Article

Fast 3D Semantic Mapping on Naturalistic Road Scenes

Xuanpeng Li [1,*], Dong Wang[1], Huanxuan Ao[1], Rachid Belaroussi[2] and Dominique Gruyer[2]

[1] School of Instrument Science and Engineering, Southeast University, 210006 Nanjing, Jiangsu, China
[2] COSYS/LIVIC, IFSTTAR, 25 allée des Marronniers, 78000 Versailles
* Correspondence: li_xuanpeng@seu.edu.cn

† This paper is an extended version of our paper published in LI, Xuanpeng, et al. Fast semi-dense 3D semantic mapping with monocular visual SLAM. In: Intelligent Transportation Systems (ITSC), 2017 IEEE 20th International Conference on. IEEE, 2017. S. 385-390.

Abstract: Fast 3D reconstruction with semantic information on road scenes is of great requirements for autonomous navigation. It involves issues of geometry and appearance in the field of computer vision. In this work, we propose a method of fast 3D semantic mapping based on the monocular vision. At present, due to the inexpensive price and easy installation, monocular cameras are widely equipped on recent vehicles for the advanced driver assistance and it is possible to acquire semantic information and 3D map. The monocular visual sequence is used to estimate the camera pose, calculate the depth, predict the semantic segmentation, and finally realize the 3D semantic mapping by combination of the techniques of localization, mapping and scene parsing. Our method recovers the 3D semantic mapping by incrementally transferring 2D semantic information to 3D point cloud. And a global optimization is explored to improve the accuracy of the semantic mapping in light of the spatial consistency. In our framework, there is no need to make semantic inference on each frame of the sequence, since the mesh data with semantic information is corresponding to sparse reference frames. It saves amounts of the computational cost and allows our mapping system to perform online. We evaluate the system on naturalistic road scenes, e.g., KITTI and observe a significant speed-up in the inference stage by labeling on the mesh.

Keywords: 3D semantic mapping; incremental fusion; global optimization; real time; naturalistic road scenes

1. Introduction

Naturalistic scene understanding plays a key background role in most vision-based mobile robots. For example, autonomous navigation in outdoor scenes asks for a rapid and comprehensive understanding of surroundings for obstacle avoidance and path planning. Vehicle movement in limited temporal and spatial contexts always requires knowledge of what something is, where it is located, and ego-vehicle's surrounding. Robotic maps, such as Occupancy grid map and OctoMap, traditionally provide geometric presentation of the environment. However, they lack the correlation in data between map points and semantic knowledge; thus, they could not be directly utilized in naturalistic road scenes.

Scene parsing is an important and promising step to address this issue. It benefits from the state-of-the-art Deep Convolutional Neural Networks (DCNNs) which contributes to better performance of 2D pixel labeling than traditional methods. Then, combined with the Simultaneous Localization and Mapping (SLAM) technology, automobile could locate itself and meanwhile recognize surrounding objects in pixel-wise level. For instance, it could make autonomous vehicle accomplish certain high-level tasks, such as "parking on the right free place" and "stopping at the crosswalk". This form of semantically annotated 3D representation provides mobile robots with functions of understanding, interaction and navigation in various scenes.

Semantic segmentation has been an active topic for a long time. Most methods have focused on increasing the accuracy of the semantic segmentation, and have seen major improvements [1–3]. However, they usually asks for high-power computing resources, which is not suitable for the embedded platform. Several recent research focuses on the balance between the computing cost and the accuracy of object detection, classification and 2D pixel labeling [4,5]. They achieves a better performance with regards to the embedded and mobile platforms.

Peer-reviewed version available at *Appl. Sci.* **2019**, *9*, 631; doi:10.3390/app9040631

Compared to the SLAM technology with scaled sensors, such as stereo and RGB-D cameras, monocular visual SLAM is a promising technology, because monocular vision is flexible, inexpensive, and most importantly, widely equipped on most recent vehicles. Scaled sensors could provide reliable measurement in their specific ranges, whereas they lack the capability of seamless switch between various-scale scenes such as indoor and outdoor. And they normally need large storage resources.

Most man-made environments, e.g., road scenes, usually exhibit distinctive spatial relations among varied classes of objects. Being able to capture, model and utilize these kinds of relations could enhance semantic segmentation performance in the 3D semantic mapping [6]. In this paper, we exploit a monocular SLAM method that provides cues of 3D spatial information and utilize state-of-the-art DCNN to build a 3D scene understanding system towards road scenes. Moreover, a Bayesian 2D-3D transfer and a map regularization process are exploited to generate a consistent reconstruction in the spatial and semantic context.

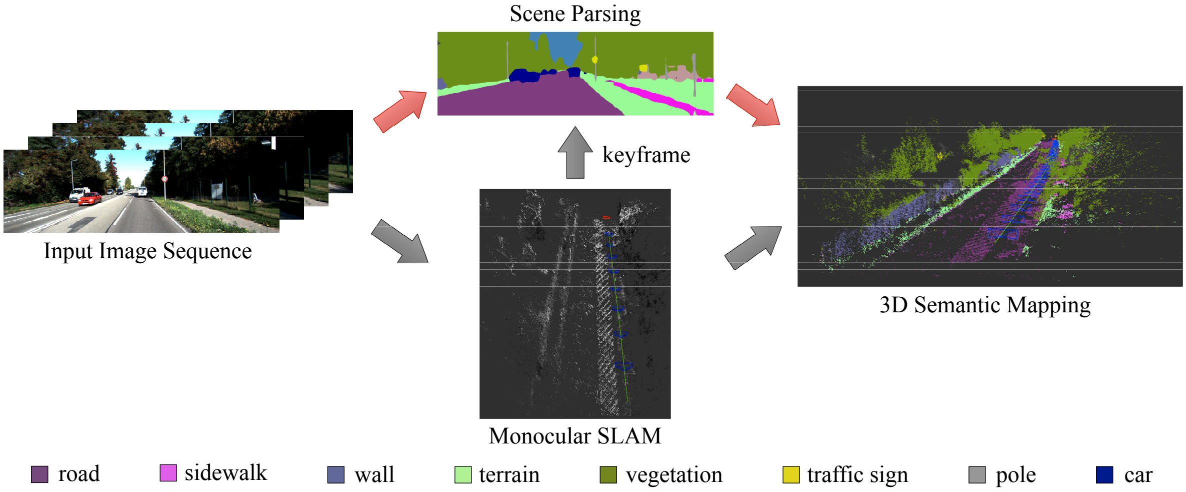


Figure 1. Overview of our system: From monocular image sequence, keyframes are selected to obtain its 2D semantic information, which then transfer to the 3D reconstruction to build the 3D semantic map.

In our monocular mapping system, the 3D map is incrementally reconstructed with a sequence of automatically selected keyframes and corresponding semantic information. There is no need to label each frame in a sequence, which could save a considerable amount of computation cost. We refer the reader to Figure 1 for an illustration. Different from the frame skipping strategy proposed by Hermans *et al.* [7] and McCormac *et al.* [8], our method could work well under fast camera motions. Since the 3D map should have global consistent depth information, it could be regularized in term of spatial structures. The regularization is aimed to remove distinctive outliers and makes components more consistent in the point cloud map, i.e., local points with same semantic label should be approached in 3D space. Two datasets, Cityscapes [9] and KITTI [10], are used to evaluate our approach. Several raw videos are taken to reconstruct 3D map with semantic labels.

This paper is presented as follows. In the following Section 2, a review of the related work is given. The problem formulation is presented in Section 3. The 3D semantic mapping is described in Section 4, including the semantic segmentation, the monocular visual SLAM, the Bayesian incremental fusion and the global regularization. Section 5 includes the results of 2D semantic inference and 3D semantic mapping. Finally, Section 6 concludes the paper and discusses possible extensions of our work.

2. Related Work

Our work is motivated by [8] which contributes an indoor 3D semantic SLAM from the RGB-D input. It aims towards a dense 3D map based on ElasticFusion SLAM [11] with semantic labeling. Pixel-wise semantic information is acquired from a Deconvolutional semantic segmentation network [12] using the scaled RGB information and the depth as the input. Depth information is also used to update surfel's depth and normal information to construct 3D dense map during loop closure. In addition, a previous work, SLAM++ [13], creates a map with semantically defined objects, but it is limited to predefined database and hand-crafted template models.

In this paper, we make use of an incremental Bayesian fusion strategy with state-of-the-art visual SLAM and semantic segmentation.

Visual SLAM usually contains sparse, semi-dense, and dense types depending on the methods of image alignment. Feature-based methods only exploited limited feature points - typically image corners and blobs or line segments, such as classic MonoSLAM [14] and ORB-SLAM [15,16]. They are not suitable for 3D semantic mapping due to rather sparse feature points. In order to better exploit image information and avoid the cost on calculation of features, direct dense SLAM system, such as the surfel-based dense slam, ElasticFusion [11] and Dense Visual SLAM [17], have been proposed recently. Whereas, direct image alignment from these dense methods is well-established for monocular, RGB-D and stereo sensors. Semi-dense methods like Large-Scale Direct-SLAM (LSD-SLAM) [18] and Semi-direct Visual Odometry (SVO) [19] provide possibility to build a synchronized 3D semantic mapping system.

Deep CNNs have proven to be effective in the field of image semantic segmentation. Long *et al.* [20] firstly introduces an inverse convolution layer to realize an end-to-end training and inference. Then, the encoder-decoder architectures with specified upsampling layers, such as max unpooling and deconvolutional layer, are proposed to avoid the problem of separate step training in the FCN network and improve the accuracy [12,21]. Zhao *et al.* [2] exploits the capability of global context information through embedding various scenery context feature in a pyramid structure. The fusion of varied scaled feature has been a popular strategy in the recent deep CNNs. The cutting-edge method, namely, DeepLab series [1,3,5], combines atrous convolutions and atrous spatial pyramid pooling (ASPP) to achieve a state-of-the-art performance on semantic segmentation. The early DeepLab models have a reasonable accuracy but require much computation overhead. Recently proposed efficient convolution neural network, such as MobileNets [22,23] boosts real-time performance of semantic segmentation without losing the accuracy too much. The state-of-the-art DeepLab-v3+ [5] contains a simple effective decoder module to refine the segmentation results especially along object boundaries. Furthermore, combining the encoder part of MobileNet-v2 in its encoder-decoder structure, DeepLab-v3+ could achieve a better trade-off between precision and runtime.

In the topic of scene understanding and mapping, recent research employ 3D priors of objects increasingly. Salas-Moreno *et al.* [13] project 3D mesh of objects to the RGB-D frame in a graphical SLAM framework. Valentin *et al.* [24] propose a triangulated meshed representation of the scene from multiple depth measurements and exploit the Conditional Random Field (CRF) to capture the consistency of 3D object mesh. Kundu *et al.* [25] exploit the CRF for joint voxels to infer the semantic information and occupancy. Sengupta and Sturgess [26] use stereo camera, estimated pose and CRF to infer the semantic octree presentation of the 3D scene. Vineet *et al.* [27] propose an incremental dense stereo reconstruction and semantic fusion technique to handle dynamic objects in the large-scale outdoor scenes. Kochanov *et al.* [28] employ scene flow measurements to incorporate temporal updates into the mapping of dynamic environment. Landrieu *et al.* [29] introduce a regularization framework to obtain spatially smooth semantic labeling of 3D point clouds from a point-wise classification, considering the uncertainty associated with each label. Gaussian Process (GP) is another popular method for map inference. Jadidi *et al.* [30] exploit GP to learn the structural and semantic correlation between map points. This technique also incorporates OcotoMap to handle sparse measurements and missing labels. In order to improve the training and query time complexities of the GP-based semantic mapping, Gan *et al.* [31] further introduce a Relevance Vector Machine (RVM) inference technique for efficient map query at any resolution.

Our semi-dense approach is also inspired by dense 3D semantic mapping methods [6,7,32,33] in both indoor and outdoor scenes. The major contributions from these work involve the 2D-3D transfer and the map regularization. Especially, Hermans *et al.* [7] propose an efficient 3D CRF to regularize 3D semantic mapping consistently considering influence between neighbors of 3D points (voxels). In this work, we adopt a similar strategy to improve the performance of the 3D semantic reconstruction in the road scenes. The key concepts are

- a 3D semantic mapping system based on the monocular vision,
- integration of monocular SLAM and scene parsing into 3D semantic representation,
- exploiting the correlation between semantic information and geometrical information to enforce spatial consistency,

- active sequence downsampling and sparse semantic segmentation so that to achieve a real-time performance and reduce the storage.

Following the comparison in [27], we list the characteristics of our approach and some relative work in TABLE 1.

Table 1. Comparison with some related work: M = monocular camera, S/D = stereo/depth camera, L = Lidar, O = outdoor, I = incremental, SDT = sparse data structures, RT = real time

Method	M	S/D	L	O	C	I	SDT	RT
Hu *et al.* [34]			√	√		√	√	√
Sengupta *et al.* [32]		√		√	√			
Hermans *et al.* [7]		√				√	√	
Kundu *et al.* [25]	√			√	√	√	√	
Vineet *et al.* [27]		√		√	√	√	√	√
Wolf *et al.* [6]		√		√		√	√	√
McCormac *et al.* [8]		√		√	√	√	√	√
Ours	√			√	√	√	√	√

3. Problem Formulation

3.1. Notation

The target is to estimate the 3D semantic map $\mathcal{M}$ comprising of a pose-graph of keyframes with semantic map taken from a monocular camera. Let $I_i : \Omega \to \mathbb{R}^3$ symbolize an H × W RGB image at the frame indexed by i. Keyframes are extracted from image sequence in light of camera's pose $\mathbf{T}_i^j$ at the i frame with respect to previous keyframe j. We define the ith keyframe to be a tuple $\mathcal{K}_i = (I_i, D_i, V_i, S_i)$, where $D_i : \Omega_{D_i} \to \mathbb{R}$ is the full-resolution inverse depth map associated with image I_i, and $V_i : \Omega_{V_i} \to \mathbb{R}$ is associated inverse depth variance map. Depth map and variance are defined in the subset of pixels as $\Omega_{D_i} \subset \Omega_i$, which means semi-dense, only available for certain image regions of large intensity gradient. The symbol $S_i : \Omega_{S_i} \to \mathbb{R}$ represents the full-resolution semantic map with maximum probability of object class from the semantic segmentation process.

The keyframes are consecutively stacked in a pose-graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where $\mathcal{V} = \{\mathcal{K}_0, \ldots, \mathcal{K}_n\}$ is the set of keyframes and $\mathcal{E} = \{\mathbf{S}_i^j \in \mathrm{Sim}(3) : \mathcal{K}_i, \mathcal{K}_j \in \mathcal{V}\}$ is the set of constraint factors. Each $\mathbf{S}_i^j = (\mathbf{T}_i^j, s_i^j)$ consists of a camera's pose $\mathbf{T}_i^j = \left(\begin{smallmatrix} R & t \\ 0 & 1 \end{smallmatrix}\right)$ from keyframe i to keyframe j, and scale factor $s_i^j > 0$. In reference to world frame W, normally regarded as the first keyframe $\mathcal{K}_0$, the pose of the keyframe indexed by i is denoted as $\mathbf{T}_W^i$. For a sequence of keyframes (n keyframes), we get the nth keyframe's pose $\mathbf{T}_W^n = \prod_1^n \mathbf{T}_{k-1}^k$.

The 3D map $\mathcal{M}$ is reconstructed by the projection of the inverse depth map of all keyframes, where each 3D point $\mathbf{P}$ can be labeled as one of the solid semantic objects in the label space $\mathcal{L} = \{l_1, l_2, \ldots, l_k\}$ like *Road*, *Building*, *Tree*, etc. We use $\mathbf{X} = \{X_1, X_2, \ldots, X_M\}$ to denote the set of random variables corresponding to the 3D points $\mathbf{P}_i : i \in \{1, \ldots, M\}$, where each variable $X_i \in \mathbf{X}$ take a value l_i from the predefined label space $\mathcal{L}$.

3.2. 3D semantic mapping

Our target is to build a 3D semantic map with semi-dense and consistent label information online while the image sequences are captured by a moving monocular forward camera. Given an image sequence, the inference of the 3D semantic map is regarded as:

$$\mathcal{M}^* = \mathrm{argmax}_{\mathcal{M}} P(\mathcal{M}|\mathcal{G}), \tag{1}$$

which can be estimated by the maximum a-posterior (MAP). Compared to the model used in [25], our observation is continuously updating, not all existing measurements. Thus, we adopt an incremental fusion strategy to estimate the 3D semantic map by incorporating new arriving keyframes. Correspondingly, the approach is decoupled into three separately running processes as shown in Figure 2.

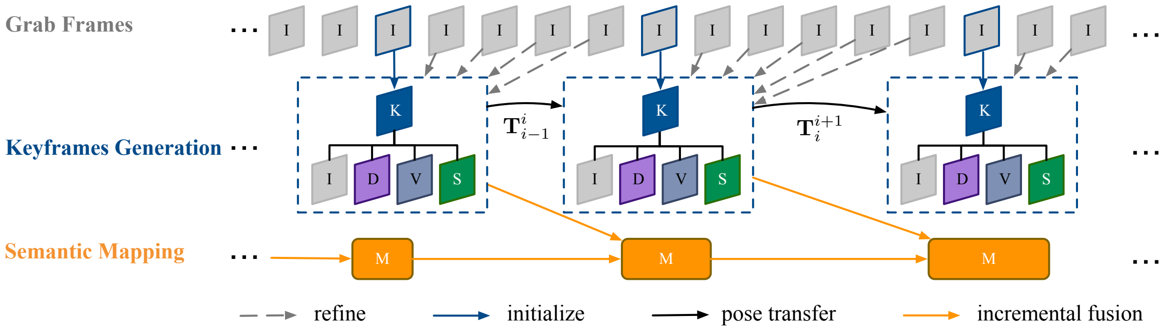


Figure 2. Framework of our method: The input is the sequence of the RGB frames, denoted as I. There are three separate processes, a keyframe selection process, a 2D semantic segmentation process , and a 3D reconstruction with semantic optimization process. Keyframes K are conditionally extracted from the sequence based on the distance between the poses. The following frames refine the depth map and the variance map of each keyframe until new keyframe is extracted. The 2D semantic segmentation module predicts the pixel-level class of the new-arriving keyframe. Finally, the keyframes are incrementally explored to reconstruct the 3D map with semantic labeling and then it is regularized by a dense CRF.

In the system, the monocular SLAM process maintains and tracks on a global map of the environment, which contains a number of keyframes connected by pose-pose constraints with associated probabilistic semi-dense depth maps. It runs in real-time on a CPU. Represented as point clouds, the map gives a semi-dense and highly accurate 3D reconstruction of the environment. Meanwhile, the second process of the 2D semantic segmentation generates the pixel-level classification on the extracted keyframes. A fast deep CNN model is explored to predict the semantic information on a GPU. In addition, an incremental fusion process for the semantic label optimization is operated in a parallel way. It builds a local optimal correspondence between semantic labeling and voxels in the 3D point cloud. To obtain a globally optimal 3D semantic segmentation, we attempt to make use of information of neighboring 3D points, involving the distance, color similarity and semantic label. It updates voxel's position and corresponding semantic label, which gives a globally consistently 3D semantic map.

4. 3D Semantic Mapping

4.1. 2D Scene Parsing

We explore the DeepLab-v3+ deep neural network proposed by Chen *et al.* [5]. Two important components in the DeepLab series are the atrous convolution and atrous spatial pyramid pooling (ASPP), which enlarge the field of view of filters and explicitly combine the feature maps at multiple scales. The improvement in the DeepLab-v3+ involves the encoder-decoder structure and the augmentation of ASPP module with image-level feature. The former is able to capture sharper object boundaries by regaining the spatial information, while the latter encodes multi-scale contextual information to capture long range information. These contributions make DeepLab successfully handle both large and small objects and achieve a better trade-off between precision and run-time.

For the semantic segmentation of road scenes, we exploit the Cityscapes dataset and the KITTI dataset and adopt the predefined 19-class label space $\mathcal{L} = \{l_1, l_2, \ldots, l_{19}\}$, which contains *Road*, *Sidewalk*, *Building*, *Wall*, and so on. We use all semantic annotated images in the Cityscapes dataset for training and fine-tune the model with the KITTI dataset.

Note that there is not any depth information involved in the training process. In the inference, we keep the original resolution of input image according to different datasets.

4.2. Semi-Dense SLAM

We explore LSD-SLAM to track camera's trajectory and build consistent, large-scale maps of the environment. LSD-SLAM is a real-time, semi-dense 3D mapping method. It has several advantages: firstly, it is a scale-aware image alignment algorithm to directly estimate the similarity transform between two keyframes against different scale environments, such as office rooms (indoor) and urban roads (outdoor). The second one is that it is a probabilistic approach to incorporate noise on the estimated depth maps into the tracking based on the propagation of uncertainty. Moreover, it could integrate easily with different kinds of sensors like monocular, stereo and panoramic cameras for various applications. These features are of benefit to a reliable tacks and maps even in challenging surroundings.

LSD-SLAM has three major components: tracking, depth map estimation and map optimization. Spatial regularization and outlier removal are incorporated in the estimation of depth map with small-baseline stereo comparisons. In addition, a direct, scale-drift aware image alignment is carried on these existing keyframes to detect scale-drift and loop closures. Due to the inherent correlation between the depth map and the tracking accuracy, depth residual is used to estimate the similarity transform $\mathfrak{sim}(3)$ constraints between keyframes. Consequently, a 3D point cloud map is built based on a set of keyframes with the estimated depth maps via minimizing the error of image alignment. The map is continuously optimized in the background using a *g2o* pose-graph optimization. The approach runs in 25Hz on an Intel i7 CPU. More details like keyframe selection and depth estimation should be referred to the work [18].

4.3. Incremental Fusion

There might be a large amount of inconsistent 2D semantic labels between consecutive frames, due to the noise of sensors, the complexity of environments in the real world and the failure of scene parsing model. Incremental fusion of semantic label from the stacked keyframes allows associating probabilistic label in a Bayesian way, when combining with the depth map propagation between keyframes in the LSD-SLAM. We will give the details about the incremental semantic fusion with the depth estimation as follows.

The camera projection transformation function $\pi(\cdot) : \mathbb{R}^3 \to \mathbb{R}^2$ is defined as

$$\mathbf{p} = \pi(\mathbf{P}) = [\alpha\frac{x}{z} + c_x, \beta\frac{y}{z} + c_y]^T, \tag{2}$$

which maps a point $\mathbf{P} = [x, y, z]^T$ in 3D space into a 2D point $\mathbf{p} = [x', y']^T$ on the digital image plane I_i in the camera coordinate system. Since this projection function is nonlinear, for the computation efficiency, the transformation should be augmented into the homogeneous coordinate system, which is defined as

$$\mathbf{p}_h = \begin{bmatrix} x'_h \\ y'_h \\ z'_h \end{bmatrix} = \begin{bmatrix} \alpha & 0 & c_x & 0 \\ 0 & \beta & c_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \mathbf{K}[\mathbf{I}\ 0]\mathbf{P}_h, \tag{3}$$

where the matrix $\mathbf{K}$ is referred to as the camera matrix. Given a 3D point $\mathbf{P}_W$ in the world reference system, the mapping to image plane I_i in the homogeneous reference system is calculated as

$$\mathbf{p}_h = \mathbf{K}\mathbf{T}^i_W\mathbf{P}_{Wh}, \tag{4}$$

where $\mathbf{T}^i_W$ the pose of the camera in the world reference system. Then, we get Euclidean coordinates $\mathbf{p} = [x'_h/z'_h, y'_h/z'_h]^T$ from the homogeneous coordinates. From this point on, any point $\mathbf{p}$ and $\mathbf{P}$ is assumed to be in homogeneous coordinates and thus we drop the h index, unless stated otherwise.

Correspondingly, given the inverse depth estimation $\hat{d}$ for a pixel $\mathbf{p} = [x', y']^T$ in I_i of the keyframe $\mathcal{K}_i$, we also have an inverse projection function below:

$$\mathbf{P} = \pi^{-1}(\mathbf{p}, \hat{d}) = [\frac{x'/\hat{d} - c_x/\hat{d}}{\alpha}, \frac{y'/\hat{d} - c_y/\hat{d}}{\beta}, \frac{1}{\hat{d}}]^T, \tag{5}$$

where $\hat{d} = D_i(\mathbf{p})$ corresponds to the point $\mathbf{p}$ existing in the depth map D_i, which projects the 2D pixel point into the 3D point in the current camera coordinate system. The inverse depth estimation of each existing keyframe is continuously refined using its following frames until new keyframe is defined. In reference to Equation 4 and 5, we can derive the 3D point in the world reference system as follows:

$$\mathbf{P}_W = {T_W^i}^{-1} \pi^{-1}(\mathbf{p}, D_i(\mathbf{p})), \tag{6}$$

where the homogeneous transformation matrix has the property: ${T_W^j}^{-1} = T_j^W$.

Once a new frame is chosen to become a keyframe $\mathcal{K}_j$, its depth map D_j is initialized by projecting points from previous keyframe into it. The information of existing, close-by keyframes is propagated to new keyframe for its initialization and semantic probabilistic refinement. The point in the depth map of new keyframe is obtained by

$$\mathbf{p} = \mathbf{K}\mathbf{T}_W^i \mathbf{T}_i^j \mathbf{P}_W \in I_j. \tag{7}$$

Here, we have a Gaussian distributed transformation between keyframes, regarded as $\mathbf{p} \in I_i \rightarrow \mathbf{P}_W \rightarrow \mathbf{p} \in I_j$.

The class label corresponding to a 3D point $\mathbf{P}$ is denoted as $X : \mathbf{P} \rightarrow l \in \mathcal{L}$. Note that the label *Sky* is removed from $\mathcal{L}$ for the 3D semantic mapping. Our target is to obtain the independent probability distribution of each 3D point over the class labels $P(X|\mathcal{K}_0^i)$ given a sequence of existing keyframes $\mathcal{K}_0^i = \{\mathcal{K}_0, \mathcal{K}_1, \ldots, \mathcal{K}_i\}$ in the pose-graph $\mathcal{G}$.

We explore a recursive Bayesian fusion to refine the corresponding probability distribution of 3D points with new keyframe's update:

$$P(X|\mathcal{K}_0^i) = \frac{1}{Z_i} P(\mathcal{K}_i|\mathcal{K}_0^{i-1}, X) P(X|\mathcal{K}_0^{i-1}), \tag{8}$$

with $Z_i = P(\mathcal{K}_i|\mathcal{K}_0^{i-1})$. Applying the first-order Markov assumption to $p(\mathcal{K}_i|\mathcal{K}_0^{i-1}, X)$, then we have:

$$P(X|\mathcal{K}_0^i) = \frac{1}{Z_i} P(\mathcal{K}_i|X) P(X|\mathcal{K}_0^{i-1}) = \frac{1}{Z_i} \frac{p(\mathcal{K}_i) P(X|\mathcal{K}_i)}{P(X)} P(X|\mathcal{K}_0^{i-1}). \tag{9}$$

We assume that $P(X)$ does not change over time and there is no need to calculate the normalization factor $P(\mathcal{K}_i)/Z_i$ explicitly.

According to the formulations above, the semantic probability distribution of all given keyframes can be recursively updated as follows:

$$P(X|\mathcal{K}_0^i) \propto P(X|\mathcal{K}_i) P(X|\mathcal{K}_0^{i-1}). \tag{10}$$

The incremental fusion can refine the semantic label of the points in the 3D space based on the pose-graph of keyframes. It could handle the inconsistent 2D semantic labels, even though its performance relies on the depth estimation. In addition, map geometry is another useful feature which could improve the performance of the 3D semantic mapping further. The following section describes how we use the dense CRF to regularize the 3D semantic map by exploring the map geometry, which could propagate semantic information between spatial neighbors.

4.4. Map Regularization

The dense CRF is widely used in the 2D semantic segmentation to enhance the performance of semantic segmentation. Some previous works [6,7,32] seek its application on the 3D map to model contextual relations between various class labels in a fully connected graph. It is a heuristic approach that assume the influence between neighbors should be proportional to their distance, visual and geometrical similarity [7].

The CRF model is defined as a graph composed of unary potentials as nodes and pairwise potentials as edges, but the size of the model makes traditional inference algorithms impractical. Thanks to Krahenbuhl and Koltun's work [35], a highly efficient approximate inference algorithm is proposed to handle this issue by defining the pairwise edge potentials as a linear combination of Gaussian kernels. We apply the efficient inference of the dense CRF to maximize label agreement between similar 3D points as follows.

Assume the 3D semantic map $\mathcal{M}$ containing M 3D points is defined as a random field. A CRF $(\mathcal{M}, \mathbf{X})$ is characterized by a Gibbs distribution as follows:

$$P(\mathbf{X}|\mathcal{M}) = \frac{1}{Z(\mathcal{M})} \exp(-E(\mathbf{X}|\mathcal{M})), \tag{11}$$

where $E(\mathbf{X}|\mathcal{M})$ is the Gibbs energy and $Z(\mathcal{M})$ is the partition function. The maximum a posteriori (MAP) labeling of the random field is

$$\mathbf{X}^* = \text{argmax}_{l \in \mathcal{L}} P(\mathbf{X}|\mathcal{M}) = \text{argmin}_{l \in \mathcal{L}} E(\mathbf{X}|\mathcal{M}), \tag{12}$$

which is converted into minimizing the Gibbs energy by the mean-field approximation and message passing scheme.

We employ the associative hierarchical CRF [32,36] which integrates the unary potential ψ_i, the pairwise potential $\psi_{i,j}$ and the higher order potential ψ_c into the Gibbs energy at different levels of the hierarchy (voxels and supervoxels) given by:

$$E(\mathbf{X}|\mathbf{C};`) = \sum_i \psi_i(X_i|\mathbf{C}) + \sum_{i<j} \psi_{i,j}(X_i, X_j|\mathbf{C};\theta) + \sum_c \psi_c(X_c|\mathbf{c}) \tag{13}$$

by the indexes $i, j \in \{1, \ldots, M\}$ correspond to different 3D points $\mathbf{P}_i, \mathbf{P}_j$ in the 3D map $\mathcal{M}$.

Unary Potential: The unary potential $\psi_i(\cdot)$ is defined as the negative logarithm of the probabilistic label for a given 3D point:

$$\psi_i(X_i|\mathbf{C}) = -\log(P(X_i \to \text{ł}|\mathcal{K}_0^t)). \tag{14}$$

This term means the cost of 3D point P_i taking an object label $l \in \mathcal{L}$ based on the incremental semantic probabilistic fusion above. The output of the unary potential for each point is produced independently, and thus, the MAP labeling produced by the unary potential alone is generally inconsistent.

Pairwise Potentials: The pairwise potential $\psi_{i,j}(\cdot)$ is modeled to be a log-linear combination of m Gaussian edge potential kernels:

$$\psi_{i,j}(X_i, X_j|\mathbf{C};\theta) = \mu(X_i, X_j) \sum_m \omega^{(m)} k^{(m)}(\mathbf{f}_i, \mathbf{f}_j;\theta), \tag{15}$$

where $\mu(\cdot)$ is a label compatibility function corresponding to the Gaussian kernel functions $k^{(m)}(\mathbf{f}_i, \mathbf{f}_j)$. $\mathbf{f}$ denotes the feature vector for the 3D point $\mathbf{P}$ including the position, the RGB appearance and the surface normal vector of the reconstructed surface. And $\mu(\cdot)$ is a Potts model given by:

$$\mu(l, l') = [l \neq l'] = \begin{cases} 1 & l \neq l' \\ 0 & l = l' \end{cases}. \tag{16}$$

This term is defined to encourage the consistency over pairs of neighboring points for the local smoothness of the 3D semantic map. We employ two Gaussian kernels for the pairwise potentials following the previous work [7]. The first one is an appearance kernel as follows:

$$k^{(1)}(\mathbf{f}_i, \mathbf{f}_j;`) = \exp\left(-\frac{|\mathbf{P}_i - \mathbf{P}_j|^2}{2\theta_{\mathbf{P},c}^2} - \frac{|\mathbf{c}_i - \mathbf{c}_j|^2}{2\theta_c^2}\right), \tag{17}$$

where $\mathbf{c}$ is the RGB color vector of the corresponding 3D points. This kernel is used to build long range connections between 3D points with a similar appearance.

The second one, a spatial smoothness kernel, is defined to enforce a local, appearance-agnositc smoothness among 3D points with similar normal vectors.

$$k^{(2)}(\mathbf{f}_i, \mathbf{f}_j;\theta) = \exp\left(-\frac{|\mathbf{P}_i - \mathbf{P}_j|^2}{2\theta_{\mathbf{P},n}^2} - \frac{|\mathbf{n}_i - \mathbf{n}_j|^2}{2\theta_n^2}\right), \tag{18}$$

where **n** are the respective surface normals. The surface normal are computed using the Triangulated Meshing using Marching Tetrahedra (TMMT) proposed in [32]. Note that the original method is towards producing a dense labeling with the stereo vision. Since the LSD-SLAM only generates semi-dense 3D point clouds, we modify the TMMT to extract a triangulated mesh within limited ranges of short distance between 3D points.

High Order Potential: The higher order term $\psi_c(X_c|\mathbf{c})$ encourages the 3D points (voxels) in the given segment to take the same label and penalizes partial inconsistency of supervoxels as described in [36]. It is defined as

$$\psi_c(X_c|\mathbf{c}) = \min_{l\in\mathcal{L}}(\gamma_c^{\max}, \gamma_c^l + k_c^l N_c^l), \tag{19}$$

where γ_c^l represents the cost if all voxels in the segment take the label l. $N_c^l = \sum_{i\in\mathbf{c}} \delta$ is the number of inconsistent 3D points with the label l which is penalized with a factor k_c, regarded as the inconsistency cost.

All parameters $\theta_{\mathbf{P},c}, \theta_c, \theta_{\mathbf{P},n}, \theta_n, \theta_{\mathbf{P},s}, \theta_s$ specify the range in which points with similar features affect each other, respectively. They can be obtained using piece-wise learning.

5. Experiments and Results

We demonstrate the performance of our approach on the KITTI dataset [10], which contains a variety of urban scene sequences involving lots of moving objects in various lighting conditions. It consists of various datasets, such as the semantic dataset, the odometry dataset, and the detection dataset. Thus, it is very challenging for the 3D reconstruction. The KITTI dataset contains a 2D semantic segmentation data of 200 labeled training images and 200 test images[1] . Its data format and metrics conform with the Cityscapes dataset [9]. The Cityscapes dataset involves 19 classes within high quality pixel-level annotations of 5000 images with a resolution of 2048 × 1024, including 2975 training images, 500 validation images, and 1525 testing images. In our experiment, we train the model on the Cityscapes and then tune it on the KITTI taking the volume size of dataset into account.

For the training of 2D semantic segmentation model, various encoder models in the DeepLab-v3+ are evaluate including *ResNet*, *Xception*, and *MobileNet*. And we find that the "poly" stochastic gradient descent is better than the "step" one on these datasets. The *TensorFlow* library is employed to do the training and inference on the workstation with 4 Nvidia Titan X GPU cards. The hyper-parameters used in training are set corresponding to the datasets and models as shown in Table 2.

Table 2. Hyper-parameters used in the training step

Dataset	Encoder	Learning Rate	Learning Power	Momentum	Weight Decay	Batch	Steps
Cityscapes	ResNet_50	0.003	0.9	0.9	0.0001	8	20000
	ResNet_101	0.003	0.9	0.9	0.0001	8	20000
	Xception_41	0.01	0.9	0.9	0.00004	8	10000
	Xception_65	0.01	0.9	0.9	0.00004	8	10000
	Xception_71	0.01	0.9	0.9	0.00004	8	10000
	MobileNet_v2	0.001	0.9	0.9	0.00004	64	10000
KITTI	ResNet_50	0.003	0.9	0.9	0.0001	8	20000
	ResNet_101	0.003	0.9	0.9	0.0001	8	20000
	Xception_41	0.01	0.9	0.9	0.00004	8	10000
	Xception_65	0.01	0.9	0.9	0.00004	8	10000
	Xception_71	0.01	0.9	0.9	0.00004	8	10000
	MobileNet_v2	0.001	0.9	0.9	0.00004	64	10000

We benchmark the performance of our semantic mapping system on the KITTI odometry dataset[2]. There are 22 sequences with the consecutive RGB frames, in which there are 11 sequences with the ground-truth poses for evaluation. The scenes contain serious illumination change, moving objects like persons and vehicles, and

[1] http://www.cvlibs.net/datasets/kitti/eval_semseg.php?benchmark=semantics2015

[2] http://www.cvlibs.net/datasets/kitti/eval_odometry.php

some turns as shown in Figure 3. These road-scene frames involves two resolutions 1242 × 375 and 1226 × 370. Our system runs on an Intel Core i7-5960K CPU and a NVIDIA Titan X GPU for online process.

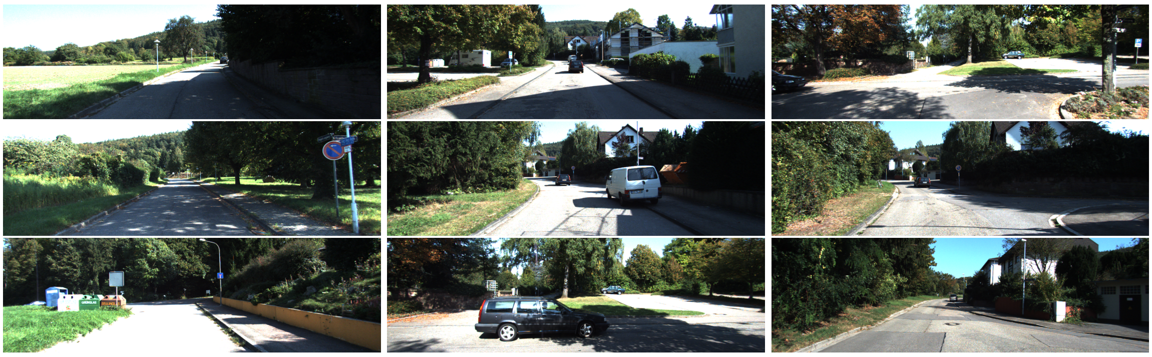
Since the KITTI sequences are mostly captured in 10 Hz, it is highly below the normal speed requirements of LSD-SLAM about 60 Hz. In addition, the LSD-SLAM is hard to handle severe turning when the platform moves. Due to the limit of the monocular LSD-SLAM, we choose 6 sequences to evaluate.

In the following sections, we show some qualitative results for our approach in 5.1 and the quantitative results of our evaluation are presented in 5.2, in which we also make the runtime analysis on our semantic mapping approach.

5.1. *Qualitative Results*

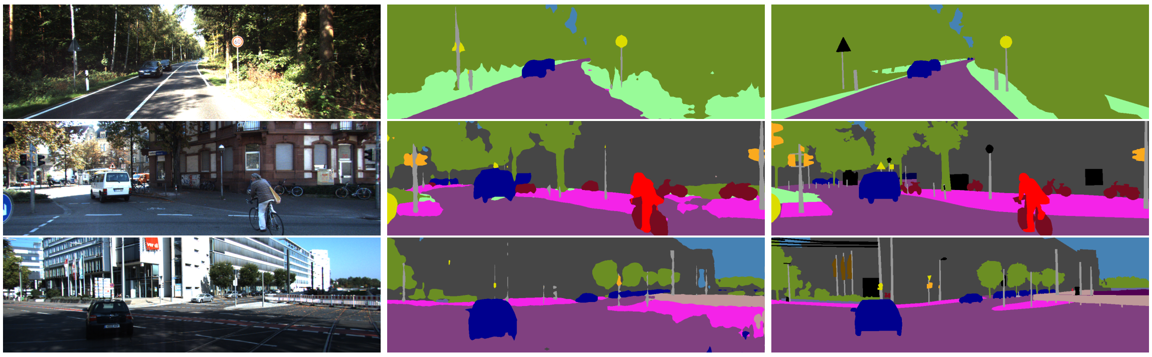
First, we present some qualitative results of the KITTI semantic dataset in Figure 4. Then, we use the trained model to make prediction on the KITTI odometry dataset, and the results are exemplified as shown in Figure 5.

Take the sequence *odometry_03* as an example of our semantic mapping approach. The sequence consists of 801 RGB frames on a urban road of about 560m and a camera calibration file. Figure 6 shows the semantic reconstruction with a close-up view including large-scale annotations such as *road*, *building* and even small-scale objects like *traffic signs*. Note we discard some keyframes at the beginning, due to random initialization of LSD-SLAM.



(a) IC **(b)** MO **(c)** T

. **Figure 3.** Instances in the *odometry_03* sequence. IC: Illumination Change, MO: Moving Objects, T: Turns



(a) Raw Image **(b)** Prediction **(c)** Ground Truth

Figure 4. Qualitative results of 2D semantic segmentation

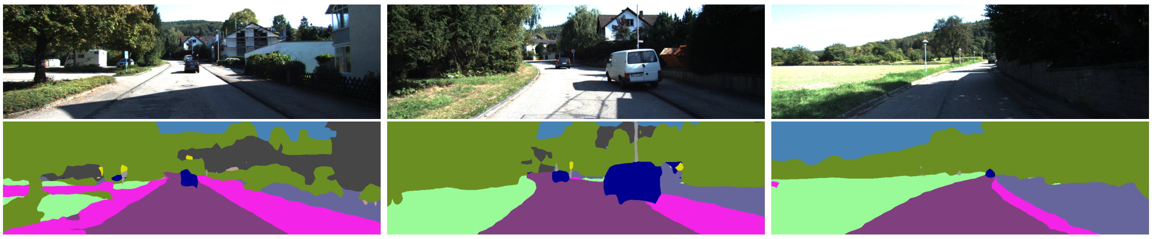


Figure 5. Instances of 2D semantic segmentation in the KITTI odometry set

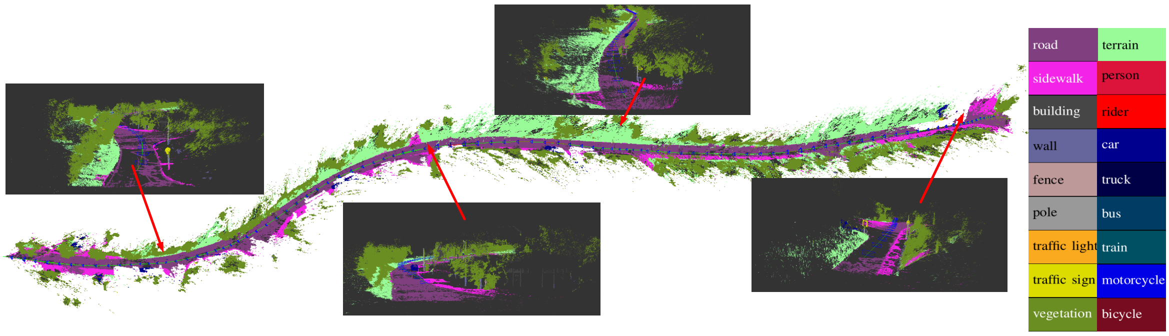


Figure 6. Qualitative results of 3D semantic mapping from the sequence *odometry_03*. Our approach not only reconstructs and labels entire outdoor scenes that include roads, sidewalks and buildings, but also accurately recovers thin objects such as traffic signs and trees.The close-up views show the details of the map.

5.2. Quantitative Results

For the quantitative performance of our approach, we focus on the 2D semantic segmentation and the runtime of the entire system, since the 3D reconstruction mainly depends on the LSD-SLAM method.

Semantic Segmentation: Table 3 shows the quantitative results of 2D semantic segmentation based on different DeepLab-v3+ models on the KITTI datasets. We evaluate these models by the mean intersection/union (mIOU) score, the model size, and the computational runtime. The mIOU score is defined as

$$\mathrm{mIOU} = \frac{1}{|\mathcal{L}|} \sum_{i=1}^{|\mathcal{L}|} \mathrm{TP}_i / (\mathrm{TP}_i + \mathrm{FP}_i + \mathrm{FN}_i) \tag{20}$$

in terms of the True/False Positives/Negatives for a given class i. We do not resize the image to evaluate the models here. Whereas, for the 3D semantic mapping process, we need to half resize the input images in order to make a trade-off between accuracy and computational speed.

During the training process, these models are initialized with the checkpoints pre-trained from various datasets including ImageNet [37] and MS-COCO [38]. In the training step on the Cityscapes dataset, we directly use the ImageNet-pretrained checkpoints as the initialization. Note we employ the *MobileNet_v2* based model which has been pre-trained on MS-COCO dataset, and the *Xception_71* based model has been pre-trained on both ImageNet and MS-COCO datasets. These pre-trained models can be accessed from the github[3].

[3] https://github.com/tensorflow/models/blob/master/research/deeplab/g3doc/model_zoo.md

Table 3. Quantitative results of various encoder parts of DeepLab-v3+ on the Cityscapes and the KITTI. I: ImageNet, M: MS-COCO, C: Cityscapes

Dataset	Encoder	Crop Size	mIOU[0.5:0.25:1.75]	Pb Size (MB)	Runtime (s)	I	M	C
Cityscapes	ResNet_50	769	63.92	107.8	-	√		
	ResNet_101	769	69.88	184.1	-	√		
	Xception_41	769	68.5	113.4	-	√		
	Xception_65	769	78.73	165.7	5.0	√		
	Xception_71	769	80.24	167.9	-	√	√	
	MobileNet_v2	513	70.7	8.8	0.8		√	
	MoblieNet_v2	769	70.9	8.8	0.8		√	
KITTI	ResNet_50	769	51.35	107.8	0.9	√		√
	ResNet_101	769	57.12	184.1	1.1	√		√
	Xception_41	769	54.2	113.4	0.88	√		√
	Xception_65	769	64.8	165.6	1.13	√		√
	Xception_71	769	66.2	167.9	1.26	√	√	√
	MobileNet_v2	513	57.74	8.8	0.2		√	√
	MobileNet_v2	769	60.73	8.8	0.2		√	√

Then we fine-tune the models on the KITTI dataset by using the pre-trained Cityscapes model. The *Xception_71* based model performs the best mIOU performance but a rather slow computational speed. The *MobileNet_v2* based model has a moderate *mIOU*, the smallest file size and the fastest speed. Note the *MobileNet_v2* based model does not employ ASPP and decoder modules for fast computation. Considering the balance between computational speed and accuracy, we choose the *MobileNet_v2* based model to carry out the 2D semantic segmentation in our approach. Table 4 shows the performance of the *MobileNet_v2* based model on the VAL/TEST split of the KITTI dataset.

Table 4. Results of our selected model on the val/test of the KITTI datasets.

method	road	sidewalk	building	wall	fence	pole	traffic light	traffic sign	vegetation	terrain	sky	person	rider	car	truck	bus	train	motorcycle	bicycle	IoU
VAL	95.7	73.9	87.1	38.1	44.2	42.7	48.6	60.3	89.1	52.3	90.1	70.1	36.5	89.1	44.6	62.2	37.4	36.1	67.7	60.3
TEST	96.1	73.7	86.2	37.9	41.4	40.1	50.3	58.3	90.2	66.8	91.3	72.4	40.3	91.8	33.7	46.4	37.1	46.0	62.4	60.9

We also make the test regarding to the effect of pre-training on the Cityscapes dataset. In Table 5, the salience has been illustrated on training the *Xception_65* and *MobileNet_v2* models. The Cityscapes pre-trained models could greatly improve the performance of 2D semantic segmentation on the KITTI dataset.

Table 5. Performance of 2D semantic segmentation with/without the Cityscapes. Using the pre-trained Cityscapes model, the accuracy of 2D semantic segmentation could be greatly improved on the KITTI semantic data.

Encoder	mIOU[0.5:0.25:1.75]	WITH Cityscapes
ResNet_101	52.46	
ResNet_101	57.12	√
Xception_65	55.99	
Xception_65	64.8	√
MobileNet_v2	51.82	
MobileNet_v2	60.73	√

Note that towards the 3D semantic mapping, since we use a novel monocular 3D mapping different from the other related work, it is not easy to make quantitative comparison here. Kundu *et al.*'s work [25] propose a joint semantic segmentation and 3D reconstruction from monocular video, but it is an offline approach with different 3D representation in the form of a 3D volumetric semantic + occupancy map.

Runtime and Storage: As shown in Table 6, our SLAM system runs about 40ms on average to process each frame, extract the keyframes and update the map. Since we reduce the size of the input image, the semantic segmentation process requires about 100ms to infer 2D semantic information parallel upon the keyframes, and the incremental fusion process needs 50ms on average. In the experiments, we find the SLAM process normally selects a keyframe from more than every 4 frames. It keeps enough timing for the 2D semantic segmentation and the incremental fusion during the 3D semantic mapping. Thus, our approach could run in real-time. Moreover, considering the speed of moving platform, in case of the speed of 60KMH, the semantic segmentation process on selected keyframes corresponds to a distance about 2 meters, which is not too sparse for an urban scene.

The lower part of this table shows the ranges of the CRF timing with different configurations due to the different size of point clouds when testing various sequences in the experiments. The CRF update runs offline due to slow inference speed on the CPU. Thus, it is only applied once at the end of the sequence. Optimized GPU implementation can be studied in future to realize the online CRF update.

Table 6. Timing results. The table lists the operation time for different components of our system. Times of three core components are averaged over all sequences and the CRF timings depends on the iterations and the point cloud sizes.

Component	Consumed Time (ms)
Semantic segmentation	100
SLAM	40
Incremental fusion	50
3D CRF 1 Iter.	800-2000
3D CRF 2 Iter.	1200-2400
3D CRF 3 Iter.	1500-3000
3D CRF 4+ Iter.	>2000

Taking the *odometry_03* sequence as example, our approach acquires 114 keyframes with 2.8E+07 3D points. Compared to the total 801 frames, the system utilizes only about $1/7$ frames for mapping. Note that smaller values of the parameters *KFDistWeight* and *KFUsageWeight* could give more constraints between keyframes so that to achieve more accurate mapping. But it has a rather limited influence on the number of keyframes, the number of 3D points and the size of storage.

6. Conclusions

We have presented a fast monocular 3D semantic mapping system which runs on a CPU coupled with a GPU. An incremental fusion method is introduced to combine 2D semantic segmentation and 3D reconstruction online. We exploit a state-of-the-art deep CNN to realize the scene parsing in the road contexts. Direct monocular SLAM provides a quick 3D mapping based on selected keyframes and corresponding depth estimation. Since the semantic segmentation only runs and propagates on the keyframes, this reduces the computational cost and improves the accuracy of semantic mapping. The offline regularization with a CRF model can enhance the mapping further.

Since the original LSD-SLAM is hard to handle the cases of sharp turns which are frequent in ordinal driving, our system is not stable in such conditions. In addition, semi-dense 3D reconstruction should be replaced by a dense model. In future work, we plan to introduce several state-of-the-art SLAM methods to improve the initialization and resistance to serious movements, i.e., rotations. Research on how labeling boosts 3D reconstruction of SLAM would be an interesting direction. The optimization of the regularization module would be another effective direct on the wide-range mapping.

Funding: This research was funded by the Natural Science Foundation of Jiangsu Province grant number No. BK20160700 and No. BK20170681.

Conflicts of Interest: The authors declare no conflict of interest. The funders had no role in the design of the study; in the collection, analyses, or interpretation of data; in the writing of the manuscript, or in the decision to publish the results.

1. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Semantic image segmentation with deep convolutional nets and fully connected crfs, 2015.

2. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. IEEE Conf. on Computer Vision and Pattern Recognition (CVPR), 2017, pp. 2881–2890.
3. Chen, L.C.; Papandreou, G.; Kokkinos, I.; Murphy, K.; Yuille, A.L. Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. *IEEE transactions on pattern analysis and machine intelligence* **2018**, *40*, 834–848.
4. Zhao, H.; Qi, X.; Shen, X.; Shi, J.; Jia, J. ICNet for Real-Time Semantic Segmentation on High-Resolution Images. European Conference on Computer Vision. Springer, 2018, pp. 418–434.
5. Chen, L.C.; Zhu, Y.; Papandreou, G.; Schroff, F.; Adam, H. Encoder-decoder with atrous separable convolution for semantic image segmentation. European Conference on Computer Vision, 2018, pp. 833–851.
6. Wolf, D.; Prankl, J.; Vincze, M. Fast semantic segmentation of 3D point clouds using a dense CRF with learned parameters. 2015 IEEE International Conference on Robotics and Automation (ICRA), 2015, pp. 4867–4873.
7. Hermans, A.; Floros, G.; Leibe, B. Dense 3d semantic mapping of indoor scenes from rgb-d images. 2014 IEEE International Conference on Robotics and Automation (ICRA), 2014, pp. 2631–2638.
8. McCormac, J.; Handa, A.; Davison, A.; Leutenegger, S. SemanticFusion: Dense 3D semantic mapping with convolutional neural networks. Robotics and Automation (ICRA), 2017 IEEE International Conference on. IEEE, 2017, pp. 4628–4635.
9. Cordts, M.; Omran, M.; Ramos, S.; Rehfeld, T.; Enzweiler, M.; Benenson, R.; Franke, U.; Roth, S.; Schiele, B. The cityscapes dataset for semantic urban scene understanding. Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, pp. 3213–3223.
10. Geiger, A.; Lenz, P.; Urtasun, R. Are we ready for autonomous driving? the kitti vision benchmark suite. Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on. IEEE, 2012, pp. 3354–3361.
11. Whelan, T.; Salas-Moreno, R.F.; Glocker, B.; Davison, A.J.; Leutenegger, S. ElasticFusion: Real-time dense SLAM and light source estimation. *The International Journal of Robotics Research* **2016**, *35*, 1697–1716.
12. Noh, H.; Hong, S.; Han, B. Learning Deconvolution Network for Semantic Segmentation. The IEEE International Conference on Computer Vision, 2015.
13. Salas-Moreno, R.F.; Newcombe, R.A.; Strasdat, H.; Kelly, P.H.; Davison, A.J. Slam++: Simultaneous localisation and mapping at the level of objects. Proceedings of the IEEE conference on computer vision and pattern recognition, 2013, pp. 1352–1359.
14. Davison, A.J.; Reid, I.D.; Molton, N.D.; Stasse, O. MonoSLAM: Real-time single camera SLAM. *IEEE transactions on pattern analysis and machine intelligence* **2007**, *29*, 1052–1067.
15. Mur-Artal, R.; Montiel, J.M.M.; Tardos, J.D. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics* **2015**, *31*, 1147–1163.
16. Mur-Artal, R.; Tardós, J.D. Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras. *IEEE Transactions on Robotics* **2017**, *33*, 1255–1262.
17. Kerl, C.; Sturm, J.; Cremers, D. Dense visual SLAM for RGB-D cameras. Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on. Citeseer, 2013, pp. 2100–2106.
18. Engel, J.; Schöps, T.; Cremers, D. LSD-SLAM: Large-scale direct monocular SLAM. European Conference on Computer Vision, 2014, pp. 834–849.
19. Forster, C.; Zhang, Z.; Gassner, M.; Werlberger, M.; Scaramuzza, D. Svo: Semidirect visual odometry for monocular and multicamera systems. *IEEE Transactions on Robotics* **2017**, *33*, 249–265.
20. Long, J.; Shelhamer, E.; Darrell, T. Fully convolutional networks for semantic segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 3431–3440.
21. Badrinarayanan, V.; Kendall, A.; Cipolla, R. SegNet: A Deep Convolutional Encoder-Decoder Architecture for Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2017**, pp. 2481–2495.
22. Howard, A.G.; Zhu, M.; Chen, B.; Kalenichenko, D.; Wang, W.; Weyand, T.; Andreetto, M.; Adam, H. Mobilenets: Efficient convolutional neural networks for mobile vision applications. *arXiv preprint arXiv:1704.04861* **2017**.
23. Sandler, M.; Howard, A.; Zhu, M.; Zhmoginov, A.; Chen, L.C. MobileNetV2: Inverted Residuals and Linear Bottlenecks. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2018, pp. 4510–4520.
24. Valentin, J.P.; Sengupta, S.; Warrell, J.; Shahrokni, A.; Torr, P.H. Mesh based semantic modelling for indoor and outdoor scenes. Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE, 2013, pp. 2067–2074.
25. Kundu, A.; Li, Y.; Dellaert, F.; Li, F.; Rehg, J.M. Joint semantic segmentation and 3d reconstruction from monocular video. European Conference on Computer Vision. Springer, 2014, pp. 703–718.

26. Sengupta, S.; Sturgess, P. Semantic octree: Unifying recognition, reconstruction and representation via an octree constrained higher order MRF. Robotics and Automation (ICRA), 2015 IEEE International Conference on. IEEE, 2015, pp. 1874–1879.

27. Vineet, V.; Miksik, O.; Lidegaard, M.; Nießner, M.; Golodetz, S.; Prisacariu, V.A.; Kähler, O.; Murray, D.W.; Izadi, S.; Pérez, P. Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. Robotics and Automation (ICRA), 2015 IEEE International Conference on. IEEE, 2015, pp. 75–82.

28. Kochanov, D.; Ošep, A.; Stückler, J.; Leibe, B. Scene flow propagation for semantic mapping and object discovery in dynamic street scenes. Intelligent Robots and Systems (IROS), 2016 IEEE/RSJ International Conference on. IEEE, 2016, pp. 1785–1792.

29. Landrieu, L.; Raguet, H.; Vallet, B.; Mallet, C.; Weinmann, M. A structured regularization framework for spatially smoothing semantic labelings of 3D point clouds. *ISPRS Journal of Photogrammetry and Remote Sensing* **2017**, *132*, 102–118.

30. Jadidi, M.G.; Gan, L.; Parkison, S.A.; Li, J.; Eustice, R.M. Gaussian processes semantic map representation. *arXiv preprint arXiv:1707.01532* **2017**.

31. Gan, L.; Jadidi, M.G.; Parkison, S.A.; Eustice, R.M. Sparse Bayesian Inference for Dense Semantic Mapping. *arXiv preprint arXiv:1709.07973* **2017**.

32. Sengupta, S.; Greveson, E.; Shahrokni, A.; Torr, P.H. Urban 3d semantic modelling using stereo vision. Robotics and Automation (ICRA), 2013 IEEE International Conference on. IEEE, 2013, pp. 580–585.

33. Martinovic, A.; Knopp, J.; Riemenschneider, H.; Van Gool, L. 3d all the way: Semantic segmentation of urban scenes from start to end in 3d. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 4456–4465.

34. Hu, H.; Munoz, D.; Bagnell, J.A.; Hebert, M. Efficient 3-d scene analysis from streaming data. 2013 IEEE International Conference on Robotics and Automation. IEEE, 2013, pp. 2297–2304.

35. Krähenbühl, P.; Koltun, V. Efficient inference in fully connected crfs with gaussian edge potentials. Advances in neural information processing systems, 2011, pp. 109–117.

36. Russell, C.; Kohli, P.; Torr, P.H. Associative hierarchical crfs for object class image segmentation. Computer Vision, 2009 IEEE 12th International Conference on. IEEE, 2009, pp. 739–746.

37. Russakovsky, O.; Deng, J.; Su, H.; Krause, J.; Satheesh, S.; Ma, S.; Huang, Z.; Karpathy, A.; Khosla, A.; Bernstein, M. Imagenet large scale visual recognition challenge. *International Journal of Computer Vision* **2015**, *115*, 211–252.

38. Lin, T.Y.; Maire, M.; Belongie, S.; Hays, J.; Perona, P.; Ramanan, D.; Dollár, P.; Zitnick, C.L. Microsoft coco: Common objects in context. European conference on computer vision. Springer, 2014, pp. 740–755.