[PLR] Incremental map-based semantic segmentation

Zador Pataki
Department of Mechanical and Process Engineering
ETH Zürich Switzerland
patakiz@student.ethz.ch

Abstract: It is crucial for a mobile robot to have a geometric and semantic understanding of its environment to be able to effectively manipulate and navigate it. Additionally, this information must be inferred in real time if the robot is exploring a not-before-seen environment. Inspired by traditional machine-learning methods, state-of-the-art approaches achieve this by semantically segmenting 2D images captured by a robot and incrementally fusing this information into a 3D dense map of its environment. In contrast to such methods, we propose an algorithm that incrementally constructs a 3D dense map and semantically segments the dense map directly. The advantage of our approach is that our classifier can leverage accumulated information of a scene captured in a dense map to make its predictions. We demonstrate the effectiveness of our proposed methods on two benchmark data-sets.

Keywords: CORL, Robots, Deep Learning, 3D Semantic Segmentation, Online

1 Introduction

In most autonomous robot tasks, the goal is to either navigate through an environment or to interact with it. In many cases the environment is unknown (or only partially known), and for this reason it is crucial for an autonomous robot to gain a geometric and semantic understanding of its surroundings. Moreover, if the goal is to do this online, then the robot must be able to update its understanding of the environment incrementally, as it gathers more information.

Inspired by traditional semantic image segmentation tasks, current approaches – referred to as view-based approaches – make segmentation predictions based on 2D snapshots of the environment and incrementally fuse the predicted labels into a dense map [1, 2]. At each increment, the classifier utilizes only the information from current snapshots of a scene to make its predictions. A disadvantage of such approaches, is that segmentation-relevant information contained in previous snapshots are unused. Rather than semantically segmenting 2D snapshots, in our approach, we attempt to overcome this limitation by inferring the labels of a 3D dense map directly, which is constructed incrementally from a series of captured depth images. The dense map contains accumulated information of the scene which is leveraged by our classifier.

Recent work [3] has shown that given a voxel map, 3D Convolutional Neural Networks (CNNs) can be used to semantically segment the scene directly with high accuracies. Furthermore, Landgraf et al. [4] compared view-based and map-based semantic segmentation approaches. They demonstrated that map-based approaches outperform view-based approaches both in the presence of and in the absence of noise. As an attempt to improve the performance of incremental semantic segmentation of a 3D scene, we propose a map-based algorithm that takes geometric information from an incrementally constructed map and infers its semantic labels directly. We believe that our framework is better suited for incremental semantic segmentation because the classifier is capable of leveraging not only 3D features, but also accumulated information stored in the map. Moreover, while view-based methods must semantically segment each individual 2D snapshot to leverage its segmentation-relevant information, in our method, semantic segmentation steps can be arbitrarily skipped and while still leveraging the relevant information from each snapshot in future steps. Due to the flexibility of the update scheme, we believe that our approach is better suited for real world applications where computational power is limited.

4rd Conference on Robot Learning (CoRL 2020), —, Japan.

The algorithm leverages a novel framework that appropriately fuses depth maps of a scene into a 3D map represented by voxel groups [5]. At each increment, updated voxel groups are used as inputs to a NN framework inspired by previous works in 3D semantic segmentation [3]. The network outputs the inputted groups with predicted semantic labels, and our framework adds these labeled groups directly back into the global voxel map.

2 Related Works

In general, approaches for semantic segmentation of 3D scenes have been divided into two broad groups: view-based (Sec. 2.1) and map-based semantic segmentation (Sec. 2.2).

To apply 3D scene semantic segmentation approaches in real-time applications, it is a necessary condition for semantic segmentation to be performed incrementally. Recently, there has been a lot of research about map-based semantic segmentation, however, most state-of-the-art incremental semantic segmentation frameworks still leverage view-based approaches.

2.1 View-based semantic segmentation

In view-based semantic segmentation of 3D scenes, labels from inputted view-wise data are estimated and then fused into the scene model. These scene models are constructed from a series of depth maps and generally represented using voxel grids. Hermans et al. [6] created a framework where, given RGB-D snapshots of a scene, a scene map is incrementally constructed, 2D RGB images are semantically segmented using randomized decision forests and then the inferred labels are fused into the map using Bayesian updates. Inspired by modern semantic segmentation approaches in the field of machine learning [7], later work has shown that the semantic segmentation performance of such approaches can be improved using CNN frameworks without being at the expensive of worsening computational time [8]. In addition to semantically segmenting RGB images, Grinvald et al. [1] developed a framework where corresponding depth images were semantically segmented in an unsupervised manner, by decomposing them into a set of segments following a geometry-based approach. The segmentation of the RGB image is then used to infer class information for the corresponding depth segments. Another contribution to view-based semantic segmentation of scenes, presented by Narita et al. [2], is a final stage of map regularization. The researchers demonstrated that map regularization can be performed online using a fully-connected Conditional Random Field model, which further improves segmentation accuracy.

Instead of modelling an entire scene, a different body of work uses view-based semantic segmentation to identify, model and track objects in a scene [9, 10]. Given RGB-D data, in this approaches, maps are constructed containing only objects of interests. The objects are tracked and their semantic segmentation and their construct models are incrementally refined.

2.2 Map-based semantic segmentation

Similarly to the view-based approach (Sec. 2.1) scenes models are constructed from a series of depth maps. The key difference between the two approaches is that in map-based semantic segmentation of 3D scenes, the classifier uses the constructed model as an input, and infers its labels directly. In general, modern approaches use 3D CNNs in the architecture of the classifier leverage the 3D information from the constructed map. In recent works, however, researchers developed classifiers which leverage not only the 3D information from the map, but also view-wise data from individual snapshots [3, 11, 12]. In contrast to previous approaches, [12] which leveraged only single RGB snapshots in addition to the 3D model, Dai and Nießner [11] proposed a framework, which leverages RGB images from multiple views. The 2D features were extracted using 2D CNNs, were projected onto the 3D voxel grid and were passed through the 3D CNN framework. In flexible map-based semantic segmentation frameworks, the classifier should be able to infer the labels of scenes with different sizes. In many approaches [4], a sliding window procedure is implemented to semantically segment different parts of the map individually. I critique of such methods is that the local 3D information during a classification step is limited by the size of the window. To handle this limitation, Dai et al. [13] proposed a NN framework in which the 3D CNN filter kernels are invariant to the overall scene size and can be deployed on arbitrarily large scenes, where the entire scenes are passed as inputs.

In a different corpus of work, researchers have developed map-based frameworks which not only semantically segment, but also complete the reconstructed scenes (semantic scene completion) [3, 14, 15, 16]. Here, scene completion refers to the completion of 3D shapes by updating the set occupied voxels in inputted voxel grids. Song et al. [14] demonstrated that simultaneously performing semantic scene completion and segmentation outperforms methods addressing each task in isolation. Inspired by their approach, Li et al. [3] developed a framework that leverages a single depth images to perform semantic scene completion, outperforming previous works on benchmark datasets. In their proposed framework, 2D information form depth images were leveraged in addition to the constructed 3D map, and a loss function was proposed and used for training which penalized the predictions on individual voxels not purely based on the correctness of the predictions, but also based on the geometric neighborhood of those voxels. More specifically, incorrect predictions on voxels on object surfaces were penalized less than thos on object edges.

Unlike in view-based methods, in map-based approaches there still exists a research gap in that to the best of our knowledge, the only incremental map-based semantic segmentation framework that exists is one developed by Wu et al. [16]. A concurrent work in which the framework leverages only voxel occupancy and large voxel blocks (64x64x64) as inputs, containing a vast amount of volumetric information in each block. However, previous work [17] has shown that using TSDFs has many benefits over using only occupancy, and we believe that relying on large blocks limits the applicability in frameworks where computational power is limited.

2.3 Compareing view-based and map-based semantic segmentation

In a work by Landgraf et al. [4], the researchers quantitatively compared the two approaches based on their performance on the same experimental framework. The two approaches are compared based on their semantic segmentation of several hundred synthetic scenes containing various scattered objects with perfect ground truth labelling.

To evaluate the effectiveness of the approaches, the semantic scene segmentation obtained by each approach is compared against the scene's ground truth using mean Intersection over Union (IoU) over all classes. The researchers concluded that in the absence of noisy data, the map-based approach achieved a higher IoU score on average. It was found, however, that the view-based method is more robust in the presence of pose noise, but despite this, it was found that overall the map-based method performs better in the presence of noise. This is because the accuracy of the view-based method strongly degraded in the presence of depth noise, while the map-based method was more robust.

These findings suggest that the map-based approach would lead to higher accuracies in an incremental semantic segmentation application. Moreover, the researchers highlighted that in their application, the map-based method is more efficient. This is because the semantic segmentation does not need to be applied individually for each captured image.

3 Methods

We propose a framework which incrementally constructs and semantically segments a 3D scene online. Our framework leverages a framework, Voxblox [5], which incrementally builds voxelized Truncated Signed Distance Fields (TSDFs)[1], from inputted point clouds. From the constructed map, at each incremental step, individual voxel blocks are extracted and used as an inputs to our NN framework, which then infer the the semantic labels of each voxel in each block. To achieve this, raw camera data is pre-processed to be applicable in the Voxblox framework, and the TSDF data generated by the Voxblox framework is then further processed to be applicable as an input to our NN framework. Our NN framework is set-up and trained in a manner in which it can be generalized to be effective when applied to not-before-seen scenes 3.2. Once the NN framework is trained, it can be applied in an incremental update scheme 3.3, to perform incremental map-based semantic segmentation of 3D scenes. The details of each of these parts are presented in the following subsections.

[1]TSDFs are 3D voxel arrays containing distance information to the nearest surface.

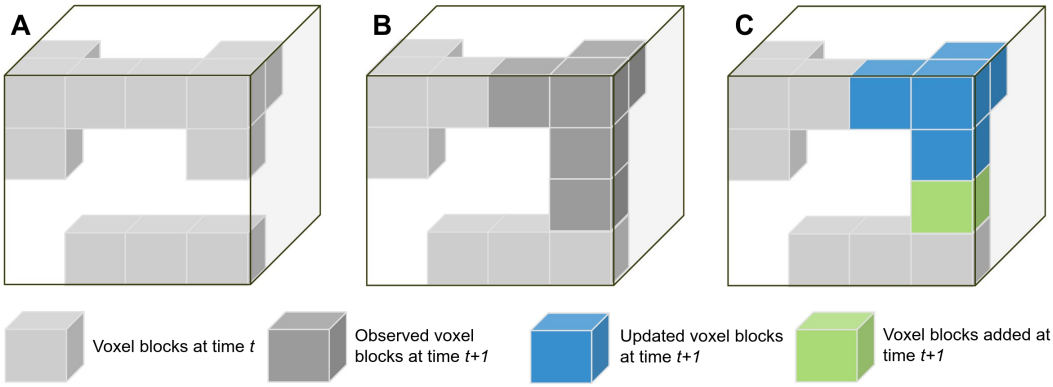


Figure 1: **Update of Global Voxel Map.** **A:** Global voxel map containing voxel blocks with inferred semantic information at time t. **B:** Global voxel map at following increment with newly observed voxel blocks highlighted. **C:** Same global voxel map as in B, except here, voxel blocks are highlighted differently. Highlighted in blue are voxel blocks that have been updated since previous inrement. Highlighted in green are voxel blocks that have been added since previous increment.

3.1 Data Pre-processing

The Voxblox framework, at each incremental stage, takes pointclouds as inputs with corresponding camera poses in the form of quaternions. Given the camera focal length, depth images are transformed into 3D pointclouds. Ground truth semantic labels are assigned to each point into corresponding pointclouds.

Given camera pose information at each view, the Voxblox framework then incrementally integrates the pointclouds from individual views into a global voxel map. At each increment, voxel blocks are either added to the global voxel map, or already existing voxel blocks in the global voxel map are updated (see Fig. 1). Each voxel block corresponds to a designated volume in the robot's environment. See Fig. 1 for information about in what manner the global voxel map is updated.

After the global voxel map is updated, data from individual voxel bocks can be extracted in the form of 3D arrays. Each voxel contains signed distance information, weight information and colour information. Voxels more than the truncation distance away from observed surfaces have zero weight distance and colour. Inside of the truncation distance, the signed distances correspond to the distance to the closest surface. Moreover, inside the truncation distance, the weights of the voxels are non-negative and we used the red colour information of the voxels to store the ground truth label information[2]. We classify voxels as occupied, if they have non-zero weights, and the magnitude of their distance value is smaller than the width voxel size (we refer to voxel size as the width of one of the sides of the cuboidal voxels). We do this because if a voxel is less than the voxel size away from the closest surface, then the voxel must contain surface information.

For training, voxel data is generated a priori – for training the NN framework we are only interested in voxel data and are not interested in depth/pointcloud data. In contrast to this, when the framework is applied in practice, data from updated voxel blocks (or from voxel blocks that had been added to the map) is extracted from the current state of the global volume map at each incremental stage. This data can be used as an input to the classifier without further processing.

3.2 Classifier

3.2.1 Neural Network Architecture

Inspired by Li et al. [3] our classifier is a 3D CNN framework which takes voxel blocks from an incrementally constructed map as input, and outputs class label probabilities of each inputted voxel (see Fig. 3 A). The NN framework consists of three parts: a voxel stream, a multi-level feature aggregation model and a reconstruction part. In the voxel stream, TSDF values of voxels are passed

[2]Note that as we do not use RGB information, the red colour channel is empty and can be used to store the ground truth information.

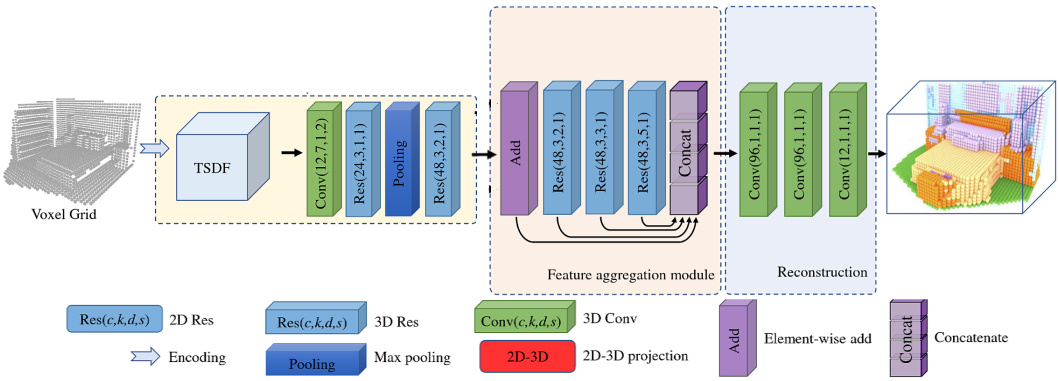


Figure 2: **Neural Network Architecture.** The network takes TSDF voxels as input. The inputs are first passed through a 3D CNN module consisting of a 3D convolutional layer, dilated residual blocks and a 3D pooling layer. The extracted features are then passed through a feature aggregation module, in which the features are passed through a series of 3D dilated residual blocks. The outputs of these blocks are then concatenated and passed through a series of 3D convolutional layers. The network outputs semantic label probabilities for each block.

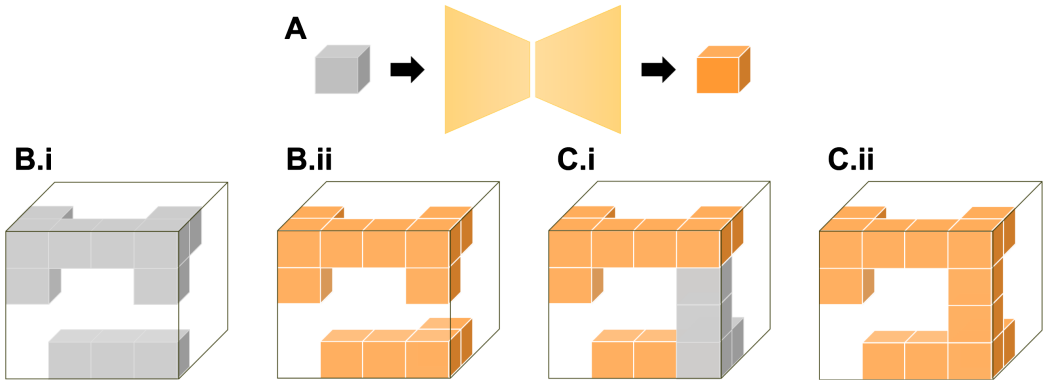


Figure 3: **Inferring labels of voxel blocks. A:** Voxel blocks without color information are passed through the NN framework. The network outputs the identical voxel block, except with voxel-wise semantic labels. **B.i:** Global volume map containing unlabeled voxel blocks during first increment before semantic segmentation. **B.ii:** Global volume map containing unlabeled voxel blocks during first increment before semantic segmentation. **C.i:** Updated voxel blocks at given increment remove previously inferred labels at corresponding block. **C.ii:** Semantic labels of updated blocks are newly inferred.

through a 3D CNN module[3]. The 3D CNN modules consist of 3D convolutional layers, 3D pooling layers and dilated residual blocks, with bottleneck versions used to increase the capacity of the network and to reduce the number of parameters. In the multi-level feature aggregation model, the 3D features are passed through a series of 3D dilated residual blocks to increase the receptive field. The outputs of each residual block are then concatenated. Finally, the outputs of the multi-level feature aggregation model are passed through three standard 3D convectional layers, generating the previously mentioned outputs. For more information about the NN architecture, see Fig. 2.

3.2.2 Training of the Neural Network

Validation-score tracking

During training, we tracked the accuracy of our classifier on a validation set, distinct from the dataset we used for training. To achieve this, the following method was used: the dataset is split into a

[3]TSDF values are often flipped in order to have strong gradients on surfaces, providing more meaningful signals for networks to learn better geometric features. In future works we plan to experiment with using flipped TSDF values as inputs.

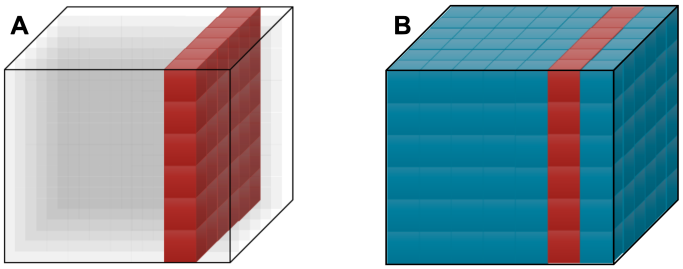


Figure 4: **Voxel block occupancy. A:** Voxel block extracted from global voxel map. Red cubes represent occupied voxels and transparent cubes represent unoccupied voxels. **B:** Voxel block outputted by classifier. Blue cubes represent unoccupied voxels with inferred labels and red cubes represent occupied voxels with inferred labels. The loss function is only takes predictions on red cubes as an input.

training set and a validation set. The split is performed proportional to the number of scnenes and not proportional to the number of voxel blocks, such that voxel blocks of a given trajectory are either all part of the training set or are all part of the validation set. The split is done such that voxel blocks from approximately 80% of all trajectories are assigned to the training set and all other voxel blocks were assigned to the validation set.

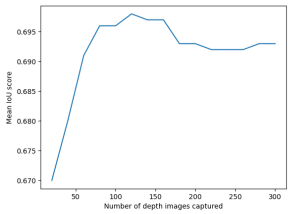
Validation accuracy is tracked by calculating the accuracy after each training epoch. If, after a certain epoch e, the validation accuracy does not increase beyond the validation accuracy at epoch e for a given number of epochs, then we perform early stopping [18] to avoid overfitting to the training data.

3.3 Incremental Update Scheme

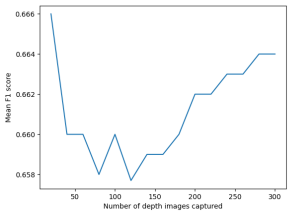
Using a trained network (trained according to the framework presented in section 3.2.2), the incremental map-based semantic segmentation framework can be applied. The incremental update scheme for incrementally constructing a semantically labelled voxel map consists of four steps at each increment (where "at each increment" means that a depth image of a scene is captured): (i) projecting a depth image into a 3D pointcloud, (ii) integrating the resulting pointcloud into the global voxel map, (iii) passing all updated voxel blocks through the NN framework, and finally, (iv) updating the global voxel map using the inferred semantic labels.

In stage (i), depth images are projected into 3D pointclouds given camera focal length. As described in section 3.1, in stage (ii), pointclouds are integrated into the global voxel map given corresponding camera pose. It must be noted that the global voxel map at increment t-1, which is being updated in the current increment t, contains not only geometric information, but also the predicted semantic labels at increment t-1. The voxblox framework overwrites previous colour information, so semantic labels from increment t-1 are removed and semantic labels inferred at increment t are only added to the map in stage (iv) (see Figs. 3 B and C).

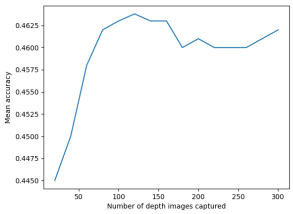
In stage (iii) the data from updated voxel blocks are extracted, and the corresponding TSDF distance values are passed through the trained NN framework, outputting class probabilities of each class used for training. In stage (iv), the class with maximum probability is assigned to each voxel. Disregarding previously inferred labels of the treated voxel blocks, the predicted labels of each block are then simply added back into the global voxel map (see Fig. 3 C). Here, the assumption is made that since the latest voxel blocks are constructed from the most accumulated information, the corresponding geometric representation of the scene is the most accurate, thus the newest predicted labels are the most optimal.[4]



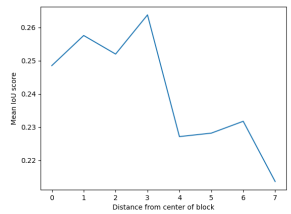
(a) Mean IoU changing with with number of depth images



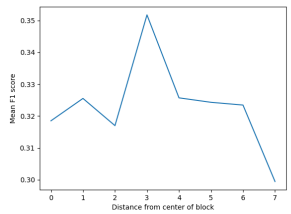
(b) Mean F1 score changing with with number of depth images



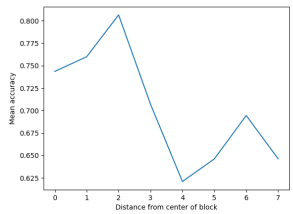
(c) Mean IoU changing with number of depth images



(d) Mean IoU changing with distance from center of block



(e) Mean F1 score changing with distance from center of block



(f) Mean accuracy changing with distance from center of block

Figure 5

4 Execution of Experiments and Results

4.1 Experimental Setup

For training depth maps, were taken from scenes of the SceneNet dataset [19]. The SceneNet dataset consists of ground truth RGB-D snapshots from photo-realistic rendered indoor scene trajectories. The data-set contains 15'000 scenes with 300 distinct snapshots per scene. On top of the RGB-D data, the data-set also includes semantic ground truth information from 13 classes, and camera pose information for each individual snapshot. For training, we used all depth images from 3'500 scenes.

The camera poses for each view are presented as camera position and the position to which the camera is pointing. This information was transformed into homogeneous transformation matrices in the world view. In our framework, each homogeneous transformation is converted to the desired quaternion format using an inbuilt Voxblox function.

During the processing of the training data (Sec. 3.1), we set each voxel block to contain 16x16x16 voxels. We set the voxel size and the truncation size of the process to 8cm and 16cm respectively.

4.2 Synthetic Scenes

For testing our network on the synthetic scenes of the SceneNet dataset, we used a test set containing 1000 scenes distinct from those of the training set.

4.2.1 Global segmentation scores

Our most optimal NN achieved a mean IoU score of 0.462 and an F1 score of 0.693. Furthermore, we provide plots displaying how the scores change with the number of depth images integrated into the map (Figs. 5a, 5b and 5c). In addition to mean IoU and F1 scores, we provide also the diagram for the accuracy.

4.2.2 Local segmentation analysis

In this subsection we present our results for our analysis on how the prediction scores change depending on how far voxels are from the center of the voxel blocks. In Figs. 5d, 5e and 5f, a voxel

[4] In future works, we plan to experiment with Bayesian and heuristic label fusion as alternatives.

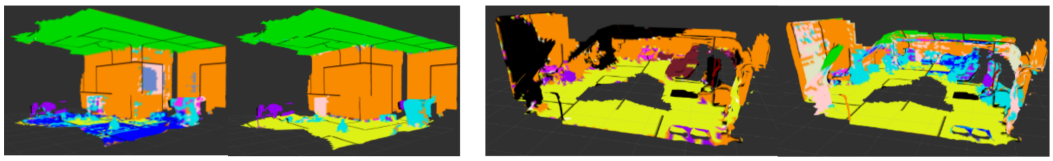


Figure 6: **A.i:** Synthetic scene with predicted labels. **A.ii:** Synthetic scene with predicted ground truth labels. **B.i:** Real scene with predicted labels. **B.ii:** Synthetic scene with ground truth labels

has zero distance of it is one of the center 4 voxels of the voxel block. The distance increases with each layer around these center blocks. The plots are presented for the mean IoU and F1 score, and for the accuracy.

4.2.3 Qualitative results

We show an example of the semantic segmentation on a synthetic scene in Fig. 6. Furthermore, we compare it to the ground truth values.

4.3 Real-world Scenes

For testing our framework on real world data, we used scenes from the ScanNet dataset [20]. Processing this data is equivalent to how we processed the synthetic data, highlighted in Section 3.1, except the pointcloud projection needed to be performed differently (for details see the cited paper).

4.3.1 Qualitative results

We show an example of the semantic segmentation on a real world scene in Fig. 6. We compare it also to the our best approximation of ground truth values. There was an imperfect overlap between the real world dataset labels and the labels which we trained our network on. Assumptions about individual classes had to be made.

4.4 Process time analysis

To study the time it takes for the process to run, we ran tests on a laptop with an Intel® Core™ i7-8565U processor, and a cpu with 16GB memory. We used no GPU. On average a process step took a on average 4.2s per increment, whil, without semantic segmentation (only the construction of the map), the process took on average 0.4s per increment.

5 Conclusion

In this work, we propose a framework which incrementally constructs a 3D map using captured depth images of a scene and semantically segments it directly at each increment. In contrast to previous incremental semantic segmentation approaches which are view-based, the main benefit of our map-based method is that the classifier is capable of leveraging accumulated information stored in the incrementally constructed dense map. Moreover, in contrast to view-based methods, to minimize process time, semantic segmentation can be performed periodically – instead of at each increment – while leveraging the same amount of accumulated information.

Our framework achieved promising IoU and F1 scores. The main modifications to the framework we would like to investigate in future works consist of the incorporation of RGB information for classification, updating the amount of geometric information inputted into the classifier when inferring the labels of a given voxel block, and finally, we would like to update the loss function. RGB images can be fused into the dense map analogously to the depth images, and can be leveraged equivalently to the depth values passed through the network. Experimental results (Sec. 4.2.2) suggest that the lack neighboring information near the boundaries of voxel blocks lead to lower segmentation accuracies. To combat this, we plan to not only pass the voxel blocks which are being segmented though the classifier through the network, but also information from neighboring blocks. The NN would

need to be updated accordingly. In the work by Li et al. [3], a loss function was proposed which is weighted based on local geometry. The researchers found that this loss function lead to higher accuracies when testing on benchmark datasets. Applying this loss function for our training scheme is also planned.

Acknowledgments

I would like to express my very great appreciation to Lukas Schmid for his great ideas and terrific guidance throughout the proces of this work. Moreover, I would like to thank Dr. Jen Jen Chung and Dr. Cesar Dario Cadena Lerma for their assistance with the writing of this paper.

References

[1] 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*, 4(3):3037–3044, Jul 2019. ISSN 2377-3774. doi:10.1109/lra.2019.2923960. URL http://dx.doi.org/10.1109/LRA.2019.2923960.

[2] Narita, Seno, Ishikawa, and Kaji. Panopticfusion: Online volumetric semantic mapping at the level of stuff and things, 2019.

[3] J. Li, Y. Liu, X. Yuan, C. Zhao, R. Siegwart, I. Reid, and C. Cadena. Depth based semantic scene completion with position importance aware loss. *IEEE Robotics and Automation Letters*, 5(1):219–226, 2019.

[4] Z. Landgraf, F. Falck, M. Bloesch, S. Leutenegger, and A. J. Davison. Comparing view-based and map-based semantic labelling in real-time slam. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 6884–6890. IEEE, 2020.

[5] 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.

[6] A. Hermans, G. Floros, and B. Leibe. Dense 3d semantic mapping of indoor scenes from rgb-d images. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 2631–2638. IEEE, 2014.

[7] J. Long, E. Shelhamer, and T. Darrell. Fully convolutional networks for semantic segmentation. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3431–3440, 2015.

[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)*, pages 4628–4635. IEEE, 2017.

[9] J. McCormac, R. Clark, M. Bloesch, A. Davison, and S. Leutenegger. Fusion++: Volumetric object-level slam. In *2018 international conference on 3D vision (3DV)*, pages 32–41. IEEE, 2018.

[10] M. Runz, M. Buffier, and L. Agapito. Maskfusion: Real-time recognition, tracking and reconstruction of multiple moving objects. In *2018 IEEE International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 10–20. IEEE, 2018.

[11] A. Dai and M. Nießner. 3dmv: Joint 3d-multi-view prediction for 3d semantic scene segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 452–468, 2018.

[12] J. Hou, A. Dai, and M. Nießner. 3d-sis: 3d semantic instance segmentation of rgb-d scans. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 4421–4430, 2019.

[13] A. Dai, D. Ritchie, M. Bokeloh, S. Reed, J. Sturm, and M. Nießner. Scancomplete: Large-scale scene completion and semantic segmentation for 3d scans. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[14] S. Song, F. Yu, A. Zeng, A. X. Chang, M. Savva, and T. Funkhouser. Semantic scene completion from a single depth image. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 1746–1754, 2017.

[15] Y. Wang, D. J. Tan, N. Navab, and F. Tombari. Forknet: Multi-branch volumetric semantic completion from a single depth image. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 8608–8617, 2019.

[16] S.-C. Wu, K. Tateno, N. Navab, and F. Tombari. Scfusion: Real-time incremental scene reconstruction with semantic completion. *arXiv preprint arXiv:2010.13662*, 2020.

[17] H. Oleynikova, A. Millane, Z. Taylor, E. Galceran, J. Nieto, and R. Siegwart. Signed distance fields: A natural representation for both mapping and planning. In *RSS 2016 Workshop: Geometry and Beyond-Representations, Physics, and Scene Understanding for Robotics*. University of Michigan, 2016.

[18] L. Prechelt. Early stopping-but when? In *Neural Networks: Tricks of the trade*, pages 55–69. Springer, 1998.

[19] A. Handa, V. Patraucean, V. Badrinarayanan, S. Stent, and R. Cipolla. Scenenet: understanding real world indoor scenes with synthetic data. arxiv preprint (2015). *arXiv preprint arXiv:1511.07041*, 2015.

[20] A. Dai, A. X. Chang, M. Savva, M. Halber, T. Funkhouser, and M. Nießner. Scannet: Richly-annotated 3d reconstructions of indoor scenes. In *Proc. Computer Vision and Pattern Recognition (CVPR), IEEE*, 2017.