2016 Fourth International Conference on 3D Vision

Multi-Label Semantic 3D Reconstruction using Voxel Blocks

Ian Cherabier[1], Christian Häne[2], Martin R. Oswald[1], Marc Pollefeys[1,3]

[1]ETH Zürich, Switzerland
{ian.cherabier, martin.oswald, marc.pollefeys}@inf.ethz.ch

[2]University of California, Berkley
chaene@eecs.berkeley.edu

[3]Microsoft, USA

Abstract

Techniques that jointly perform dense 3D reconstruction and semantic segmentation have recently shown very promising results. One major restriction so far is that they can often only handle a very low number of semantic labels. This is mostly due to their high memory consumption caused by the necessity to store indicator variables for every label and transition. We propose a way to reduce the memory consumption of existing methods. Our approach is based on the observation that many semantic labels are only present at very localized positions in the scene, such as cars. Therefore this label does not need to be active at every location. We exploit this observation by dividing the scene into blocks in which generally only a subset of labels is active. By determining early on in the reconstruction process which labels need to be active in which block the memory consumption can be significantly reduced. In order to recover from mistakes we propose to update the set of active labels during the iterative optimization procedure based on the current solution. We also propose a way to initialize the set of active labels using a boosted classifier. In our experimental evaluation we show the reduction of memory usage quantitatively. Eventually, we show results of joint semantic 3D reconstruction and semantic segmentation with significantly more labels than previous approaches were able to handle.

1. Introduction

Combining dense 3D reconstruction and semantic segmentation, two of the core tasks in computer vision, is a recent idea that is receiving more and more attention. Performing these tasks jointly leads to improvements for both of them. Knowing the semantic class of an object gives prior evidence about its shape. Inversely, having access to the shape and location of an object helps refining the segmentation. Moreover, it leads to a more complete representation of the scene by combining geometric and semantic information.

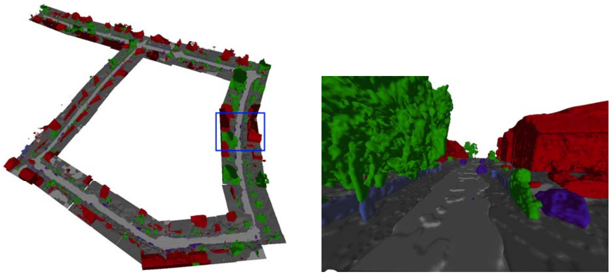

(a) Full scene (b) Close-up

Figure 1. A top view of a sequence from the KITTI data set and a close-up of the indicated region.

Volumetric methods provide a natural framework for semantic 3D reconstruction [10]. Besides determining if a voxel is in the free or occupied space, we need to determine the semantic class of the object in the latter case, i.e. the binary labeling problem that is usually considered in volumetric methods for dense reconstruction becomes a multi-labeling problem. One label corresponds to free space and the others to specific semantic categories such as *building* or *ground*.

If we represent the scene as a voxel grid where all voxels have equal size, as in [10], we need to store an indicator variable for every label at every voxel. The main difficulty when going towards many labels that rapidly emerges is the scalability in terms of number of labels: the more labels we consider in the scene, the more memory is needed to store the corresponding variables. This is especially problematic as we do not only need to store indicator variables for every label at every voxel, but we also need to store indicator variables for the transitions between the different labels, leading

978-1-5090-5407-7/16 $31.00 © 2016 IEEE
DOI 10.1109/3DV.2016.68

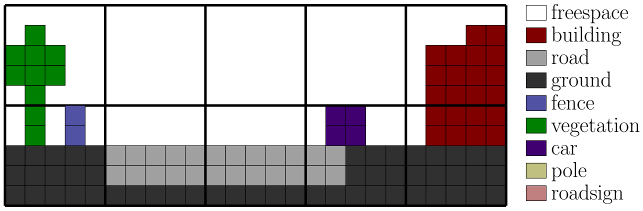


Figure 2. 2D illustration of our voxel block approach. Only a subset of the labels is present per voxel block.

to a quadratic complexity in the number of labels. A way to reduce memory consumption is by using an octree representation as in [1]. However, they only propose a way to scale the spacial extent of the scene but not the number of labels used, and hence they only use up to 6 labels.

Instead of having all the potential labels active at every possible position in the scene, we propose to divide the scene into blocks in which only a set of relevant labels is active. An illustration of this is given in Fig. 2. The motivation behind this is that for many semantic classes it can be determined early on that they are not present in a specific block. If we have no evidence of the presence of a car within a block we can deactivate this label right from the beginning of the numerical optimization. Compared to the standard approach where we would try to directly reconstruct the shape of the non-present car this leads to a much more efficient processing. It especially tackles our main goal of reducing the quadratic complexity in the number of labels which is present in [10]. Our approach is not limited to this specific method but could be adapted to other semantic 3D reconstruction methods.

By determining early on in the reconstruction procedure where the semantic labels are located, and hence which labels are needed in which blocks, we improve the performance and memory consumption significantly. Therefore, before starting the joint semantic segmentation and 3D reconstruction process, we initialize the blocks such that only a set of relevant labels is activated. We present a method using boosted classifiers on the input data. This initialization does not always give perfect results: in a given block, some labels might be missed, or due to noise, labels might be activated which are not present. Therefore, we propose to include updates of the active labels within the blocks during the iterative optimization procedure which is utilized for the reconstruction process. This further reduces the resource usage as we quickly deactivate labels which are not present and hence do not need to update their variables any more in the iterative optimization. We show extensive evaluations on the KITTI dataset [6].

The rest of the paper is organized as follows. Section 2 gives a brief overview of existing work in the domain of semantic 3D reconstruction. In Section 3 we review the method of [10] which we use as a base framework. Our proposed block approach is detailed in Section 4. Finally, we present an experimental evaluation in Section 5.

2. Related Work

Two major computational difficulties arise when dealing with semantic 3D reconstruction: scalability in the spatial extent of the scene, and scalability in the number of labels.

Standard volumetric 3D reconstruction without any semantics has been explored extensively in computer vision [4, 12, 20, 27]. This led to a variety of methods for dealing with the spatial scalability. It is possible for instance to use a data adaptive discretization of the space in form of a Delaunay triangulation [15], an octree data structure to store the geometry [18] or voxel hashing [21]. A hashing based method combined with regularized depth maps from computational stereo, is used for dense street reconstruction in [24].

When using semantic information to enrich the models the problem of scalability in the number of labels is an additional problem that needs to be tackled. The links between semantic and 3D have gained a lot of attention over the last years [2], and effort has been put into fusing both types of information. Most methods choose to map 2D image segmentation to an existing 3D model [25, 23, 26]. [26] combined such an approach with voxel hashing, thus achieving large scale reconstruction.

Another option is to combine geometry and semantics by jointly estimating a depth map and semantic segmentation [17, 8]. Related to semantic are also the similarities of objects in a scene, which can give cues to improve the reconstruction. For instance in [30], repetitive patterns of urban scenes are used in order to detect and exploit object similarities, such as houses, thus leading to an overall better 3D model.

True joint volumetric 3D reconstruction and semantic segmentation was introduced in [10]. While [10] densely labels the volume into semantic classes, [13] only labels the surface voxels into semantic classes. The presence of a dense semantic labeling of the volume becomes apparent when looking at hidden, unobserved surfaces. The road can continue underneath a car or the building facade behind the vegetation even though these surfaces have not been visible in the input data. [14] solves a volumetric labeling problem using a conditional random field (CRF) formulation with image data from a single monocular camera which is mounted on a driving car.

One of the key components in [10] is the use of semantic class dependent surface area penalization as a regularization term. The utilized energy formulation [28], is originating in the spatially continuous, variational, setting. It is therefore not affected by metrication artifacts which are common in discrete graph-based formulations [19]. As mentioned above one of the key benefits is that all the hidden transitions are modeled and hence can have their specific priors on the surface direction. Unfortunately, this comes with the high price of having a quadratic number of variables in the number of labels per voxel in the optimization problem. Therefore, the major limitation of this approach is scalability, in terms of size of the scenes and in terms of number of different semantic labels.

Scaling up the work of [10] in terms of spatial extent is made more difficult by the fact that hidden surfaces need to be reconstructed, which are not handled by a traditional octree and voxel hashing data structure. Recently, [1] proposed to use an adaptive octree data structure with coarse-to-fine optimization to apply [10] on large scale scenes. However, the scalability in terms of number of labels remains an open problem for which we propose a solution in this work.

2.1. Contributions

Our contribution is in summary a method which improves the scalability of semantic 3D reconstruction in terms of the number of labels. In particular we make the following individual contributions:

- A framework that divides the scene into blocks and which takes advantage of its sparseness in terms of location of the different semantic labels.
- A method to initialize the blocks, i.e. determining the set of labels that are active in each block based on the input data.
- A method for updating the set of active labels in each block during the iterative optimization of the energy.

3. Semantic 3D Reconstruction

In this section we state the formulation of the semantic 3D reconstruction problem of [10] and analyze its benefits and memory consumption. We will use this formulation for our new block framework in the subsequent section.

3.1. Energy Formulation

We recall that the semantic 3D reconstruction problem is casted in a volumetric framework. It is thus formulated as a multi-labeling problem in a volumetric domain. We use the formulation of [10], which is stated as a discretized version of a spatially continuous energy formulation. The energy allows for non-metric and anisotropic surface area penalization. More details about the derivation of the energy can be found in [28]. In the following we summarize the parts relevant for the understanding of the manuscript.

We denote Ω the discrete volumetric lattice corresponding to the scene, and s the voxel positions. We consider $L+1$ labels with 0 indicating free space and the rest indicating occupied space with specific semantic categories. For each label $i \in \{0, \dots, L\}$ we introduce an indicator variable $x_s^i \in [0,1]$, with the meaning $x_s^i = 1$ indicates that label i is assigned to voxel s. Formulating the labeling as an energy minimization problem we can write:

$$E\left(\mathbf{x}\right) = \sum_{s \in \Omega} \left(\sum_i \rho_s^i x_s^i + \sum_{i,j:\, i<j} \phi^{ij}\left(x_s^{ij} - x_s^{ji}\right) \right) \quad (1)$$

subject to the following marginalization, normalization and non-negativity constraints

$$\begin{aligned} & x_s^i = \sum_j (x_s^{ij})_k, \quad x_s^i = \sum_j (x_{s-e_k}^{ji})_k \quad (k \in \{1,2,3\}) \\ & \sum_i x_s^i = 1, \quad x_s^i \geq 0, \quad x_s^{ij} \geq 0 \end{aligned} \quad (2)$$

where $e_k \in \mathbb{R}^3$ is the k^{th} canonical 3D unit vector. The $x_s^{ij} \in [0,1]^3$ encode the transition between classes i and j, where $x_s^{ij} - x_s^{ji} \in [-1,1]^3$ is the transition gradient which is aligned with the local surface orientation between i and j at voxel s. We also introduced the data cost ρ_s^i for label i at voxel s, which is built upon evidence from depth maps and pixel-wise semantic classification scores for the input images. Finally $\phi^{ij} : \mathbb{R}^3 \rightarrow \mathbb{R}_0^+$ is a convex positively 1-homogeneous transition-dependent anisotropic surface area penalization function [10, 28, 5]. It encodes the predominant directions of semantic classes by assigning costs to label transitions based on the involved labels and the surface direction, e.g. a building should have vertical walls, while the ground is mostly flat and horizontal.

3.2. Data Cost

In this section we briefly review the way the data term ρ_s^i is defined. The explanation closely follows [10].

The data term is defined through the input depth maps and the pixel-wise semantic classifier scores. It describes the local per voxel cost for assigning label i to voxel s. The construction is cumulative, the information from all input images is added up per voxel. Each voxel s is projected to each input view. For the rest of this section we will consider one such line-of-sight which goes through pixel p in one of the input image I. The data cost which is added to voxel s for image I depends on the position of the voxel on the line-of-sight. Looking at the depth map of I two cases can occur: the depth at p is either observed or unobserved

Observed Depth Assume a depth $\hat{d}$ is observed at pixel p. This means that the surface of an object is very likely to be at depth $\hat{d}$. We define an interval of length 2δ around $\hat{d}$ on the viewing ray where the data cost is such that for a voxel along the ray at a depth $d \in \left[\hat{d} - \delta; \hat{d} + \delta\right]$:

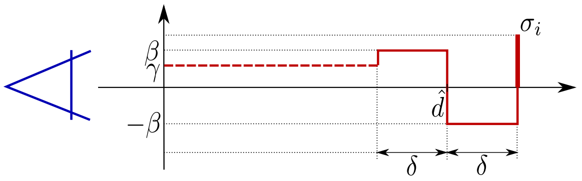


Figure 3. Different data costs which are assigned to voxels along a viewing ray for a observed depth $\hat{d}$.

$$\rho^i_{d+\delta} = \sigma_i \qquad \rho^i_d = \begin{cases} 0 & \text{if } i = 0 \\ \beta \cdot \text{sign}\left(\hat{d} - d\right) & \text{if } i \neq 0 \end{cases} \tag{3}$$

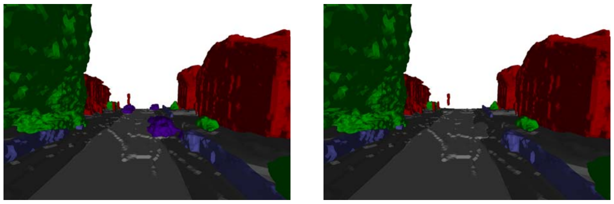
where the cost σ_i depends on the classifications scores: σ_i is low if label i is likely. This cost corresponds to an exponentially distributed noise assumption on the inlier depth values. Figure 3 illustrates this data cost along a viewing ray. The observed depth does not only provide information around the observed depth. Observing a surface at some distance from the camera in theory also implies that we see free space along the whole line-of-sight. However, adding the weight for free space along the whole line-of-sight is not robust against outliers. Therefore, to not completely discard this information but still be robust against outliers we use it with a much lower weight, *i.e.* $\gamma \ll \beta$. In practice this helps to speed up convergence.

Unobserved Depth In this case we have no information about the depth of the potentially visible object at this pixel. However, if we have a strong indication that sky is the semantic class of that pixel we get an indication that the whole ray should be free space. Therefore we define a cost that simply favors the class free space over all the others if sky is the class that obtained the best, i.e. lowest score. If sky is not the most likely label we do not use the evidence from this pixel.

$$\rho^0_s = \min\left\{0, \sigma_0 - \min_{i \neq 0} \sigma_i\right\} \tag{4}$$

3.3. Benefits of Semantic 3D Reconstruction

As mentioned in section 2, there are currently two ways to acquire semantically annotated 3D reconstructions. The first way is to project the 2D labels on the reconstructed volume [26, 25, 23]. In this approach, the semantic information is not used to improve the 3D reconstruction but rather to augment the model using more information. The second way, the "volumetric semantic 3D reconstruction" is to use volumetric techniques [10] not only to represent the geometry, but also the semantic information. Such methods facilitate a much more complete understanding of the scene, e.g. hidden surfaces can be reconstructed. For instance, if a street is reconstructed on which there are many cars, they could be removed, revealing the street underneath the cars which has not been observed (see Fig. 4).



(a) Full scene (b) Without cars

Figure 4. Volumetric semantic reconstruction allows to easily remove objects in a scene, such as the three cars in this example.

Another benefit is that it allows us to introduce different priors on the boundary surfaces between the different semantic labels. Hence, it enables the reconstruction to be completed even if there is not much data support. An example of such a shape prior is a smoothness function that favors building to be connected to ground rather than floating in the air.

3.4. Memory Consumption

The major limitation of the formulation using (1) is its high memory consumption which prevents the reconstruction of scenes with many different semantic labels. This is mostly due to the nature of the indicator variables which imply that a variable is stored for every label at every voxel and moreover a quadratic amount of variables in the number of labels per voxel for the transitions. Methods based on an octree representation [1] provide a first solution for reducing the memory consumption by tackling the scalability in terms of size of the scene. However, they do not deal with the problem of scalability with respect to the number of labels. To give an idea of the problem of memory consumption, imagine a scene with $L + 1$ different semantic labels. Then the required memory consumption to compute a minimizer of energy (1) is linear $\mathcal{O}\left(L\right)$ for the data term and quadratic $\mathcal{O}\left(L^2\right)$ for the regularizer. In the next section we propose an approach for reducing the amount of memory needed by discarding unneeded variables right from the beginning.

4. Dividing the Scene: the Blocks Approach

In this section we introduce a new approach for solving the semantic 3D reconstruction which reduces the memory consumption. We recall that our goal is to reduce memory consumption related to the number of labels, in order to allow for scalability. The main idea of our approach is to subdivide our scene into blocks, with each block allowing a set of relevant labels instead of all labels. It is thus necessary to first initialize these blocks, which we define as determining the sets that are allowed in each block. Since this initialization is not expected to be without errors, we also introduce criteria for updating the blocks by either removing highly unlikely labels or allowing labels that were missing from the initial relevant set. The rest of this section details every part of our framework.

4.1. Energy in the Blocks Framework

As in section 3.1 we focus on the reconstruction of a scene Ω with L different classes of objects. We now divide the scene into N_B disjoint blocks, i.e. $\forall B_i \neq B_j \in \{0, \ldots, N_B - 1\} : B_i \cap B_j = \emptyset$ and $\cup_i B_i = \Omega$. We choose a regular division of the scene. For each block B_k with $k \in \{0, \ldots, N_B - 1\}$ we introduce an indicator function.

$$\begin{aligned} \mathbb{1}_{B_k} \colon \{0, \ldots, L\} &\to \{0, 1\} \\ i &\mapsto \begin{cases} 1 & \text{if } i \in B_k \\ 0 & \text{else} \end{cases} \end{aligned} \tag{5}$$

This indicator function is used to encode whether a label i is activated in block B. We can now rewrite equation (1).

$$\begin{aligned} E = \sum_k \sum_i \mathbb{1}_{B_k}(i) \Bigg[\sum_{s \in B_k} \rho_s^i x_s^i + \\ \sum_{j \colon i<j} \mathbb{1}_{B_k}(j) \left[\phi^{ij} \left(\mathbf{x}_s^{ij} - \mathbf{x}_s^{ji} \right) \right] \Bigg] \end{aligned} \tag{6}$$

where the constraints remain the same.

4.2. Initialization of the Blocks

After defining a division of the scene into blocks we need to initialize the correct block indicator function. We propose to train a classifier for non free space semantic labels in order to detect possible locations of corresponding objects based on evidence from the data cost.

We define a bounding box F for every non free space semantic label. When the box is placed in Ω it is associated with a feature vector f of size $L + 1$ such that for all $i \in \{0, \ldots, L\}$, $f_i = \sum_{s \in F} \rho_s^i$.

Since we are evaluating our framework on the KITTI dataset, we can use a fraction of one of the sequences for training. For every object class we extract a training set (around 15 positive samples and 30 negative samples) and then train boosted classifiers on these sets. The classifiers are trained using the Darwin Library [7] (version 1.9).

At the initialization phase we use a sliding box approach to determine the position of objects belonging to the corresponding class. The box is moved through the scene as a sliding box and at each position a feature vector is extracted. This feature vector is then evaluated with the classifier. If the object is detected, it is activated in the blocks which contain the sliding box.

This initialization step is subject to limitations. There might be misdetections in the initialization phase, i.e. a label could be activated which is not present in the block. Similarly, a semantic label might be not detected in a block in which it should be activated due to occlusion for instance. At initialization it is important that a sufficient number of labels are activated in blocks, in order to avoid getting too far from the original problem. Therefore we train our classifier and tune the parameters in such a way that recall is favored over precision, hence ensuring label sufficiency.

4.3. Updating the Blocks

We want the block to have the ability to make their set of activated labels evolve during the optimization in order to counter the limitations of the initialization and to take into account information from the optimization.

If we tune the parameters during the training step to favor recall over precision, two failure cases may occur. In the first case a class is activated in a block which is not needed, and therefore can be deactivated in the block. In the second case an object of a certain class tries to propagate from one block to another in which it is deactivated. In such a case the class needs to be activated. For instance a building in the scene could be placed across the border between two blocks, but the class *building* was removed from one of these blocks.

Deactivating Classes In the case of a class that was activated by mistake, we expect that after some iterations of the optimization, the evidence that corresponding objects cannot be reconstructed in the block is strong. Consider a block B in which a label i is activated. To evaluate this evidence, we introduce a new indicator function.

$$\mathbb{1}_i \colon \Omega \to \{0, 1\} \tag{7}$$

$$s \mapsto \begin{cases} 1 & \text{if } \arg\max_{j \in \{0, \ldots, L\}} x_s^j = i \\ 0 & \text{else} \end{cases} \tag{8}$$

for all $i \in \{0, \ldots, L\}$. It indicates whether label i is the most likely label to be assigned to voxel i. We can then use a threshold $\tau_{disable}$ such that:

- $\sum_{s \in B} \mathbb{1}_i(s) \leq \tau_{disable}$: label i should be deactivated in block B
- $\sum_{s \in B} \mathbb{1}_i(s) > \tau_{disable}$: label i should not be deactivated in block B

In practice we found that it is better to choose a rather conservative model, with $\tau_{disable} = 0$, such that a label is deactivated from the block only if there is no evidence for a label occurrence in the block.

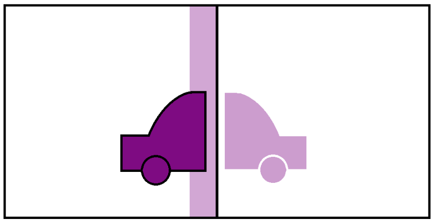


Figure 5. Enabling a new label in a block: a car present in the left block is being reconstructed close to the border. However, there is strong evidence that it should be, as shown by the light purple car. It can be detected by looking at the voxels in the light purple band.

Activating Classes We consider a block B_k in which a label i is deactivated, and one of its neighbors in the six-neighborhood, which we will denote B_{k+1} and in which it is activated. Our assumption is that if an object labeled i is being reconstructed close to the block border, there are good chances that this object might be also reconstructed in B_k. We therefore want to activate it in B_k.

In order to determine if such a situation occurs, we focus on the first layers of voxels, typically 3 or 4 layers of voxels, in B_{k+1} which are next to the boundary between B_k and B_{k+1}, which we denote $\mathcal{D}_{k \leftarrow k+1}$ (see Fig. 5 for an illustration). The reason why we need to consider multiple layers of voxels instead of focusing only on the boundary voxels, is due to the boundary conditions. As explained later in section 4.4, we use Neumann boundary conditions between blocks which do not share some label, for instance label i. The boundary conditions can lead to a global solution which introduces a layer of free space between the object labeled i and the boundary, hence the need to look a little further away. We then introduce a threshold τ_{enable} such that:

- $\sum_{s \in \mathcal{D}_{k \leftarrow k+1}} x_s^i \leq \tau_{enable}$: label i should not be activated in block B_k
- $\sum_{s \in \mathcal{D}_{k \leftarrow k+1}} x_s^i > \tau_{enable}$: label i should be activated in block B_k

The selection of $\tau_{disable}$ depends on the resolution of the scene.

4.4. Optimization

We use the primal-dual algorithm [3] to optimize the energy. Lagrange multipliers are introduced for the constraints and the regularization term is written in the primal-dual saddle-point form (c.f. [10, 28]). The blocks approach does not affect the optimization much. Instead of having primal and dual variables for every label in every voxel, we only have variables for the active labels of the blocks in which the voxel is placed.

The specific case that needs to be discussed is the case when the set of activated labels in a block is updated. In this case, both primal and dual variables are either added or removed, and the optimization problem thus changes. We tackle this situation by reinitializing the problem using the following rules:

	Reserved Memory (Gb)	Gain (%)
Reference	5.16 (16.4%)	0%
Less Transitions	3.91 (12.5%)	24.2%
Blocks	1.825 (5.8%)	64.5%

Table 1. Overview of memory consumption for different approaches showing the benefits of removing classes and transitions.

- All variables corresponding to labels that are already present keep their old values
- Newly added variables are initialized to 0

5. Experiments

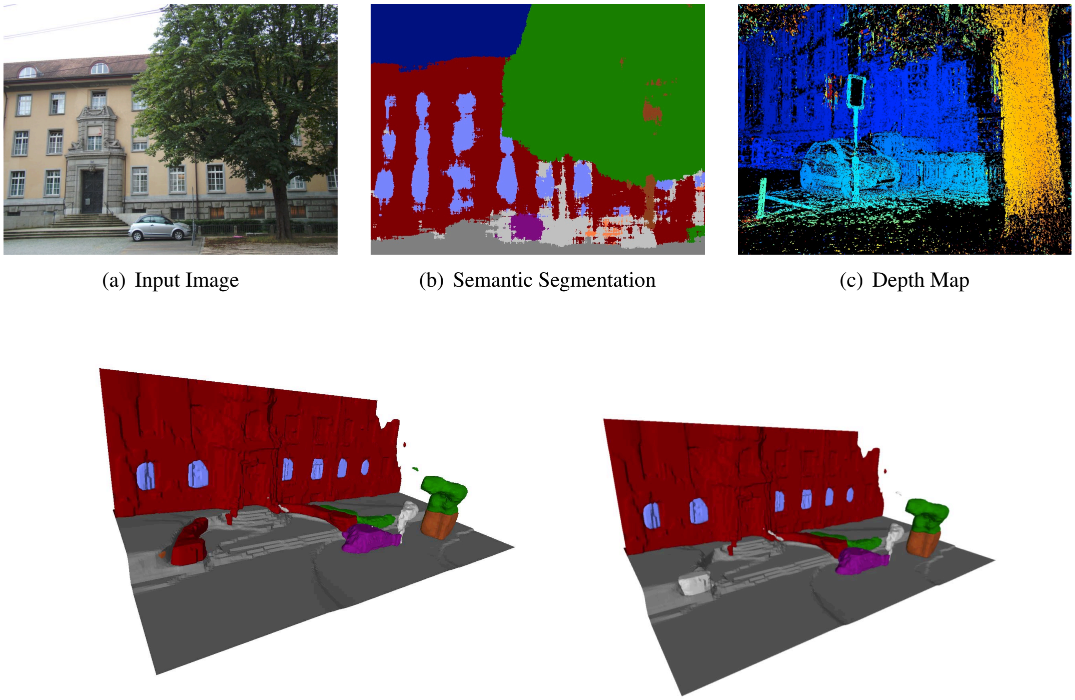
We begin by quantitatively evaluating our approach on a small dataset to illustrate the method and show the gain in memory and speed of convergence. We then show that our block approach can handle more labels than presented in previous joint volumetric reconstruction and semantic segmentation formulations.

5.1. Quantitative Evaluation

The dataset consist of a sequence of 164 images of the facade of a building. We obtained the camera poses using a publicly available structure from motion pipeline [29], and the depth maps are obtained using the publicly available plane sweep stereo matching implementation [9] with zero mean normalized cross correlation and subsequent semi-global matching on the cost volume [11]. The 2D semantic segmentations were computed using the Automatic Labeling Environment [16].

Fig. 6 shows an example of input data for our semantic 3D reconstruction next to the resulting scene. The scene is composed of 9 labels, which are *sky*, *ground*, *building*, *window*, *vegetation*, *tree trunk*, *car*, *clutter*, and *fence*. In order to show the benefits of our method, we compare it to the method in [10], and also to a modified version of the latter in which some transitions are not allowed. We pointed out that memory consumption is in the order of $\mathcal{O}\left(L^2\right)$, where L is the number of semantic labels, due to the necessity to store indicator variables for the transitions between objects of different semantics. A good way to reduce memory consumption is to remove very unlikely transitions such as a transition between *car* and *building*.

Table 1 shows the gain in memory. We use a regular division of the scene into 8 blocks. In comparison to the reference reconstruction of [10], we observe that we reduced the memory consumption by a factor of almost 3. We also



(a) Input Image (b) Semantic Segmentation (c) Depth Map

(d) Baseline Reconstruction (e) Reconstruction with Blocks

Figure 6. Example scene with 9 labels demonstrating that the deactivation of labels in individual blocks leads to similar reconstruction results as for the baseline approach which uses all labels everywhere.

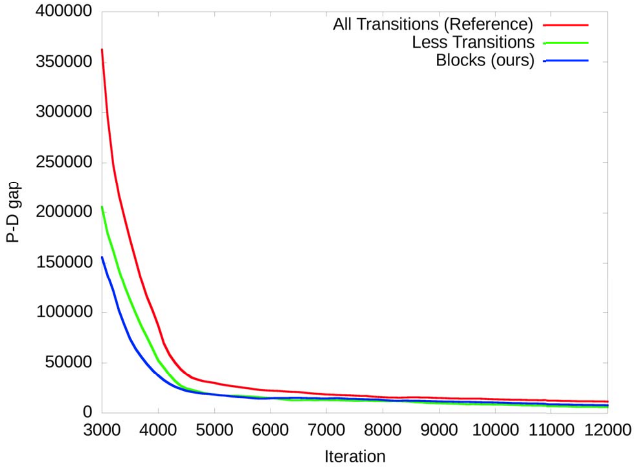


Figure 7. Comparison of the speed of convergence of the semantic 3D reconstruction for the 3 approaches.

observe that our method is more efficient than the basic approach of removing only very unlikely transitions.

We also compared the speed of convergence of the three methods, using the primal-dual gap as a measure of convergence. The results are shown in Fig. 7. We observe that our approach decreases faster at the beginning of the optimization, though final convergence is reached with the same number of iterations. Note that often a good 3D model is obtained before full convergence.

	Average Memory (Gb)	Gain
Reference	30.9	0%
Blocks	10.9	67.6%

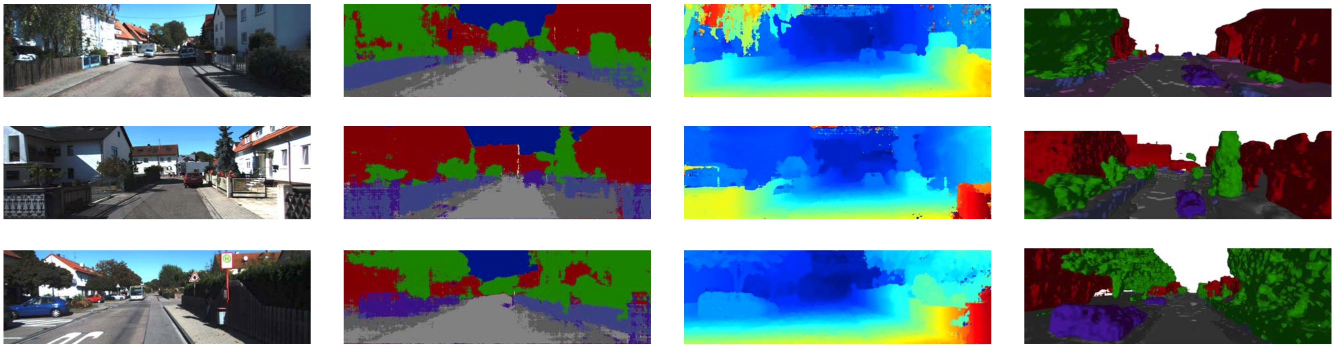
Table 2. Memory consumption of our block approach in comparison to baseline method.

5.2. Results on a Large Dataset

We used our method to apply semantic 3D reconstruction to a sequence of the KITTI dataset [6]. We reconstructed a long sequence of the urban environment by dividing the sequence into smaller stretches that were independently reconstructed and then put together to form the street. The overlap region between the stretches varied between 20% and 25%. Fig. 1 shows a top view of the reconstructed sequence and a close-up on one of the stretches. Fig. 8 shows more close-ups together with the corresponding input data.

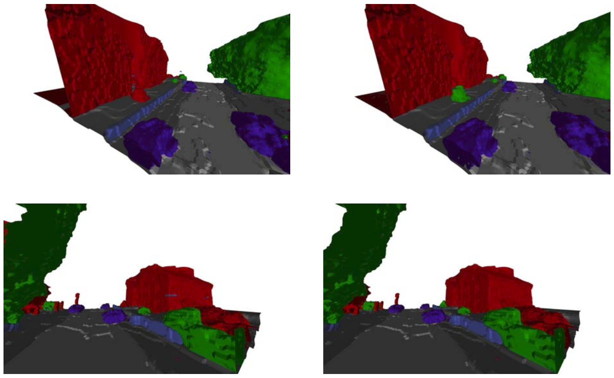
We used stretches of resolution 320 × 240 × 160, for 9 labels. The speed of convergence is comparable with that of [10], however we observed gains in memory consumption as shown in Table 2. We used a division into 10 × 5 × 10 blocks.

We reduced the memory consumption by approximately a third on such stretches. The differences to the model ob-



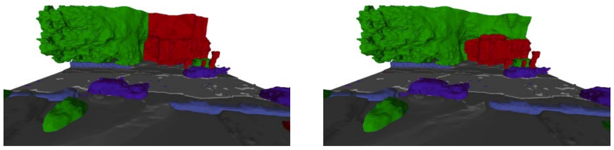
(a) Input Image (b) Semantic Segmentation (c) Depth Map (d) Reconstruction

Figure 8. This figure shows further results and corresponding inputs for the KITTI dataset.



(a) Block model (b) Full model

Figure 9. Comparison of the semantic reconstruction of a portion of a street with and without the blocks framework.



(a) Block model (b) Full model

Figure 10. Comparison of reconstructions obtained with our method (left) and the method of [10] (right): the apparent difference occurs because of a very weak data term.

tained with the method of [10] are minimal as can be seen in Fig. 9. The main difference comes from a small bush that is labeled as building in the blocks approach. This is due to an error in the initialization of the blocks. However, the main structure of the scene is the same, and little loss is observed.

In certain cases we observed apparent differences between our model and the full model as illustrated in Fig. 10. As we can see there is an apparent difference above the building. There is an ambiguity in this region between vegetation and building. The reason is obvious when we look at the input pictures in Fig. 6(a), we see the street from one direction. The top of the building is sometimes hidden by trees, and therefore there is not much observed data. Since there is no strong data observed, the reconstruction mostly stems from regularization. What happens in the full model is that the vegetation propagates in the full model which is not the case for the block model. In the block model the vegetation label is deactivated on top of the house, thus favoring propagation of the building over vegetation.

6. Conclusion

We introduced a framework for reducing the memory consumption of semantic 3D reconstruction methods. Our approach allows for better scalability in the number of labels. By taking advantage of the fact that all semantic labels are not needed everywhere in the scene, we propose to divide the scene into blocks, in which only relevant labels are active. The set of active labels in each block is initialized and can evolve during the optimization of the 3D model. We further showed that our method yields a significant gain in memory compared to other volumetric methods [10], hence allowing for better scalability in the number of the labels.

While we increased the number of labels already in this work we plan to further increase the number of classes by applying the method to scenes which yield a larger diversity of semantic classes. The method is currently dependent on initialization and update of the set of labels in every blocks. Better initialization should lead to less updates and hence faster convergence for the models.

Another line of research is to combine our block approach with the method of [1] to allow for spatially larger scenes with many labels. Using a data term defined as ray potential could further improve the quality of the reconstructions [22].

Acknowledgement We gratefully acknowledge partial funding by the Swiss National Science Foundation project no. 157101 on 3D image understanding for urban scenes.

References

[1] M. Bláha, C. Vogel, A. Richard, J. D. Wegner, T. Pock, and K. Schindler. Large-Scale Semantic 3d Reconstruction: an Adaptive Multi-Resolution Model for Multi-Class Volumetric Labeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 2, 3, 4, 8

[2] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla. Segmentation and recognition using structure from motion point clouds. In *Proceedings of the European Conference on Computer Vision*, pages 44–57. Springer, 2008. 2

[3] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 40(1):120–145, 2011. 6

[4] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *Proceedings of the 23rd annual conference on Computer graphics and interactive techniques*, pages 303–312. ACM, 1996. 2

[5] S. Esedolu and S. J. Osher. Decomposition of images by the anisotropic Rudin-Osher-Fatemi model. *Communications on pure and applied mathematics*, 57(12):1609–1626, 2004. 3

[6] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 3354–3361. IEEE, 2012. 2, 7

[7] S. Gould. DARWIN: A framework for machine learning and computer vision research and development. *Journal of Machine Learning Research*, 13(Dec):3533–3537, 2012. 5

[8] F. Güney and A. Geiger. Displets: Resolving stereo ambiguities using object knowledge. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4165–4175, 2015. 2

[9] C. Häne, L. Heng, G. H. Lee, A. Sizov, and M. Pollefeys. Real-time direct dense matching on fisheye images using plane-sweeping stereo. In *2014 2nd International Conference on 3D Vision*, volume 1, pages 57–64. IEEE, 2014. 6

[10] C. Häne, C. Zach, A. Cohen, R. Angst, and M. Pollefeys. Joint 3d Scene Reconstruction and Class Segmentation. In *2013 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 97–104, June 2013. 1, 2, 3, 4, 6, 7, 8

[11] H. Hirschmüller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 2 - Volume 02*, CVPR '05, pages 807–814, Washington, DC, USA, 2005. IEEE Computer Society. 6

[12] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Proceedings of the Fourth Eurographics Symposium on Geometry Processing*, SGP '06, pages 61–70, Aire-la-Ville, Switzerland, Switzerland, 2006. Eurographics Association. 2

[13] B.-S. Kim, P. Kohli, and S. Savarese. 3d scene understanding by Voxel-CRF. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1425–1432, 2013. 2

[14] A. Kundu, Y. Li, F. Dellaert, F. Li, and J. M. Rehg. Joint semantic segmentation and 3d reconstruction from monocular video. In *Proceedings of the European Conference on Computer Vision*, pages 703–718. Springer, 2014. 2

[15] P. Labatut, J.-P. Pons, and R. Keriven. Efficient multi-view reconstruction of large-scale scenes using interest points, delaunay triangulation and graph cuts. In *2007 IEEE 11th international conference on computer vision*, pages 1–8. IEEE, 2007. 2

[16] L. Ladický. *Global Structured Models towards Scene Understanding*. PhD thesis, Oxford Brookes University, Apr. 2011. 6

[17] L. Ladický, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. H. S. Torr. Joint Optimization for Object Class Segmentation and Dense Stereo Reconstruction. *International Journal of Computer Vision*, 100(2):122–133, Nov. 2012. 2

[18] S. Laine and T. Karras. Efficient sparse voxel octrees. *IEEE Transactions on Visualization and Computer Graphics*, 17(8):1048–1059, 2011. 2

[19] J. Lellmann, B. Lellmann, F. Widmann, and C. Schnörr. Discrete and continuous models for partitioning problems. *International journal of computer vision (IJCV)*, 104(3):241–269, 2013. 2

[20] V. Lempitsky and Y. Boykov. Global Optimization for Shape Fitting. In *2007 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8, June 2007. 2

[21] M. Nießner, M. Zollhfer, S. Izadi, and M. Stamminger. Real-time 3d reconstruction at scale using voxel hashing. *ACM Transactions on Graphics*, 32(6):1–11, Nov. 2013. 2

[22] N. Savinov, C. Häne, L. Ladický, and M. Pollefeys. Semantic 3d Reconstruction with Continuous Regularization and Ray Potentials Using a Visibility Consistency Constraint. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2016. 8

[23] S. Sengupta, E. Greveson, A. Shahrokni, and P. H. Torr. Urban 3d semantic modelling using stereo vision. In *Robotics and Automation (ICRA), 2013 IEEE International Conference on*, pages 580–585. IEEE, 2013. 2, 4

[24] M. Tanner, P. Pinies, L. M. Paz, and P. Newman. DENSER Cities: A System for Dense Efficient Reconstructions of Cities. *arXiv preprint arXiv:1604.03734*, 2016. 2

[25] J. Valentin, S. Sengupta, J. Warrell, A. Shahrokni, and P. Torr. Mesh based semantic modelling for indoor and outdoor scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2067–2074, 2013. 2, 4

[26] V. Vineet, O. Miksik, M. Lidegaard, M. Nießner, S. Golodetz, V. A. Prisacariu, O. Kahler, D. W. Murray, S. Izadi, P. Peerez, and P. H. S. Torr. Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. In *Robotics and Automation (ICRA), 2015 IEEE International Conference on*, pages 75–82. IEEE, 2015. 2, 4

[27] C. Zach. Fast and high quality fusion of depth maps. In *Proceedings of the international symposium on 3D data processing, visualization and transmission (3DPVT)*, volume 1, page 2. Citeseer, 2008. 2

[28] C. Zach, C. Häne, and M. Pollefeys. What is optimized in convex relaxations for multilabel problems: Connecting discrete and continuously inspired map inference. *IEEE transactions on pattern analysis and machine intelligence*, 36(1):157–170, 2014. 2, 3, 6

[29] C. Zach, M. Klopschitz, and M. Pollefeys. Disambiguating visual relations using loop constraints. In *Computer Vision and Pattern Recognition (CVPR), 2010 IEEE Conference on*, pages 1426–1433, June 2010. 6

[30] C. Zhou, F. Güney, Y. Wang, and A. Geiger. Exploiting Object Similarity in 3d Reconstruction. In *2015 IEEE International Conference on Computer Vision (ICCV)*, pages 2201–2209, Dec. 2015. 2