

Article

# Semantic 3D Reconstruction with Learning MVS and 2D Segmentation of Aerial Images

Zizhuang Wei <sup>1,3†</sup>, Yao Wang <sup>1,3†</sup>, Hongwei Yi<sup>1,3</sup>, Yisong Chen<sup>1,2,3</sup> and Guoping Wang<sup>1,2,3\*</sup>

<sup>1</sup> Graphics & Interaction Lab, School of Electronics Engineering and Computer Sciences, Peking University, Beijing 100871, China

<sup>2</sup> Beijing Engineering Technology Research Center of Virtual Simulation and Visualization, Peking University, Beijing 100871, China

<sup>3</sup> Key Lab of Machine Perception and Intelligent, MOE; Department of Computer Sciences, Peking University, Beijing 100871, China

\* Correspondence: wgp@pku.edu.cn;

† These authors contributed equally to this work.

Version January 22, 2020 submitted to Journal Not Specified

**Abstract:** Semantic modeling is a challenging task that has received widespread attention in recent years. With the help of mini Unmanned Aerial Vehicles (UAV), multi-view high-resolution aerial images of large-scale scenes can be conveniently collected. In this paper, we propose a semantic Multi-View Stereo (MVS) method to reconstruct 3D semantic models from 2D images. Firstly, 2D semantic probability distribution is obtained by Convolutional Neural Network (CNN). Secondly, the calibrated cameras poses are determined by Structure from Motion (SfM), while the depth maps are estimated by learning MVS. Combining 2D segmentation and 3D geometry information, dense point clouds with semantic labels are generated by a probability-based semantic fusion method. In the final stage, the coarse 3D semantic point cloud is optimized by both local and global refinements. By making full use of the Multi-View Consistency, the proposed method efficiently produces a fine-level 3D semantic point cloud. The experimental result evaluated by re-projection maps achieves 88.4% Pixel Accuracy on Urban Drone Dataset (UDD). In conclusion, our graph-based semantic fusion procedure and refinement based on local and global information can suppress and reduce the re-projection error.

**Keywords:** semantic 3D reconstruction; deep learning; multi-view stereo; probabilistic fusion; graph-based refinement

## 1. Introduction

Semantic 3D reconstruction makes Virtual Reality (VR) and Augmented Reality (AR) much more promising and flexible. In computer vision, 3D reconstruction and scene understanding receive more and more attention these days. 3D models with correct geometry structure and semantic segmentation are crucial in urban planning, automatic piloting, robot vision and many other fields. For urban scenes, semantic labels are used to visualize targets such as buildings, terrain, and roads. 3D point cloud with semantic labels makes the 3D map more simple to understand, thereby propelling the subsequent research and analysis. 3D semantic information also shows potential in automatic piloting. For a self-driving vehicle, one of the most important things is to distinguish whether the road is passable or not. Another essential thing for an autonomous automobile is to localize other vehicles in real-time so that it can adapt to their speed, or exceed them if necessary. In the field of robotics, scene understanding is a standard task to recognize surrounding objects. The semantic of the surrounding environment plays a vital role in applications like loop closure and route planning.

30 Although 3D semantic modeling has been widely concerned in recent years, the approaches of  
31 extracting semantic information as a post-processing of point cloud reconstruction generally lead  
32 to inconsistent or incorrect results. Performing semantic segmentation on point cloud data is more  
33 hard than on 2D images. One major problem is the lack of 3D training data, since labeling a dataset  
34 in 3D is much more laborious than in 2D. Another challenge is the unavoidable noises in 3D point  
35 clouds, which makes it difficult to accurately distinguish which category a point belongs to. Thus, it  
36 is necessary to develop new semantic 3D reconstruction approaches by simultaneously estimating  
37 3D geometry and semantic information over multiple views. In the past few years, many researches  
38 on image semantic segmentation have achieved promising results by deep learning techniques [1–4].  
39 Deep learning methods based on well-trained neural networks can help us do pixel-wise semantic  
40 segmentation on various images. Meanwhile, Deep learning based methods are not only able to  
41 extract semantic information but also practical in solving Multi-View Stereo (MVS) problems. Recently,  
42 learning based MVS algorithms [5,6] have been proposed to generate high precision 3D point clouds  
43 for large-scale scenes. These results inspire us much and give rise to the research of semantic 3D  
44 reconstruction. In this paper, we mainly focus on developing accurate, clear and complete 3D semantic  
45 models of urban scenes.

46 Once satisfactory depth and semantic maps are acquired, 3D semantic models can be easily  
47 generated. 3D laser scanners can detect depth directly but only perform well in short-distance indoor  
48 scenes. Compared with 3D laser scanners, the purely RGB-based method to reconstruct 3D models from  
49 2D images is cheaper, faster, and more generalized. Recently, Unmanned Aerial Vehicles (UAV) have  
50 become applicable to collect multi-view high-resolution aerial images of large-scale outdoor scenes.  
51 The calibrated camera poses can be obtained from the images by the traditional Structure-from-Motion  
52 (SfM) technique. After that, 3D point clouds are determined by fusing 2D images according to  
53 multi-view geometry.

54 However, due to the occlusions, the complexity of environments, and the noise of sensors, both  
55 2D segmentation and depth estimation results contain errors. As a result, many inconsistencies may  
56 occur when projecting the multi-view 2D semantic labels to the corresponding 3D points. There are  
57 still plenty of works to do to obtain accurately segmented 3D semantic models. With the booming  
58 of deep learning methods, 2D segmentation tasks now reach a high performance, which makes it  
59 possible to acquire a large-scale 3D semantic model easily. Nevertheless, errors within depth maps and  
60 semantic maps may lead to inconsistency. This can be alleviated by considering 3D geometry and 2D  
61 confidence maps together in an optimization module. Moreover, 3D models with coarse segmentation  
62 still need further refinement to filter error points. In a nutshell, the main contributions of our work are  
63 three folds:

- 64 • We present an end-to-end learning based semantic 3D reconstruction framework, which reaches  
65 high Pixel Accuracy on Urban Drone Dataset (UDD) [7].
- 66 • We propose a probability-based semantic MVS method, which combines the 3D geometry  
67 consistency and 2D segmentation information to generate better point-wise semantic labels.
- 68 • We design a joint local and global refinement method, which is proved effective by computing  
69 re-projection errors.

## 70 2. Related Work

71 Right before the renaissance of deep learning, it was a hard task to get a good pixel-wise  
72 segmentation map on images. SY Bao *et al.* [8] take object-level semantic information to constrain  
73 camera extrinsic. Some other methods perform the segmentation directly on the point cloud or  
74 meshes, according to their geometric characteristics. Martinovic A *et al.* [9] and Wolf D *et al.* [10]  
75 take the random forest classifier to do point segmentation, while Häne C *et al.* [11][12] and Savinov  
76 N *et al.* [13] treat it as an energy minimization problem in a Conditional Random Field(CRF). Ray  
77 potential(likelihood) is frequently adopted in semantic point cloud generation.

The flourishing of CNN based semantic segmentation methods quickly outperforms traditional methods in image semantic segmentation tasks, such as Fully Convolutional Network(FCN) [1] and Deeplab [3]. High-level computer tasks like scene understanding and semantic 3D reconstruction are now steady and rudimentary processes. The goal of 3D semantic modeling is to assign a semantic label to each 3D point rather than each 2D pixel. Several learning-based approaches follow the end-to-end manner, analyzing the point cloud and giving segmentation results directly on 3D. Voxel-based methods like ShapeNets [14] and VoxNet [15] are proposed naturally. Some methods learn a spatial encoding of each point and then aggregate all individual point features to a global point cloud signature [16] [17]. However, current deep learning-based segmentation pipelines cannot handle noisy large-scale 3D point clouds. Thus, a feasible method is required to firstly perform pixel-wise semantic segmentation on 2D images and then back-project these labels into 3D space using the calibrated cameras to be fused. The methods above handle the point cloud directly, which means they carry a costly computational burden. In other words, they cannot manage large-scale 3D scenes without first partitioning the scene. More than that, because the morphological gap between point clouds in different scenarios is too large. These algorithms may be poorly generalized.

There are several methods doing semantic segmentation on 2D image and making use of multi-view geometric relationships to project semantic labels into 3D space. For RGBD based approaches, once good semantic maps of each image are acquired, the semantic point clouds can easily be fused. V Vineet *et al.* [18] take advantage of random forest to classify 2D features to get semantic information, while Zhao C *et al.* [19] use FCN with CRF-RNN to perform segmentation on images. J McCormac *et al.* [20] and Li X *et al.* [21] propose incremental semantic label fusion algorithms to fuse 3D semantic map. For RGB based approaches, also addressed as Structure from Motion(SfM) and MVS, each point in the generated 3D structure corresponds to pixels on several images. Following the prediction of 2D labels, the final step is to assign each 3D pixel a semantic label [20] [22]. The refinement process is as essential as the generation process of the semantic point cloud itself. Chen Y *et al.* [7] and EK Stathopoulou *et al.* [23] filter the mismatching by semantic labels of feature points. With the motivation of denoising, R Zhang *et al.* [24] propose a Hough Transform based algorithm called FC-GHT to detect plane on point cloud for further semantic label optimization. EK Stathopoulou *et al.* [23] use semantic information as a mask to wipe out the meshes belonging to the semantic class *sky*. These methods have two primary drawbacks. Firstly, they only use the final semantic maps, which means the probability of other categories is discarded. Secondly, they contain no global constraints integrated into their algorithms. In response, we propose some ideas for improvement.

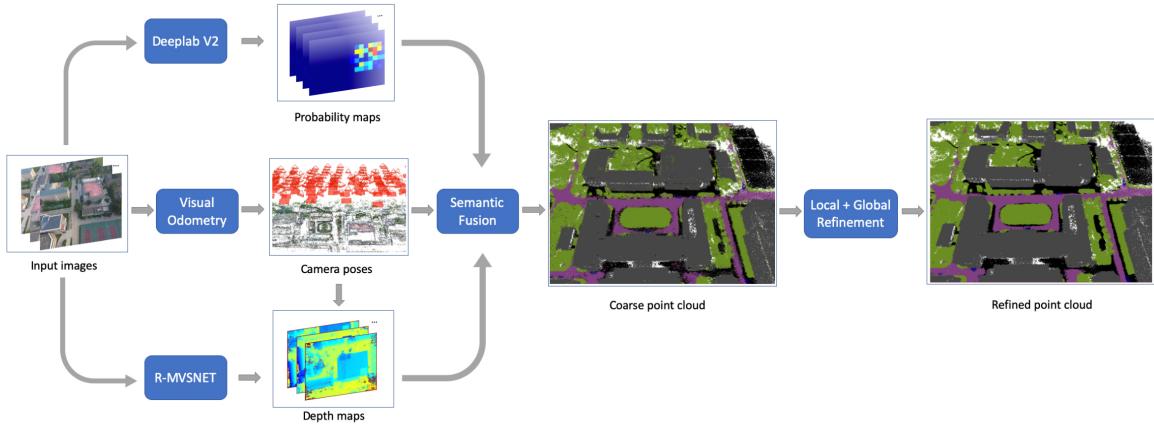
### 3. Method

#### 3.1. Overall Framework

The overall framework of our method is depicted in Figure 1. In Deeplab v2 [3] based 2D segmentation branch, we discard the last argmax layer of the network. We save pixel-wise semantic probability maps for every image instead. With the help of COLMAP-SfM [25], we simultaneously estimate the camera parameters and depth ranges for the source images. In order to acquire 3D geometry for large scale scenes, we utilize learning-based MVS method R-MVSNet [6] to estimate depth maps for multiple images. After 2D segmentation and depth estimation, we obtain a dense semantic point cloud by the semantic fusion method according to multi-view consistency. Finally, we propose a graph-based point cloud refinement algorithm integrating both local and global information as the last step of our pipeline.

#### 3.2. 2D Segmentation

In this research, Deeplab v2 [3] with Residue Block is adopted as our segmentation network. The pre-trained weights of ResNet-101 [26] on Imagenet [27] are used as our initial weights. We adopt the residual block to replace the ordinary 2D convolution layer to improve the training performance. We



**Figure 1.** General pipeline of our work. Three branches are implemented to process the reconstructed data set. The upper branch is the semantic segmentation branch to predict the semantic probability map; the middle branch is SfM to calculate the 3D odometry and camera poses; the lower branch is to estimate the depth map. Then semantic fusion is applied to fuse them into coarse point cloud. The last step is to refine the point cloud by local and global methods.

also modify the softmax layer that classifies the images to fit the label space of UDD [7] dataset. With the network all set up, the training set of UDD [7] is employed for transfer learning.

The label space of UDD [7] is denoted as  $\mathcal{L} = \{l_0, l_1, l_2, l_3, l_4\}$ , which contains *Vegetation*, *Building*, *Road*, *Vehicle* and *Background*. After the transfer learning process, we predict the semantic maps for every image in the reconstruction dataset. Furthermore, we save the weight matrix before the last argmax layer. This matrix  $P(\mathcal{L})$  represents the probability distributions of every pixel in the semantic label space.

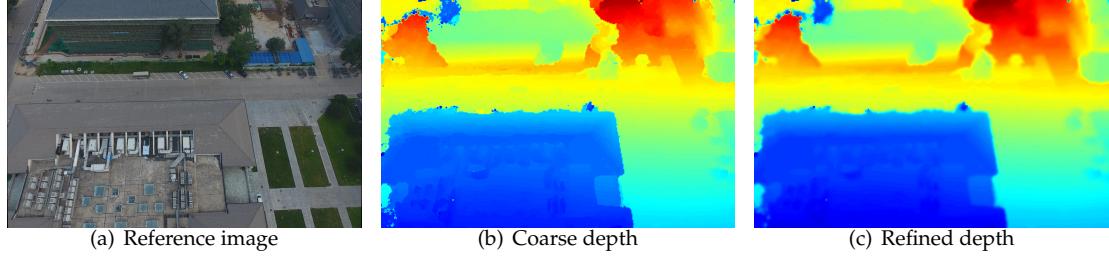
### 3.3. Learning-based MVS

In order to acquire 3D geometry for large scale scenes, we explore learning-based MVS method to estimate depth maps for multiple images. R-MVSNet [6], a deep learning architecture with capability to handle multi-scale problem, has advantages in processing high-resolution images and large-scale scenes. Moreover, R-MVSNet utilizes the Gated Recurrent Unit (GRU) to sequentially regularize the 2D cost maps, which reduces the memory consumption and makes the network flexible. Thus, we follow the framework of R-MVSNet to generate corresponding depths of the source images and train it on DTU [28] dataset. Camera parameters and image pairs are determined by the implementation of COLMAP-SfM [25], while depth samples are chosen within  $[d_{min}, d_{max}]$  using the inverse depth setting. The network returns a probability volume  $P$  where  $P(x, y, d)$  is the probability estimation for the pixel  $(x, y)$  at depth  $d$ , then the expectation depth value  $d(x, y)$  is calculated by the probability weighted sum over all hypotheses:

$$d(x, y) = \sum_{d=d_{min}}^{d_{max}} P(x, y, d) \cdot d. \quad (1)$$

However, as with most depth estimation methods, the coarse pixel-wise depth data  $d(x, y)$  generated by R-MVSNet may contain errors. Therefore, before point cloud fusion by the depth maps, it is necessary to perform denoising process on the depth data. In this paper, we apply bilateral filtering method to improve the quality of depth maps with edge preservation, the refined depth data  $d'(x, y)$  is obtained by:

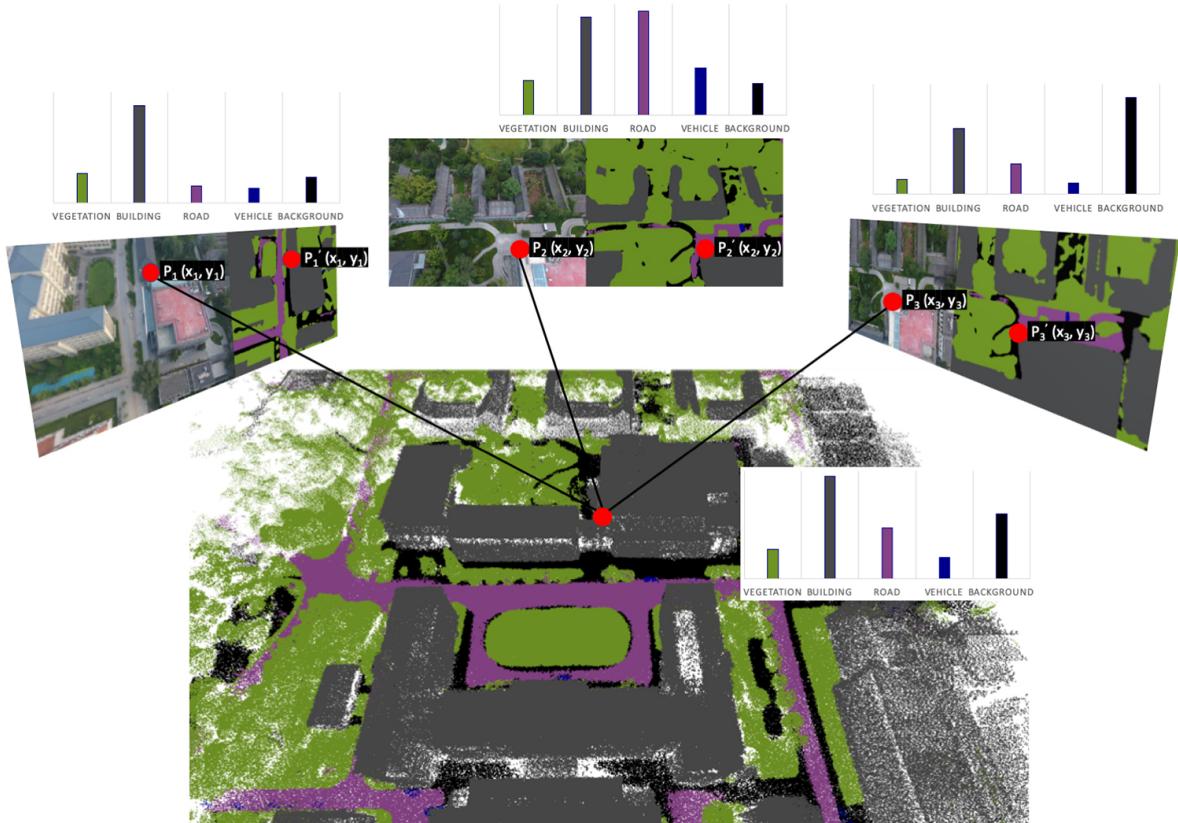
$$d'(x, y) = \frac{\sum_{i,j} \omega(x, y, i, j) \cdot d(x, y)}{\sum_{i,j} \omega(x, y, i, j)} \quad (2)$$



**Figure 2.** Visualization of the depth map estimated by learning-based MVS method. **Left:** The input image. **Middle:** Depth estimation by R-MVSNet [6]. **Right:** Refined depth by bilateral filtering.

where  $\omega(x, y, i, j) = \exp\left(-\frac{(x-i)^2 + (y-j)^2}{2\sigma_f^2} - \frac{\|d(x,y) - d(i,j)\|^2}{2\sigma_g^2}\right)$  is the weighted coefficient,  $\sigma_f$  and  $\sigma_g$  are the variance of domain kernel  $f(x, y, i, j) = \exp\left(-\frac{(x-i)^2 + (y-j)^2}{\sigma_f^2}\right)$  and range kernel  $g(x, y, i, j) = \exp\left(-\frac{\|d(x,y) - d(i,j)\|^2}{\sigma_g^2}\right)$  respectively. As shown in Figure 2, the depth map becomes more smooth with edge preservation after bilateral filtering.

### 153 3.4. Semantic Fusion



**Figure 3.** Illustration of our semantic fusion method, the 3D semantic labels are determined by multi-view information — the 3D point's label is decided by the correspondence accumulated probability of 2D pixels in each image.

With the learning 2D segmentation and depth estimation, pixel-wise 2D semantic labels and depth maps are obtained for each view. However, because of the occlusions, complexity of environments and noise of sensors, both image segmentation results and depth maps might have a large amount of

157 inconsistencies between different views. Thus, we further cross filter the depth maps by their neighbor  
 158 views, then produce the 3D semantic point clouds by combining 2D segmentation and depth maps  
 159 with multi-view consistency.

160 Similar to other depth map based MVS methods [6,29], we utilize geometric consistency to cross  
 161 filter the multi-view depth data. Given the pixel  $(x, y)$  from image  $I_i$  with depth  $d(x, y)$ , we project  
 162  $(x, y)$  to the neighbor image  $I_j$  through  $d(x, y)$  and camera parameters. In turn, we re-project the  
 163 projected pixel back from the neighbor image  $I_j$  to the original image  $I_i$ , the re-project depth on  $I_i$  is  
 164  $d_{reproj}$ . We consider the pixel consistent in the neighbor view  $I_j$  when  $d_{reproj}$  satisfies:

$$\frac{|d(x,y) - d_{reproj}|}{d(x,y)} < \tau. \quad (3)$$

165 According to the geometric consistency, we filter the depths which are not consistent in more than  
 166  $k$  views. In this paper, we take  $\tau = 0.01$  and  $k = 3$ .

167 After cross filtering, the depths are projected to 3D space to produce 3D point clouds. Since our  
 168 purpose is to assign point-wise semantic labels for the 3D model, we propose a probabilistic fusion  
 169 method to aggregate multi-view 2D semantic information. With the fine-tuned CNN, a pixel-wise  
 170 label probability distribution  $P(\mathcal{L})$  has been calculated for each source image. Given a 3D point  $X$   
 171 which visible in  $N$  views, the corresponding probability on view  $i$  for label  $l_j$  is  $p_i(l_j)$ , we accumulate  
 172 the multi-view probability distribution of each view as follows:

$$P(l_j) = \frac{1}{N} \sum_{i=1}^N p_i(l_j), l_j \in \mathcal{L}, \quad (4)$$

173 where  $P(l_j)$  denotes the probability of point  $X$  labeling by  $l_j$ . In this way, we transfer the  
 174 probability distribution of multi-view images into 3D space. Generally, the predicted 3D semantic  
 175 label can be determined by the argmax operation as:

$$l(X) = \underset{l_j}{\operatorname{argmax}}(P(l_j)), l_j \in \mathcal{L}, \quad (5)$$

176 where  $l(X)$  is the 3D semantic label of  $X$ . As depicted in Figure 3, the probabilistic fusion method  
 177 effectively reduces errors since it integrates information from multiple images.

### 178 3.5. Point Cloud Refinement

179 Through the semantic fusion method, the 3D point cloud is classified into point-wise semantic  
 180 labels. However, there are still few scattered points with error labels due to incorrect semantics or  
 181 depths of source images. To remove these unexpected semantic errors, we explore both local and global  
 182 refinement strategies for point cloud refinement. *KD-Tree* data structure is employed to accelerate the  
 183 query speed of the point cloud from  $O(n)$  to  $O(\log(n))$ .

184 Generally, Adjacent point clouds often have some correlation and are more likely to be segmented  
 185 into the same class. Hence, we utilize the local refinement method for each point by combining the  
 186 hypothesizes with the neighbor points. Given a 3D point  $X$  from the dense semantic model, through  
 187 the *KD-Tree* structure established by the whole point cloud, the K-Nearest Neighbor of  $X$  could be  
 188 determined in a short time.  $P_i(l_j), i = 1, \dots, k$  represents the probability for neighbor point  $i$  labeling  
 189 by  $l_j$ , the new semantic label  $l'(X)$  is updated by:

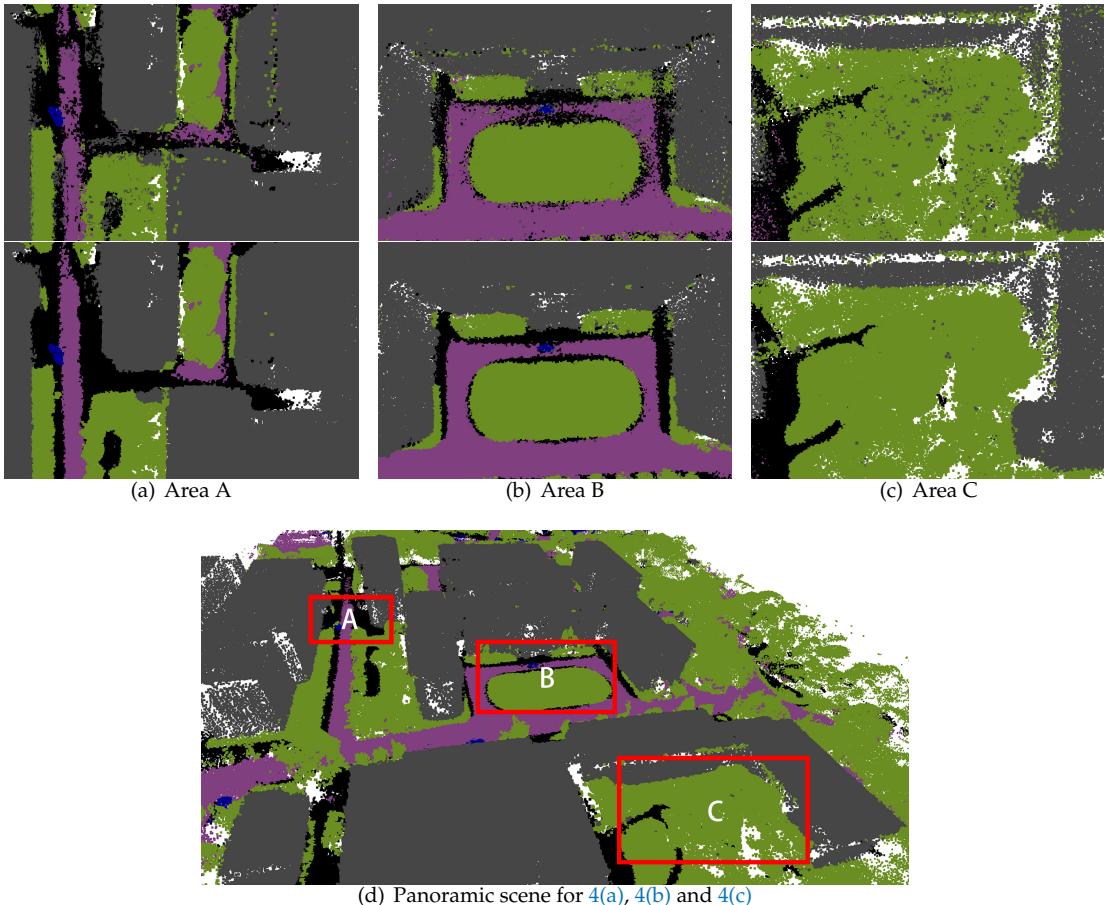
$$l'(X) = \underset{l_j}{\operatorname{argmax}}\left(\frac{1}{k} \sum_{i=1}^k P_i(l_j)\right), l_j \in \mathcal{L}. \quad (6)$$

190 However, the local refinement method only takes the local adjacency into consideration with  
 191 the global information ignored. For overall optimization, we further apply a graph-based global  
 192 refinement method by minimizing an energy function. For every 3D point in the point cloud  $V$ , a

<sup>193</sup> graph  $G$  is established by connecting them with their K-Nearest Neighbor. Then the energy function is  
<sup>194</sup> defined as:

$$E(L) = \sum_{\langle X_p, X_q \rangle \in D} B(l(X_p), l(X_q)) + \lambda \cdot \sum_{X \in V} R(l(X)), \quad (7)$$

<sup>195</sup> where  $L = \{l(X) | X \in V\}$  is the semantics of  $V$  and  $D$  is the set of all neighbor pairs. Similar to  
<sup>196</sup> [30],  $B(l(X_p), l(X_q)) = 1$  and  $R(l(X)) = \frac{1}{k} \sum_{i=1}^k P_i(l_j)$  are the boundary term and inner region term  
<sup>197</sup> respectively, while  $\lambda \geq 0$  is a constant. Finally, the energy  $E(L)$  is minimized by a max-flow algorithm,  
<sup>198</sup> as implemented in [31]. The refined point cloud is illustrated in Figure 4. Compared with the coarse  
<sup>199</sup> result, our method wipes out semantic outliers and noises.



**Figure 4.** Comparison between the point clouds before and after refinement. For 4(a), 4(b) and 4(c),  
**Top:** Coarse result. **Bottom:** Refined result.

## <sup>200</sup> 4. Experimental Evaluation

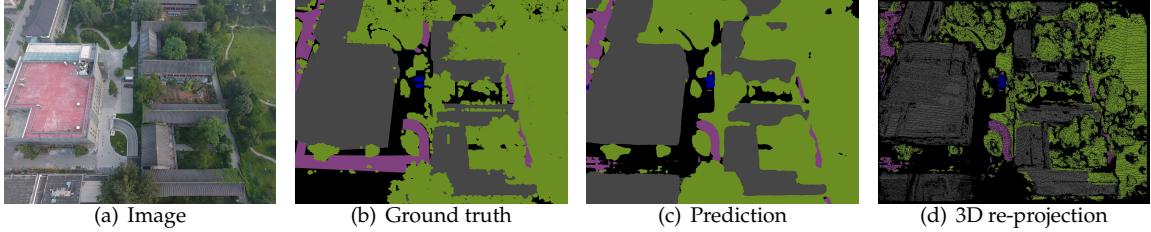
### <sup>201</sup> 4.1. Experimental Protocol

<sup>202</sup> **Dataset** We carry out the training process of semantic segmentation on UDD<sup>1</sup>, a UAV collected  
<sup>203</sup> dataset with five categories, containing 160 and 40 images in the training and validation sets,  
<sup>204</sup> respectively. The categories are *Building*, *Vegetation*, *Road*, *Vehicle* and *Background*. The performance is

<sup>1</sup> <https://github.com/MarcWong/UDD>

205 measured on its test set called PKU-M1, which is a reconstruction dataset also collected by a mini-UAV  
206 at low altitude. PKU-M1 consists of 288 RGB images at  $4000 \times 3000$  resolution. We down-sample the  
207 result to  $1000 \times 750$  to accelerate the prediction speed.

208 **Training** UDD [7] is trained by Deeplab V2 network structure implemented on TensorFlow [32].  
209 We use the Stochastic Gradient Descending [33] optimizer with weight decaying parameter  $5e - 5$ .  
210 Learning rate is initialized to  $1e - 3$  with a momentum of 0.99. The entire experiment is conducted on a  
211 Ubuntu 18.04 server, with an Intel core i7-9700K CPU, 32GB memory and a single Titan X Pascal GPU.



**Figure 5.** Visualization of PKU-M1. Since the re-projection map 5(d) is quite sparse, we use Pixel Accuracy to compare the re-projection map and the ground truth map. **Grey:** Building, **Green:** Vegetation, **Blue:** Vehicle, **Pink:** Road, **Black:** Background. Best viewed in color.

212 **Measurements recap** Assume the number of non-background classes is  $k$ , the confusion matrix  
213  $\mathbf{M}$  for foreground categories can be denoted as below:

$$\mathbf{M} = \begin{pmatrix} c_{11} & c_{12} & \dots & c_{1k} \\ c_{21} & c_{22} & \dots & c_{2k} \\ \dots & \dots & \dots & \dots \\ c_{k1} & c_{k2} & \dots & c_{kk} \end{pmatrix} \quad (8)$$

214 For a specific foreground semantic label  $l_x \in \mathcal{L}$ , the problem can be formulated to a binary  
215 classification problem, where:

$$TruePositive(TP) = c_{xx}, \quad (9)$$

$$TrueNegative(TN) = \sum_{i=0}^k \sum_{j=0}^k c_{ii}, i \neq x, j \neq x, \quad (10)$$

$$FalsePositive(FP) = \sum_{i=0}^k c_{xi}, i \neq x, \quad (11)$$

$$FalseNegative(FN) = \sum_{i=0}^k c_{ix}, i \neq x. \quad (12)$$

216 Then Pixel Accuracy, Precision, Recall, and F1-score can be deducted as below:

$$PixelAccuracy(PA) = \frac{\sum_{i=0}^k c_{ii}}{\sum_{i=0}^k \sum_{j=0}^k c_{ij}}, \quad (13)$$

$$Precision = \frac{TP}{TP + FP}, \quad (14)$$

$$Recall = \frac{TP}{TP + FN}, \quad (15)$$

$$F1-score = 2 \times \frac{Precision \times Recall}{Precision + Recall}. \quad (16)$$

217 *4.2. Evaluation Process*

218 We choose proper measurements to quantitatively evaluate the 2D segmentation performance and  
 219 3D semantic model. We randomly labeled 16 images in PKU-M1 to test the segmentation performance.  
 220 An example of PKU-M1 is shown in Figure 5. Table 1 gives class-wise statistics, where *Building*  
 221 category is segmented very well, but *Vegetation*, *Road* and *Vehicle* is segmented relatively poor. Since  
 222 hand-crafted 3D semantic labeling is now still a challenging and tedious task, especially for large-scale  
 223 scenarios, we have to evaluate the 3D semantic model indirectly. Noticed that each 3D point is assigned  
 224 a semantic label during the semantic fusion process, the label can be projected back to each camera  
 225 coordinate by the geometric relation. We call this step re-projection. Then we can indirectly evaluate  
 226 the 3D semantic point cloud by re-projection images in a simpler manner. However, the re-projection  
 227 map 5(d) is quite sparse. Only foreground labels, which include *Vegetation*, *Building*, *Vehicle*, and  
 228 *Road*, are countable for evaluation. So several common measurements for 2D segmentation are not  
 229 suitable in our cases, such as MIoU (Mean Intersection over Union) and FWIoU (Frequent Weighted  
 230 Intersection over Union). In our experiment, we choose Pixel Accuracy (Equation 13) and class-wise  
 231 F1-score (Equation 16) for evaluation.

Table 1. Evaluation of 2D semantic segmentation.

Category	Accuracy(%)	Precision(%)	Recall(%)	F1 score(%)
Building	95.60	98.25	94.87	96.53
Vegetation	89.85	76.96	71.24	73.99
Vehicle	97.95	67.09	22.02	33.15
Road	87.91	52.58	73.84	61.42

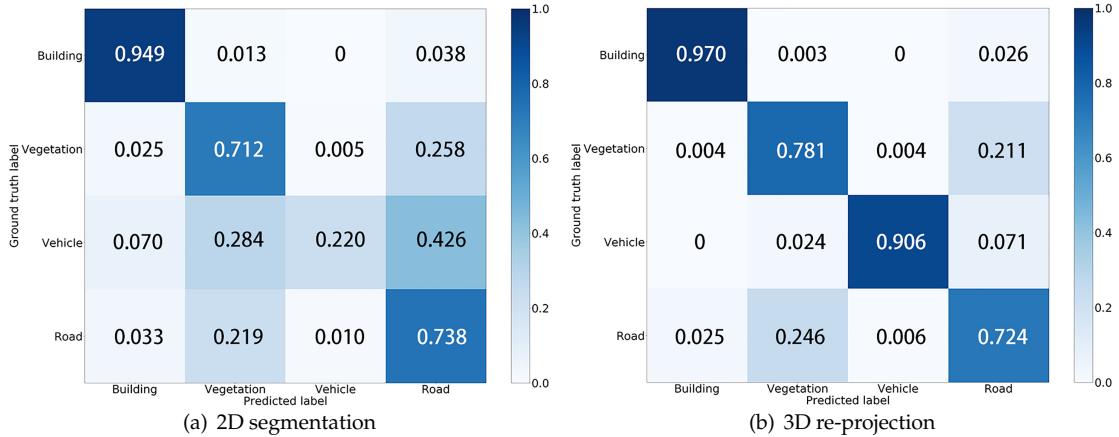


Figure 6. 6(a) is the Confusion Matrix for 2D segmentation, 6(b) is the Confusion Matrix for re-projection images. Four categories are evaluated, which are *Building*, *Vegetation*, *Road* and *Vehicle*. It shows that the re-projection map from 3D semantic points behaves higher accuracy compared with 2D segmentation , due to considering multi-view information.

232 *5. Results and Discussion*

233 *5.1. Quantitative Results*

234 With the semantic fusion process introduced in Section 3.4, the coarse semantic 3D point cloud is  
 235 generated. Its quantitative result is denoted as 3D baseline in Table 2. To be more specific, most points in  
 236 3D baseline are correct, yet with outliers and errors. The evaluation result of 3D baseline's re-projection  
 237 map demonstrates that 3D baseline is much better than 2D in both PA and F1-score. Figure 6(a) and

**Table 2.** Quantitative results of different methods for semantic categories.

Method	Pixel Accuracy(%)				
	Building	Vegetation	Vehicle	Road	All
2D prediction	95.60	89.85	97.95	87.91	85.66
3D baseline	<b>97.51</b>	90.06	99.76	75.59	87.76
Local	96.20	91.38	99.74	68.61	88.24
Global	96.16	91.40	99.45	<b>71.44</b>	88.21
<b>Global+Local</b>	96.19	<b>91.40</b>	<b>99.76</b>	68.16	<b>88.40</b>

Method	F1-score(%)			
	Building	Vegetation	Vehicle	Road
2D prediction	96.53	73.99	33.15	61.42
3D baseline	97.00	74.69	63.66	75.79
Local	97.13	74.87	62.72	75.63
Global	97.15	74.69	73.17	75.03
<b>Global+Local</b>	<b>97.85</b>	<b>76.07</b>	<b>81.40</b>	<b>76.57</b>

238 Figure 6(b) illustrate this fact vividly, where Vehicle is segmented badly in 2D segmentation and  
239 segmented much better in 3D baseline.

240 Furthermore, as shown in Table 2, The Pixel Accuracy of 3D baseline is 87.76%, and the F1-score  
241 of Vehicle, Vegetation, and Road are relatively low. The refinement methods introduced in Section  
242 3.5 denote as Local, Global and Local+Global in Table 2. Local, Global, and Local+Global methods  
243 in Table 2 have been fully tested, and we put the best results under various parameters to this  
244 table. With refinement, the F1-score of Vehicle significantly rises, while Building, Vegetation, and  
245 Road also increase. In addition, the Local+Global optimization approach is better than the Local or  
246 Global approach in each semantic category. It leads to the conclusion that the Local+Global approach  
247 outperforms any single Local or Global approach.

## 248 5.2. Discussion

249 In the following part, the discussion of our semantic fusion method will be arranged in three  
250 aspects: the Down-sample Rate, the parameter chosen for the K-Nearest Neighbor algorithm, and the  
251 Decision strategies between Soft and Hard.

### 252 5.2.1. Parameter Selection for K-Nearest Neighbor

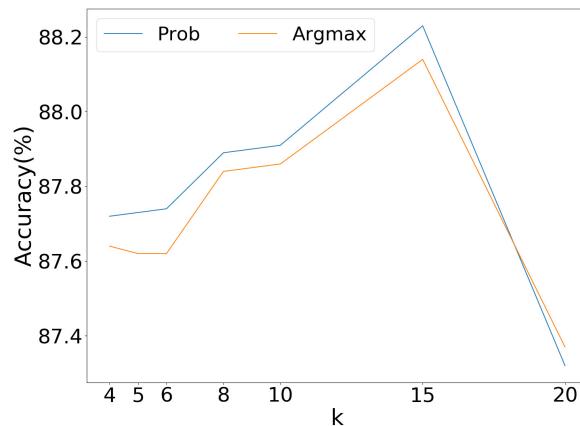
253 There are two criteria for judging neighbor points. As the name K-Nearest Neighbor itself  
254 indicates, the maximum amount of neighbors is  $k$ . Besides that, the absolute distance in 3D space  
255 should also be limited. We down-sample the point could again with a rate of 0.001 to build a small  
256 KD-tree. Then we calculate the average distance of these points, setting the value to be the threshold  
257 of absolute distance. As indicated in Figure 7, the Pixel Accuracy firstly increases with the growth of  
258  $k$ , and reaches its peak with  $k = 15$ . After crossing the peak, accuracy decreases as  $k$  increases. This  
259 is because as  $k$  increases, the local method negatively optimizes for small areas such as vehicles and  
260 narrow roads.

### 261 5.2.2. Soft VS. Hard Decision Strategy

262 The decision strategies based on probability like Bayesian and Markov Decision are soft, while  
263 threshold and argmax layer are hard decision strategies. There is no doubt that hard decision processes  
264 discard some information. As demonstrated in Figure 7, Prob outperforms Argmax under the same  $k$   
265 in most circumstances. The best result of Prob is also greater than Argmax as well. It reveals that the  
266 soft decision strategy leads to better performance.

<sup>267</sup> 5.2.3. Down-Sample Rate

<sup>268</sup> Since the dense point cloud's scale of a specific outdoor scene collected by UAV is usually around  
<sup>269</sup> 20M or bigger, global-wise algorithms cannot handle all points at once. For instance, PKU-M1 contains  
<sup>270</sup> 27M points. Table 3 shows a trend that the Pixel Accuracy generally reaches its peak at a Down-sample  
<sup>271</sup> rate of 1, equivalent to no down sampling at all. Increasing the Down-sample rate makes the filtered  
<sup>272</sup> point cloud denser, which intends the neighbors of a single point to become closer. The closer points  
<sup>273</sup> are, the more likely they belong to the same semantic class. So it is sensible that the increasing of the  
<sup>274</sup> Down-sample rate avails the final Pixel Accuracy. If the performance of a point cloud with lower  
<sup>275</sup> sampling rate is higher than a point cloud, it is reasonable to believe that the former method is better.



**Figure 7.** Ablation study on parameter selection for K-Nearest Neighbor and Soft VS. Hard Decision Strategy. For both Prob and Argmax method,  $k=15$  is the best parameter. In most circumstances, the soft decision strategy Prob dominates hard decision strategy Argmax.

**Table 3.** Ablation study on Down-sample rate.

Method	k-Nearest Neighbor	Down-sample Rate	Pixel Accuracy(%)
2D prediction	0	1	85.66
3D baseline	0	0.1	87.76
Local	15	0.1	88.14
Local	15	0.2	88.02
Local	15	0.5	88.21
<b>Local</b>	<b>15</b>	<b>1</b>	<b>88.24</b>

<sup>276</sup> **6. Conclusions**

<sup>277</sup> In this paper, we propose a semantic 3D reconstruction method to reconstruct 3D semantic  
<sup>278</sup> models by integrating 2D semantic labeling and 3D geometry information. In implementation, we  
<sup>279</sup> utilize deep learning for both 2D segmentation and depth estimation. Then, the semantic 3D point  
<sup>280</sup> cloud is obtained by our probability-based semantic fusion method. Finally, we apply the local and  
<sup>281</sup> global approaches for point cloud refinement. Experimental results shows that our semantic fusing  
<sup>282</sup> procedure with refinement based on local and global information is able to suppress noise and reduce  
<sup>283</sup> the re-projection error. This work paves the way for realizing finer-grained 3D segmentation and  
<sup>284</sup> semantic classifications.

<sup>285</sup> **Author Contributions:** conceptualization, Z.W and Y.W.; methodology, Z.W.; software, Y.W. and Z.W.; validation,  
<sup>286</sup> Y.W. and H.Y.; formal analysis, G.W. and Y.C.; investigation, Y.W and Z.W.; resources, G.W. and Y.C.; data curation,  
<sup>287</sup> Y.W.; writing—original draft preparation, Z.W. and Y.W.; writing—review and editing, H.Y. and Y.C.; visualization,  
<sup>288</sup> Y.W. and Z.W.; supervision, G.W.; project administration, Y.W. and H.Y.; funding acquisition, G.W.

**Funding:** This research was supported by [The National Key Technology Research and Development Program of China] grant number [2017YFB1002705, 2017YFB0203002, and 2017YFB1002601], and [National Natural Science Foundation of China (NSFC)] grant number [61632003, 61661146002, and 61872398], and [Equipment Development Project] grant number [315050501]

**Conflicts of Interest:** The authors declare no conflict of interest.

## References

1. 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.
2. 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**, *39*, 2481–2495.
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* **2017**, *40*, 834–848.
4. Zhao, H.; Shi, J.; Qi, X.; Wang, X.; Jia, J. Pyramid scene parsing network. Proceedings of the IEEE conference on computer vision and pattern recognition, 2017, pp. 2881–2890.
5. Yao, Y.; Luo, Z.; Li, S.; Fang, T.; Quan, L. Mvsnet: Depth inference for unstructured multi-view stereo. Proceedings of the European Conference on Computer Vision (ECCV), 2018, pp. 767–783.
6. Yao, Y.; Luo, Z.; Li, S.; Shen, T.; Fang, T.; Quan, L. Recurrent mvsnet for high-resolution multi-view stereo depth inference. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, pp. 5525–5534.
7. Chen, Y.; Wang, Y.; Lu, P.; Chen, Y.; Wang, G. Large-scale structure from motion with semantic constraints of aerial images. Chinese Conference on Pattern Recognition and Computer Vision (PRCV). Springer, 2018, pp. 347–359.
8. Bao, S.Y.; Savarese, S. Semantic structure from motion. CVPR 2011. IEEE, 2011, pp. 2025–2032.
9. 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.
10. 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). IEEE, 2015, pp. 4867–4873.
11. Häne, C.; Zach, C.; Cohen, A.; Pollefeys, M. Dense semantic 3d reconstruction. *IEEE transactions on pattern analysis and machine intelligence* **2016**, *39*, 1730–1743.
12. Häne, C.; Zach, C.; Cohen, A.; Angst, R.; Pollefeys, M. Joint 3D scene reconstruction and class segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2013, pp. 97–104.
13. Savinov, N.; Ladicky, L.; Häne, C.; Pollefeys, M. Discrete optimization of ray potentials for semantic 3d reconstruction. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2015, pp. 5511–5518.
14. Wu, Z.; Song, S.; Khosla, A.; Yu, F.; Zhang, L.; Tang, X.; Xiao, J. 3d shapenets: A deep representation for volumetric shapes. Proceedings of the IEEE conference on computer vision and pattern recognition, 2015, pp. 1912–1920.
15. Maturana, D.; Scherer, S. Voxnet: A 3d convolutional neural network for real-time object recognition. 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2015, pp. 922–928.
16. Qi, C.R.; Su, H.; Mo, K.; Guibas, L.J. Pointnet: Deep learning on point sets for 3d classification and segmentation. Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2017, pp. 652–660.
17. Qi, C.R.; Yi, L.; Su, H.; Guibas, L.J. Pointnet++: Deep hierarchical feature learning on point sets in a metric space. Advances in neural information processing systems, 2017, pp. 5099–5108.
18. Vineet, V.; Miksik, O.; Lidegaard, M.; Nießner, M.; Golodetz, S.; Prisacariu, V.A.; Kähler, O.; Murray, D.W.; Izadi, S.; Pérez, P.; others. Incremental dense semantic stereo fusion for large-scale semantic scene reconstruction. 2015 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2015, pp. 75–82.

- 340 19. Zhao, C.; Sun, L.; Stolkin, R. A fully end-to-end deep learning approach for real-time simultaneous 3D  
341 reconstruction and material recognition. 2017 18th International Conference on Advanced Robotics (ICAR).  
342 IEEE, 2017, pp. 75–82.
- 343 20. McCormac, J.; Handa, A.; Davison, A.; Leutenegger, S. Semanticfusion: Dense 3d semantic mapping with  
344 convolutional neural networks. 2017 IEEE International Conference on Robotics and automation (ICRA).  
345 IEEE, 2017, pp. 4628–4635.
- 346 21. Li, X.; Wang, D.; Ao, H.; Belaroussi, R.; Gruyer, D. Fast 3D Semantic Mapping in Road Scenes. *Applied  
347 Sciences* **2019**, *9*, 631.
- 348 22. Zhou, Y.; Shen, S.; Hu, Z. Fine-level semantic labeling of large-scale 3d model by active learning. 2018  
349 International Conference on 3D Vision (3DV). IEEE, 2018, pp. 523–532.
- 350 23. Stathopoulou, E.; Remondino, F. Semantic photogrammetry: boosting image-based 3D reconstruction  
351 with semantic labeling. *International Archives of the Photogrammetry, Remote Sensing and Spatial Information  
352 Sciences* **2019**, *42*.
- 353 24. Zhang, R.; Li, G.; Li, M.; Wang, L. Fusion of images and point clouds for the semantic segmentation of  
354 large-scale 3D scenes based on deep learning. *ISPRS journal of photogrammetry and remote sensing* **2018**,  
355 *143*, 85–96.
- 356 25. Schonberger, J.L.; Frahm, J.M. Structure-from-motion revisited. Proceedings of the IEEE Conference on  
357 Computer Vision and Pattern Recognition, 2016, pp. 4104–4113.
- 358 26. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. Proceedings of the IEEE  
359 conference on computer vision and pattern recognition, 2016, pp. 770–778.
- 360 27. Krizhevsky, A.; Sutskever, I.; Hinton, G.E. Imagenet classification with deep convolutional neural networks.  
361 Advances in neural information processing systems, 2012, pp. 1097–1105.
- 362 28. Jensen, R.; Dahl, A.; Vogiatzis, G.; Tola, E.; Aanæs, H. Large scale multi-view stereopsis evaluation.  
363 Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2014, pp. 406–413.
- 364 29. Galliani, S.; Lasinger, K.; Schindler, K. Massively parallel multiview stereopsis by surface normal diffusion.  
365 Proceedings of the IEEE International Conference on Computer Vision, 2015, pp. 873–881.
- 366 30. Sedlacek, D.; Zara, J. Graph cut based point-cloud segmentation for polygonal reconstruction. International  
367 Symposium on Visual Computing. Springer, 2009, pp. 218–227.
- 368 31. Boykov, Y.; Kolmogorov, V. An experimental comparison of min-cut/max-flow algorithms for energy  
369 minimization in vision. *IEEE Transactions on Pattern Analysis & Machine Intelligence* **2004**, pp. 1124–1137.
- 370 32. Abadi, M.; Agarwal, A.; Barham, P.; Brevdo, E.; Chen, Z.; Citro, C.; Corrado, G.S.; Davis, A.; Dean, J.; Devin,  
371 M.; others. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. *Software available  
372 from tensorflow.org* **2015**, *1*.
- 373 33. Bottou, L. Large-scale machine learning with stochastic gradient descent. In *Proceedings of COMPSTAT'2010*;  
374 Springer, 2010; pp. 177–186.

375 **Sample Availability:** Samples of the compounds ..... are available from the authors.

376 © 2020 by the authors. Submitted to *Journal Not Specified* for possible open access  
377 publication under the terms and conditions of the Creative Commons Attribution (CC BY) license  
378 (<http://creativecommons.org/licenses/by/4.0/>).