Indoor Semantic Mapping with Efficient Convolutional Neural Networks for Resource-constrained SLAM System

Jiahao Zhang1*, Zhiqiang Wang2 and Qing Zhu3

1Information Faculty, Beijing University of Technology, Beijing, 100124, China
2Information Faculty, Beijing University of Technology, Beijing, 100124, China
3Information Faculty, Beijing University of Technology, Beijing, 100124, China
*Corresponding author’s e-mail: zebra_jiahao@163.com

Abstract. Scene parsing and robotic manipulation using visual sensing are of significant importance for mobile and robotics related applications. And increasing efforts are being made to associate semantic concepts to geometric entities in robot’s surroundings. However, they are not well suitable for resource-constrained and mobile platforms. Toward an efficient semantic mapping solution, we propose a specifically light-weight segmentation network for building a dense 3D semantic map in real-time. The propose architecture is based on depth-wise convolution and channel shuffle to reduce computation cost. An improved Atrous Spatial Pyramid Pooling (ASPP) and encoder-decoder structures are presented and multi-scale features are merged for better prediction. Experiments are compared with other methods to prove the effectiveness and superiority.

1. Introduction
Compared with geometric maps, semantic mapping is more rewarding to enhance robot’s autonomy and robustness, facilitate more complex tasks, move from path-planning to task-planning, and enable advanced human-robot interaction. However, the task of including semantic information in SLAM (Simultaneous Localization And Mapping) is still in its infancy, it still lacks a cohesive formulation [1]. And it’s also a challenging research topic for semantic mapping on the resource-constrained and mobile platforms. In order to exploit robotic mapping with semantic sensing ability under a limited computational budget, given by target platform and application scenarios, we combine the dense SLAM system ElasticFusion [2] with a specifically light-weight designed CNN, it also could improve the frame-wise segmentation result.

In this work, our framework consists of two modules, semantic segmentation performed by CNNs and dense annotated 3D map built by SLAM system. And most of our attention is focused on the former due to fact that the general trend of CNNs has been to make deeper and more complicated networks in order to achieve higher accuracy. CNN-based semantic segmentation mainly exploits fully convolutional networks (FCNs) [3], which is an important cornerstone of applying the CNN structure to the field of image semantic segmentation. Later, a variety of advanced networks have been proposed for semantic segmentation. And deeplab series [4], [5] achieve better performance by using dilated convolution to enlarge the receptive filed for dense labelling. In many real-world mobile applications such as drones, robots, self-driving car and augmented reality, they still face the critical issue of yielding real-time performance on a computationally limited platform. This motivates us to...
propose an efficient network architecture, which is based on group convolution, depth-wise convolution and dilated convolution.

In slam, a map is a description of the environment, and the construction of different application scenarios is not the same. For a home sweeping robot, only a 2D map is enough for task. However, in our experiment, we need a dense map by merging data from a mobile sensor. It contains the continuous surfaces that allow precise viewpoint-invariant localization and the potential for detailed semantic scene understanding. ElasticFusion could be able to use surface prediction every frame for true incremental simultaneous localization and dense mapping due to its typical dense frontend, which has proven to be highly efficient technique capable of running in real-time even on resource constrained mobile devices. And it applies surface loop closure optimizations early and often in order to stay near to the mode of the map distribution. The surface representation is particularly suitable for fusing semantic labels into the dense map in viewpoint.

We evaluate the performance of our systems on the NYUv2 dataset [6], which could run semantic segmentation in real-time, so that it can perform well even under fast camera motions. And the size of our CNN model is only 2.8MB, compared with the DeconvNet of SemanticFusion [7] and PSPNet [8], respectively, 1007MB and 197MB. In particular, our approach is more suitable for resource-constrained and mobile platforms.

2. Related Work

2.1. Semantic slam

At present, the research of semantic slam includes three main stages: (i) semantic mapping, which maps the results of semantic segmentation to dense maps, (ii) the semantic information is to help slam; and, (iii) slam can also improve the level of semantic understanding. In terms of semantic mapping, Semantic Fusion proposed by John McCormac et al., in which the convolutional neural network predicts object class labels, Elastic Fusion provides pose estimation, and finally combines Bayesian update and CRF to achieve probabilistic multiplication of predicted values from multiple perspectives. In the case of semantics helping SLAM, Bowman et al. [9], [10] introduced EM estimation to convert semantic SLAM into probability problem. The result of object detection is used as the input of the front end of SLAM, complementing the features of ORB to improve the robustness of localization. In addition to using semantic information to improve accuracy of localization, it can also be used for bundle adjustment. Salehi et al. [11] used neural networks to infer the semantic and structural information of the environment, and used a Bayesian framework to inject the results into a bundle adjustment process that constrains the 3d point cloud to texture-less 2d build models. In the situation of SLAM helping semantics, in addition to using the geometric consistency between the images obtained in the SLAM system to promote image semantic segmentation, SLAM can also help to construct large-scale datasets with corresponding relationships between images, which can reduce the difficulty of marking deep learning datasets. Kerl et al. [12] obtained the camera trajectory using RGB-D SLAM and warped the predictions of RGB-D images into ground-truth annotated frames in order to enforce multi-view consistency during training.

2.2. Semantic segmentation

In the past few years, convolutional neural networks have great advantages in improving the accuracy of image semantic segmentation. Later, there has rising interest in building light-weight and efficient neural networks. In the networks that achieve high accuracy, DeepLab series used dilated convolution to enlarge the receptive field of neural networks and multi-scale feature extraction for global and local features and contexts. PSPNet exploited the capability of global context information by different-region-based context aggregation via pyramid scene parsing network. In terms of real-time, ENet [13] focused on making efficient use of scarce resources available on embedded platforms by designing a
novel neural network architecture. For efficiency problems, in addition to the direct design of the small networks mentioned above, the usual method is to perform model compression, which is to compress on the already trained model. Compared to the processing on the already trained model, we choose a more efficient network calculation method for convolution, so that the network parameters are reduced without losing network performance. For example, MobileNet [14] used depth-wise separable convolutions to build light weight deep neural networks. ShuffleNet [15], [16] utilized pointwise group convolution and channel shuffle to greatly reduce computation cost while maintaining accuracy.

3. Method
Our semantic mapping pipeline mainly consists of two components: an efficient convolutional neural network, as illustrated in Figure 1, and a real-time SLAM system ElasticFusion. The input RGB images and depth frames can be obtained from a mobile RGB-D sensor. The light-weight networks, which is specifically designed by depth-wise separable convolutions, receives RGB images to compute the class probability of per pixel. Simultaneously, the SLAM system receives RGB-D images to build a comprehensive dense globally consistent surfel-based map of room scale environments, without pose graph optimization or any post-processing steps. Finally, class probabilities assigned to each surfel by recursive Bayesian update in order to combine semantic segmentation information with dense visual map. The following sections discuss these components in more detail.

3.1. CNN architecture
We focus on solving the efficiency limitation that is essentially present in the semantic segmentation component of this system. To balance performance and efficiency given resource-constrained and mobile platforms, our architecture, ShuffleSegNet combines concepts from ShuffleNet and DeepLab. As shown in Figure 1, the ShuffleSegNet utilizes 15 ShuffleNetV2 units, which used depthwise convolution and channel shuffle to extract features and greatly reduce computation cost, but with the addition of improved Atrous Spatial Pyramid Pooling (improved ASPP) module which are designed for the problem of segmenting objects at multiple scales.

Depthwise separable convolution factorizes a standard convolution into a depthwise convolution followed by a pointwise convolution, greatly reducing computational complexity. However, the pointwise convolution accounts for most of the computational complexity, which is especially disadvantageous for lightweight models, so we adopt a channel shuffle operation to address the issue. But ShuffleV2 used a fast downsampling design because the high resolution of the image introduces a large read and write overhead in the scene of fast network computing. For our semantic segmentation task, this design it not suitable. The resolution of the image is very important for semantic segmentation. We appropriately refine the network backbone to reduce the number of downsamping, and take into account the computational efficiency, using more structural units to calculate low resolution images.
To obtain detailed information related to object boundaries, we propose a novel encoder-decoder structure, as shown in Figure 2. The encoder features are first bilinearly upsampled by a factor of 8 and then concatenated with the corresponding low-level features from 3 ShuffleV2 units that have the same spatial resolution by upsampling individually. Before the concat operation we apply $1 \times 1$ convolutions to reduce the number of channels.

On the other hand, the spatial pyramid pooling is effective to resample features at different scales for accurately and efficiently regions of an arbitrary scale. And the depthwise convolution also could be used in the ASPP module by replacing the original general convolution with it, further reducing the computation cost. Our improved ASPP consists of five $1 \times 1$ convolutions, three $3 \times 3$ depthwise convolutions with rates = (9,13,17), the image-level features, and a shortcut connection, as shown in Figure 3. Specifically, the different atrous rate convolution use the same information from one $1 \times 1$ convolution, rather than each one has its own $1 \times 1$ convolution.

3.2. SLAM mapping

ElasticFusion is suitable for the SLAM component of our system. For the $t$-th incoming RGB-D frames, the camera pose is estimated by ICP algorithm and Direct Method to perform point clouds registration. Then using a space deformation approach to ensure local and global loop surface consistency in the map. After the surfel is merged into the global model, the point that can be seen in the current perspective is obtained through the OpenGL projection, and is used to register the image of the next frame.

It is worth noting that camera tracking combines geometric and photometric pose estimation. We aim to find the motion parameters $\xi$ that minimize the joint cost $E_{\text{track}}$ form point-to-plane error $E_{\text{icp}}$ and photometric error $E_{\text{rgb}}$:

$$E_{\text{icp}} = \sum_k \left( (v^k - \exp(\hat{\xi}) T v^k) \cdot n^k \right)^2,$$

where $v^k$ is the back-projection of the k-th vertex in latest depth frame $D_t^k$, $n^k$ is the corresponding of $v^k$. $T$ is the transform matrix and $\exp(\hat{\xi})$ is the exponential map. The photometric pose estimation minimize the cost the intensity difference between the current colour image $C_t^a$ and the last frame $C_{t-1}^a$:

$$E_{\text{rgb}} = \sum_{u \in \Omega} \left( I(u,C_t^a) - I(\pi(K \exp(\xi) T p(u,D_t^k)),C_{t-1}^a) \right)^2,$$

$$E_{\text{icp}} = E_{\text{icp}} + \omega_{\text{rgb}} E_{\text{rgb}},$$

To optimise the joint cost we use the Gauss-Newton non-linear least-squares method with a three level coarse-to-fine pyramid scheme.

4. Experiments

4.1. Dataset and network training
After training our ShuffleSegNet with the PASCAL VOC2012 data for 20K iterations, we finetuned the network with the training dataset of the NYUv2 dataset for 50K iterations. We use a mini-batch of 3 images and the experiments were implemented on a laptop with an Intel Core i5-8300H 2.3GHz CPU and an Nvidia GTX1060 GPU. We use the poly learning rate policy and set base learning rate to 0.0001 and power to 0.9. Momentum and weight decay are set to 0.9 and 0.0005 respectively.

4.2. Run-time performance and accuracy
In this section, we experimentally demonstrate the run-time performance of our method by quantitatively comparing the time, FLOPs, parameters and model size through Table 1. Regarding to the efficiency, the model size of ShuffleSegNet is much smaller than DeconvNet, and it is nearly twice as fast as itself without decode module. As shown in Table 2, although the decode module can improve the accuracy to 66.0% average class accuracy, which achieves 0.5% higher compared to itself without decode module, we can see that using the decode module will increase the complexity of the image semantic segmentation, and the speed will be slower. We can choose to use the decode module according to the requirements of the task and resource allocation. Additionally, figure 5 shows qualitative results of our dense semantic mapping.

| Method                  | Time (ms) | FLOPs (G) | Para (M) | Model size (MB) |
|-------------------------|-----------|-----------|----------|-----------------|
| DeconvNet [7]           | 81.16     | 28.81     | 480.2    | 1007            |
| Ours without decoder    | 44.27     | 1.69      | 0.6      | 2.8             |
| Ours                    | 83.79     | 2.45      | 0.8      | 3.6             |

| Method                  | Class avg. |
|-------------------------|------------|
| RGBD-SF [7]             | 58.9       |
| RGBD-SF-CRF [7]         | 59.2       |
| Eigen-SF [7]            | 63.2       |
| Eigen-SF-CRF [7]        | 63.6       |
| Ours without decoder    | 65.5       |
| Ours                    | 66.0       |

As we can see the first scene in Figure 4, our network can recognize the light switch on the wall, and the decode module can get more detailed edge information. Similarly, in the second scenario our approach has a more regular edge. The third scenario shows that after a large number of rgb images are missing, decode module can also get some information (the TV category is displayed in yellow). Overall, our ShuffleSegNet can have a small amount of parameters, it also can achieve better effect than DeconvNet.
5. Conclusion

In this paper, we proposed an efficient and light-weight semantic mapping network for mobile resource-constrained semantic SLAM system. Through our experiments, our ShuffleSegNet achieves trade-off on NYUv2 dataset in terms of segmentation and implementing efficiency. In the case of sufficient resources, the decode module can be used to improve the accuracy of the image segmentation. On the other hand, we could just combine the ShuffleV2 units and the improved ASPP to obtain semantic information in the case of resource-constrained. The semantic slam we built in the article is just a simple form. The semantic information is not only reflected on the label, but also needs to be able to perform some constraints with the SLAM to improve the accuracy of the SLAM. In the future work, we will focus more on the integration of SLAM modules and semantic information.

References

[1] Cadena, C., Carlone, L., Carrillo, H., Latif, Y., Scaramuzza, D., & Neira, J., et al. (2016) Simultaneous localization and mapping: present, future, and the robust-perception age. IEEE Transactions on Robotics, 32:1309-1332.

[2] Whelan, Thomas, Leutenegger, Stefan, Glocker, & Ben, et al. (2016) Elasticfusion: real-time dense slam and light source estimation. International Journal of Robotics Research, 35:1697-1716.

[3] Ren, Shaoqing, He, Kaiming, Girshick, Ross, & Sun, Jian. (2015) Faster r-cnn: towards real-time object detection with region proposal networks. IEEE Transactions on Pattern Analysis & Machine Intelligence, 39:1137-1149.

[4] Chen, Liang-Chieh, Papandreou, George, Kokkinos, Iasonas, Murphy, Kevin, & Yuille, Alan L. (2016) Deeplab: semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs. IEEE Transactions on Pattern Analysis & Machine Intelligence, 40:834-848.

[5] Chen, L. C., Papandreou, G., Schroff, F., & Adam, H. (2017) Rethinking atrous convolution for semantic image segmentation. arXiv preprint arXiv:1706.05587.
[6] Silberman, N., Hoiem, D., Kohli, P., & Fergus, R. (2012). Indoor Segmentation and Support Inference from RGBD Images. In: European Conference on Computer Vision (ECCV). Springer, pp. 746-760.

[7] McCormac, J., Handa, A., Davison, A., & Leutenegger, S. (2017) Semanticfusion: dense 3d semantic mapping with convolutional neural networks. In: International Conference on Robotics and automation (ICRA). Singapore. pp. 4628-4635.

[8] Zhao, Hengshuang, Shi, Jianping, Qi, Xiaojian, Wang, Xiaogang, & Jia, Jiaya. (2017) Pyramid scene parsing network. In: Proceedings of the IEEE conference on computer vision and pattern recognition. (CVPR) Puerto. pp. 2881-2890.

[9] Atanasov, N., Zhu, M., Daniilidis, K., & Pappas, G. J. (2016) Localization from semantic observations via the matrix permanent. The International Journal of Robotics Research, 35: 73-99.

[10] Bowman, S. L., Atanasov, N., Daniilidis, K., & Pappas, G. J. (2017) Probabilistic data association for semantic SLAM. 2017 IEEE International Conference on Robotics and Automation (ICRA). Singapore. pp. 1722-172.

[11] Salehi, A., Gay-Bellile, V., Bourgeois, S., Allezard, N., & Chausse, F. (2017) Large-scale, drift-free slam using highly robustified building model constraints. Vancouver. pp. 1586-1593.

[12] Ma, L., Stückler, Jörg, Kerl, C., & Cremers, D. (2017) Multi-view deep learning for consistent semantic mapping with rgb-d cameras. Vancouver. pp. 598-605.

[13] Paszke, A., Chaurasia, A., Kim, S., & Culurciello, E. (2016) Enet: a deep neural network architecture for real-time semantic segmentation. arXiv preprint arXiv:1606.02147.

[14] Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., & Weyand, T., et al. (2017) Mobilenets: efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.

[15] Zhang, X., Zhou, X., Lin, M., & Sun, J. (2017) Shufflenet: an extremely efficient convolutional neural network for mobile devices. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR). Puerto. pp. 6848-6856.

[16] Ma, N., Zhang, X., Zheng, H. T., & Sun, J. (2018) Shufflenet v2: practical guidelines for efficient cnn architecture design. In: Proceedings of the European Conference on Computer Vision (ECCV). Munich. pp.116-131.