2021 IEEE International Conference on Robotics and Automation (ICRA 2021)
May 31 - June 4, 2021, Xi'an, China

Lightweight Semantic Mesh Mapping for Autonomous Vehicles

Markus Herb[1,2], Tobias Weiherer[1], Nassir Navab[2] and Federico Tombari[2,3]

***Abstract*— Lightweight and semantically meaningful environment maps are crucial for many applications in robotics and autonomous driving to facilitate higher-level tasks such as navigation and planning. In this paper we present a novel approach to incrementally build a meaningful and lightweight semantic map directly as a 3D mesh from a monocular or stereo sequence. Our system leverages existing feature-based visual odometry paired with learned depth prediction and semantic image segmentation to identify and reconstruct semantically relevant environment structure. We introduce a probabilistic fusion scheme to incrementally refine and extend a 3D mesh with semantic labels for each face without intermediate voxel-based fusion. To demonstrate its effectiveness, we evaluate our system in outdoor driving scenarios with monocular depth prediction and stereo and present quantitative and qualitative reconstruction results with comparison to ground truth. Our results show that the proposed approach achieves reconstruction quality comparable to current state-of-the-art voxel-based methods while being much more lightweight both in storage and computation.**

I. Introduction

Navigating unknown environments is a challenging task for robots and humans alike. Particularly in complex surroundings such as urban traffic, maps are an invaluable resource for path planning and for finding the correct way. With the deployment of autonomous vehicles, detailed maps of the road infrastructure are expected to play a vital role to enable high levels of automation, as these provide prior information about the environment complementing onboard sensors. In addition to purely geometric information for obstacle avoidance, semantic information is crucial for interpreting the environment in higher-level planning modules. For the use case of autonomous driving, maps should include elements such as lane markings, drivable road area, traffic signs and traffic lights to follow the rules of the road and enable re-localization within the map.

Over the years many different methods for dense 3D map reconstruction have been introduced, most commonly using either voxel-grids [1] or surfel [2] representations. While these methods offer tremendous reconstruction quality, their deployment in mobile robots including autonomous vehicles is mainly limited by high computational and storage requirements for high-fidelity reconstructions. Accurate reconstruction of fine details such as road markings, poles or traffic signs requires a large number of voxels or surfels. This presents a major challenge for deployment of such systems on robots with limited compute resources in large scale outdoor environments, especially to facilitate applications such as collaborative or crowd-sourced mapping.

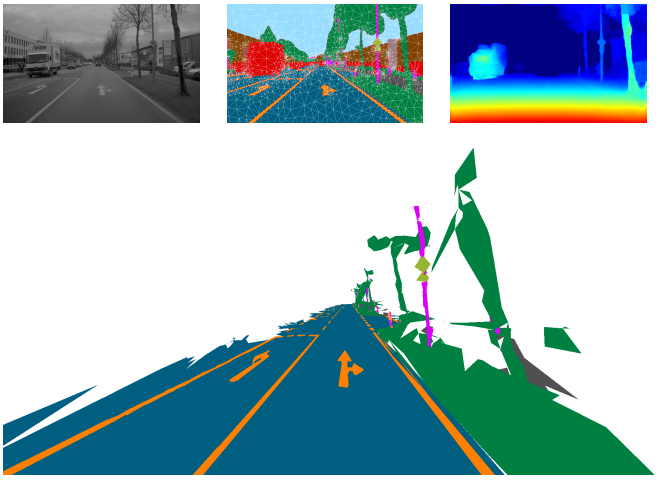


Fig. 1: Dense semantic mesh map reconstruction (bottom) obtained from direct incremental fusion of semantics (top center) and predicted depth (top right) in each keyframe (top left)

To address the limitations of existing approaches for dense mapping, we propose a novel direct mesh reconstruction method using monocular or stereo video, that accurately reconstructs a semantically labelled and lightweight 3D map, while still being computationally efficient. Our system combines semantic image segmentation and depth from deep monocular depth prediction or stereo for each keyframe to recover a semantically labelled 3D mesh as seen in Fig. 1, by directly reconstructing and incrementally updating the mesh from keyframe observations. In contrast to existing methods, we do not use an intermediate discretization into voxels or surfels for reconstructing the 3D mesh, resulting in a small and lightweight map while avoiding potential artifacts caused by spatial discretization.

Our method is grounded on the idea of lifting a triangulated mesh of a 2D semantic segmentation into 3D space, where each mesh face is semantically labelled, such that the mesh resolution adapts to the semantic objects in the scene. We refine and extend the 3D mesh over time by matching the 3D mesh to the 2D semantic mesh in each keyframe, for which we introduce our novel probabilistic matching scheme for 3D to 2D semantic object matching. Furthermore, we model each vertex in the reconstruction as a probability distribution to account for spatial uncertainty and allow fusion of depth estimates. We evaluate our method on two different outdoor driving datasets with stereo and monocular

[1]AUDI AG, Ingolstadt, Germany
[2]Technical University of Munich, Department of Informatics, Germany
[3]Google, Zurich, Switzerland

depth prediction against semantically labelled ground truth and demonstrate that our approach achieves reconstruction results on par with the state of the art while being much more efficient.

II. Related Work

Mapping for robotic applications has been researched intensively for decades within the context of Simultaneous Localization and Mapping (SLAM). Given the vast amount of work in the field, we focus our review on the most related vision-based systems with either semantic or lightweight map representations.

A. Semantic Mapping and Localization

Numerous semantic SLAM approaches were proposed in recent years, due to the large number of applications in robotics and fueled by advancements in semantic segmentation of RGB and LiDAR data using neural networks. While unstructured sparse or semi-dense semantic reconstructions were demonstrated to be beneficial for tasks such as metric localization [3], such maps are typically too difficult to interpret for other robotic tasks such as motion planning due to their sparseness.

Dense semantic reconstruction methods were proposed leveraging monocular cameras [4], stereo cameras [5][1][6], LiDAR [7][2], RGB-D [8] or learned monocular depth prediction [9]. For dense reconstructions, voxel-grids storing either an occupancy map or a truncated signed distance function (TSDF), from which a mesh can be extracted using the Marching Cubes algorithm, are commonly used [10][11]. Other representations include point clouds or surfels, which can be further densified into a mesh using local meshing techniques such as [12] or SurfelMeshing [13]. Such methods can accurately model the environment, but this usually comes at the cost of high computational and storage demand, often requiring GPUs for practical use. While memory-optimized voxel-representations were proposed [14][15], large scale voxel-grids still require relatively large voxel sizes to meet performance requirements, leading to reconstruction artifacts due to discretization. Efficient Surfel-based mapping has also been proposed by Wang *et al.* [16], but their output is not an entirely dense surface due to their use of super-pixels for surfel initialization.

B. Object-based Mapping

Another line of research represents semantic maps by means of individual objects in the scene using pre-defined object models [17], volumetric models [18][19], meshes [20] or quadrics [21]. In addition to these approaches, which are focused mostly on indoor scenes, SegMap [22] was proposed as an object-segment map for outdoor scenes using LiDAR, which however focused more on compact object descriptors for relocalization rather than detailed semantic environment representation.

While object-based systems are well suited for scenes with many distinct components or simple geometric primitives (e.g. planes) usually present in indoor scenes, outdoor scenes often contain only few static and distinct components, such as signs or poles. Much more common are continuous structures such as road markings, barriers, guard rails, fences or buildings, which are often referred to as *stuff* in the context of semantic segmentation. These components typically cannot be described easily as a single object due to their large and complex spatial extent, which motivates our use of a semantic mesh as map representation.

C. Mesh- and Edge-based Mapping

Recently, a number of works tackled the issue of directly creating lightweight maps based on geometric primitives such as meshes. In [23] a method to incrementally reconstruct a manifold mesh from sparse SLAM feature points in real-time is proposed. Rosinol *et al.* [24] also use feature points from visual-inertial odometry and enforce constraints for planar regions optimizing both the mesh reconstruction and odometry, however their approach is targeted primarily at improving the recovered trajectory and only keeps a limited mesh as map. While mesh reconstruction using sparse feature points as vertices fit well into existing feature-based SLAM pipelines, this severely limits the reconstruction quality as the mesh vertices are typically too sparse to represent detailed geometry well.

ScalableFusion [25] directly reconstructs a 3D mesh from RGB-D data in indoor scenes by decoupling the mesh geometry from the color texture to obtain high-resolution colored meshes. Also Rosu *et al.* [26] presented an approach for semantic mapping from LiDAR by storing the semantic information as a texture applied to the geometrical mesh. While decoupling geometry from the semantic texture produces high-quality reconstructions, this requires a reliable initial mesh reconstruction. As demonstrated by [26], storing high-resolution semantic textures for meshes is also very memory intensive for large scale outdoor reconstruction, which limits its use in mobile systems with compute constraints. To obtain a lightweight map representation suited for mobile robots, we instead choose to explicitly couple geometrical structure with semantic meaning by assigning semantic classes directly to mesh faces, which is much more lightweight to store and does not require GPU usage.

Lastly, this work is inspired by our prior work on semantic-edge based mapping [27], in which the outer contours of semantic objects were reconstructed as 3D edges, which is particularly compact though lacking spatial awareness.

III. Semantic Mesh Mapping

An overview of our incremental mesh mapping system is depicted in Fig. 2. Individual parts of the pipeline will be explained in the following subsections.

A. Visual Odometry

Our semantic mesh mapping system is designed to be used in combination with a keyframe-based visual odometry or SLAM system. We base our work on the state-of-the-art feature-based ORB-SLAM2 [28] from which we use the

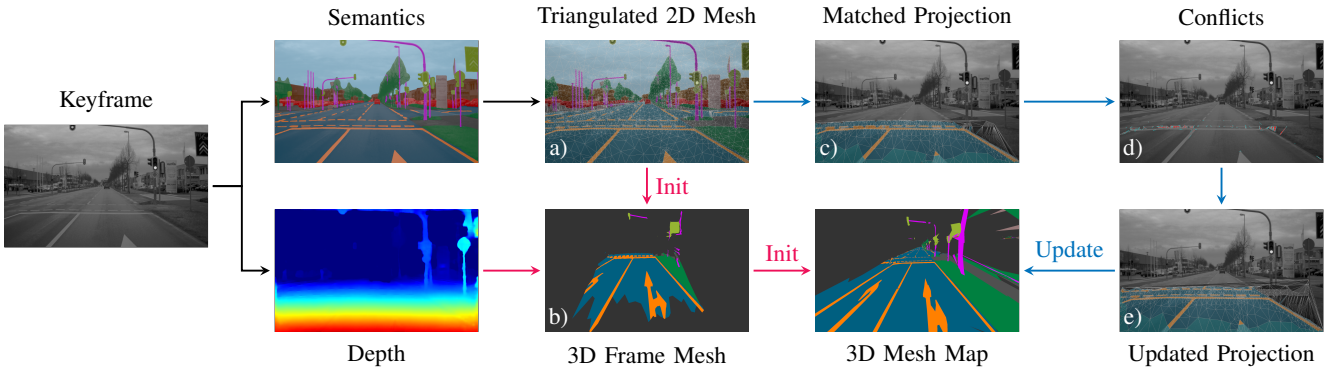


Fig. 2: Pipeline overview of our incremental mesh mapping approach. Input to each update step is a keyframe with semantic segmentation and dense depth including uncertainty. Red arrows denote steps which are part of the map initialization from the first frame, blue arrows denote update steps afterwards.

keyframes as input to our mapping system. Given this loosely coupled architecture, it is easy to combine the mesh mapping system with other SLAM or odometry systems.

B. Keyframe Semantics and Depth

For each keyframe from the visual odometry system, we compute a 2D semantic image segmentation as well as a depth map. For semantics, we use PSPNet [29] with mean-frequency class balancing [30] to favor under-represented but important classes such as thin poles or road markings. Depth is obtained either through stereo or depth prediction from monocular images, both of which we will demonstrate in our evaluation. To obtain an estimate of the uncertainty in the depth prediction, we adapt the method proposed by Tateno *et al.* [9], leveraging the prediction in nearby keyframes.

C. Frame Mesh Extraction

Given the semantic segmentation of a keyframe, we extract a triangulated 2D mesh $\mathcal{M}_\mathsf{F} = (\mathcal{V}_\mathsf{F}, \mathcal{F}_\mathsf{F})$ of 2D vertices $\mathcal{V}_\mathsf{F}$ and faces $\mathcal{F}_\mathsf{F}$, with each face labeled according to its semantic class. For each segment in the frame semantic segmentation, we extract the outer contour of the segment and simplify the contours of adjacent segments to reduce the number of contour points. We then compute a Constrained Delaunay Triangulation using CGAL [31] given all extracted contour points and constrain the edges adjacent to segments of differing semantic classes to ensure that semantic boundaries are represented in the resulting triangulated mesh. To avoid overly large or acute faces, we apply an additional Delaunay Refinement, which ensures that all faces have a defined maximum edge length and minimum angle.

We further define individual object instances by grouping all faces belonging to the same semantic segment. Since computed depth values at depth discontinuities around object boundaries are typically inaccurate and blurry, we apply a two-step process to improve the estimated depth for such vertices. First, we estimate the position and normal for each face from the depth in the center of each face. Then we refine the estimated depth for each vertex by minimizing the distance of the vertex to the adjacent face planes of the same object instance using a single Gauss-Newton iteration. Finally we only keep valid faces at a distance not farther than 20m to avoid uncertain depth measurements too far away. A resulting triangle mesh is shown in Fig. 2a).

D. Mesh Map Initialization

With detected object contours and estimated depth and uncertainties, we can lift the 2D frame triangulation $\mathcal{M}_\mathsf{F}$ to a 3D mesh $\mathcal{M}_\mathsf{M} = (\mathcal{V}_\mathsf{M}, \mathcal{F}_\mathsf{M})$ of vertices $\mathcal{V}_\mathsf{M}$ and faces $\mathcal{F}_\mathsf{M}$ for each object instance, depicted in Fig. 2b). To model the spatial uncertainty of vertices, we model each vertex $v \sim \mathcal{N}(\mathbf{X}_v, \Sigma_v)$ as a normal distribution with mean $\mathbf{X}_v \in \mathbb{R}^3$ and covariance $\Sigma_v \in \mathbb{R}^{3\times3}$. Each face f is assigned a discrete semantic label distribution $\mathcal{L} \in \mathbb{R}^k$ over k classes.

We estimate the covariance in the camera frame Σ_c by propagating the 2D position and inverse depth uncertainties σ_u, σ_v and σ_ρ through the unprojection function $\pi^{-1}(u, v, \rho) \mapsto \mathbf{X} \in \mathbb{R}^3$ and propagate the covariance as

$$\Sigma_c = J_{\pi^{-1}}(u, v, \rho)\ \mathrm{diag}(\sigma_u^2, \sigma_v^2, \sigma_\rho^2)\ J_{\pi^{-1}}^\mathsf{T}(u, v, \rho) \qquad (1)$$

with $J_{\pi^{-1}}$ being the Jacobian of π^{-1} evaluated at its arguments and $\mathrm{diag}(\cdot)$ a diagonal matrix. We then transform the vertex distribution to global coordinates.

E. Semantic Mesh Matching

In order to update mesh vertex positions and faces for each object, we need to match the current keyframe to the existing map mesh. For this, we match existing 3D mesh vertices $\mathcal{V}_\mathsf{M}$ to the current keyframe 2D semantic triangulation $\mathcal{M}_\mathsf{F}$, taking into account the uncertainty of the vertex position. To match a vertex $V = (\mathbf{X}_V, \Sigma_V)$ to the current keyframe, we first transform the distribution from world to camera frame and project to the frame using $\mathbf{x}_v = \pi_\mathsf{K}(\mathbf{X}_V)$. To obtain the corresponding 2D distribution, we again propagate the covariance through π as

$$\Sigma_v = J_\pi(\mathbf{X}_V)\Sigma_V J_\pi^\mathsf{T}(\mathbf{X}_V) \qquad (2)$$

with J_π being the jacobian of π evaluated at $\mathbf{X}_V$. For the matching, we consider three distinct cases, as a vertex can either be an *inner* vertex fully inside an object surface, a

border vertex at the outer border to an object of differing semantic class or a *boundary* vertex at the range boundary where we do not know which of the other two it is. For *inner* vertices, we use the point projection as matching point. For *border* vertices, we try to match the vertex to a corresponding face edge in the 2D triangulation $\mathcal{M}_\mathsf{F}$ with matching semantics. In order to determine the matching point $\tilde{x}_v$, we search for a point along compatible edges that minimizes the Mahalanobis distance for the 2D vertex distribution. This process is also outlined in Fig. 3.

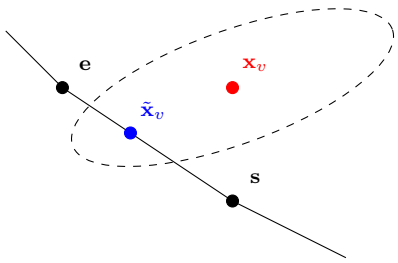


Fig. 3: Probabilistic vertex-edge matching of vertex projection x_v with covariance (dashed) to semantic mesh edge $(\mathbf{s}, \mathbf{e})$. The matched point $\tilde{\mathbf{x}}_v$ minimizes the Mahalanobis distance to x_v along the edge segment (s, e). Remaining mesh structure omitted for clarity.

For each edge with start point $\mathbf{s}$ and end point $\mathbf{e}$, we find the point $\mathbf{p} = \mathbf{s} + t \cdot \mathbf{v}, \mathbf{v} = (\mathbf{e} - \mathbf{s}), t \in [0, 1]$ along the edge that minimizes the squared Mahalanobis distance

$$d_M = \mathbf{p}^\mathsf{T} I \mathbf{p} = (\mathbf{s} + t\mathbf{v})^\mathsf{T} I (\mathbf{s} + t\mathbf{v}) \tag{3}$$

with information matrix $I = {\Sigma_v}^{-1}$.

After initial matching, we verify each match for consistency of the measured and expected depth. We accept a match if the expected inverse depth $\tilde{\rho}$ computed by transforming the vertex into the camera frame is compatible with the measured inverse depth distribution (ρ, σ_ρ^2) according to the χ^2 test

$$\frac{(\tilde{\rho} - \rho)^2}{\sigma_\rho^2} < \mathrm{th}_\rho \tag{4}$$

for a value of threshold $\mathrm{th}_\rho = 3.84$ at 95% as in [32]. Alternatively, if the vertex depth is much greater than the measured depth, such that the χ^2 test does not pass and the semantics in the frame match differs in its semantic class, we label the vertex as occluded. In case of *boundary* vertices, we try both the *inner* and *border* matching strategies and choose the matching with higher consistency.

F. Semantic Mesh Fusion

After matching the 3D map vertices to the current frame, we want to update the 3D map mesh $\mathcal{M}_\mathsf{M}$ from our current frame segmentation $\mathcal{M}_\mathsf{F}$. Since updating a 3D mesh is a non-trivial process, we divide our mesh reconstruction into an *active* mesh triangulation as a 2D mesh $\mathcal{M}_\mathsf{A}$ lifted to 3D space and *inactive* part as a 3D surface mesh $\mathcal{M}_\mathsf{I}$. The active mesh $\mathcal{M}_\mathsf{A}$ contains all faces that are at least partially in the current field of view, while faces that go completely out of view are committed to the inactive surface mesh $\mathcal{M}_\mathsf{I}$ using their full 3D vertex distributions. Each vertex in the active mesh thus is made up of the 2D triangulated mesh vertex in the current frame and the corresponding 3D vertex distribution that is updated over time to refine its position. For clarity we will refer only to a single mesh in the following, even though the process is repeated for every distinct object instance.

The incremental mesh update is performed by propagating the active mesh from frame to frame until it leaves the active area and this process is split into three main steps. These include 1) forming the 2D mesh projection $\mathcal{M}_\mathsf{P}^i$ for the current frame i from the previous active mesh $\mathcal{M}_\mathsf{A}^{i-1}$, 2) conflict detection, updating and extension of the current projection and 3) finally the update of individual vertices to form the propagated active mesh $\mathcal{M}_\mathsf{A}^i$. The individual steps are marked as c)-e) in Fig. 2 and will be explained in the following.

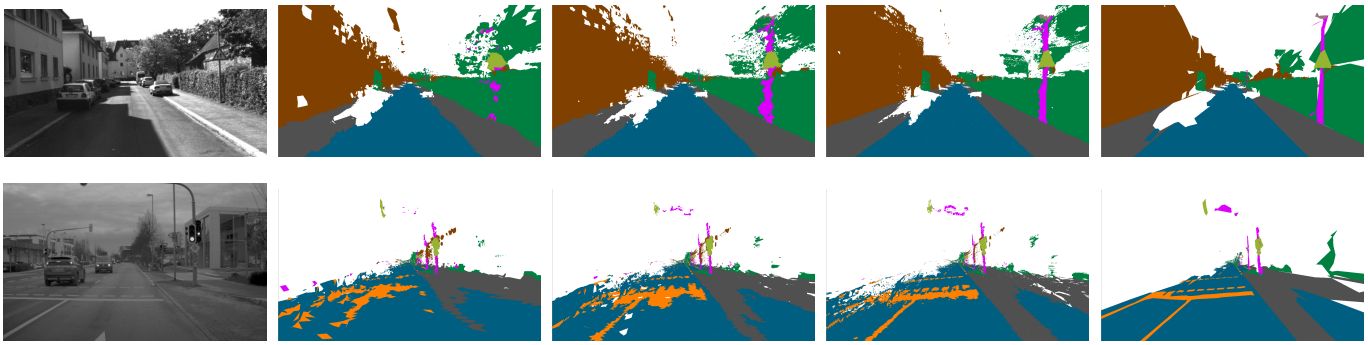
1) Matched Mesh Projection: We compute a mesh projection $\mathcal{M}_\mathsf{P}^i$ in frame i, which will eventually replace our current active mesh, by projecting the active mesh $\mathcal{M}_\mathsf{A}^{i-1}$ from frame $i-1$ into the current frame. To create the projected triangulation, depicted in Fig. 2c), we insert vertices into the projected mesh using their matched point locations while keeping their 3D vertex distributions. We then propagate the label distribution for each face from the active mesh to the projection. Faces that are fully outside the field of view are committed to the inactive mesh using their current vertex distribution and the mode of their label distribution.

2) Conflict Detection, Update & Extension: Having obtained the matched mesh projection $\mathcal{M}_\mathsf{P}^i$, representing the expected view of the current frame, we compare the current frame semantic mesh $\mathcal{M}_\mathsf{F}^i$ with the matched reprojection $\mathcal{M}_\mathsf{P}^i$. For each triangle face in $\mathcal{M}_\mathsf{P}$, we find overlapping faces in $\mathcal{M}_\mathsf{F}$ with differing label and compute the intersection of both faces. The computed intersections represent the areas of conflict between existing mesh and current frame detections. Conflicted polygons are depicted in Fig. 2d).

For each connected component of conflicted faces, we extract the outer contour as a conflict boundary, which we insert as constrained edges into $\mathcal{M}_\mathsf{P}^i$ to reflect the changed mesh structure. The label distribution $\mathcal{L}$ of each face gets updated by increasing the score of the face label detected in the current frame by the weight of the face as computed in [26]. Following this, we collapse edges of tiny faces caused by insertion of the conflict boundaries to avoid an overly complex mesh.

Finally, we insert new vertices and faces from the current frame mesh $\mathcal{M}_\mathsf{F}^i$ into the mesh projection that are now within the applied depth limit of 20m and initialize their label distributions. The result of the full update on $\mathcal{M}_\mathsf{P}^i$ step can be seen in Fig. 2e).

3) Vertex Update: After updating the mesh structure, we update the vertex distribution of all mesh vertices given the matched vertex location and depth estimate in the current frame. As during map initialization, we estimate a measured vertex distribution in global coordinates $v_m = (X_m, \Sigma_m)$



(a) Scene image (b) TSDF 20cm (c) TSDF 10cm (d) TSDF 5cm (e) Ours

Fig. 4: Qualitative mesh reconstruction results for two challenging street scenes from KITTI (top) and our data (bottom). Large voxel sizes (4b, 4c) lead to discretization artifacts in the TSDF reconstruction used in Kimera [33].

given the matched point location (u, v) and inverse depth estimate ρ. To avoid over-estimating the confidence of fused distributions due to highly correlated measurements from similar viewpoints, we use fast Covariance Intersection (CI) [34] instead of naive convex combination of distributions. This yields update formulas of

$$\tilde{\Sigma} = \left(\omega_m {\Sigma_m}^{-1} + \omega_p {\Sigma_p}^{-1}\right)^{-1} \tag{5}$$

$$\tilde{\mathbf{X}} = \tilde{\Sigma}\left(\omega_m {\Sigma_m}^{-1}\mathbf{X}_m + \omega_p {\Sigma_p}^{-1}\mathbf{X}_p\right) \tag{6}$$

with weighting factors ω_m, ω_p computed as in [34] given prior vertex distribution v_p, measured distribution v_m and posterior distribution $\tilde{v}$. Updating the 3D vertex distributions finalizes the mesh update, resulting in the current mesh projection $\mathcal{M}_{\mathrm{P}}^i$ becoming the current active mesh $\mathcal{M}_{\mathrm{A}}^i$.

IV. Evaluation

A. Datasets

We evaluate our approach on the public KITTI dataset [35] as well as on our own data recorded in an urban environment. For KITTI, we use the odometry sequences, for which we use the labeled LiDAR point clouds from SemanticKITTI [36] as ground-truth. We train our segmentation network on CityScapes [37] and fine-train on the official KITTI labels. We use GA-Net [38] to predict the depth from stereo.

For our own dataset instead we use an automotive-grade monocular greyscale camera and a precise RTK-GPS system mounted to the vehicle to record urban driving sequences of around 3km length, for which we obtain a vectorized high-definition map generated from hand-labeled camera and lidar data as ground truth. We use struct2depth [39] to predict depth from monocular frames using velocity supervision [40] for absolute scale. All experiments were run on an Intel i7-6820HQ laptop CPU using 32GB of RAM.

B. Baseline

We compare our approach against TSDF-based reconstruction methods, represented by the semantic reconstruction component of Kimera [33], which extends Voxblox [10] with semantic information. We chose voxel sizes of 5cm, 10cm and 20cm for evaluation and set the truncation threshold to 8 times the voxel size. For comparability, we run the reconstruction on a single core using the *fast* integration method of Voxblox and integrate up to 20m distance.

C. Qualitative Results

We demonstrate qualitative reconstruction results of two complex street scenes in Fig. 4. In the reconstructions of the three baseline configurations and our method, we see that all methods can reconstruct the main parts of the scene such as road, sidewalk, or buildings. However, it is well observable that there are strong discretization artifacts visible in the TSDF reconstructions with larger voxel sizes of 10 and 20cm, particularly for details such as lane markings or poles. Our reconstruction achieves similar reconstruction results as the 5cm TSDF reconstruction for fine details with sharp edges, but requires a much lower number of vertices and faces thanks to the semantic-aware reconstruction. More qualitative results can be seen in our supplementary video.

D. Evaluation Metric

We base our evaluation metric on the Chamfer distance, commonly used as 3D reconstruction metric in computer vision, which measures the distance to the closest reconstructed point from each reference point. Since we are dealing with semantic mesh reconstruction in which the correct semantics is a crucial aspect, we propose a Semantic Chamfer distance, where distances are computed for points of the same semantic class. To compute the distance for a pair of reconstructed and ground truth mesh, we randomly sample points on each mesh surface with an average density of 2500 points per m^2 surface area to obtain point clouds $\mathcal{R}, \mathcal{G}$ for the reconstructed and ground truth mesh, respectively. For each point $g \in \mathcal{G}$ with class c_g and each point $r \in \mathcal{R}$ with class c_r, we compute the Semantic Chamfer Distance $e_{g\to\mathcal{R}}$ and $e_{r\to\mathcal{G}}$ respectively as

$$e_{g\to\mathcal{R}} = \min_{r\in\mathcal{R}, c_g = c_r} \|r - g\| \qquad e_{r\to\mathcal{G}} = \min_{g\in\mathcal{G}, c_r = c_g} \|g - r\|.$$

Using these definitions, we adopt the precision-recall evaluation from [41] to obtain a *precision*, *recall* and *F-Score* for each reconstructed class.

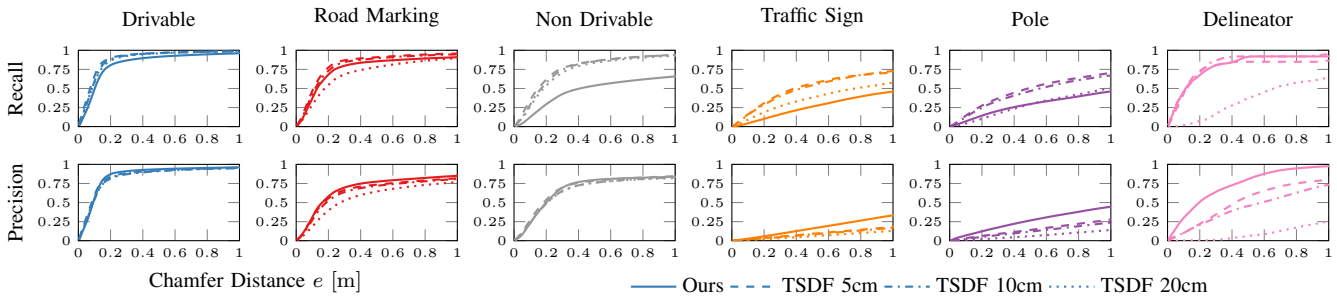


Fig. 5: Precision (bottom) and recall (top) distribution by object class on our urban driving sequence.

TABLE I: Reconstruction F-Score listed by object category for distance threshold $d = 25\text{cm}$ on KITTI sequence 00. Best score per class marked in **bold**, second-best in *italics*.

	Road	Side.	Sign	Pole	Barrier	Building
TSDF 20cm	90.6	75.7	30.3	16.1	39.9	44.2
TSDF 10cm	*91.2*	*77.7*	39.8	51.9	45.0	*51.1*
TSDF 5cm	**91.6**	**79.2**	*49.4*	**71.4**	*51.1*	**57.7**
Ours	89.5	74.1	**53.7**	*58.7*	**53.6**	50.1

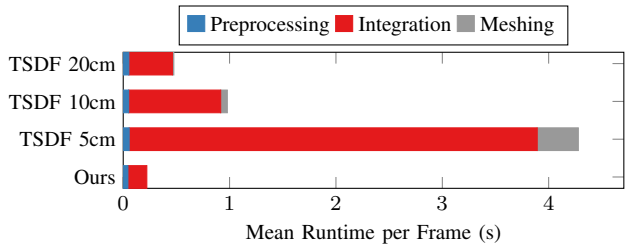


Fig. 7: Mean runtime per frame on a single core laptop CPU

E. Quantitative Results

We show precision and recall distributions in Fig. 5 for six different class categories in our urban driving sequence. For KITTI, we list the respective F-scores in Table I. Generally, we can observe that our method achieves similar overall reconstruction performance on both datasets as the TSDF reconstruction, with some classes outperforming all baselines. In the precision-recall plots, we observe that our method tends to achieve higher precision, but lower recall, particularly for small and thin objects. We further can see from the KITTI results, that fine structures such as signs and poles achieve much higher scores overall thanks to the more accurate depth from stereo compared to the predicted monocular depth used in our data. For the case of non-drivable and side-walk, which can often be occluded by other objects, we observe that voxel raycasting still performs better, because it can handle occlusions more consistently.

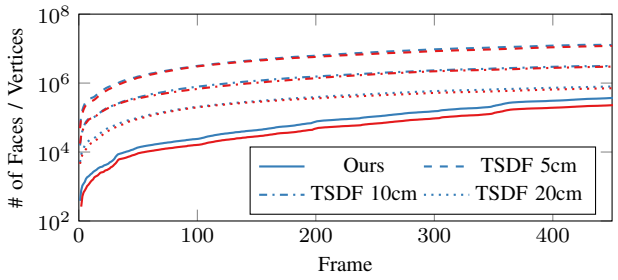


Fig. 6: Number of mesh faces (red) and vertices (blue) after each processed frame. Note the logarithmic scale on y axis.

In addition to reconstruction quality, we show the total vertex and face counts over time in Fig. 6 and mean runtime in Fig. 7. We observe that increasing voxel resolution results in strongly increased reconstructed mesh size and runtime. In contrast, our approach produces mesh reconstructions that are two orders of magnitude smaller while also being almost 10 times faster than the 5cm voxel baseline. With a mean runtime of just over 200ms per frame on a single laptop CPU core, our approach achieves real-time keyframe processing, demonstrating the potential of the approach on resource constrained hardware platforms.

V. Conclusion

We have presented a novel approach for dense semantic map reconstruction from monocular video using semantic image segmentation and learned depth prediction. The main novelty is represented by direct reconstruction of a 3D semantic mesh by combining 2D semantics and depth and tracking the full 3D vertex distribution in the the reconstructed mesh. Quantitative and qualitative evaluation demonstrates that our method can achieve dense mesh reconstruction with similar quality as state-of-the-art voxel-based reconstructions for challenging outdoor scenes. By actively incorporating the semantics in the reconstruction process, our approach requires orders of magnitude less vertices while also being significantly faster compared to a high resolution TSDF reconstruction, achieving real-time rates on single CPU cores. This makes our approach particularly suited for dense map reconstruction in mobile robots with limited resources. Further, it enables collaborative dense mapping in outdoor scenarios such as autonomous driving using fleets of robots with limited communication bandwidth.

For future works, incorporating panoptic segmentation or 3D tetrahedral meshes is an interesting avenue to improve the performance for occluding and self-occluding objects.

REFERENCES

[1] V. Vineet, O. Miksik, M. Lidegaard, M. Niebner, S. Golodetz, V. A. Prisacariu, O. Kahler, D. W. Murray, S. Izadi, P. Peerez, and P. H. S. Torr, "Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction," in *2015 IEEE International Conference on Robotics and Automation (ICRA)*, 2015.

[2] X. Chen, A. Milioto, E. Palazzolo, P. Giguere, J. Behley, and C. Stachniss, "SuMa ++ : Efficient LiDAR-based Semantic SLAM," in *IEEE/RSJ Int. Conf. Intell. Robot. Syst. (IROS)*, 2019.

[3] C. Toft, E. Stenborg, L. Hammarstrand, L. Brynte, M. Pollefeys, T. Sattler, and F. Kahl, "Semantic Match Consistency for Long-Term Visual Localization," *2018 European Conference on Computer Vision (ECCV)*, 2018.

[4] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg, "Joint Semantic Segmentation and 3D Reconstruction from Monocular Video," in *European Conference on Computer Vision (ECCV)*, 2014.

[5] S. Sengupta, E. Greveson, A. Shahrokni, and P. H. S. Torr, "Urban 3D semantic modelling using stereo vision," in *2013 IEEE International Conference on Robotics and Automation*, 2013.

[6] S. Yang, Y. Huang, and S. Scherer, "Semantic 3D occupancy mapping through efficient high order CRFs," in *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[7] L. Sun, Z. Yan, A. Zaganidis, C. Zhao, and T. Duckett, "Recurrent-OctoMap: Learning State-Based Map Refinement for Long-Term Semantic Mapping With 3-D-Lidar Data," *IEEE Robotics and Automation Letters*, vol. 3, no. 4, 2018.

[8] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "SemanticFusion: Dense 3D semantic mapping with convolutional neural networks," in *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 2017.

[9] K. Tateno, F. Tombari, I. Laina, and N. Navab, "CNN-SLAM: Real-Time Dense Monocular SLAM with Learned Depth Prediction," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[10] H. Oleynikova, Z. Taylor, M. Fehr, R. Siegwart, and J. Nieto, "Voxblox: Incremental 3D Euclidean Signed Distance Fields for On-Board MAV Planning," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2017.

[11] A. Millane, Z. Taylor, H. Oleynikova, J. Nieto, R. Siegwart, and C. Cadena, "C-blox: A Scalable and Consistent TSDF-based Dense Mapping Approach," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018.

[12] Z. C. Marton, R. B. Rusu, and M. Beetz, "On fast surface reconstruction methods for large and noisy point clouds," in *2009 IEEE International Conference on Robotics and Automation*, 2009.

[13] T. Schops, T. Sattler, and M. Pollefeys, "SurfelMeshing: Online Surfel-Based Mesh Reconstruction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2019.

[14] M. Nießner, M. Zollhöfer, S. Izadi, and M. Stamminger, "Real-time 3D Reconstruction at Scale using Voxel Hashing," *ACM Transactions on Graphics (TOG)*, 2013.

[15] E. Vespa, N. Funk, P. H. J. Kelly, and S. Leutenegger, "Adaptive-Resolution Octree-Based Volumetric SLAM," in *2019 International Conference on 3D Vision (3DV)*, 2019.

[16] K. Wang, F. Gao, and S. Shen, "Real-time Scalable Dense Surfel Mapping," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019.

[17] 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 *2013 IEEE Conference on Computer Vision and Pattern Recognition*, 2013.

[18] J. Mccormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, "Fusion++: Volumetric Object-Level SLAM," in *2018 International Conference on 3D Vision (3DV)*, 2018.

[19] M. Grinvald, F. Furrer, T. Novkovic, J. J. Chung, C. Cadena, R. Siegwart, and J. Nieto, "Volumetric Instance-Aware Semantic Mapping and 3D Object Discovery," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, 2019.

[20] Q. Feng, M. Yue, M. Shan, and N. Atanasov, "Localization and Mapping using Instance-specific Mesh Models," in *IEEE/RSJ International Conference Intelligent Robots and Systems (IROS)*, 2019.

[21] L. Nicholson, M. Milford, and N. Sunderhauf, "QuadricSLAM: Dual Quadrics from Object Detections as Landmarks in Object-oriented SLAM," *IEEE Robotics and Automation Letters*, 2018.

[22] R. Dubé, A. Cramariuc, D. Dugas, J. Nieto, R. Siegwart, and C. Cadena, "SegMap: 3D Segment Mapping using Data-Driven Descriptors," in *Robotics: Science and Systems XIV*, 2018.

[23] E. Piazza, A. Romanoni, and M. Matteucci, "Real-time CPU-based large-scale 3D mesh reconstruction," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018.

[24] A. Rosinol, T. Sattler, M. Pollefeys, and L. Carlone, "Incremental Visual-Inertial 3D Mesh Generation with Structural Regularities," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019.

[25] S. Schreiberhuber, J. Prankl, T. Patten, and M. Vincze, "ScalableFusion: High-resolution Mesh-based Real-time 3D Reconstruction," in *2019 International Conference on Robotics and Automation (ICRA)*, 2019.

[26] R. A. Rosu, J. Quenzel, and S. Behnke, "Semi-supervised Semantic Mapping Through Label Propagation with Semantic Texture Meshes," *International Journal of Computer Vision*, 2019.

[27] M. Herb, T. Weiherer, N. Navab, and F. Tombari, "Crowd-sourced Semantic Edge Mapping for Autonomous Vehicles," in *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2019.

[28] R. Mur-Artal and J. D. Tardós, "ORB-SLAM2: An Open-Source SLAM System for Monocular, Stereo, and RGB-D Cameras," *IEEE Transactions on Robotics*, vol. 33, no. 5, 2017.

[29] H. Zhao, J. Shi, X. Qi, X. Wang, and J. Jia, "Pyramid Scene Parsing Network," in *2017 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2017.

[30] D. Eigen and R. Fergus, "Predicting Depth, Surface Normals and Semantic Labels with a Common Multi-scale Convolutional Architecture," in *2015 IEEE International Conference on Computer Vision (ICCV)*, 2015.

[31] The CGAL Project, *{CGAL} User and Reference Manual*, 4th ed. CGAL Editorial Board, 2019. [Online]. Available: https://doc.cgal.org/4.14.2/Manual/packages.html

[32] R. Mur-Artal and J. Tardos, "Probabilistic Semi-Dense Mapping from Highly Accurate Feature-Based Monocular SLAM," in *Robotics: Science and Systems XI*, 2015.

[33] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an Open-Source Library for Real-Time Metric-Semantic Localization and Mapping," in *2020 IEEE International Conference on Robotics and Automation (ICRA)*, 2020.

[34] D. Franken and A. Hupper, "Improved fast covariance intersection for distributed data fusion," in *2005 7th International Conference on Information Fusion*, 2005.

[35] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? The KITTI vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012.

[36] J. Behley, M. Garbade, A. Milioto, J. Quenzel, S. Behnke, C. Stachniss, and J. Gall, "SemanticKITTI: A Dataset for Semantic Scene Understanding of LiDAR Sequences," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019.

[37] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele, "The Cityscapes Dataset for Semantic Urban Scene Understanding," in *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2016.

[38] F. Zhang, V. Prisacariu, R. Yang, and P. H. Torr, "GA-Net: Guided Aggregation Net for End-To-End Stereo Matching," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2019.

[39] V. Casser, S. Pirk, R. Mahjourian, and A. Angelova, "Depth Prediction Without the Sensors: Leveraging Structure for Unsupervised Learning from Monocular Videos," in *Thirty-Third AAAI Conference on Artificial Intelligence (AAAI'19)*, 2019.

[40] V. Guizilini, R. Ambrus, S. Pillai, A. Raventos, and A. Gaidon, "3D Packing for Self-Supervised Monocular Depth Estimation," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020.

[41] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and temples," *ACM Transactions on Graphics*, vol. 36, no. 4, 2017.