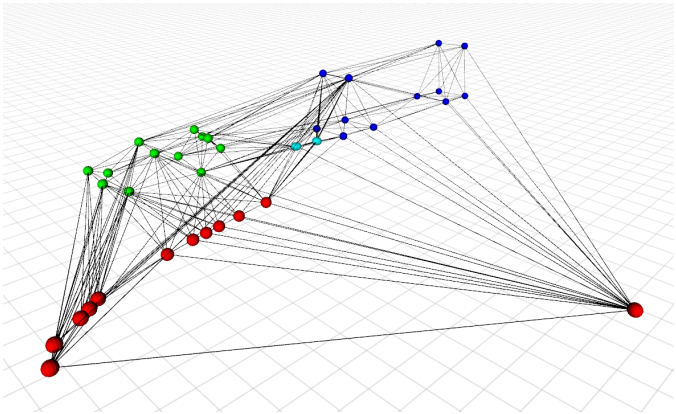

Fig. 3: Tetrahedral decomposition of 3-D space allowing interpolation of sparse properties such as car parts (green), bus parts (blue), and camera trajectory (red) into a dense classification of space. Auxiliary points (cyan) are added at the intersections of camera rays and tetrahedra faces.

look for the intersection of visible constraint edge and triangles edges other than tetrahedra faces. If an object keypoint o_i is visible from a camera point c_j, there should be an edge going through free space between them. We first use Delaunay triangulation to subdivide the convex hull of the point set including object keypoints $\{o\}$ and camera points $\{c\}$. We add the intersections of the rays that connect camera points with visible points with the tetrahedra faces to the point set. We refer to these additional intersection points as auxiliary points. See Fig. 2 for an example. To restrict the number of auxiliary points, we choose only intersections that are outside of the object convex hulls. We set the added auxiliary points $\{a\}$ as free space points, same as the camera points. Then, together with the added auxiliary free space points and the semantic keypoints and camera points, we can re-triangulate the whole space defined by the convex hull of $\{o, c, a\}$.

For the dense reconstruction, we assign a label $c \in \mathbb{R}^{C+1}$ for each point $s \in \mathbb{R}^3$, where C is the number of categories the classifier can distinguish among. Given a point s_0 included in the convex hull we generated, we can check the only tetrahedron that contains s_0 and use barycentric coordinates to represent s_0 as a combination of the vertices of the tetrahedron $v_{1:4}$

$$s_0 = \sum_{i=1}^{4} p_i v_i \quad c_{s_0} = \sum_{i=1}^{4} p_i c_{v_i} \tag{5}$$

where $\sum_{i=1}^{4} p_i = 1$. Notice that for existing vertices v, c_v is a vector with one element as 1 and the others as 0. A threshold is chosen to decide whether a point belongs to an object or the free space.

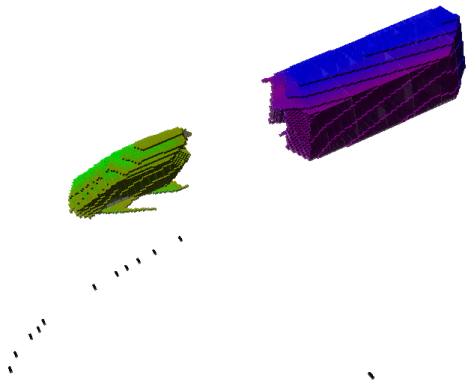

Fig. 4: Dense segmentation of space into free, car (green), and bus (blue) classes based on the tetrahedral model in Fig. 3. Colors indicate the probabilities the points belong to certain object classes, including the camera poses (black).

IV. Evaluation

To detect object bounding boxes we used the YOLOv3 network [22] pretrained on the COCO dataset [14]. We used a pretrained stacked hourglass model [18] to obtain semantic keypoint heatmaps and the followed the procedure in Sec. III-B to localize the keypoints in 3-D space. The proposed method was evaluated in a Gazebo simulation [1], with a car and a bus as foreground and a street view with houses as background. The estimated semantic keypoint positions and the Delaunay triangulation are shown in Fig 3. Fig 4 shows the dense spatial segmentation resulting from a barycentric interpolation on the tetrahedral model. Future work will focus on interpolating variance information from the graphical model to the dense representation and on an extension to a monocular visual inertial system.

Acknowledgments

We gratefully acknowledge support from ARL DCIST CRA W911NF-17-2-0181 and NSF CRII RI IIS-1755568.

References

[1] Gazebo. http://gazebosim.org/. Accessed: 2010-09-30.

[2] M. Bloesch, S. Omari, M. Hutter, and R. Siegwart. Robust visual inertial odometry using a direct EKF-based approach. In *IEEE/RSJ IROS*, pages 298–304, 2015.

[3] S. L. Bowman, N. Atanasov, K. Daniilidis, and G. J. Pappas. Probabilistic data association for semantic slam. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 1722–1729. IEEE, 2017.

[4] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *IEEE CVPR*, 2005.

[5] F. Dellaert. Factor graphs and gtsam: A hands-on introduction. Technical report, Georgia Institute of Technology, 2012.

[6] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. In *arXiv:1607.02565*, 2016.

[7] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object Detection with Discriminatively Trained Part-Based Models. *IEEE Trans. on PAMI*, 32(9):1627–1645, 2010.

[8] C. Forster, Z. Zhang, M. Gassner, M. Werlberger, and D. Scaramuzza. SVO: Semi-Direct Visual Odometry for Monocular and Multi-Camera Systems. *IEEE T-RO*, 2016.

[9] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *IEEE CVPR*, 2014.

[10] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *IEEE CVPR*, 2016. doi: 10.1109/CVPR.2016.90.

[11] B. K. Horn. Closed-form solution of absolute orientation using unit quaternions. *JOSA A*, 4(4):629–642, 1987.

[12] A. Krizhevsky, I. Sutskever, and G. H. ImageNet Classification with Deep Convolutional Neural Networks. In *NIPS*, pages 1097–1105. 2012.

[13] C.-Y. Lee, S. Xie, P. Gallagher, Z. Zhang, and Z. Tu. Deeply-supervised nets. In *Artificial Intelligence and Statistics*, pages 562–570, 2015.

[14] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick. Microsoft coco: Common objects in context. In *European conference on computer vision*, pages 740–755. Springer, 2014.

[15] Y. Ling and S. Shen. Building maps for autonomous navigation using sparse visual slam features. In *Intelligent Robots and Systems (IROS), 2017 IEEE/RSJ International Conference on*, pages 1374–1381. IEEE, 2017.

[16] J. L. Long, N. Zhang, and T. Darrell. Do convnets learn correspondence? In *Advances in Neural Information Processing Systems*, pages 1601–1609, 2014.

[17] R. Newcombe. *Dense Visual SLAM*. PhD thesis, Imperial College London, 2012.

[18] A. Newell, K. Yang, and J. Deng. Stacked hourglass networks for human pose estimation. In *European Conference on Computer Vision*, pages 483–499. Springer, 2016.

[19] G. Pavlakos, X. Zhou, A. Chan, K. G. Derpanis, and K. Daniilidis. 6-dof object pose from semantic keypoints. In *Robotics and Automation (ICRA), 2017 IEEE International Conference on*, pages 2011–2018. IEEE, 2017.

[20] E. Piazza, A. Romanoni, and M. Matteucci. Real-time cpu-based large-scale three-dimensional mesh reconstruction. *IEEE Robotics and Automation Letters*, 3 (3):1584–1591, 2018.

[21] T. Qin, P. Li, and S. Shen. VINS-Mono: A robust and versatile monocular visual-inertial state estimator. In *arXiv preprint:1708.03852*, 2017.

[22] J. Redmon and A. Farhadi. Yolov3: An incremental improvement. *CoRR*, abs/1804.02767, 2018. URL http://arxiv.org/abs/1804.02767.

[23] A. Romanoni and M. Matteucci. Incremental reconstruction of urban environments by edge-points delaunay triangulation. In *Intelligent Robots and Systems (IROS), 2015 IEEE/RSJ International Conference on*, pages 4473–4479. IEEE, 2015.

[24] R. F. Salas-Moreno, R. A. Newcombe, H. Strasdat, P. H. Kelly, and A. J. Davison. Slam++: Simultaneous localisation and mapping at the level of objects. In *Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on*, pages 1352–1359. IEEE, 2013.

[25] K. Simonyan and A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. *arXiv:1409.1556*, 2014.

[26] S. Tulsiani and J. Malik. Viewpoints and keypoints. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1510–1519, 2015.

[27] T. Whelan, R. Salas-Moreno, B., A. Davison, and S. Leutenegger. ElasticFusion: Real-Time Dense SLAM and Light Source Estimation. *IJRR*, 35(14):1697–1716, 2016. doi: 10.1177/0278364916669237.

[28] X. Zhou, A. Karpur, L. Luo, and Q. Huang. Starmap for category-agnostic keypoint and viewpoint estimation. *arXiv preprint arXiv:1803.09331*, 2018.

[29] M. Zhu, X. Zhou, and K. Daniilidis. Single image pop-up from discriminatively learned parts. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 927–935, 2015.