3D Semantic Label Transfer and Matching in Human-Robot Collaboration

Szilvia Szeier[1], Benjámin Baffy[1], Gábor Baranyi[1], Joul Skaf[1], László Kopácsi[1,2], Daniel Sonntag[2], Gábor Sörös[3], and András Lőrincz[1]

[1] Department of Artificial Intelligence, Eötvös Loránd University, Budapest, Hungary
[2] Department of Interactive Machine Learning, German Research Center for Artificial Intelligence (DFKI), Saarbrücken, Germany
[3] Nokia Bell Labs, Budapest, Hungary

Abstract. Semantic 3D maps are highly useful for human-robot collaboration and joint task planning. We build upon an existing real-time 3D semantic reconstruction pipeline and extend it with semantic matching across human and robot viewpoints, which is required if class labels differ or are missing due to different perspectives during collaborative reconstruction. We use deep recognition networks, which usually perform well from higher (human) viewpoints but are inferior from ground robot viewpoints. Therefore, we propose several approaches for acquiring semantic labels for unusual perspectives. We group the pixels from the lower viewpoint, project voxel class labels of the upper perspective to the lower perspective and apply majority voting to obtain labels for the robot. The quality of the reconstruction is evaluated in the Habitat simulator and in a real environment using a robot car equipped with an RGBD camera. The proposed approach can provide high-quality semantic segmentation from the robot perspective with accuracy similar to the human perspective. Furthermore, as computations are close to real time, the approach enables interactive applications.

1 Introduction

Human-robot collaboration faces several challenges today. Let us take the example of a simple physical rehabilitation scenario with a ground robot that observes a human performing physical exercises. The exercise supervision requires detailed body pose recognition in 3D, which often relies on a 3D human body model. Time constraints for correcting instructions can be high, and verbal communication might be also a necessity. The robot needs to recognize the person and the objects in the environment, and reconstruct the geometry of the space for path planning and navigation. It is also essential to understand the relationships between objects, so complete 3D semantic reconstruction is necessary in such cases.

There can be both advantages and disadvantages if a home robot is small. A tiny robot can find dropped and forgotten objects, enter hard-to-reach places,

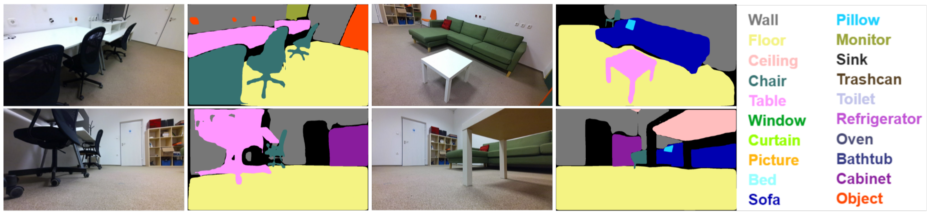


Fig. 1: **Predicted semantic segmentation from different perspectives.** Semantic segmentation algorithms can fail on images taken from unusual perspectives, such as those captured by small ground robots or drones. (Best viewed in color.)

and occupies little space in the home. At the same time, it can encounter objects that are not in the training database or see them from an unusual viewpoint [1] (see examples in Figure 1). It is highly likely that a single object detection model would not work from arbitrary viewpoints without a long training process, and training images may not be available at all. In such cases, semantic matching between a robot and a human can be performed instead.

One option is to pass the observation to a human partner and request semantic information, e.g., via verbal communication. Another option can be to use semantic scene completion [2] to infer the labels of the unknown regions. However, this would require complex models that cannot run in real time on typical robots today. A third option can be the usage of a semantic map of a higher viewpoint to estimate the semantics of the lower viewpoint. This way, we can also provide the system with low-angle training samples to improve itself.

Building upon the semantic reconstruction works of [3] and [4], we propose ways to generate semantic segmentations from lower viewpoints given an existing semantic reconstruction. We study the effects of changing the camera heights and compare different segmentation models. We evaluate the proposed label projection approaches on synthetic data coming from the Habitat [5] simulation framework and on a real-world dataset. Finally, we show that our method can provide semantic segmentation from the lower viewpoints with similar accuracy as the semantic reconstruction from the upper viewpoint. Our implementation is available at `https://github.com/szilviaszeier/semantic_matching`.

2 Related works

Robotic systems have been assisting us more and more in our daily lives. Initially, these agents were hard-coded to complete a specified task without any ability to generalize. However, in recent years more emphasis has been put on their autonomy. Spatial artificial intelligence (SAI) can be divided into four distinct but interdependent layers with increasing complexity: spatial perception, pose tracking, geometry understanding, and semantic understanding. Information about a given environment can be stored within a 3D semantic reconstruction, and later this semantic 3D map can be utilized by the same or even by other agents to complete various tasks. Reconstruction of an agent's environment lies at the core of our work. In this section, we briefly discuss relevant reconstruction methods and introduce the problem of segmentation failing from unusual perspectives, such as that of small ground robots.

2.1 3D Semantic Reconstruction

3D semantic reconstruction of indoor environments refers to the task of recreating the geometry and appearance of our surroundings. The reconstruction can be produced using different techniques. Some utilize a LiDAR scanner and reconstruct the environment with a structure-from-motion algorithm [6], while others achieve this by traversing the area with an RGBD or stereo camera while tracking its location and orientation. In our work, we focus on RGBD cameras.

Due to the complexity and high dimensionality of this task, many solutions only work offline [7, 8]. In robotic applications, real-time approaches like BundleFusion [9] become much more relevant. For example in a navigation task, the robot needs to be aware of immediate changes in its environment to be able to generate a collision-free trajectory [10].

Another aspect is the way in which the semantic labels of the reconstruction are created. Some methods utilize 2D semantic segmentation algorithms [11], while others perform the segmentation on the 3D geometry itself [12, 13]. There are many publicly available 2D semantic segmentation methods making this approach an attractive choice. On the other hand, 3D segmentation methods have the advantage of being able to utilize the geometry of a given object, but as these techniques are more time-consuming, there are much fewer variants.

When it comes to semantic reconstruction, typical methods isolate components of and fuse the results later in a pipeline architecture. However, this is not always the case. The authors of [14] proposed an approach to jointly infer 3D geometry and 3D semantic labels with the use of a depth fusion network. This method leverages the 2D semantic prior to enhance 3D reconstruction accuracy, meaning that out-of-distribution viewpoints would not be supported, which is the main problem we aim to tackle here.

2.2 Object Detection

In our work, we need to segment object masks in 2D images. Such algorithms often have to make a trade-off between inference speed (one-stage methods) and accuracy (two-stage methods). One-stage approaches include the YOLO series [15–17], while main representatives of the two-stage methods are the R-CNN series, including faster R-CNN [18] and R-FCN [19]. Most of the available object detection algorithms are trained on datasets such as [20–22], which mainly contain images taken from a standard (human) perspective. This means that when applied to images taken from an unusual viewpoint, such as that of a small ground robot or even a drone, the detection is unreliable or fails.

3 Methods

In this section, we individually detail our contributions towards 3D semantic matching across multiple viewpoints. We first describe the structure of our pipeline, briefly mentioning previous components adopted from [4].

3.1 General Pipeline

In the previous work [4], the main focus was on real-time semantic reconstruction of indoor environments. The reconstruction pipeline includes UcoSLAM [23] as the visual SLAM method used for pose tracking, and Mask R-CNN [24] as the 2D semantic segmentation algorithm trained on the SUNRGBD dataset [25]. A point cloud filtering procedure was also introduced to filter out erroneous pose estimates that may result from false re-localization or drifts in the pose. The final 3D model of the environment is constructed using the Voxblox [26] framework by integrating measurements from each sensor into a global map. The reconstruction can be colored or semantically labeled, depending on the inputs. As we focus on semantic reconstructions, we paint the model with semantic labels.

We extended the pipeline of [4] with different components. The proposed architecture is shown in Figure 2. We aim to obtain a semantic reconstruction from unusual viewpoints starting from a partial human-perspective reconstruction. To capture data from an odd perspective, we built a small ground robot to use within our experimentation. The starting reconstruction is provided by the reconstruction pipeline using a typical 2D semantic segmentation algorithm. Originally Mask R-CNN was used for semantic labeling, but we also included the recent segmentation algorithm called Mask Transfiner [27]. We trained this model on a mixture between the SUNRGBD [25] and the ADE20K [28] dataset. For comparability, the Mask R-CNN was also trained and evaluated on the same dataset.

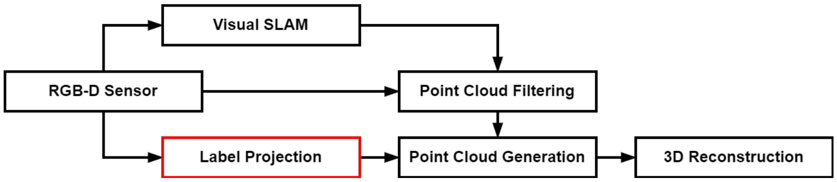


Fig. 2: **Overall pipeline.** We build upon the work of [4]. We added a label projection module (highlighted in red) to infer the missing semantic labels from a lower viewpoint, given an existing 3D semantic reconstruction.

From the starting reconstruction, our goal is to extend and improve the model to unusual perspectives. The semantic segmentation algorithms are not reliable from these odd perspectives, so to obtain a labelling, we proposed several different projection methods, which serve as a mapping from the starting reconstruction to the perspective of the ground robot. These contribution are detailed in the following sections.

3.2 Label Projections

The label projection module aims to provide 2D semantic segmentation to the point cloud generation module. However, when the camera is close to the floor, the accuracy of the pre-trained semantic segmentation degrades. Therefore, we propose methods to provide high-quality semantic segmentation when the camera position is not optimal, by incorporating labels from an existing 3D semantic mesh. The following sections present the proposed methods. As a baseline, we include two 2D semantic segmentation neural networks pre-trained on the upper-view dataset.

Superpixel-based Projection For semantics-independent clustering of the RGB image, we use superpixel technology. Superpixel segmentation is a clustering algorithm that aims to cluster regions of an image based on some similarity metrics such as color, texture, and proximity. For superpixel segmentation, we use Fast-SLIC [29], which is an optimized version of SLIC [30] (Simple Linear Iterative Clustering) for CPU-constrained devices.

This approach consists of the following steps: (i) We project the semantic reconstruction from the upper viewpoint onto the lower-view image plane. (ii) We use Fast-SLIC on the RGB frame from the lower viewpoint to find coherent image regions and aggregate the projected semantic labels. (iii) We use majority voting to assign semantic labels to each superpixel. This approach will be referred to as “2D SLIC”. While this method only works with 2D images, the following methods also use 3D geometry.

Superpixel-based Projection in 3D This method extends the previous 2D superpixel method. We still run the superpixel segmentation on the RGB frame, but the label assignment is done in 3D. Algorithm 1 describes the method in detail. For the rest of the paper, we refer to this method as “3D projected SP”.

Supervoxel-based Projection In addition to the generalization of the label assignment, one can extend superpixel segmentation to 3D. In this case, we refer to the resulting 3D regions as supervoxels. Out of the several approaches for supervoxel segmentation, we use an extension of SLIC called maskSLIC [31].

To implement this approach, we need to convert the point cloud to a voxel grid. Let $P \in \mathbb{R}^{N \times 3}$ be a point cloud, $\bar{P} = \frac{1}{N}\sum_{j=1}^{N} P^{(j)}$ the center of the point cloud and $\mathcal{V} \in \mathbb{R}$ the voxel size. Then, we can convert point cloud P to a voxel grid with the following formula:

$$V = \frac{P - \bar{P}}{\mathcal{V}}.$$

```
Algorithm 1: Superpixel-based Projection in 3D
  Input: Lower-view RGBD image ($I \in [0,255]^{H \times W \times 3}$ and $D \in [0,255]^{H \times W}$),
         upper-view semantic reconstruction (including point cloud
         $P_{upper} \in \mathbb{R}^{N \times 3}$ and labels $L_{upper} \in \{0,1,\dots,L\}^N$, where $L \in \mathbb{N}$
         denotes the number of semantic labels).
  Output: Lower-view semantic segmentation ($S \in \{0,1,\dots,L\}^{H \times W}$)
1 Run superpixel segmentation (SLIC) on the RGB image ($I$) to find coherent
   image regions ($SP$).
2 Using the depth image ($D$), project the superpixel segmentation ($SP$) to 3D
   to get a point cloud ($P_{lower}$).
3 [Optional] Downsample the point cloud ($P_{lower}$).
4 Match the lower-view point cloud ($P_{lower}$) with the upper-view semantic
   reconstruction ($P_{upper}, L_{upper}$) and determine the corresponding semantic
   label of each superpixel ($L_{lower}$).
5 Project the labels of the lower-view semantic point cloud ($L_{lower}$) onto the
   image plane to get the semantic segmentation ($S$).
```

As we increase the number of dimensions, we need to decrease the shape of the voxel grid to keep the runtime low. To this end, we introduce heavy subsampling on the point cloud before the voxelization. Furthermore, we use SLIC with a foreground mask to ignore empty voxels, and we limit the number of supervoxels. Algorithm 2 describes the method in more detail. This method is hereafter referred to as “3D SLIC” method.

```
Algorithm 2: Supervoxel-based Projection
  Input: Lower-view RGBD image ($I \in [0,255]^{H \times W \times 3}$ and $D \in [0,255]^{H \times W}$),
         upper-view semantic reconstruction (including point cloud
         $P_{upper} \in \mathbb{R}^{N \times 3}$ and labels $L_{upper} \in \{0,1,\dots,L\}^N$, where $L \in \mathbb{N}$
         denotes the number of semantic labels).
  Output: Lower-view semantic segmentation ($S \in \{0,1,\dots,L\}^{H \times W}$)
1 Using the lower-view RGB ($I$) and depth image ($D$), create a colored 3D point
   cloud ($P_{lower}$).
2 Downsample the point cloud ($P_{lower}$) and keep track of each point's original
   location.
3 Convert the lower-view point cloud ($P_{lower}$) to a voxel grid ($V_{lower}$).
4 Run SLIC on the voxel grid ($V_{lower}$) with masking to get supervoxels ($SV$).
5 Match the lower-view point cloud ($P_{lower}$) with the upper-view semantic
   reconstruction ($P_{upper}, L_{upper}$) and determine the corresponding semantic
   label of each supervoxel ($L_{lower}$).
6 Project the labels of the lower-view semantic point cloud ($L_{lower}$) onto the
   image plane to acquire the semantic segmentation ($S$).
```

3D Clustering-based Projection We propose another clustering approach, which starts by finding planes (e.g., floor, ceiling, and walls) and then cluster the rest of the scene. For this, we use RANSAC [32] and DBSCAN [33] iteratively. Algorithm 3 details the proposed approach. This method can handle point clouds of hundreds of thousands of points, but subsampling the point cloud is advantageous in terms of speed. The number of supervoxels cannot be adjusted directly, it depends on the point density, and therefore it cannot distinguish regions with fine details. Furthermore, this method is sensitive to noise without fine-tuning the parameters. For the rest of the paper, we refer to this approach as "DBSCAN".

```
Algorithm 3: 3D Clustering-based Projection
Input: Lower-view depth image ($D \in [0, 255]^{H \times W}$), upper-view semantic
       reconstruction (including point cloud $P_{upper} \in \mathbb{R}^{N \times 3}$ and labels
       $L_{upper} \in \{0, 1, \ldots, L\}^N$, where $L \in \mathbb{N}$ denotes the number of semantic
       labels).
Output: Lower-view semantic segmentation ($S \in \{0, 1, \ldots, L\}^{H \times W}$)
Using the lower-view depth image ($D$), create a 3D point cloud ($P_{lower}$).
Downsample the point cloud ($P_{lower}$) and keep track of each point's original
  location.
while $P_{lower}$ contains planes do
    Segment plane: Use RANSAC to find planes and select the largest
      coherent region by applying DBSCAN ($P_{cluster}$).
Cluster the rest of the point cloud ($P_{lower}$, those points that were not part of
  any plane) using DBSCAN ($P_{cluster}$).
Match the lower-view clustered point cloud ($P_{cluster}$) with the upper-view
  semantic reconstruction ($P_{upper}, L_{upper}$) and determine the corresponding
  semantic label of each cluster ($L_{cluster}$).
Trace back and project the labels of the clustered semantic point cloud
  ($L_{cluster}$) onto the image plane to get the semantic segmentation ($S$).
```

Per-point matching We project the lower-view RGBD frame to 3D, then we compare the points from the resulting point cloud with the upper-view point cloud and assign the corresponding upper-view labels.

Let $P_{lower} \in \mathbb{R}^{M \times 3}$ the lower-view and $P_{upper} \in \mathbb{R}^{N \times 3}$ the upper-view point cloud, and $L_{upper} \in \{0, 1, \ldots, L\}^N$ the corresponding labels, where $L \in \mathbb{N}$. We assign the nearest upper-view label to each lower-view point:

$$L_{lower}^{(j)} := L_{upper}^{(i)},$$

where $L_{lower} \in \{0, 1, \ldots, L\}^M$ is the labels of the lower-view point cloud, $\|.\|$ denotes the $\mathcal{L}_2$ norm, and $i \in [1, N], j, k \in [1, M], j \neq k$ are indices, such that

$$\left\| P_{lower}^{(i)} - P_{lower}^{(j)} \right\|_2 < \left\| P_{lower}^{(i)} - P_{lower}^{(k)} \right\|_2 .$$

Finally, we project L_{lower}, the lower-view semantic labels onto the image plane to obtain the semantic segmentation. We refer to this approach as "3D neighborhood" method.

3.3 Benchmark Measures

When comparing meshes, we transform them into semantic voxel grids and then compare them with the following specified metrics.

Jaccard We calculate an intersection over union (IoU) where the union is the total number of voxels within the two voxel grids that are occupied within at least one of them, and the intersection is the number of voxels for which the position and the color match in the two voxel grids. We call this the "color intersection". The ratio of these two values gives the Jaccard index.

Sorensen The Sorensen similarity index is calculated as

$$\frac{2 \cdot |X \cap Y|}{|X| + |Y|},$$

where the numerator specifies the double of the color intersection and the denominator is the total number of voxels within the compared voxel grids.

Color Acc. The color accuracy is the ratio between the color intersection and the intersection of occupied voxels.

Mean Acc. The main benchmark measure is the mean accuracy, which we define as the mean of the other three measures, the Jaccard, the Sorensen, and the Color Acc.

4 Experimental Results

To prove the validity of our work, tested our methods both with synthetic data and in real scenarios. We first present the results on synthetic input created with the Habitat simulator, and show the real-world tests afterwards.

4.1 Experimental setup

We trained the semantic segmentation models (Mask R-CNN [24] and Transfiner [27]) on a mixture between the SUNRGBD [25] and ADE20K [28] datasets.

We used the label projection methods with the following parameters: For the 2D SLIC and the 3D projected SP method, we set the number of superpixels to 512, and in the case of the 3D projected SP we only kept every 4^{th} point of the point cloud. In the case of the 3D SLIC method, we subsampled the point cloud by setting the voxel size $\mathcal{V}$ to 0.1, and we limited the number of supervoxels to 128. Finally, in the case of the DBSCAN label projection method, a voxel size $\mathcal{V}$ of 0.04 was used for the downsampling, and we set the ϵ parameter of DBSCAN to 0.1, which determines the maximum distance between points within a neighborhood. The rest of the parameters can be found in the codebase[4].

We measured the runtime of each label projection method on a server equipped with an AMD EPYC 7401P CPU and 3 NVIDIA GeForce RTX 2080 Ti GPUs. Despite the optimizations, 3D SLIC and DBSCAN could only run at around 1 frame per second (FPS). However, the 3D projected SP and the 3D neighborhood could perform at around 20 and 10 FPS, respectively (see Table 1).

2D SLIC	3D SLIC	3D projected SP	DBSCAN	3D neighborhood
2.29	1.04	9.57	0.84	20.06

Table 1: **Speed of the label projection methods.** The measurements are in frame per second (FPS). For more detail, see Section 4.1.

Ground Robot For our experiments, we built a small ground robot. We 3D-printed the frame of OpenBot [34], collected and assembled the electrical components from scratch, and adapted the Arduino code to fit our needs. The brain of our agent is the NVIDIA Jetson Nano [35] embedded computer, and we equipped the robot with an Azure Kinect RGBD camera.

Test data We conducted the experiments in both a simulated and a real-world environment.

To generate the simulated dataset, we used the Habitat [5] simulation framework. While this simulator supports multiple datasets, we only used it with the Replica [36] dataset, which contains several photo-realistic models of indoor scenes. We used the following environments: `frl_apartment_4`, `room_2`, and `office_3`. Depending on the task, we generated different trajectories. We ran the experiments with every scene and reported the average results. We opted for a simulator as it has the added benefit that the ground truth camera poses can be queried as opposed to an online visual SLAM method.

To test the proposed methods, we also recorded a custom real-world dataset. The dataset consists of upper- and lower-view video feeds recorded with an Azure Kinect camera in 2 different rooms. Given the video feeds, we reconstructed the 3D scene and annotated the resulting mesh by hand.

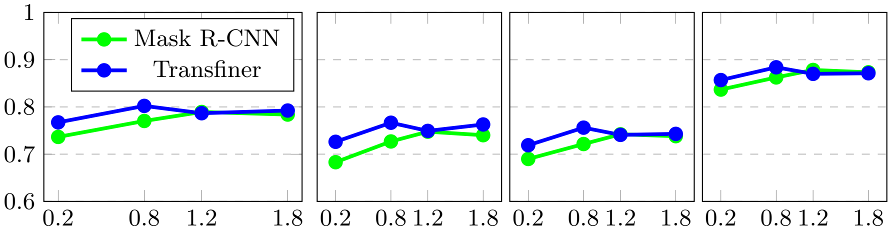
4.2 Influence of the viewpoint

The Habitat simulation framework was used to investigate the effects of unconventional perspectives. For each scene, a trajectory was generated by periodically tilting the camera up and down by 20 degrees. The same trajectory was used

[4] `https://github.com/szilviaszeier/semantic_matching`

for each experiment, but the camera height was varied. Figure 3 shows how the segmentation accuracy is affected by changing the camera height.

Two segmentation models were used, a Mask R-CNN and a Transfiner pre-trained on the SUNRGBD and the ADE20K [25, 28]. The models performed the worst at 0.2 m height, with mean accuracies of 0.737 and 0.767, while the best results were obtained at 0.8 m height, with mean accuracies of 0.770 and 0.802. Further increment of the height resulted in a slight decay in accuracy.



(a) Mean Acc. (b) Jaccard (c) Sorensen (d) Color Acc.

Fig. 3: **Segmentation accuracy as the height of the camera changes.** The same trajectories were used during the evaluations, only the height of the camera was changed. The trajectories were generated by tilting the camera up and down by 20°. Both models were trained on the same dataset. Mean Acc. represents the mean of the other three (Jaccard, Sorensen and Color Acc.) metrics.

4.3 Semantic reconstruction of synthetic scenes

To generate synthetic input data, we used the Habitat simulation environment and the following scenes from the Replica dataset: `frl_apartment_4`, `room_2`, `office_3`. We manually controlled the virtual camera along two distinct trajectories within each room, an upper and a lower one, and saved the generated images. In both cases, the aim was to explore the whole room. We ran the semantic reconstruction method from the upper trajectory using different models. Then, we executed the proposed label projection methods given the upper-view reconstruction to generate the lower-view semantic reconstruction. The results are shown in Table 2.

As a baseline, we present the accuracy of the semantic reconstruction using only the semantic segmentation models. We also indicate the accuracy of the label projection methods with the ground truth upper-view reconstruction to demonstrate the upper limit (in terms of the benchmark measures) of the proposed methods.

In all cases, the proposed label projection methods outperformed the baseline methods, i.e., semantic reconstruction using deep learning models applied to the odd viewpoints. The 3D neighborhood method achieved the highest accuracy

Model	View	Method	Jaccard	Sorensen	Color Acc.	Mean Acc.
Ground Truth	Upper	-	-	-	-	-
	Lower	-	-	-	-	-
		2D SLIC	0.813	0.868	0.955	0.879
		3D SLIC	0.892	0.922	0.975	0.930
		3D projected SP	0.894	0.921	0.969	0.928
		DBSCAN	0.859	0.865	0.953	0.892
		3D neighborhood	**0.917**	**0.944**	**0.985**	**0.949**
Mask R-CNN	Upper	-	0.778	0.754	0.867	0.800
	Lower	-	0.737	0.720	0.841	0.766
		2D SLIC	0.728	0.717	0.852	0.766
		3D SLIC	0.781	0.746	0.861	0.796
		3D projected SP	0.794	0.752	0.860	0.802
		DBSCAN	0.793	**0.757**	**0.871**	0.807
		3D neighborhood	**0.800**	**0.757**	0.865	**0.808**
Transfiner	Upper	-	0.794	0.758	0.867	0.806
	Lower	-	0.734	0.708	0.832	0.758
		2D SLIC	0.718	0.707	0.851	0.759
		3D SLIC	0.790	0.746	0.862	0.799
		3D projected SP	0.798	0.749	0.860	0.802
		DBSCAN	0.796	0.753	**0.871**	0.807
		3D neighborhood	**0.804**	**0.754**	0.866	**0.808**

Table 2: **Results on the synthetic dataset.** The “View” denotes the camera position. The trajectories differ between the Upper and the Lower view. From the Upper viewpoint, we create a semantic map using per-frame semantic segmentation from different models. (The Ground Truth model indicates results when the upper-view semantic reconstruction is provided by the simulator.) From the Lower viewpoint, we either use a pre-trained model or the Upper-view semantic reconstruction with our proposed label projection methods to provide per-frame semantic segmentation to create a Lower-view semantic reconstruction.

regardless of which model was used for the upper-view semantic reconstruction. The DBSCAN approach achieved similar color accuracy when used in conjunction with deep learning models but fell short when using the ground truth mesh due to its inability to detect fine details without thorough fine-tuning. The 3D SLIC and 3D projected SP methods performed similarly. They outperformed DBSCAN with the ground truth and provided comparable results when used with deep learning models.

Figure 4 shows the semantic reconstructions using each approach. The label projection methods were guided by the ground truth upper-view semantic mesh.

4.4 Semantic reconstruction of real scenes

Experiments done on the synthetic dataset do not provide a clear picture of the real-world performance of our methods as the data lacks noise and measurement errors. Therefore, we tested the proposed methods on real-world data as well.

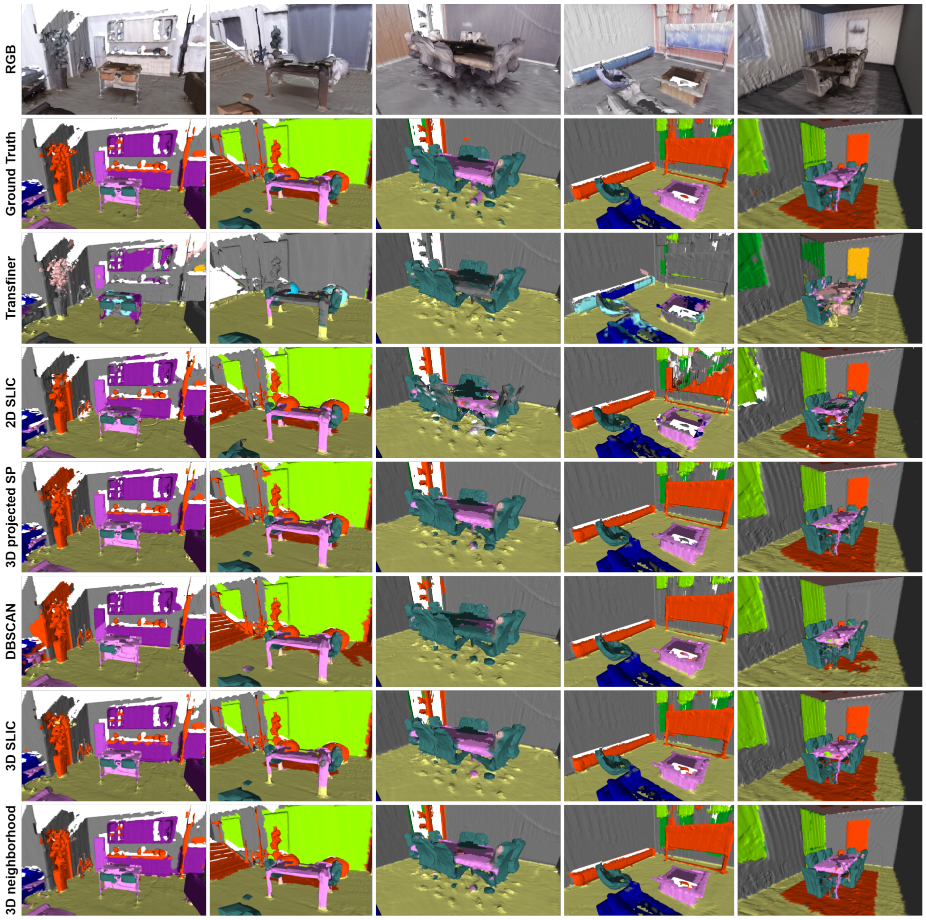


Fig. 4: **Qualitative results.** Each column shows a different environment from the Replica dataset. The label projection methods were guided by the ground truth upper-view semantic reconstructions. We use the same color scheme to represent the labels as in Figure 1. (Best viewed in color.)

Similarly to Section 4.3, we also had recordings from two different points of view: a lower and an upper. The recording from the upper viewpoint represents the human perspective, which was used to create semantic reconstruction to guide the label projection methods. We used the same deep learning models without changing the weights. In Table 3, we compare the scores of the semantic reconstruction using our different label projection strategies.

Model	View	Method	Jaccard	Sorensen	Color Acc.	Mean Acc.
Mask R-CNN	Lower	-	0.379	0.522	0.868	0.590
		2D SLIC	0.460	0.604	0.898	0.654
		3D SLIC	0.513	0.644	0.892	0.683
		3D projected SP	0.635	0.727	0.878	0.747
		DBSCAN	0.547	0.680	**0.919**	0.715
		3D neighborhood	**0.665**	**0.749**	0.891	**0.768**
Transfiner	Lower	-	0.384	0.527	0.864	0.592
		2D SLIC	0.509	0.648	0.912	0.690
		3D SLIC	0.565	0.682	0.891	0.713
		3D projected SP	0.697	0.767	0.882	0.782
		DBSCAN	0.540	0.677	**0.927**	0.715
		3D neighborhood	**0.718**	**0.777**	0.881	**0.792**

Table 3: **Results on the real-world dataset.** We used the same methods as described in Table 2.

As in the simulated experiments, the 3D neighborhood method achieved the highest mean accuracy, while DBSCAN achieved the highest color accuracy. In contrast to the previous results, DBSCAN performed worse compared to the other methods. This is due to the lack of robustness to noise, which could be mitigated by fine-tuning the parameters. However, this would require ground truth samples, thereby rendering further evaluation unfair as other methods do not require supervision. The 3D SLIC approach also achieves significantly lower average accuracy than 3D projected SP due to the measurement noise and the steps taken to reduce the execution time for real-time runs, i.e., the strong subsampling of the point cloud and the significantly lower number of supervoxels. However, the 3D projected SP approach achieved a significantly higher mean accuracy as it could be executed at a higher resolution.

5 Conclusions

Several solutions have been proposed for the semantic reconstruction of indoor environments. However, for semantic segmentation algorithms, the ability to recognize objects in the case of drastic perspective changes has not yet been fully addressed. In this work, we proposed a modular pipeline for semantically reconstructing indoor environments from unusual perspectives, such as one from a small ground robot. To achieve our goal, we utilize a superpixel technique and its variations, and the geometry of the surroundings. Our pipeline starts with a partial semantic reconstruction from the human perspective, which gets extended to the new, unusual perspective.

We experimented in both simulated and real-world scenarios with two different 2D semantic segmentation networks. The proposed label projection methods can provide semantic segmentation from lower viewpoints with accuracy similar to the human perspective. Thereby, label transfer and the fine-tuning of semantic segmentation networks to these perspectives becomes possible.

The resulting reconstruction is a class-level semantic segmentation of the 3D geometry, which means that we can differentiate between categories but not between the instances themselves. In the future, we plan to incorporate instance-level segmentation and enable panoptic 3D reconstructions of the environments.

Author Contributions

Szilvia Szeier: Conceptualization, Methodology - 2D and 3D Label Projection methods, Software, Writing, Visualization. **Benjámin Baffy**: Methodology - 2D Label Projection method, Software, Investigation, Visualization. **Gábor Baranyi**: Investigation, Software - model training **Joul Skaf**: Investigation, Software - model training, Visualization. **László Kopácsi**: Conceptualization, Methodology - 3D Label Projection methods, Writing. **Daniel Sonntag**: Conceptualization, Supervision, Project administration. **Gábor Sörös**: Conceptualization, Software, Paper revision. **András Lőrincz**: Conceptualization, Methodology, Writing - original draft, Supervision, Revision, Project administration.

Acknowledgements

This work was partially supported by (i) the European Commission funded project “Humane AI: Toward AI Systems That Augment and Empower Humans by Understanding Us, our Society and the World Around Us” (grant #820437), (ii) project RRF-2.3.1-21-2022-00004 of the European Union within the framework of the Artificial Intelligence National Laboratory, (iii) the MOBOT project (grant #2020-1.1.2-PIACI-KFI-2020-00115) on the development of AI supported automatic warehouse storage system based on machine vision, and (iv) the project “Application Domain Specific Highly Reliable IT Solutions” implemented with the support of the National Research, Development and Innovation Fund of Hungary, financed under the Thematic Excellence Programme TKP2020-NKA-06 (National Challenges Subprogramme) funding scheme. The authors thank Robert Bosch, Ltd. Budapest, Hungary for their generous support to the Department of Artificial Intelligence of the Eötvös Loránd University.

References

1. X. Li, J. Liu, J. Baron, K. Luu, and E. Patterson, “Evaluating effects of focal length and viewing angle in a comparison of recent face landmark and alignment

methods," *EURASIP Journal on Image and Video Processing*, vol. 2021, no. 1, pp. 1–18, 2021.

2. L. Roldao, R. De Charette, and A. Verroust-Blondet, "3d semantic scene completion: a survey," *International Journal of Computer Vision*, pp. 1–28, 2022.
3. A. Rosinol, M. Abate, Y. Chang, and L. Carlone, "Kimera: an open-source library for real-time metric-semantic localization and mapping," in *IEEE Intl. Conf. on Robotics and Automation*, ICRA, 2020.
4. D. Rozenberszki, G. Soros, S. Szeier, and A. Lorincz, "3D semantic label transfer in human-robot collaboration," in *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 2602–2611, 2021.
5. M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, D. Parikh, and D. Batra, "Habitat: A platform for embodied AI research," in *IEEE/CVF International Conference on Computer Vision*, ICCV, 2019.
6. W. Zhen, Y. Hu, H. Yu, and S. Scherer, "LiDAR-enhanced structure-from-motion," in *IEEE International Conference on Robotics and Automation*, ICRA, 2020.
7. S. Agarwal, Y. Furukawa, N. Snavely, I. Simon, B. Curless, S. M. Seitz, and R. Szeliski, "Building Rome in a day," *Communications of the ACM*, vol. 54, 2011.
8. J. L. Schönberger and J.-M. Frahm, "Structure-from-motion revisited," in *Conference on Computer Vision and Pattern Recognition*, CVPR, 2016.
9. A. Dai, M. Nießner, M. Zollöfer, S. Izadi, and C. Theobalt, "BundleFusion: Real-time globally consistent 3D reconstruction using on-the-fly surface re-integration," *ACM Transactions on Graphics*, 2017.
10. 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, July 2019.
11. J. McCormac, A. Handa, A. Davison, and S. Leutenegger, "SemanticFusion: Dense 3D semantic mapping with convolutional neural networks," *IEEE International Conference on Robotics and Automation*, 2017.
12. Z. Murez, T. van As, J. Bartolozzi, A. Sinha, V. Badrinarayanan, and A. Rabinovich, "Atlas: End-to-end 3D scene reconstruction from posed images," in *ECCV*, 2020.
13. D. Zhang, J. Chun, S. Cha, and Y. M. Kim, "Spatial Semantic Embedding Network: Fast 3D Instance Segmentation with Deep Metric Learning," in *arXiv preprint arXiv:2007.03169*, 2020.
14. D. Menini, S. Kumar, M. R. Oswald, E. Sandström, C. Sminchisescu, and L. Van Gool, "A real-time online learning framework for joint 3d reconstruction and semantic segmentation of indoor scenes," *IEEE Robotics and Automation Letters*, vol. 7, no. 2, pp. 1332–1339, 2021.
15. J. Redmon and A. Farhadi, "Yolov3: An incremental improvement," *arXiv preprint arXiv:1804.02767*, 2018.
16. A. Bochkovskiy, C.-Y. Wang, and H.-Y. M. Liao, "Yolov4: Optimal speed and accuracy of object detection," *arXiv preprint arXiv:2004.10934*, 2020.
17. C.-Y. Wang, A. Bochkovskiy, and H.-Y. M. Liao, "Scaled-YOLOv4: Scaling cross stage partial network," in *IEEE Conference on Computer Vision and Pattern Recognition*, pp. 13029–13038, June 2021.
18. S. Ren, K. He, R. Girshick, and J. Sun, "Faster R-CNN: Towards real-time object detection with region proposal networks," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, 06 2015.

19. J. Dai, Y. Li, K. He, and J. Sun, "R-fcn: Object detection via region-based fully convolutional networks," *Advances in neural information processing systems*, vol. 29, 2016.
20. T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C. L. Zitnick, "Microsoft coco: Common objects in context," in *ECCV*, pp. 740–755, Springer, 2014.
21. J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei, "Imagenet: A large-scale hierarchical image database," in *2009 IEEE conference on computer vision and pattern recognition*, pp. 248–255, 2009.
22. M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The pascal visual object classes (voc) challenge," *International journal of computer vision*, vol. 88, no. 2, pp. 303–338, 2010.
23. R. Muñoz-Salinas and R. Medina-Carnicer, "UcoSLAM: Simultaneous localization and mapping by fusion of keypoints and squared planar markers," *Pattern Recognition*, vol. 101, 2020.
24. K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *IEEE International Conference on Computer Vision*, 2017.
25. S. Song, S. P. Lichtenberg, and J. Xiao, "SUN RGB-D: A RGB-D scene understanding benchmark suite," in *IEEE Conference on Computer Vision and Pattern Recognition*, CVPR, 2015.
26. 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.
27. L. Ke, M. Danelljan, X. Li, Y.-W. Tai, C.-K. Tang, and F. Yu, "Mask transfiner for high-quality instance segmentation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 4412–4421, 2022.
28. B. Zhou, H. Zhao, X. Puig, T. Xiao, S. Fidler, A. Barriuso, and A. Torralba, "Semantic understanding of scenes through the ade20k dataset," *International Journal of Computer Vision*, vol. 127, no. 3, pp. 302–321, 2019.
29. A. Kim, "Fast-slic." `https://github.com/Algy/fast-slic`. Last accessed: 2021-11-01.
30. R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk, "Slic superpixels compared to state-of-the-art superpixel methods," *IEEE transactions on pattern analysis and machine intelligence*, vol. 34, no. 11, pp. 2274–2282, 2012.
31. B. Irving, "maskSLIC: regional superpixel generation with application to local pathology characterisation in medical images," *arXiv preprint arXiv:1606.09518*, 2016.
32. M. A. Fischler and R. C. Bolles, "Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography," *Commun. ACM*, vol. 24, no. 6, p. 381–395, 1981.
33. M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, KDD'96, p. 226–231, AAAI Press, 1996.
34. M. Müller and V. Koltun, "OpenBot: Turning smartphones into robots," in *International Conference on Robotics and Automation*, ICRA, 2021.
35. NVIDIA, "Jetson nano." `https://developer.nvidia.com/embedded/jetson-nano`. Last accessed: 2022-05-08.
36. J. Straub, T. Whelan, and L. M. et al., "The Replica dataset: A digital replica of indoor spaces," *arXiv preprint arXiv:1906.05797*, 2019.