

Panoptic Multi-TSDFs: a Flexible Representation for Online Multi-resolution Volumetric Mapping and Long-term Dynamic Scene Consistency

Lukas Schmid¹, Jeffrey Delmerico², Johannes L. Schönberger²,
Juan Nieto², Marc Pollefeys^{2,3}, Roland Siegwart¹, and Cesar Cadena¹

Abstract—For robotic interaction in environments shared with other agents, access to volumetric and semantic maps of the scene is crucial. However, such environments are inevitably subject to long-term changes, which the map needs to account for. We thus propose *panoptic multi-TSDFs* as a novel representation for multi-resolution volumetric mapping in changing environments. By leveraging high-level information for 3D reconstruction, our proposed system allocates high resolution only where needed. Through reasoning on the object level, semantic consistency over time is achieved. This enables our method to maintain up-to-date reconstructions with high accuracy while improving coverage by incorporating previous data. We show in thorough experimental evaluation that our map can be efficiently constructed, maintained, and queried during online operation, and that the presented approach can operate robustly on real depth sensors using non-optimized panoptic segmentation as input.

I. INTRODUCTION

Having a geometric and semantic understanding of the world is a crucial capability for autonomous systems to interact with their environment in tasks ranging from collision avoidance and path planning to mobile manipulation or object search. In many applications, these tasks are desirable in environments that are shared with other agents. However, these inevitably induce long-term dynamic changes in the environment that the robot map needs to account for.

In particular, volumetric representations such as occupancy [1] or Truncated Signed Distance Fields (TSDF) [2] have found a lot of success. By dividing the map into a dense grid of voxels, they are able to explicitly represent free space and differentiate between known and unknown regions in the map, which is crucial for online planning. However, this fixed grid structure makes these methods very memory intensive and renders them inflexible when trying to account temporal changes.

Recently, a number of works extended dense maps to also provide semantic information [3]–[9]. Typically, semantic image predictions obtained by Convolutional Neural Networks (CNN) are fused into a global map to estimate the maximum a posteriori labels for each voxel. However, these methods still assume that the environment is static in order to integrate semantics into the fixed geometry.

This work was supported by funding from the Microsoft Swiss Joint Research Center and the European Union’s Horizon 2020 research and innovation programme under grant agreement No 101017008.

¹ Autonomous Systems Lab, ETH Zürich, Zürich, Switzerland

² Mixed Reality and AI Lab, Microsoft, Zürich, Switzerland

³ Computer Vision and Geometry Lab, ETH Zürich, Zürich, Switzerland
schmluk@ethz.ch

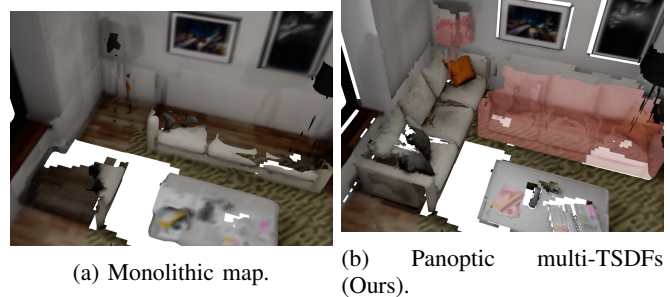


Fig. 1: Qualitative comparison. Our method shows persistent and absent meshes in solid and shaded red, respectively. Our object-oriented approach preserves semantic consistency over time, accurately capturing new objects (left sofa) and removing absent objects (right sofa) as a whole. In contrast, the voxel-based approach keeps artifacts in the map and fails to capture up-to-date geometry. In addition, individual objects (on the table) are not merged together and can be reconstructed at higher resolutions.

In this work, we aim to invert this paradigm and explore how semantic information can be leveraged to improve the modeling of geometry and achieve temporal consistency. The central idea of our approach is that the world typically does not change at random but in a semantically consistent way. We thus propose a novel semantic volumetric map representation that uses the object as the minimal unit of change, rather than the voxel. Based on the panoptic segmentation paradigm [10], we differentiate between object instances, background classes, and free space. In light of recent success of submapping approaches for spatially consistent volumetric mapping [11,12] and moving object reconstruction [5,6,13], we represent the world as a collection of submaps. Each submap contains a locally consistent panoptic entity, i.e. each object, piece of background, or free space is reconstructed individually, such that the collection of submaps together recovers the full volumetric map. We show that this panoptic map representation enables memory efficient multi-resolution volumetric mapping and is able to capture long-term dynamic scene changes during online mapping. We make the following contributions:

- We propose panoptic multi-TSDFs as a novel, flexible multi-resolution volumetric map representation to capture long-term object-level scene changes.
- We present a method for panoptic multi-TSDF integration and map management for temporally consistent mapping during online operation.
- We thoroughly evaluate our approach in simulation and on real world datasets. The code and data is available

as open-source¹.

II. RELATED WORK

A. Dense Semantic Mapping

Dense semantic mapping aims at estimating the semantic label of each surface element. Early works [14] fuse frame-wise geometric segmentations into a global surfel-map. McCormac *et al.* [4] extend surfel-based mapping [15] by fusing 2D semantic predictions and refining using a global Conditional Random Field (CRF).

Similarly, [9] fuse CNN predictions in a Bayesian way into a volumetric map based on voblox [2]. Grinvald *et al.* [8] extend [2] by combining geometric segmentation with instance predictions from MaskRCNN [16] to refine label boundaries. A panoptic approach also based on [2] is presented in [3], where CNN background labels are combined with instance predictions [16] to achieve the panoptic labeling. While the TSDF-based methods can supply the volumetric information needed for planning, all of the above approaches make the limiting assumption that the environment is static.

B. Object-centric Mapping

A different family of approaches focuses on reconstructing selected individual objects. This was pioneered in SLAM++ [17], where models of known objects are fitted to sensor data and act as nodes in graph-based sparse SLAM. This constraint is relaxed in Fusion++ [18]. Similar to us, each object is reconstructed in its own TSDF volume and segmented by estimating a foreground probability, giving flexibility to the system to account for pose estimation errors. However, only selected objects are reconstructed, thus not providing the volumetric information required for planning. Furthermore, the environment is considered static.

A number of works leverage this approach to capture short-term dynamics, i.e. objects moving in front of the camera. Rünz *et al.* [6] track objects using geometric and photometric alignment. They are segmented based on motion cues or semantic segmentation and reconstructed using surfel fusion. In a similar approach, MaskFusion [7] combines geometric and instance segmentation [16] for improved object recognition. Strecke *et al.* [5] extend [18] to moving objects, estimating camera and object poses in an expectation-maximization scheme. In a TSDF approach based on [19], MID-Fusion [13] combines the segmentation of [7] with motion cues to reconstruct multiple moving objects. Long *et al.* [20] additionally include motion tracking to reconstruct a single large moving object.

While significant progress in reconstructing selected individual objects was made, these approaches are usually confined to small environments with few tracked objects. Non-moving objects and background are not considered and assumed static, thus making these approaches not well suited to capture long-term changes.

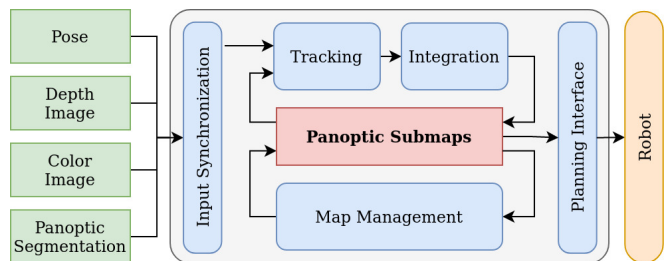


Fig. 2: System overview.

C. Change Detection

TSDFs have also found success in capturing long-term changes. Finman *et al.* [21] generate multiple reconstructions [22] and identify changes via surface point cloud differencing. Fehr *et al.* [23] directly combine different observations into a multi-layer TSDF and use volumetric differencing to extract the static background and movable objects. A recent approach [24] additionally leverages semantic information to identify support surfaces and movable objects to improve change detection. However, all of these methods can only operate post-hoc and are computationally demanding, making them unsuitable for online operation of robots in shared environments.

D. Online Long-term Mapping

Another line of works tackles the problem of incremental long-term mapping. Krajník *et al.* [25] augment 2D occupancy maps to estimate the temporal presence of each voxel as a frequency. Lázaro *et al.* [26] represent the world as 2D point cloud submaps and apply a map management strategy, similar measurements are fused and differing data is overwritten by the most recent estimate. Tang *et al.* [27] build a graph of submaps connected by their poses. When new submaps can not be re-localized, they are also added as temporal information to the spatial graph. Alternatively, Macenski *et al.* [28] add a temporal decay to voxels. As old voxels are removed the map is kept up to date but also loses previous information. Mason and Marthi [29] propose an object-based approach, where an object is any point cloud supported by a plane. Persistence of these sparse objects is then tracked by comparing their convex hulls projected onto the support plane.

A limitation of these approaches is that they lack the expressiveness, i.e. volumetric and semantic information, needed for robot interaction and do not provide semantic consistency when accounting for changes, thus leaving undesirable artifacts in the map.

III. APPROACH

The central idea of our approach is to leverage panoptic segmentation information as the governing factor in representing, building, and maintaining volumetric maps during online operation. The goal of our work is not to optimize the semantic labeling, but rather to explore how high-level information can be leveraged for multi-resolution 3D reconstruction and temporal consistency. An overview of our system is given in Fig. 2. The inputs to our pipeline are depth and color images, e.g. from a RGBD sensor. We take

¹https://github.com/ethz-asl/panoptic_mapping

robot poses from an external estimator, allowing for a broad range of sensors and systems, e.g. [30]–[32], to be employed. Lastly, panoptic segmentation can be predicted from the color and depth information. To demonstrate the robustness of our method with respect to imperfect segmentation, we directly use the output of [33] as input to our system. Nonetheless, our method can readily integrate other segmentation improvements such as [3,4,7,8,13,14].

A. Map Representation

Our map representation is based on the observation that the world typically does not change at random but in a semantically consistent way. To capture this feature, we use the object as the minimal unit of change and propose to represent the world as a collection of panoptic entities, structured as submaps. In this formulation, we differentiate between three panoptic labels, being *objects*, *background*, and *free space*. Each submap contains the geometry of one entity, i.e. of an object instance, a background class, or free space, such that all submaps together constitute the full volumetric map. To guarantee temporal consistency of each submap, we further differentiate between *active* and *inactive* submaps, denoting active those currently being tracked and built, and inactive submaps from past observations.

For efficient processing, a hierarchical structure illustrated in Fig. 3 is employed. On the highest level lies the submap collection, where a spatial index is maintained for constant time scaling in large-scale environments. Each submap contains all related data such as panoptic, instance, and class labels, as well as transformation and tracking data. To represent geometry, we choose to use TSDF grids [2] for their ability to fuse multiple observations. The space containing an object is partitioned into blocks, where only blocks containing surface information are allocated, except for free space submaps. For efficient traversal of the submap collection, each object has a sphere spanned by the blocks as bounding volume. Each block contains a dense grid of voxels that store the TSDF values representing the surface.

This hierarchical structure allows for efficient queries of the submap collection at all stages of the pipeline. In addition, each object can be reconstructed at a different resolution and only takes up the memory required to represent its surface, while the full volumetric information can be recovered from the collection. Most importantly, semantic consistency is maintained by performing reasoning, e.g. about persistence over time, on the object level. This further makes our approach very flexible to also account e.g. for short-term dynamics via object tracking [5]–[7,13] or global consistency [11,12]. However, this is left for future work. Finally, because all submaps are fully data-parallel, the following operations can be distributed over multiple cores.

B. Label Tracking

To ensure the consistency of instance labels over multiple frames and temporal consistency of the map, incoming frames are tracked against the current map. Since ray-casting into many submaps quickly becomes intractable, we

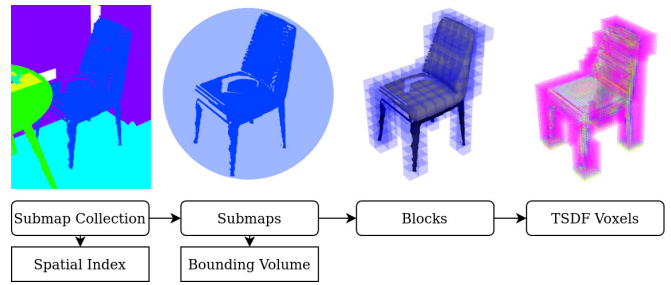


Fig. 3: Hierarchical map representation. Each submap (color) contains locks of TSDF voxel grids (colored pink to green based on the TSDF values). Submap bounding volumes and a global spatial index are maintained for efficient map traversal.

incrementally compute the iso-surface of each submap using Marching Cubes [34]. All active submaps whose bounding volume intersects with the view frustum are gathered and iso-surface points of blocks in the view frustum are projected into the image plane. Points whose rendered depth lies within a tolerance of $\xi_d = \nu$, where ν denotes the voxel size of the TSDF, of the measured depth are considered valid and fill in a patch of size ν . Each input segment is associated to the submap that has the highest Intersection over Union (IoU) between predicted and rendered mask and the same class label. To avoid spurious associations, a minimum IoU of $\xi_{IoU} = 0.1$ is required.

For masks that were not associated, a new submap is allocated. Compared to other approaches [5,13,18] that allocate a fixed grid size such that the object is contained in it, we choose ν as a function of the semantic label of each detection. This allows to select ν e.g. between $[\nu_{small} \in \{2 \dots 4\}, \nu_{large} \in \{5 \dots 10\}]$ cm based on how complex the expected geometry of the object class is and leave the efficiency optimization to our hierarchical map structure. We keep $\nu_{freespace} = 30$ cm for all settings. To guarantee local consistency, submaps are only active as long as they are successfully tracked, leading to multiple submaps with potentially varying resolutions describing the same object in case of detection or tracking failure. These submaps are later filtered during map management. To avoid instantiating too many false positives, submaps need to be tracked for $\tau_{new} = 3$ frames to be kept. Similarly, submaps that have not been detected for $\tau_{active} = 5$ frames are deactivated. This ensures that data is only integrated into currently tracked submaps and previous data can not be corrupted when e.g. changes in the environment have occurred.

C. Integration

To update the volumetric map, each measurement is fully integrated into all active submaps. Since ray-casting as in [2] quickly becomes intractable for many submaps, we use our hierarchical map representation for fast projective updates. For each submap, all blocks within the truncation distance $\delta = 2\nu$ of points belonging to their masks are allocated. Similar to [18], we separately reconstruct geometry and semantics. To best estimate the surfaces, we perform TSDF updates to *all* allocated voxels, adapting the weighting function of [2]:

$$w_{in}(v) = \frac{f_x * f_y * \nu^2}{z(v)^4} \quad (1)$$

Where f_x and f_y are the focal lengths of the camera and $z(v)$ is the depth of voxel v in the image.

To refine which surfaces are part of the submap, each voxel v has a belonging probability $P_b(v)$. Since network probabilities can be overconfident [5,18], we employ a memory efficient binary estimate $P_b(v)$ of the count probability $P_b^*(v)$ using weights p :

$$P_b^*(v) = \frac{1}{|\mathbb{F}_t|} \sum_{f=1}^{|\mathbb{F}_t|} \mathbb{I}_{\{label(u_f(v))=label(S(v))\}} \quad (2)$$

$$P_b(v) = \frac{\sum_{f=1}^{|\mathbb{F}_t|} p(|\mathbb{F}_t| - f) \mathbb{I}_{\{label(u_f(v))=label(S(v))\}}}{\sum_{f=1}^{|\mathbb{F}_t|} p(|\mathbb{F}_t| - f)} \quad (3)$$

$$p(f) = 1/2^{\lfloor f/128 \rfloor} \quad (4)$$

where \mathbb{F}_t is the set of frames where submap $S(v)$ was tracked, $u_f(v)$ is v projected into image f , and \mathbb{I} is the indicator function. This way, $P_b(v)$ can always be efficiently stored in only 16 bits.

Blocks that do not contain relevant information, i.e. $\#$ voxel v s.t. $|sdf(v)| < \delta \wedge P_b(v) > 0.5$, are pruned.

D. Map Management

Inactive submaps are frozen except for their change state $C(S) \in \{\text{persistent, unobserved, absent}\}$. To compare two submaps, we interpolate the SDF distance $sdf(p)$ and weight $w(p)$ of each iso-surface point $p \in \mathbb{P}$ of the reference map in the other map. For each observed point, the distance should be close to 0 if the point is a surface. If $|sdf(p)| < \xi_{sdf}$, where $\xi_{sdf} = \nu$ is the error tolerance, the point counts as agreeing with the surface. Otherwise, $sdf(p) < -\xi_{sdf}$ indicates intersections with object maps and $sdf(p) > \xi_{sdf}$ indicates conflicts with free space maps. Each point is weighted with the combined TSDF weight:

$$\hat{w}(p) = \sqrt{\min\left(\frac{w(p)}{\xi_w}, 1\right) * \min\left(\frac{w_{ref}(p)}{\xi_w}, 1\right)} \quad (5)$$

We empirically set the max weight $\xi_w = 100$. Submaps count as conflicting or matching if the weight-adjusted number of points exceeds a threshold $\tau_{abs} = 20$ or $\tau_{rel} = 2\%$ of $|\mathbb{P}|$.

When performing change detection, all inactive submaps that overlap with active submaps are compared against the latter. If they conflict with any of them, their state is set to *absent*. Otherwise, if they match with any of them, their state is set to *persistent*. This way, erroneous matches or rejections, e.g. through sensing noise, are corrected for later on. Far back in time submaps are *unknown*, and can become absent or persistent again when observed.

When submaps are deactivated and match with inactive submaps of the same class, they are fused together, allowing re-use of prior measurements and connections of components separated by e.g. occlusions or missed detections.



Fig. 4: Flat dataset. Run 1 (left), run 2 (center), and changes (right), showing new (green), removed (red), and modified (blue) objects.

E. Map Queries

To utilize the map for robotic interaction, efficient queries are important. To achieve this, we make use of our hierarchical map representation to only consider submaps and blocks that intersect with a query point p . Similarly, we use a temporal hierarchy to query spatio-temporal information. If the point is observed in an active submap, we directly use the highest resolution submap. Otherwise, $sdf(p)$ is the minimum of the distances to the surface of any persistent submap. Lastly, free space submaps are queried before resorting to yet unobserved submaps to predict expectations. Only present, i.e. *active* or *persistent* submaps, are counted for evaluation.

IV. EVALUATIONS

To properly evaluate temporal consistency, the true geometry of the whole scene needs to be known at every time step. Since this is hard to obtain in the real world, we employ a simulated environment, where complete ground truth is available. This data, termed the *flat dataset*, consists of two trajectories in a flat, where 8 objects are moved, 5 are added and 4 are removed between the runs, highlighted in in Fig. 4. The data was generated using the high fidelity simulation of [12]. We make this dataset available for future comparisons. To verify our method using real sensors and scenes, we perform experiments on the RIO dataset [35] with the limited ground truth available. We use the provided ground truth or optimized poses for state estimation.

A. Multi-resolution

Fig. 5 shows the reconstruction error as Mean Absolute Distance (MAD) versus the map size for varying voxel sizes, evaluated in the flat dataset. We compare against Voxblox [2] and Supereight² [19], which are the geometry representations of many semantic mapping frameworks [3,8,9,13]. We further compare against the sensor-based multi-resolution approach of [36].

The use of Ground Truth (GT) segmentation highlights the potential capabilities of our method, almost cutting the reconstruction error in half while consuming similar memory to Supereight for low resolutions, and achieving similar quality to [36] while reducing the memory $\times 23$ for high resolutions. This suggests significant benefits of semantically informed multi-resolution over the purely geometry-based

²We thank Emanuele Vespa for support and discussion while setting up Supereight.

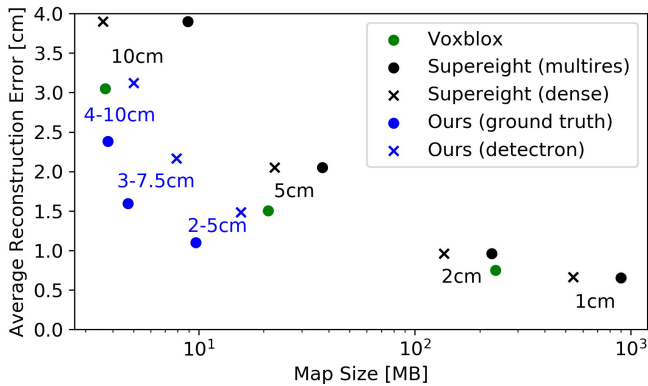


Fig. 5: Error vs map size for different voxel sizes indicated as text. For Supereight (multires) the minimum voxel size is given.

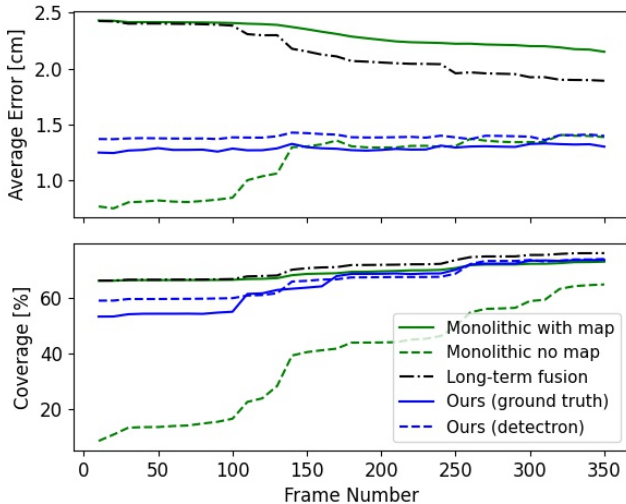


Fig. 6: Long-term mapping performance shown for the second run in the flat dataset, starting from a map of the first.

approach. Even with naive segmentation inputs, where only few small objects are detected and reconstructed precisely and false detections increase the memory consumption, our hierarchical map still saves memory compared to the baselines.

B. Long-term Temporal Consistency

Fig. 6 shows the MAD error and coverage as percentage of ground truth points with an error < 5 cm during the second trajectory of the flat dataset, given a prior map from the first run. We compare against a monolithic map as in [3,8,9] and consider both continuing mapping based on the previous state, or starting from scratch. This separation is only done for comparison, as our system accounts for changes continuously. We further compare against our implementation of [26] for volumetric maps, labeled Long-term fusion, where voxels that have a distance update > 5 cm are overwritten. For fair comparisons that only focus on the temporal component, we use identical TSDF integration and constant $\nu = 5$ cm for all approaches.

By including only active or persistent submaps, the reconstruction accuracy of our approach remains consistently at the discretization accuracy of ~ 1.4 cm. The map built from scratch initially shows a lower error, since the second run starts in a kitchen with few complex surfaces. When

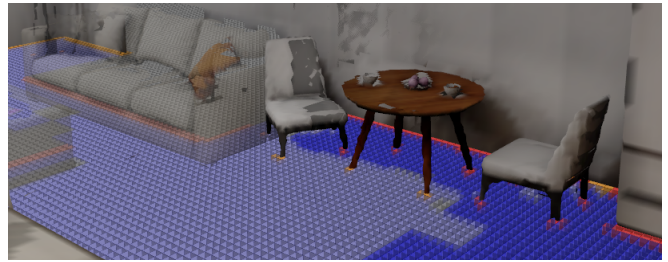


Fig. 7: Spatio-temporal map queries. Meshes of present submaps are drawn in solid and points observed as occupied or free are shown in solid red and blue, respectively. Yet unobserved points are shown in orange if they are occupied by persistent objects, or in shaded red and blue if they are expected to be occupied or free based on previous data (shaded meshes). Unknown points are gray.

starting from the previous map, none of the changes are captured, resulting in high errors. As more observations are made, the error is reduced. This happens faster when for long-term fusion strategy. However, due to the lack of semantic consistency, artifacts in the map keep the error well above the discretization level. Through inclusion of previous data that match current observations, our method explores significantly faster, converging to full coverage. The small gap between GT and Detectron highlights the robustness of our method to imperfect segmentation, since it only assumes semantic consistency of each submap and does not require completeness or accuracy.

C. Semantic Consistency

Qualitative comparisons are shown in Fig. 1. Since in our formulation semantically consistent submaps are the minimal unit of change, object consistency is preserved over time. In comparison, the voxel-based approach results in artifacts in the map and objects being merged together.

D. Spatio-temporal Look-ups for Online Planning

Fig. 7 shows spatio-temporal occupancy look-ups on our proposed map representation. Although our map can be queried at any point in space, a 2D slice is visualized. Fig. 7 highlights both the spatio-temporal information retrieved from single map queries, as well as multi-resolution preserving thin geometry. Map look-ups on this collection consisting of 130 submaps took an average of $3\mu s$.

E. Real World Experiments

Experiments on the RIO dataset [35] verify our method on real world data. We randomly chose reference scans 466 and 27 for evaluation, where different parts of indoor scenes are reconstructed over 2 and 4 runs, respectively. To account for the increased sensor noise and the reduced Detectron2 detections, we set $\xi_{sdf} = 2\nu$ and $\tau_{new} = 1$. All other settings remain unchanged. Since no complete ground truth is available as in simulation, we evaluate two different approximations.

We approximate a ground truth point cloud by combining all optimized meshes of the runs. To compensate for the temporal changes, we compute the coverage as the number of ground truth points that are observed in the map, and

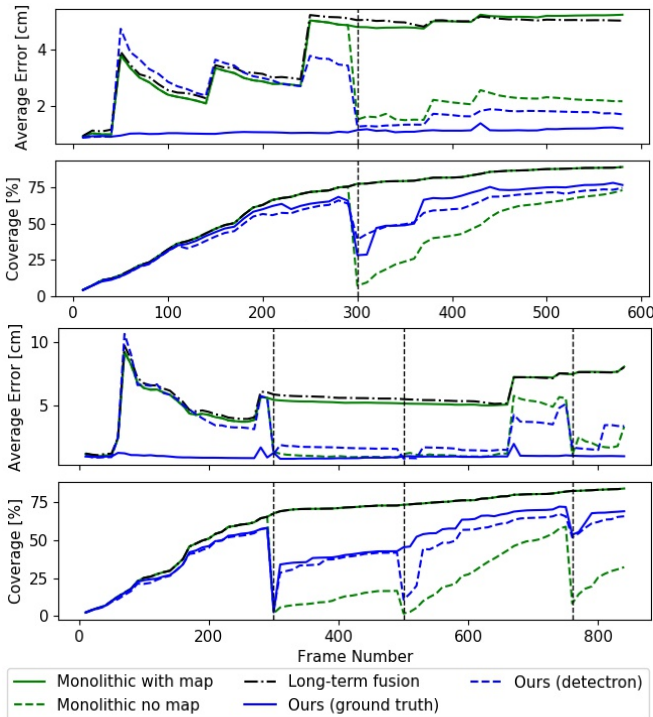


Fig. 9: Adapted average error and coverage over all runs, separated by dashed lines, for scan 466 (top) and 27 (bottom). Like in simulation, our method quickly covers the scene while keeping the error low.

the reconstruction accuracy as the distance from each iso-surface point to the closest ground truth point. Fig. 9 shows the findings, which for both scenes are very similar. During the first run all errors are comparable. For our method, the reconstruction quality can be increased by eliminating high error observations that disagree with other measurements. This is particularly pronounced when using ground truth segmentation, further highlighting the potential of leveraging semantic information for scene reconstruction. Like the simulated results, the temporal evolution after the first session demonstrates the capability of our method to represent the scene with similar or better quality than when starting from scratch, while extrapolating to a significantly larger coverage based on previous observations. The monolithic methods using the previous map show high errors and coverage. However, the coverage overestimates the true value since all

TABLE I: Computation times per operation in ms.

| Setting | Resolution | Tracking | Integration | Management | FPS* |
|--------------------|------------|------------|--------------|--------------|------|
| Flat, ground truth | 2-5 cm | 70.2 ± 8.4 | 104.3 ± 14.4 | 199.1 ± 54.1 | 5.1 |
| | 4-10 cm | 63.9 ± 4.4 | 89.2 ± 7.8 | 182.1 ± 44.3 | 5.8 |
| Flat, detectron | 2-5 cm | 57.8 ± 5.5 | 91.1 ± 11.1 | 192.1 ± 54.3 | 5.9 |
| | 4-10 cm | 54.7 ± 5.1 | 80.3 ± 7.4 | 183.5 ± 49.2 | 6.5 |
| RIO, detectron | 2-5 cm | 21.8 ± 4.6 | 21.8 ± 5.9 | 33.2 ± 23.8 | 21.3 |
| | 4-10 cm | 16.8 ± 3.4 | 13.3 ± 3.5 | 9.5 ± 4.5 | 32.2 |

* Final frame rate is computed performing change detection every 10 frames.

points from all times are included in the ground truth. The temporal consistency of our method is further illustrated in qualitative comparisons in Fig. 8.

F. Computational Performance

Tab. I shows the mean and standard deviation of execution times per operation. Data is obtained in the second run of Sec. IV-B, with a sensor resolution of 640×480 . Computation was performed on a laptop grade Intel Core i7-8550U CPU @1.80GHz. We do not account for the panoptic segmentation, which runs at 66ms per frame according to [33], although on a NVIDIA V100 GPU.

Even though our implementation is not thoroughly optimized, we achieve frame rates around 5 to 6 Hz, making our system amenable for real time operation on compute constrained mobile robots. The frame rates using real segmentation are slightly higher, since typically fewer objects are detected. Since RIO uses a sensor resolution of 224×172 and fewer objects, our method speeds up significantly, highlighting the flexibility of our approach to adapt to various settings.

V. CONCLUSIONS

In this work, we proposed panoptic multi-TSDFs, a novel representation for multi-resolution volumetric mapping. By leveraging higher-level information for 3D reconstruction, our proposed system allocates high resolution only where needed. Our submap-based approach achieves semantic consistency over time, enabling high reconstruction accuracy while increasing coverage by incorporating and fusing previous data where appropriate. We showed in thorough experimental validation that our map representation can be efficiently constructed, maintained, and queried during online operation on compute constrained hardware and operates

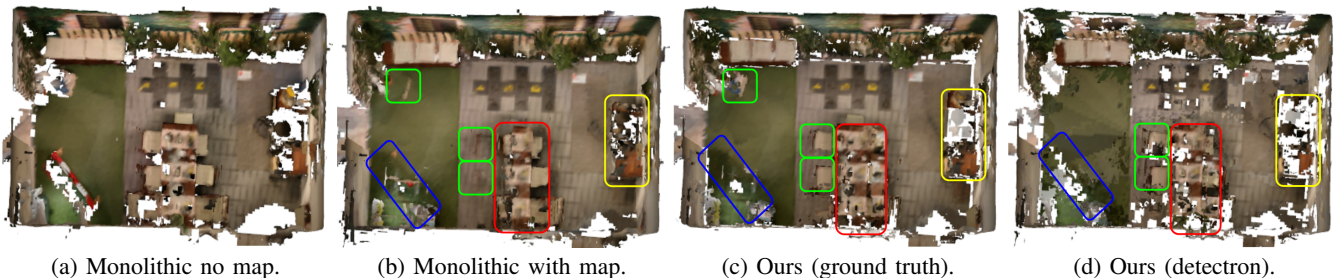


Fig. 8: Reconstructed meshes after run 2 in scan 466. For (c) and (d), *persistent* submaps are drawn solid and *unknown* submaps are shaded. Notably, (b) fails to fully reconstruct the moved table (red), which is only about half the size as in (a). New objects (green), such as the chairs near the table, are not captured in (b) whereas they are preserved in (c) and (d). Multiple observations at different times are merged into a blob (yellow) in (b), where our method preserves individual objects. The thin pole on (blue) is not captured by any method. The mesh in (d) appears more noisy due to the noisy Detectron2 detections.

robustly on real depth data and imperfect segmentation. We make our framework and data available as open-source.

In future work, our approach can be readily combined with methods for segmentation refinement and to also account for short-term dynamics. Recognition and re-localization of changed objects could further boost performance.

REFERENCES

- [1] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "Octomap: An efficient probabilistic 3d mapping framework based on octrees," *Autonomous robots*, vol. 34, no. 3, pp. 189–206, 2013.
- [2] 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 Int. Conf. on Intelligent Robots and Systems*, 2017, pp. 1366–1373.
- [3] G. Narita, T. Seno, T. Ishikawa, and Y. Kaji, "PanopticFusion: Online Volumetric Semantic Mapping at the Level of Stuff and Things," *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, pp. 4205–4212, Mar. 2019.
- [4] J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "Semanticfusion: Dense 3d semantic mapping with convolutional neural networks," in *IEEE Int. Conf. on Robotics & Automation*, 2017, pp. 4628–4635.
- [5] M. Strecke and J. Stueckler, "EM-Fusion: Dynamic Object-Level SLAM With Probabilistic Data Association," in *IEEE/CVF Int. Conf. on Computer Vision*, Oct. 2019, pp. 5864–5873.
- [6] M. Rünz and L. Agapito, "Co-fusion: Real-time segmentation, tracking and fusion of multiple objects," in *IEEE Int. Conf. on Robotics & Automation*, 2017, pp. 4471–4478.
- [7] M. Runz, M. Buffer, and L. Agapito, "MaskFusion: Real-Time Recognition, Tracking and Reconstruction of Multiple Moving Objects," in *IEEE International Symposium on Mixed and Augmented Reality*, Oct. 2018, pp. 10–20.
- [8] 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, pp. 3037–3044, 2019.
- [9] A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: An Open-Source Library for Real-Time Metric-Semantic Localization and Mapping," in *IEEE Int. Conf. on Robotics & Automation*, May 2020, pp. 1689–1696.
- [10] A. Kirillov, K. He, R. Girshick, C. Rother, and P. Dollár, "Panoptic segmentation," in *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2019, pp. 9404–9413.
- [11] V. Reijgwart, A. Millane, H. Oleynikova, R. Siegwart, C. Cadena, and J. Nieto, "Voxgraph: Globally consistent, volumetric mapping using signed distance function submaps," *IEEE Robotics and Automation Letters*, vol. 5, no. 1, pp. 227–234, 2019.
- [12] L. Schmid, V. Reijgwart, L. Ott, J. Nieto, R. Siegwart, and C. Cadena, "A unified approach for autonomous volumetric exploration of large scale environments under severe odometry drift," *IEEE Robotics and Automation Letters*, vol. 6, no. 3, pp. 4504–4511, 2021.
- [13] B. Xu, W. Li, D. Tzoumanikas, M. Bloesch, A. Davison, and S. Leutenegger, "Mid-fusion: Octree-based object-level multi-instance dynamic slam," in *IEEE Int. Conf. on Robotics & Automation*, 2019, pp. 5231–5237.
- [14] K. Tateno, F. Tombari, and N. Navab, "Real-time and scalable incremental segmentation on dense SLAM," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, vol. 2015-Decem, Dec. 2015, pp. 4465–4472.
- [15] T. Whelan, S. Leutenegger, R. Salas-Moreno, B. Glocker, and A. Davison, "Elasticfusion: Dense slam without a pose graph," in *Proc. of Robotics: Science and Systems*, 2015.
- [16] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *IEEE/CVF Int. Conf. on Computer Vision*, 2017, pp. 2961–2969.
- [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 *Proc. of IEEE Conf. on Computer Vision and Pattern Recognition*, 2013, pp. 1352–1359.
- [18] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger, "Fusion++: Volumetric Object-Level SLAM," in *Int. Conf. on 3D Vision*, Sept. 2018, pp. 32–41.
- [19] E. Vespa, N. Nikolov, M. Grimm, L. Nardi, P. H. J. Kelly, and S. Leutenegger, "Efficient octree-based volumetric slam supporting signed-distance and occupancy mapping," *IEEE Robotics and Automation Letters*, vol. 3, no. 2, pp. 1144–1151, 2018.
- [20] R. Long, C. Rauch, T. Zhang, V. Ivan, and S. Vijayakumar, "RigidFusion: Robot Localisation and Mapping in Environments with Large Dynamic Rigid Objects," *IEEE Robotics and Automation Letters*, vol. 6, no. 2, pp. 3703–3710, Oct. 2021.
- [21] R. Finman, T. Whelan, M. Kaess, and J. J. Leonard, "Toward lifelong object segmentation from change detection in dense rgb-d maps," in *European Conf. on Mobile Robots*, 2013, pp. 178–185.
- [22] T. Whelan, M. Kaess, M. Fallon, H. Johannsson, J. Leonard, and J. McDonald, "Kintinuous: Spatially extended kinectfusion," 2012.
- [23] M. Fehr, F. Furrer, I. Dryanovski, J. Sturm, I. Gilitschenski, R. Siegwart, and C. Cadena, "Tsd-f-based change detection for consistent long-term dense reconstruction and dynamic object discovery," in *IEEE Int. Conf. on Robotics & Automation*, 2017, pp. 5237–5244.
- [24] E. Langer, T. Patten, and M. Vincze, "Robust and efficient object change detection by combining global semantic information and local geometric verification," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2020, pp. 8453–8460.
- [25] T. Krajník, J. Pulido Fentanes, M. Hanheide, and T. Duckett, "Persistent localization and life-long mapping in changing environments using the frequency map enhancement," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2016, pp. 4558–4563.
- [26] M. T. Lázaro, R. Capobianco, and G. Grisetti, "Efficient long-term mapping in dynamic environments," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2018, pp. 153–160.
- [27] L. Tang, Y. Wang, X. Ding, H. Yin, R. Xiong, and S. Huang, "Topological local-metric framework for mobile robots navigation: a long term perspective," *Autonomous Robots*, vol. 43, no. 1, pp. 197–211, 2019.
- [28] S. Macenski, D. Tsai, and M. Feinberg, "Spatio-temporal voxel layer: A view on robot perception for the dynamic world," *International Journal of Advanced Robotic Systems*, vol. 17, no. 2, p. 1729881420910530, 2020.
- [29] J. Mason and B. Marthi, "An object-based semantic world model for long-term change detection and semantic querying," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2012, pp. 3851–3858.
- [30] S. Izadi, D. Kim, O. Hilliges, D. Molyneaux, R. Newcombe, P. Kohli, J. Shotton, S. Hodges, D. Freeman, A. Davison, et al., "Kinectfusion: real-time 3d reconstruction and interaction using a moving depth camera," in *Proc. of the ACM symposium on User Interface Software and Technology*, 2011, pp. 559–568.
- [31] M. Bloesch, M. Burri, S. Omari, M. Hutter, and R. Siegwart, "Iterated extended kalman filter based visual-inertial odometry using direct photometric feedback," *Int. Journal of Robotics Research*, vol. 36, no. 10, pp. 1053–1072, 2017.
- [32] X. Zuo, P. Geneva, W. Lee, Y. Liu, and G. Huang, "Lic-fusion: Lidar-inertial-camera odometry," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, 2019, pp. 5848–5854.
- [33] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," <https://github.com/facebookresearch/detectron2>, 2019.
- [34] W. E. Lorensen and H. E. Cline, "Marching cubes: A high resolution 3d surface construction algorithm," *ACM siggraph computer graphics*, vol. 21, no. 4, pp. 163–169, 1987.
- [35] J. Wald, A. Avetisyan, N. Navab, F. Tombari, and M. Niessner, "Rio: 3d object instance re-localization in changing indoor environments," in *IEEE/CVF Int. Conf. on Computer Vision*, 2019.
- [36] E. Vespa, N. Funk, P. H. Kelly, and S. Leutenegger, "Adaptive-resolution octree-based volumetric slam," in *Int. Conf. on 3D Vision*, 2019, pp. 654–662.