IOP Conference Series: Materials Science and Engineering

PAPER • OPEN ACCESS

Deep Learning Based Semantic Labelling of 3D Point Cloud in Visual SLAM

To cite this article: Xuxiang Qi *et al* 2018 *IOP Conf. Ser.: Mater. Sci. Eng.* **428** 012023

IOP Conf. Series: Materials Science and Engineering **428** (2018) 012023 doi:10.1088/1757-899X/428/1/012023

Deep Learning Based Semantic Labelling of 3D Point Cloud in Visual SLAM

Xuxiang Qi[1, 2], Shaowu Yang[1, 2, *] and Yuejin Yan[2]

[1]State Key Laboratory of High Performance Computing, National University of Defense Technology, 410073, Changsha, China
[2]College of Computer, National University of Defense Technology, 410073, Changsha, China

* Corresponding author: shaowu.yang@nudt.edu.cn

Abstract. Three-dimensional (3D) point cloud understanding is important for autonomous robots. However, point clouds are normally irregular and discrete. It is challenging to obtain semantic information from them. In this paper, we present a method to build a dense semantic map, which utilizes both two-dimensional (2D) image labels and 3D geometric information. The dense point cloud is built by using a state-of-the-art RGB-D SLAM system. It is further segmented into meaningful clusters using a graph-based method. Then, image keyframes during the SLAM process are used to extract semantic image labels by a convolution neural network (CNN). Finally, these semantic labels are projected to the point cloud clusters to achieve a 3D dense semantic map. The effectiveness of our method is validated on a popular public dataset.

1. Introduction

Scene understanding is crucial to autonomous driving and mobile robotics. Recent research focuses on scene reconstruction to build a 3D sparse or dense map, such as KinectFusion [1], ElasticFusion [2] and DynamicFusion [3]. However, there is no semantic information in those maps, from which robots cannot obtain a semantic-level understanding of surroundings. In fact, recent years have witnessed great progress in 2D image semantic segmentation. With the help of CNN, we can analysis 2D level semantic information in images, such as the work in FCN [4], U-Net [5], SegNet [6], RefineNet [7], PSPNet [8] and DeepLab [9-11]. Meanwhile, 3D point cloud semantic segmentation is a hot issue in computer vision, and it has made great progress in the recent years, including PointNet [12], PointNet++ [13] and PointCNN [14]. However, these approaches merely make use of 3D information to analyse point cloud.

Actually, point cloud can be generated by RGB-D SLAM, e.g. ORB-SLAM2 [15], using cheap RGB-D sensors like Asus Xtion. In RGB-D SLAM, RGB images have rich texture information, and point clouds have geometrical information. In 3D semantic segmentation and mapping, most of existing methods use either RGB images or point clouds as input. However, few methods make use of both 2D and 3D information. In this paper, we introduce a system that fuses 2D and 3D information to build a dense semantic map. Our main contributions of this paper can be summarized as follows:

- An efficient segmentation method for 3D point cloud
- A semantic labelling method for 3D point cloud that fuses both 2D image information and 3D geometric information

IOP Conf. Series: Materials Science and Engineering **428** (2018) 012023 doi:10.1088/1757-899X/428/1/012023

- A 3D dense semantic mapping system

2. Related Work

2.1. Semantic SLAM

Traditional simultaneous localization and mapping (SLAM) systems mainly focus on using low-level geometric features, such as points, lines, and planes, which cannot provide semantic information. Semantic SLAM can give semantic information of environments. It can help robots to understand surrounding scenes in both geometrical and content level. Salas-Moreno et al. proposed the SLAM++ [16], which can perform object detection in RGB-D tracking and mapping. The study by John McCormac et al. on semantic SLAM was SemanticFusion [17]. The method uses a convolution neural network to produce class probability maps, and fuses these predictions into the 3D map. Keisuke Tateno et al. presented a real-time dense monocular CNN-SLAM [18]. With the aid of CNN, CNN-SLAM can perform not only depth prediction, but also semantic segmentation. DA-RNN [19] introduces a new recurrent neural network (RNN) architecture for semantic labelling on RGB-D videos, which utilizes information in multiple viewpoints to improve segmentation performance. Tong [20] combines different SLAM systems facilitated by a scene detection method.

2.2. 2D Object Detection and Semantic Segmentation

An essential component to get semantic information is object detection, which can localize object instances in images. Girshick et al. [21] presented R-CNN, which proposed to apply CNN to object detection. Other similar methods have been proposed in recent years, like Fast R-CNN [22], Faster R-CNN [23], Mask-RCNN [24] and YOLO [25-26]. R-CNN uses selective search algorithm for generating region proposals, which runs very slow. Faster R-CNN replaces the slow selective search algorithm with a fast neural net. Mask R-CNN improves the region of interest (ROI) pooling layer and extends Faster R-CNN to pixel-level image segmentation.

Semantic segmentation is to understand an image at a pixel level, which can label each pixel with a class identity. Similar to object detection, state-of-the-art semantic segmentation approaches also rely on CNN. FCN [4] by Long et al. is the first end-to-end system, which popularizes CNN architecture for semantic segmentation. U-Net [5] is a popular encoder-decoder architecture which can make use of annotated samples more efficiently and have a higher accuracy. SegNet [6] is a similar encoder-decoder architecture. SegNet copies indices from max-pooling for up-sampling, which makes it more memory efficient. RefineNet [7] proposes a method called RefineNet block which fuses both high resolution and low resolution features. It solves the problem of significant decrease in image resolution when we repeat the sub-sampling operation. PSPNet [8] introduces a pyramid pooling method to aggregate the context. DeepLab [9-11] utilizes dilated convolutions to increase the field of view.

2.3. 3D Point Cloud Segmentation and Semantic Analysis

Point cloud segmentation is the process of dividing point clouds into different regions, each of which has similar properties. It is an essential step towards scene understanding from point clouds. Driven by specific applications, like environment modelling in robotics, 3D point cloud segmentation becomes a very active research topic. Point Cloud Library (PCL) [27] is a popular library which provides open-source segmentation algorithms. Early approach [28] uses RANSAC to detect planes from the point clouds, and then it divides objects with Euclidean separation. Region growing [29] algorithm was proposed in 2D image processing work. Later, it was used in the work related to 3D point cloud. Rabbani et al. [30] presented a method for segmentation of point clouds using smoothness constraint, which finds smoothly connected areas in point clouds. Vo et al. [31] introduced a novel octree-based region-growing algorithm for the fast surface patch segmentation of 3D point clouds in urban environments. Stein et al. [32] divided the point cloud into some individual segments using Locally Convex Connected Patches (LCCP) algorithm, which uses normal vector to judge local convexity.

IOP Conf. Series: Materials Science and Engineering **428** (2018) 012023 doi:10.1088/1757-899X/428/1/012023

Golovinskiy et al. [33] proposed a min-cut based method of segmenting objects in point clouds, which can be adapted for both automatic and interactive segmentation.

Different from 2D images, 3D point clouds are irregular and unordered. Thus, the common way in 2D image processing like convolution is ill-suited for them. Recently, deep neural network based methods are proposed for 3D point cloud classification and segmentation, such as PointNet [12] and PointNet++ [13]. The PointNet is able to learn directly from unordered point clouds, which combines local point features and global information to perform 3D segmentation. Based on PointNet, PointNet++ introduces a hierarchical neural network to learn local features with increasing contextual scales, which can learn deep point set features efficiently and robustly. PointCNN [14] presents a novel approach named X-transformation, which can take advantage of CNNs for point cloud processing. However, these methods use only point cloud information, which is difficult to be extended to semantic labelling.

3. Deep Learning Based Semantic labelling in 3D Point Cloud

Our system is based on ORB-SLAM2 [15] and consists of 3 modules: visual SLAM module, point cloud segmentation module and semantic labelling module. We use a keyframe-based update strategy to generate point cloud data. When a keyframe is inserted, we use a modify deep learning framework to process it to obtain object labels. Furthermore, we segment the point cloud with a graph-based method. Finally, we project image labels to the point cloud segments to achieve semantic labelling of the point cloud. The above process is shown in figure 1.

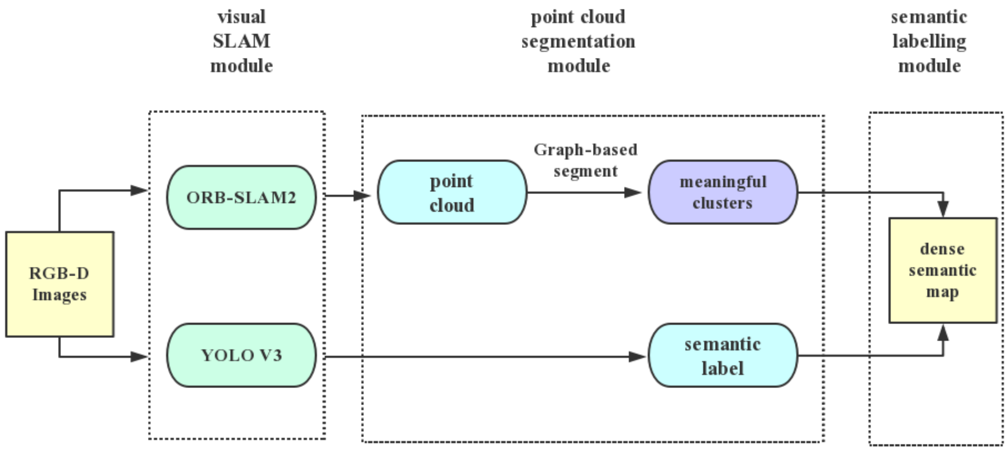


Figure 1. Overview of our semantic 3D mapping system

3.1. The visual SLAM module

The visual SLAM module generates the point cloud data from the RGB-D dataset, which will be further processed by the point cloud segmentation module. The visual SLAM module works in three threads: A tracking thread, a local mapping thread and a loop closing thread.

The tracking thread takes charge of extracting and matching ORB features in gray-scale images converted from raw RGB images for localizing the camera. Besides, it decides when to insert a new keyframe.

The local mapping thread focuses on building the local 3D sparse map. It optimizes both local map and the keyframe poses by performing local bundle adjustment (BA). To generate point cloud data, only keyframes are utilized, while other frames are used to compute camera poses. Point clouds are generated by transforming the 3D points in the depth images from the camera coordinate system to the world coordinate system. We also obtain a rough semantic map in this thread, which will be described in Sect. 3.3.

The loop closing thread detects appearance loops and then corrects the accumulated drifts by pose graph optimization. It is accomplished by using a bag-of-words method among the keyframes.

3.2. The point cloud segmentation module

In the point cloud segmentation module, we only rely on geometric information to segment the point cloud, while the image texture corresponding to the point cloud has no effect on the segmentation result. First, we use the supervoxel method [34] to segment the original point cloud as shown in figure 2. In this way, we can not only reduce the computational cost but also convert point cloud into surface patches according to the similarities of the points. The result of supervoxel can be represented by an adjacency graph $G=\{\nu,\varepsilon\}$ where $\nu_i \in \nu$ are patches and ε connect adjacent patches (ν_i, ν_j).

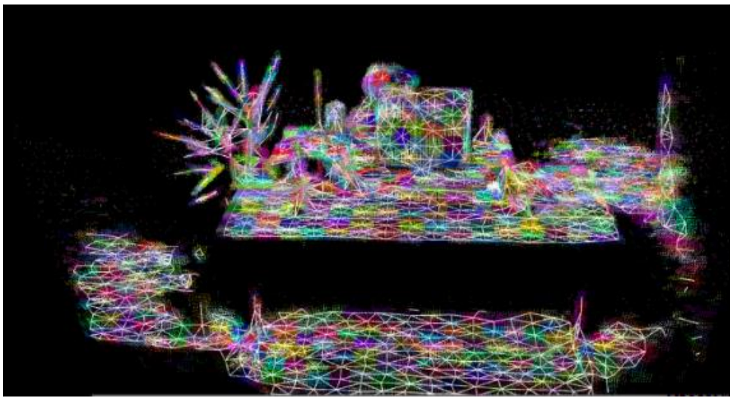


Figure 2. supervoxels of point clouds

After the process of supervoxel, there is a centroid c_i, and a normal vector n_i in each surface patch. The scene segmentation can be framed as a graph partitioning problem. Make it clear in figure 3, nodes are the surface patches, each of node belongs to an object. We are supposed to determine whether edges are ON or OFF.

It is a common way to compute the similarity of nodes by using Euclidean distance [35] with mean shift algorithm [36]. However, the method has a high computational complexity. Normal vector [32] is a reflect of local convexity information, which can be used for clustering. However, it becomes unreliable when there is a higher percentage of noise in point clouds. Thus, we propose to fuse support plane to suppress this negative effect. Supposing that we have K support planes $\{s_1, s_2, ..., s_k\}$ in a point cloud, and surface patches have been obtained in these planes. We define a variable $\{b_i\}_1^N, b_i \in [0, K], b_i = K$ indicates that surface patch belongs to surface plane s_k. Then, we extract planes of all objects and distribute surface patches to them.

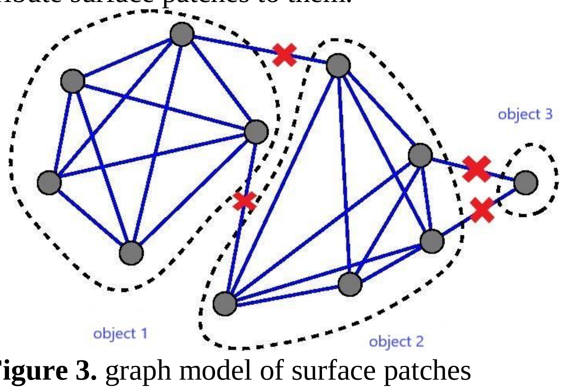


Figure 3. graph model of surface patches

After we get surface patches, we use RANSAC [28] to process patches to generate plane candidates $PC=\{pc_1, pc_2, ...pc_m\}$, and then compute $d(c_i, pc_m)$, which is the distance of the surface patch centroid c_i to plane pc_m. With a threshold δ, we can get all patches within the distance δ to plane pc_m, named $\Pi=\{v_i \in V \mid d(c_i, pc_m) < \delta\}$. Then, we define:

$$D(p_{cm}) = \begin{cases} 1 - \dfrac{\Pi}{\eta} & \Pi < \eta \\ \exp(1 - \dfrac{\eta - \Pi}{\eta}) & \Pi \geq \eta \end{cases} \tag{1}$$

$D(p_{cm})$ are possible planes for objects in the point cloud. In the experiments, we set $\eta = 45$ and $\delta = 4.5cm$. Then the plane extraction problem becomes to minimize the energy formulation:

$$P^* = \arg\min E(P), P \subset PC\,. \tag{2}$$

And our fitting energy formulation is:

$$E(P) = \sum_{pc_m \in E} D(pc_m)\,. \tag{3}$$

After we finish plane extraction, we get planes L and surface patches K .We can directly use the fast graph-based [37] method to perform segmentation.

3.3. The semantic labelling module

Original map generated by ORB-SLAM2 has no semantic information (see figure 4(a)), which is merely a set of irregular and unordered points. We perform semantic labelling in point cloud with the aid of a deep learning framework.

We use a modify YOLO v3 [26] framework to detect objects in keyframes extracted by the SLAM module. Image boundaries of the labels are not accurate enough in 2D images (see figure 4(b)), but we can project 2D semantic labels to 3D point cloud to achieved reasonable 3D semantic mapping, as shown in figure 4(c). However, it is a rough semantic map and not accurate enough to use. As mentioned before, we have already performed point cloud segmentation, then we can fuse segmentation and rough semantic map to improve mapping performance. Original labels come from RGB images, here we make use of both 2D label and 3D segment result to perform semantic segment. Finally, each object region in the point cloud gets a specific semantic label. More details of the process will be further described in Sect. 4.3.

(a) original point cloud

(b) object detection

(c) point cloud with semantic labels, using specific colors

Figure 4. Semantic labelling from 2D image to 3D point cloud

IOP Conf. Series: Materials Science and Engineering **428** (2018) 012023 doi:10.1088/1757-899X/428/1/012023

4. Experiments and Results

We evaluate our system with the popular TUM dataset [38]. Without training the neural network for semantic labelling in TUM dataset, we only focus on validating our 3D semantic mapping method with a few kinds of known objects.

4.1. Experimental setup

The system runs on a laptop with Ubuntu 14.04 64-bit operating system, an Intel Core i5-6300HQ (4 cores @2.3GHZ) CPU, 16G DDR4 RAM and a GTX690M GPU. Our visual SLAM module is able to reach real-time performance. With the help of GPU, YOLO v3 is able to detect images at 8 frames per second.

4.2. Point cloud segment results

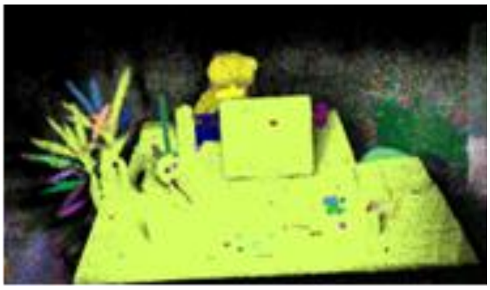
In this experiment, we evaluate our point cloud segmentation method qualitatively by comparing the segmentation results with other existing methods. Locally Convex Connected Patches (LCCP) method only uses local convexity or concavity for segmentation, which ignores global geometrical information.

Lack of global information may lead to wrong segment result, as we can see in figure 5(a), LCCP method is unable to segment small objects such as the keyboard from indoor scene.

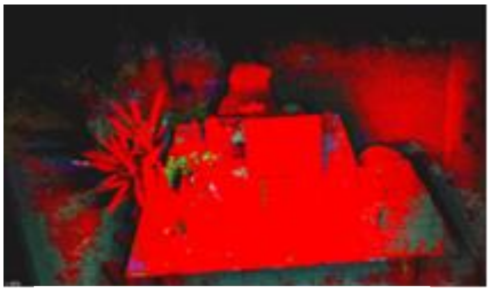
Region Growing method initializes the seed points by their curvature value, and then each region from these points will grow by adding neighbour points. However, the method is sensitive to noise data. In the segment result, we can see that background object wall and foreground object desktop are mixed (see figure 5(b)).

Thus, we use supervoxel method to reduce the time costing, and then make use of both global information such as supporting planes and local information to get a better segmentation performance. We can check whether different methods can segment out objects in the point cloud. As it shows in figure 5(c), we can see that our method is able to segment the majority objects in this indoor scene.

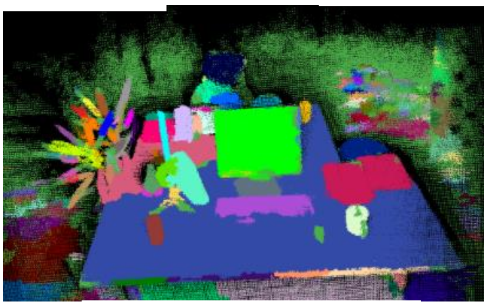
Furthermore, based on an efficient graph-based approach, our segmentation method is faster than LCCP and Region Growing method. In this experiment, our SLAM system uses 82060 RGB images and 82060 depth images to build a 3D point cloud map, which contains 952563 points. Our system is able to finish segmentation within 5 seconds. Time costs of these methods are shown in table 1.



(a) LCCP



(b) Region Growing



(c) Ours

Figure 5. Segment results

IOP Conf. Series: Materials Science and Engineering **428** (2018) 012023 doi:10.1088/1757-899X/428/1/012023

Table 1. Time cost comparison, in milliseconds.

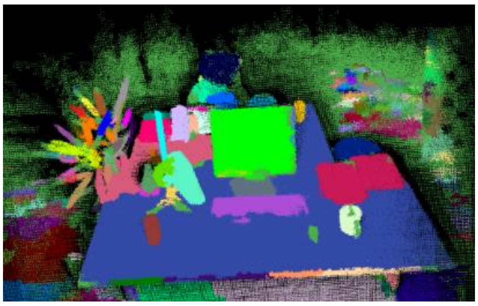
Method	Time-costing
LCCP	7138
Region Growing	13214
Ours	4638

4.3. Semantic segmentation results in the point cloud

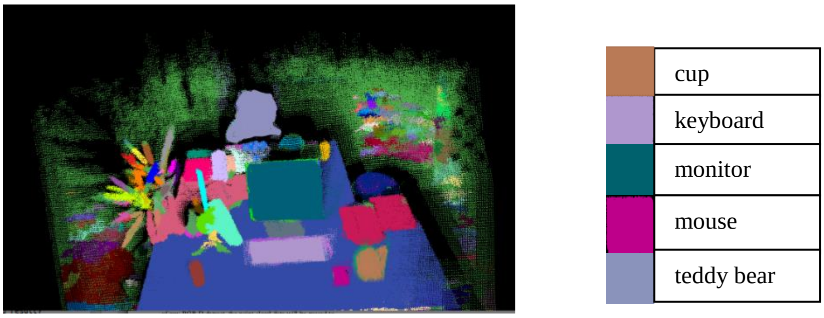
Our visual SLAM module uses YOLO v3 to obtain object labels and project them to point cloud. The point cloud has XYZ and RGB information, which represents Euclidean coordinates and the color data for each point (see figure 6(a)). Besides, the point cloud has semantic information from 2D images. After the point cloud segmentation, we also obtain a segment label for each point (see figure 6(b)). However, this segment label is only a random label without semantic meaning. Then we can fuse two point clouds to combine semantic information with the segment labels, in this way we obtain the final semantic point cloud as shown in figure 6(c).

In fact, two point clouds have the same number of points, and we can use PCL library to fuse them. After fusion, we can not only correct the segment result, but also perform semantic labelling. In this way, we convert 2D image labels to 3D point cloud labels. By the way, our semantic labelling module cannot reach a real-time performance, but it can be a helpful module for a visual SLAM system.

(a) point cloud with semantic label



(b) segment result



(c) semantic map

Figure 6. fuse label and segmentation

YOLO v3 was trained on *coco*[1] dataset which contains 80 classes. However, we only detect several classes in our indoor scene (cup, keyboard, monitor, mouse, teddy bear) as objects of interest. Most of the classes in *coco* dataset are outdoor objects classes, which do not exist in our experiment.

5. Conclusion

In this work, we present a novel approach to combine 2D object labels and 3D point cloud segmentation to achieve 3D semantic mapping in visual SLAM. We propose an effective method of graph-based 3D point cloud segmentation. Another contribution of this paper is that we fuse 2D and 3D information to build a dense semantic map. We share the source code on github[2]. Besides, a video demonstration can be found online[3].

References

[1] Newcombe, Richard A., et al. "KinectFusion: Real-time dense surface mapping and tracking." Mixed and augmented reality (ISMAR), 2011 10th IEEE international symposium on. IEEE, 2011.
[2] Whelan, Thomas, et al. "ElasticFusion: Dense SLAM without a pose graph." Robotics: Science and Systems, 2015.
[3] Newcombe, Richard A., Dieter Fox, and Steven M. Seitz. "Dynamicfusion: Reconstruction and tracking of non-rigid scenes in real-time." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
[4] Long, Jonathan, Evan Shelhamer, and Trevor Darrell. "Fully convolutional networks for semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2015.
[5] Ronneberger, Olaf, Philipp Fischer, and Thomas Brox. "U-net: Convolutional networks for biomedical image segmentation." International Conference on Medical image computing and computer-assisted intervention. Springer, Cham, 2015.
[6] Badrinarayanan, Vijay, Alex Kendall, and Roberto Cipolla. "Segnet: A deep convolutional encoder-decoder architecture for image segmentation." IEEE transactions on pattern analysis and machine intelligence 39.12 (2017): 2481-2495.
[7] Lin, Guosheng, et al. "Refinenet: Multi-path refinement networks for high-resolution semantic segmentation." IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017.
[8] Zhao, Hengshuang, et al. "Pyramid scene parsing network." IEEE Conf. on Computer Vision and Pattern Recognition (CVPR). 2017.
[9] Chen, Liang-Chieh, et al. "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs." IEEE transactions on pattern analysis and machine intelligence 40.4 (2018): 834-848.
[10] Chen, Liang-Chieh, et al. "Rethinking atrous convolution for semantic image segmentation." arXiv preprint arXiv:1706.05587 (2017).
[11] Chen, Liang Chieh, et al. "Encoder-Decoder with Atrous Separable Convolution for Semantic Image Segmentation." (2018).
[12] Qi, Charles R., et al. "Pointnet: Deep learning on point sets for 3d classification and segmentation." Proc. Computer Vision and Pattern Recognition (CVPR), IEEE 1.2 (2017): 4.
[13] Qi, Charles Ruizhongtai, et al. "Pointnet++: Deep hierarchical feature learning on point sets in a metric space." Advances in Neural Information Processing Systems. 2017.
[14] Li, Yangyan, et al. "PointCNN." arXiv preprint arXiv:1801.07791 (2018).
[15] Mur-Artal, Raul, and Juan D. Tardós. "Orb-slam2: An open-source slam system for monocular, stereo, and rgb-d cameras." IEEE Transactions on Robotics 33.5 (2017): 1255-1262.

[1] *http://cocodataset.org*
[2] *https://github.com/qixuxiang/orb-slam2_with_semantic_label*
[3] *http://v.youku.com/v_show/id_XMzYyOTMyODM2OA*

[16] Salas-Moreno, Renato F., et al. "Slam++: Simultaneous localisation and mapping at the level of objects." Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE, 2013.
[17] McCormac, John, et al. "SemanticFusion: Dense 3D semantic mapping with convolutional neural networks." Robotics and Automation (ICRA), 2017 IEEE International Conference on. IEEE, 2017.
[18] Tateno, Keisuke, et al. "CNN-SLAM: Real-time dense monocular SLAM with learned depth prediction." arXiv preprint arXiv:1704.03489 (2017).
[19] Xiang, Yu, and Dieter Fox. "DA-RNN: Semantic mapping with data associated recurrent neural networks." arXiv preprint arXiv:1703.03098 (2017).
[20] Tong, Zhehang, Dianxi Shi, and Shaowu Yang. "SceneSLAM: A SLAM framework combined with scene detection." Robotics and Biomimetics (ROBIO), 2017 IEEE International Conference on. IEEE, 2017.
[21] Girshick, Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." Proceedings of the IEEE conference on computer vision and pattern recognition. 2014.
[22] Girshick, Ross. "Fast r-cnn." arXiv preprint arXiv:1504.08083 (2015).
[23] Ren, Shaoqing, et al. "Faster r-cnn: Towards real-time object detection with region proposal networks." Advances in neural information processing systems. 2015.
[24] He K, Gkioxari G, Dollár P, et al. Mask R-CNN[J]. 2017.
[25] Redmon, Joseph, et al. "You only look once: Unified, real-time object detection." Proceedings of the IEEE conference on computer vision and pattern recognition. 2016.
[26] Redmon, Joseph, and Ali Farhadi. "YOLOv3: An incremental improvement." arXiv preprint arXiv:1804.02767 (2018).
[27] Rusu, Radu Bogdan, and Steve Cousins. "3d is here: Point cloud library (pcl)." Robotics and automation (ICRA), 2011 IEEE International Conference on. IEEE, 2011.
[28] Schnabel, Ruwen, Roland Wahl, and Reinhard Klein. "Efficient RANSAC for point‐cloud shape detection." Computer graphics forum. Vol. 26. No. 2. Blackwell Publishing Ltd, 2007.
[29] Adams, Rolf, and Leanne Bischof. "Seeded region growing." IEEE Transactions on pattern analysis and machine intelligence 16.6 (1994): 641-647.
[30] Rabbani, Tahir, Frank Van Den Heuvel, and George Vosselmann. "Segmentation of point clouds using smoothness constraint." International archives of photogrammetry, remote sensing and spatial information sciences 36.5 (2006): 248-253.
[31] Vo, Anh-Vu, et al. "Octree-based region growing for point cloud segmentation." ISPRS Journal of Photogrammetry and Remote Sensing 104 (2015): 88-100.
[32] Stein, Simon Christoph, et al. "Object Partitioning Using Local Convexity." CVPR. 2014.
[33] Golovinskiy, Aleksey, and Thomas Funkhouser. "Min-cut based segmentation of point clouds." Computer Vision Workshops (ICCV Workshops), 2009 IEEE 12th International Conference on. IEEE, 2009.
[34] Papon, Jeremie, et al. "Voxel cloud connectivity segmentation-supervoxels for point clouds." Computer Vision and Pattern Recognition (CVPR), 2013 IEEE Conference on. IEEE, 2013.
[35] Aldoma, Aitor, et al. "Tutorial: Point cloud library: Three-dimensional object recognition and 6 dof pose estimation." IEEE Robotics & Automation Magazine 19.3 (2012): 80-91.
[36] Comaniciu, Dorin, and Peter Meer. "Mean shift: A robust approach toward feature space analysis." IEEE Transactions on pattern analysis and machine intelligence 24.5 (2002): 603-619.
[37] Boykov, Yuri, Olga Veksler, and Ramin Zabih. "Fast approximate energy minimization via graph cuts." IEEE Transactions on pattern analysis and machine intelligence 23.11 (2001): 1222-1239.
[38] Sturm, Jürgen, et al. "A benchmark for the evaluation of RGB-D SLAM systems." Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on. IEEE, 2012.