Published 09/23/2017

doi:10.4271/2017-01-1969

3D Scene Reconstruction with Sparse LiDAR Data and Monocular Image in Single Frame

Yuanxin Zhong, Sijia Wang, Shichao Xie, Zhong Cao, Kun Jiang, and Diange Yang
Tsinghua University

ABSTRACT

Real-time reconstruction of 3D environment attributed with semantic information is significant for a variety of applications, such as obstacle detection, traffic scene comprehension and autonomous navigation. The current approaches to achieve it are mainly using stereo vision, Structure from Motion (SfM) or mobile LiDAR sensors. Each of these approaches has its own limitation, stereo vision has high computational cost, SfM needs accurate calibration between a sequences of images, and the onboard LiDAR sensor can only provide sparse points without color information. This paper describes a novel method for traffic scene semantic segmentation by combining sparse LiDAR point cloud (e.g. from Velodyne scans), with monocular color image. The key novelty of the method is the semantic coupling of stereoscopic point cloud with color lattice from camera image labelled through a Convolutional Neural Network (CNN). The presented method comprises three main process: (I) perform semantic segmentation on color image from monocular camera by using CNN, (II) extract ideal surfaces and other structural information from point cloud, (III) improve the image segmentation with the extracts and label the point cloud with the image segments. The whole process is done in a single frame, and the output of the system is labelled point cloud which can be used in construction of semantic object convex and alignment between frames. We demonstrate the effectiveness of our system on the KITTI dataset providing sufficient camera and LiDAR data, and present qualitative and quantitative results indicating the improvements in segmentation comparing to methods merely using either image or LiDAR data.

CITATION: Zhong, Y., Wang, S., Xie, S., Cao, Z. et al., "3D Scene Reconstruction with Sparse LiDAR Data and Monocular Image in Single Frame," *SAE Int. J. Passeng. Cars – Electron. Electr. Syst.* 11(1):2018, doi:10.4271/2017-01-1969.

INTRODUCTION

Scene understanding and reconstruction has been a difficult but popular research topic in recent years. Autonomous vehicle makes decisions based on how it understands its surroundings. Semantic segmentation is one of effective methods to achieve it, informing the vehicle with specific categorical information. Segmentation with rich semantic information even enables vehicle to predict scenes in the next period. For decades, most of current methods are using computer vision techniques based on images captured by color or gray cameras, for that images contain rich visual patterns and they are harmonized with human intuition. However, the more researches are carried out, more clearly it is that the scene perception and comprehension of human vision system are of high complexity with the integration of motion, hearing and other sensations. Therefore, fusion of different sensors contributes to efficient perceptual process.

Sensors with 3D information, such as LiDAR, stereo and depth camera, have emerged as significant data sources of real time scene reconstruction. The depth information collected from the sensors is a reliable object discriminator being able to divide a scene into several disjoint partitions, and therefore decreases the difficulty of automated situation understanding. There exist methods that generate spatial information from flat data (usually referred to image), for example SfM and optical flow, yet these methods along with camera can hardly provide precise long-range depth information in out-door condition while they are relatively low-cost. LiDAR, among these sensors and methods, provides rich and accurate point cloud in a moment. There are two main categories of LiDAR, airborne and terrestrial, the latter of which is available for autonomous vehicle. Nevertheless, point cloud lacks color and texture information even if it has been colored with multi-view technology, due to its space discreteness. Despite of the intensity information provided by LiDAR, sparse point cloud from LiDAR, which covers a distance of less than 100m, could hardly contain enough features, while the pattern constructed from color and texture has been proved to be effective in detailed semantic classification. In Addition, LiDAR can cover a distance of only about 100m from its center, while one can see thought over a kilometer with cameras in a good visibility condition.

In this work, taking the advantage of the Velodyne on-board LiDAR and monocular camera, we propose a method of fusing them in a single frame which makes it possible to get semantic segments in real time. The input is a point cloud collected by LiDAR and a colored image captured continuously by monocular camera whose location is fixed with the LiDAR, and the output is a semantically segmented point cloud, every point of which has been marked with a semantic label. Data source in this paper is fetched from the KITTI Vision Benchmark Suite [1] (details will be shown in section IV). The structure of the system can be described as following parts (as Figure 1 shows): 1) using CNN, one of deep learning methods, to execute

semantic segmentation in monocular colored image, 2) extract point clusters from the point cloud based on shape features, 3) fusing the image segments and spatial clusters by point projection and back-projection, and rectify the incorrect labels (especially near object edge) with a hierarchical voting algorithm.

The main contributions of this paper can be concluded into two parts: First, CNN is used in LiDAR-camera fusion point cloud labeling for the first time to the best of our knowledge. Neural network gives an excellent segmentation result of a single-frame image, and in this paper, the excellence is brought to the sparse point cloud. Second, a novel algorithm for hierarchical voting is proposed in semantic fusion of segments. The proposed algorithm will improve the reservation of semantic details and ensure the object integrity. The remainder of this paper will describe the procedures of segmentation, aligning and augmentation. Concluding remarks are presented in the last section.

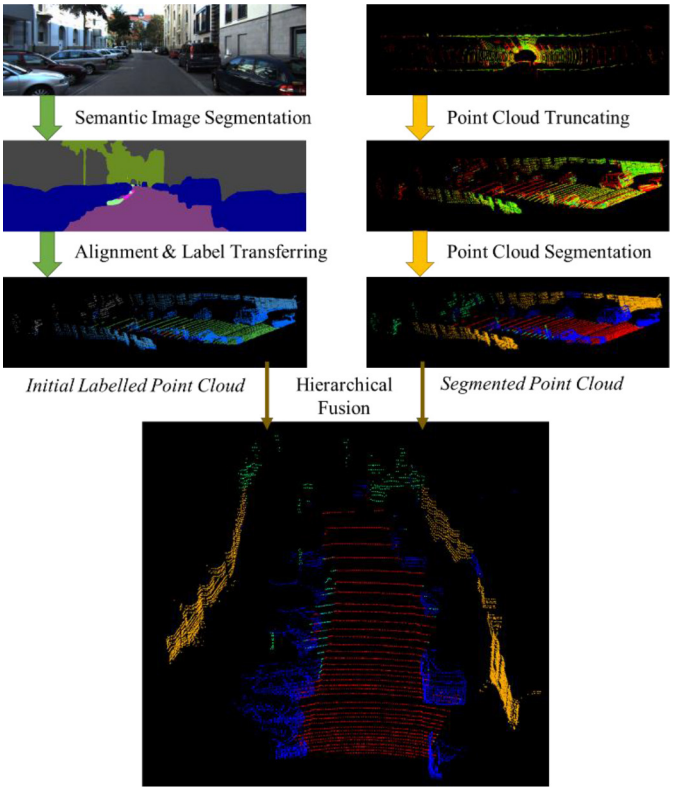


Figure 1. Pipeline overview of the method

RELATED WORK

A large amount of literature have been working on segmentation of point cloud. The classical method of segmentation is using feature of primitive shapes as done in [2, 3, 4]. The segments are simple and unlabeled. Barnea et al. [5] use shape features in segmentation of panoramic point cloud. Liu et al. [6] segment point cloud in residential area using primitive feature of houses. Another way to do segmentation is to represent 3D information by range image, and use methods in image segmentation. Zhu et al. [7] use SVM to segment range image captured by L4. Chen et al. [8] project point cloud into range image and segment the image using random forest. These methods can sometimes perform well with dense 3D information, but it is hard to implement them on sparse LiDAR point cloud.

To natively semantically segment point cloud, machine learning methods have been used in sparse point cloud. Valentin et al. [9] use conditional random field in segmentation of 3D mesh constructed from point cloud. Habermann et al. [10] use artificial neural network directly to do segmentation in point cloud, however it is done with a tiny label class. 3D convolutional network is implemented by Maturana et al. [11] to detect safe landing zone in a planar area. These methods use features directly from sparse point clouds which lack of dense information.

In recent works, many researchers have tried to combine image with sparse point cloud to improve segmentation. Zhao et al. [12] use fuzzy logic depending on priori to fuse segments generated by markov random field and point cloud. Stereo images are used to generate both semantic segments and point cloud in the paper [13] by Kundu et al., and then fused into labeled 3D segments.

SEGMENTS GENERATING

In this paper, we do segmentation both in images and point clouds. In order to unify the description, we treat the clusters generated from point cloud as 3D segments, for that they are also parts of the entire data. Image and point cloud are both fine segmented and over-segmentation is not avoided for that it does not worsen the result. After the segments are generated, a semantic fusion is carried out between corresponding segments.

Image Semantic Segmentation

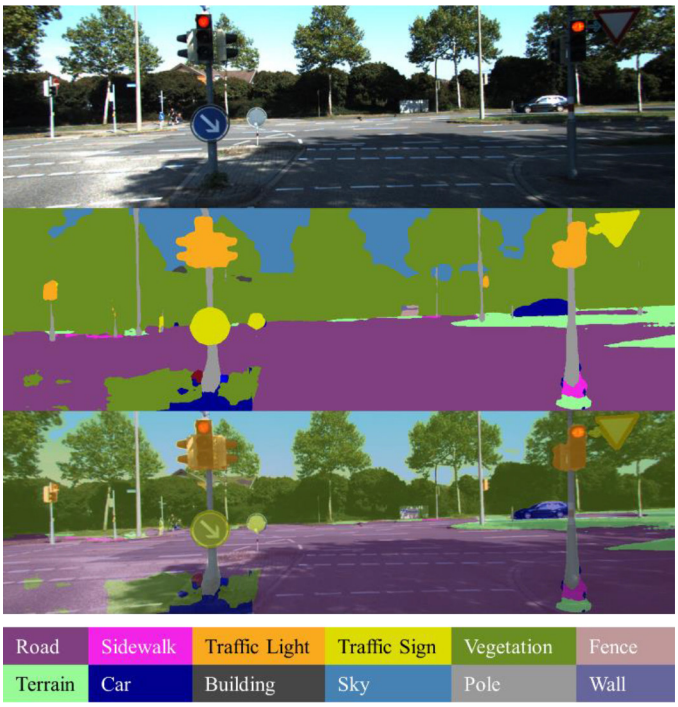


Figure 2. Example of image semantic segmentation with PSPNet. The top image is the original image, one in the middle is the segmented image with segments colored by labels, the bottom one is a combined image showing the locations of segments in the image.

Scene parsing based on semantic segmentation is such a fundamental topic in computer vision that a considerable amount of works have been carried out on it. With the rapid development of neural network, many segmentation frameworks based on it have revealed their outstanding learning ability. In our study, image segmentation is the base step to generate semantic labels, so we utilize the Pyramid Scene Parsing Network (PSPNet) [14], one of the state-of-art image segmentation frameworks, to carry out reliable segmentation on a large semantic label class. PSPNet first introduced Pyramid Pooling Module as global contextual prior to gather context information, it is able to collect several levels of information with unapparent increase in computation cost. PSPNet tested on Cityscape dataset [15] shows an excellent performance with a Jaccard Index of 80.2. The net weights used in out method were pre-trained on Cityscape dataset and we did not fine-tuned them on KITTI dataset, for that the scenes in both datasets are similar in urban surroundings and weather condition. However, the cameras used in the two dataset are different in luminance and vibrance which make the segmentation results in KITTI not as well as ones in Cityscape. Except for that, the main drawback of the network is that it does not work well with the edges of object, which will cause some troubles when the result is used under extreme conditions. Yet, our method is able to fix the major faults in segmentation.

Point Cloud Segmentation

Segmentation of point cloud has been studied in many previous works. However, within the range of our knowledge, there is no satisfying and simple method that can natively segment a single-frame sparse LiDAR point cloud without destroying the semantic integrity of objects. The segmentation in this paper is mainly implemented on Point Cloud Library (PCL) platform [16], and like many other methods, the whole process is divided into several stages.

Down-Sampling

The point cloud collected by rotating laser scanner (e.g. Velodyne) is dense near the collector in circumferential direction, while cloud far from the collector is sparse. So at first, down-sampling is carried out with a voxel grid filter to reduce computation burden while features of objects are not eliminated and the performance of segmentation does not decline. The points in the same voxel grid are replaced by the centroid of these points. Size of each voxel grid is 0.1×0.1×0.01m. The height of the grid is smaller because the cloud is relatively dense in Z-direction, and roughness of the ground is a key feature that drivers care about. Down-sampling is effective near the center of the rotation scanner (as Figure 3 shows), and usually the compression ratio can be as small as 70%.

Large Planes Extraction

In reality, most objects lie on planes or relatively flat surfaces (e.g. the road, tables, platforms), so when the support surfaces are removed from point cloud, objects are isolated from each other which makes the clustering in next stage easier to carry out. The planes are iteratively extracted using RANSAC estimator, until the proportion of points remaining is under certain threshold (here we take the threshold as 0.3). The planes extracted can usually be considered as complete segments, while there are cases that a plane contains different parts such as road and grassland, or the plane is even actually a collection of disconnected points. However, this step ensures that the ground has been removed in this stage.

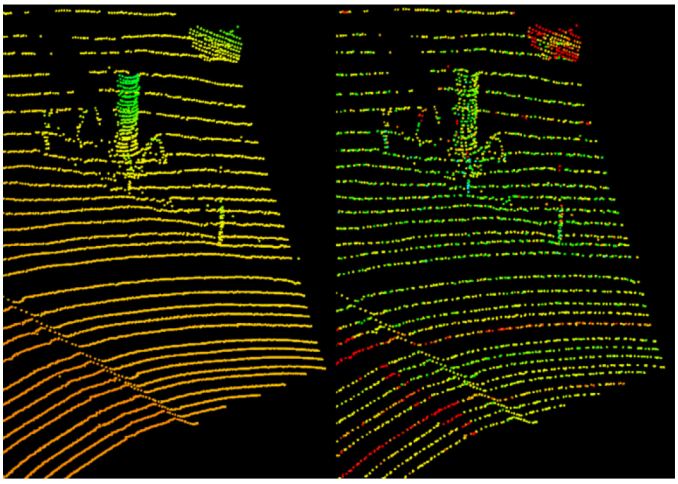


Figure 3. Down-sampling process. The left image shows a part of the original point cloud with 19056 points, the right image shows the same part of the down-sampled point cloud with 13477 points.

Object-Based Clustering

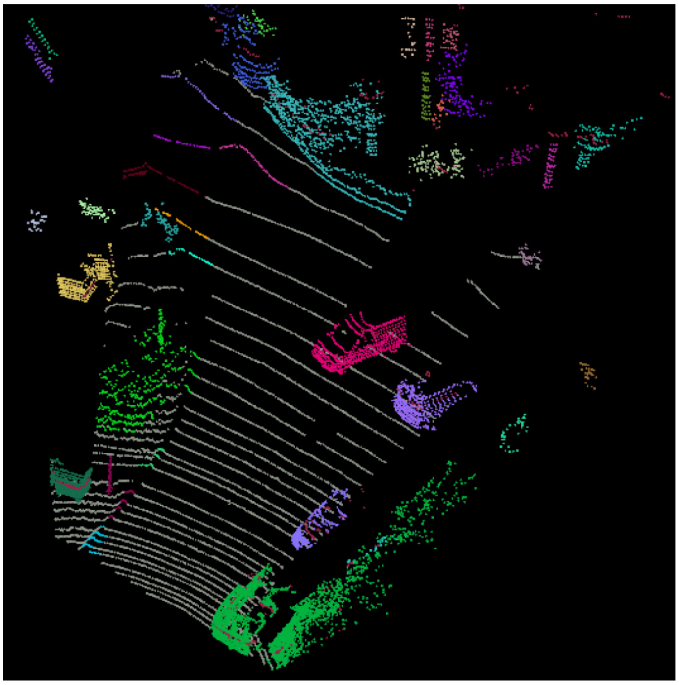


Figure 4. An example result of Euclidean clustering. Points with different colors belong to different clusters. There are 2 planes and 43 clusters in the point cloud. Most objects are separated after clustering.

After the two stages above, the objects can be clustered using simple strategy. Frequently-used methods on point cloud clustering are Euclidean clustering and region growing. The latter method need normal and curvature of a point calculated in order to decide whether the point is a seed. The former one is used in this paper (as shown in Algorithm 1) and other non-trivial methods [17, 18] are not implemented here, for the Euclidean clustering performs well enough

in the plane-removed point cloud and its computational cost is less. The minimum count of points in a cluster is set to 20 to get a fine segmentation result.

Input: P: set of points in point cloud; d_{thres}: distance threshold
Output: $C \leftarrow \emptyset$: set of point cloud clusters;
  $Q \leftarrow \emptyset$: queue for unprocessed points
  for each $p_i \in P$ **do**
    $Q \leftarrow Q \cup P$
    for each $p_j \in Q$ **do**
      $P_j^N \leftarrow$ point neighbors of p_j in a sphere with radius $r < d_{thres}$
      for each $p_k \in P_j^N$ **do**
        if p_k has not been processed **then**
          $Q \leftarrow Q \cup \{p_k\}$
        end if
      end for
    end for
    if all points in Q has been processed **then**
      $C \leftarrow C \cup \{Q\}$
      Q $\leftarrow \emptyset$
    end if
  end for

Algorithm 1. Euclidean Clustering

In our experiments, distance threshold is set to 0.5m by default. The threshold is chosen by priori experience, and it will be more effective if the threshold is chosen according to the distance from the cluster to the origin, since farther from the origin the points are, sparser the cluster will be, and many points in an object will be seen as isolated when they are extremely sparse.

JOINT FEATURE AUGMENTATION

Alignment between Point Cloud and Flat Image

In our method, registration between point cloud and image is accomplished by using extrinsic parameters of LiDAR and video camera. In the KITTI dataset, the parameters are generated from joint calibration of LiDAR and camera using calibration chess boards. Through them, the points in point cloud can be projected into camera coordinate system and fetch labels from the image through the projection. The projection is performed between a point cloud and an image with the same timestamp ensuring time synchronization.

Suppose $p_l = (x_l, y_l, z_l)^T$ is a point in a point cloud frame from laser scanner, then the projected point $p_c = (x_c, y_c, z_c)^T$ in camera coordinate system corresponding to it can be calculated by:

$$p_c = \boldsymbol{P} \cdot \boldsymbol{R}_{0rect} \cdot (\boldsymbol{R} \cdot p_l + \boldsymbol{T}) + \boldsymbol{T}_c \tag{1}$$

or using homogeneous form $p_l' = (x_l, y_l, z_l, 1)^T$, the formula is transformed into following one:

$$p_c = [\boldsymbol{P} \quad \boldsymbol{T}_c]_{3\times4} \cdot \begin{bmatrix} \boldsymbol{R}_{0rect} & \boldsymbol{0} \\ \boldsymbol{0} & \boldsymbol{1} \end{bmatrix}_{4\times4} \cdot \begin{bmatrix} \boldsymbol{R} & \boldsymbol{T} \\ \boldsymbol{0} & \boldsymbol{1} \end{bmatrix}_{4\times4} \cdot p_l' \tag{2}$$

where

$\boldsymbol{P}$ = projection matrix of the camera (related to intrinsic parameters)

$\boldsymbol{T}_c$ = transition matrix from selected camera to center camera

$\boldsymbol{R}_{0rect}$ = rectifying rotation matrix of the 0th camera

$\boldsymbol{R}$ = rotation matrix from point cloud to camera coordinate system

$\boldsymbol{T}$ = transition matrix from point cloud to camera coordinate system
Shapes of the matrices are $\boldsymbol{P}$:3×3, $\boldsymbol{T}_c$:3×1, $\boldsymbol{R}_{0rect}$:1×1, $\boldsymbol{R}$:3×3, $\boldsymbol{T}$:3×1

In KITTI dataset, the origin of camera coordinate system is located at center of 0th camera. All the matrices mentioned in fomula (1) and (2) are provided in calibration files from the dataset. Matrices $\boldsymbol{P}$, $\boldsymbol{T}_c$ and $\boldsymbol{R}$, $\boldsymbol{T}$ are given in compound form.

Noting that the view range of camera does not cover that of LiDAR point cloud, since the camera provide a view of approximately only 80 degrees in azimuth. So the points beyond the camera FoV are abandoned and not taken into consider in our method. In definitions in formula (2), (x_c, y_c) of a point p_c in camera coordinate corresponds to (u, v) in uv-coordinate system of an image, and z_c of the point represents the distance from the point to the image plane. During the projection, points that fall behind the plane ($z_c < 0$) or fall out of the range of the image ($x_c \notin [0, width]$ *or* $y_c \notin [0, height]$) are discarded. By this way, the point cloud is truncated to the same FoV of camera (as shown in right part of Figure 1).

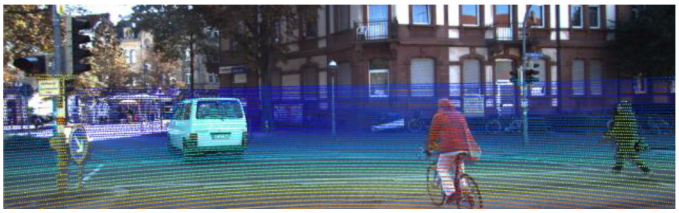


Figure 5. A test image showing the projection from point cloud to image coordinate. Color of each projected point represents its distance to the center of the LiDAR.

Label Transfering Using Gauss Kernel

When the projected points are calculated, the label of an original point can be fetched by using gauss filter voting in adjacent 9 pixels.

$$L(p_i) = \underset{l \in C}{\operatorname{argmax}} \left\{ \sum_{p_j \in A_i \ s.t. \ L(p_j)=l} \exp\left(-\frac{\|p_i - p_j\|^2}{2\sigma^2} \right) \right\} \tag{3}$$

where

L: $\mathbb{R}^2 \rightarrow C$ = the function that label a spatial point from label class C

p_i = the target point for labelling

A_i = the set of nearest 9 adjacent point-representing pixels.

$\|V\|$ = the Euclidean norm of vector V

σ = scale parameter which is set to 1 in our experiments

The label given by the method depends mainly on the label of nearest pixel while adjacent pixels are taken into consideration as well. Therefore the algorithm makes the transferring robust when there are tiny fault areas surrounded by correct ones.

Hierarchical Voting for Segment Fusion

After the label transferring, point cloud is labeled by image segments and thus the cloud is divided into semantic segments. However, the segments do not align with the clusters which are based on spatial connectivity, so a fusion of them is needed. Here we propose a hierarchical voting algorithm for labelling cross region of segments.

Before the point cloud is inputted, the priori knowledge of inclusion relationship in the label class is needed. That is to say, for given two labels l_1 and l_2, whether one of them is semantically included in or part of anther one should be clear. The voting algorithm runs with the help of the relationship. The relation can be represented by a directed acyclic graph, where a node is a label, and an edge represents the parent-child relationship of the labels.

A classical voting process is a maximum extracting procedure. Each of the points that falls into the clusters votes with its label from image, and the label that gain most votes is set as the label of the cluster. Normally, all votes have the same weight. However, due to the sparsity of point cloud, voting weight is customized in this paper to make the edge points vote less and farther points vote more. Suppose the voting point is p_v, then the voting weight $w(p_v)$ is calculated through following formula:

$$w(p_v) = \tanh\left(\frac{\|p_v - p_c\|}{\max_{p_i \in C}\|p_i - p_c\|}\right) \cdot \sqrt{\|p_v\|} \tag{4}$$

where

C = current cluster that voting is conducted on

p_c = the centroid of the cluster

Input: C: set of points in target cluster generated by Algorithm 1, each point has a label from image;
n_l: total number of labels;
Output: $L_n(p)$: new label of a point with initial value same as $L(p)$
Function: $L(p)$: represent the label of the point from image
$w(p)$: get the customized voting weight of the point
A(l): get semantic ancestors of the label
$W \leftarrow \{0, 0..0\}$: array with length of n_l and initial value 0 that stores voting weights
for each $p_i \in C$ **do**
$W[L(p_i)] \leftarrow W[L(p_i)] + w(p_i)$
for each $l_a \in$A($L(p_i)$) **do**
$W[l_a] \leftarrow W[l_a] + w(p_i)$
end for
$l_m \leftarrow$ argmax(W)
if $L_n(p_i)$ is not l_m and $l_m \notin$ A($L(p_i)$) **then**
$L_n(p_i) \leftarrow l_m$
end if
end for

Algorithm 2. Hierarchical Voting

In hierarchical voting process, an inlier point not only votes for its label, but also for semantic ancestors of its label. And when the label l_m with most votes is calculated, label of a point in the cluster is covered if its label is not l_m or a child of l_m. Comparisons are shown in Figure 6.

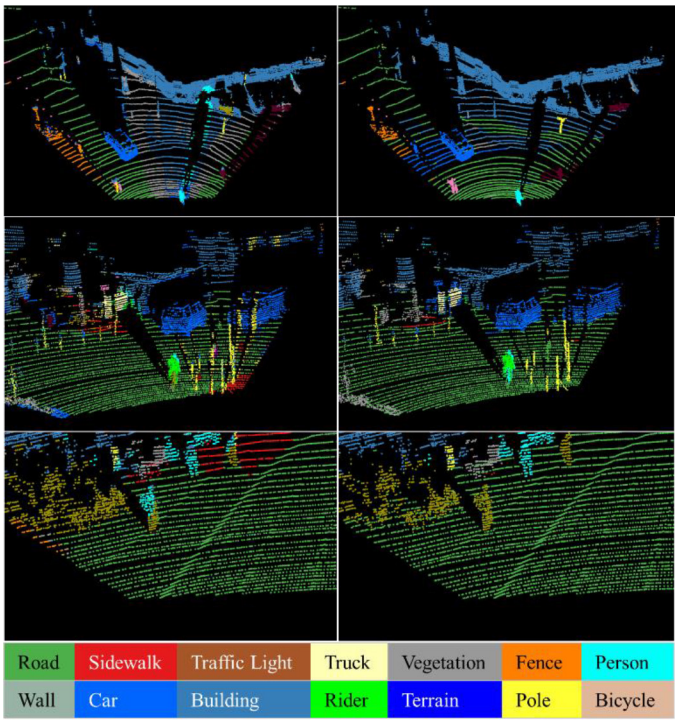


Figure 6. Segments from image that destroy the integrity of object is fixed by voting process. The images on the left are origin point clouds whose labels are directly transferred from image. The images on the right are the point clouds after segment fusion

Using hierarchical algorithm, labels that are hard to distinguish (e.g. road and terrain) can be fused without harming the differentiation between labels. For instance, the plane cluster extracted from point cloud is usually labeled as road and terrain, however, there will be a huge ratio of fault labels if classical max-voting algorithm, where every pixel votes with same weight, is used. On the contrary, hierarchical voting makes the two label remain unchanged. In addition, the voting is able to fix conflicts between semantic label and object integrity and build a bridge of two sensors in different measure views. In hierarchical voting, a bicycle and its rider will not be unified to be either bicycle or rider as classical voting algorithm does.

EXPERIMENT EVALUATION

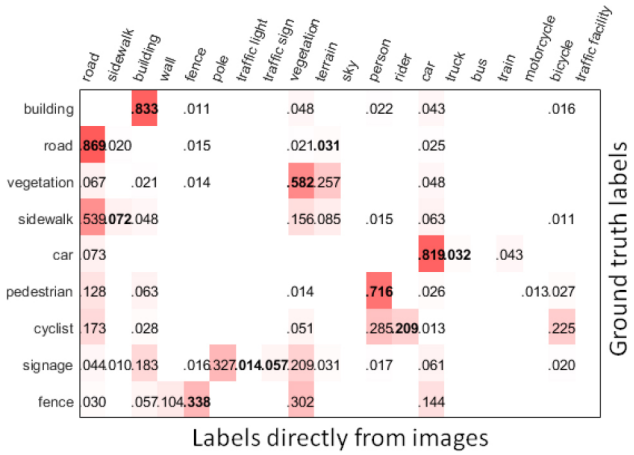
The test data is fetched from object tracking and detecting parts of the KITTI dataset. Its recording platform is a Volkswagen Passat B6 equipped with an 8-core i7 computer with RAID storage system, running Ubuntu Linux and a real-time database. Sensors used in recording and referred in this paper are a Velodyne HDL-64E laser scanner and Point Grey Flea 2 color cameras whose resolution is 1.4M pixels. For every single frame, a binary file containing point coordinates, a color image file and a text file containing calibration matrices are included in the data suite.

In the hierarchical voting procedure, original label class from Cityscape is extended. The new labels are listed in Table 1. To evaluate segmentation result, hand-labeled ground truth from Zhang et al. [19] is used. The annotations by Zhang cover a series of images and point clouds from 8 sequences in tracking task of KITTI dataset. Noting that the labels in the annotations are partly different from the labels from Cityscape, and the mapping between the two label classes can be found in the evaluation chart shown in Figure 7.

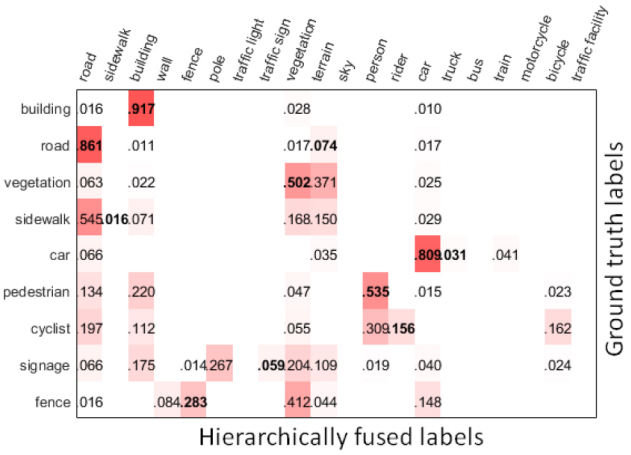
Table 1. The labels used in our segmentation. Labels in italic are extended ones. The upper label is parent of lower one in a major row (bounded by bold lines).

Traffic facility				Pole	Terrain
Traffic lights	Traffic sign	*Traffic pole*			Road
Car			Person	*Two-wheels vehicle*	
Truck	Bus		Rider	Motorcycle	Bicycle
Wall	Sky	Fence	Sidewalk	Train	Vegetation

Experiments are carried out in the referred data sequences with 252 frames and 333K points in total. Each point is labeled with our method and compared to the ground truth label. The result of experiments are shown in Table 2 and Figure 7. There is no other available method in hand for us to compare with in terms of point cloud segmentation performance, so major results are presented. It can be found that the semantic labeling accuracy is satisfying using CNN, especially for road, buildings, people and cars. Using our hierarchical fusion algorithm, the IoU performance is improved and labeling result is centralized. However, the accuracy improvement is not prominent because, according to our view, the relationship of labels is not well-constructed and the segmentation of point cloud is accomplished by trivial methods, which can be ameliorated in future study.



a.



b.

Figure 7. Comparison of semantic segmentation of the point cloud using (a) origin labels directly from image and (b) labels from hierarchical fusion. The horizontal axis lists the labels of segmentation result through our methods, and the vertical axis lists the labels from hand-labeled ground truth. Given label *x* from horizontal axis and *y* from vertical axis, decimals in the chart shows the ratio of points with label *y* that are actually labelled as *x*, to the total points of label *y*. Numbers in bold style mean that the labelling from *x* to *y* is true in reality (true-positive labels).

Table 2. The experiment results.

Method	Mean IoU	Point-wise accuracy	Time cost per frame
Directly from segmented image	67.56%	80.13%	2.17s @ TITAN X GPU
Fusion with spatial clusters	67.73%	80.07%	2.17s + 14.41s @ 2.50GHz CPU

According to time performance of the two methods, it takes couples of seconds to process a single frame, which makes it hardly possible to be used directly in real automobile, but the method has potential in real-time application thanks to the online algorithm in the method. The time cost of the method mainly consists of two parts, image processing and point cloud processing, and they can be improved by using more efficient neural network structure and reducing iteration count in algorithms respectively. As well, in this paper, optimizing is implemented in neither software nor hardware because we mainly focus on accuracy performance. To ameliorate the performance, a better and faster segmentation algorithm is expected to improve the accuracy and integrity of segments which can help rectifying more faults in image segmentation and point cloud segmentation.

CONCLUSIONS

In this paper, we present a novel and effective system for semantic segmentation of sparse point cloud frames. Semantic labels are firstly generated by CNN (PSPNet) in the image frame, then projected into 3D space initially, and furthermore corrected by the LiDAR points. A label fusion algorithm with hierarchical characteristic is proposed as well. We use label hierarchy to reserve the detailed semantic information when an object is covered by several labels from image. In addition, customized weight function is proposed to improve

existing voting algorithms. Finally, the experiment results validate the effectiveness of proposed system of semantic segmentation and improvement in object integrity protection.

For future work, performance of the proposed system can be ameliorated by generating better image and point cloud segments with higher accuracy. Alternative choice of exploration could be fusion algorithms where the point clouds and images are both semantically segmented, or the two components are segmented jointly.

REFERENCES

1. Geiger, A., Lenz, P. and Urtasun, R., "Are we ready for autonomous driving? The KITTI vision benchmark suite," IEEE Conference on Computer Vision and Pattern Recognition, 2012, doi:10.1109/cvpr.2012.6248074.
2. Schnabel, R., Wessel, R., Wahl, R, and Klein, R., "Shape Recognition in 3D Point Clouds," 2008.
3. Yang, B., and Dong, Z., "A Shape-Based Segmentation Method for Mobile Laser Scanning Point Clouds," ISPRS Journal of Photogrammetry and Remote Sensing 81: 19-30, 2013, doi:10.1016/j.isprsjprs.2013.04.002.
4. Douillard, B., Underwood, J., Kuntz, N., Vlaskine, V. et al., "On the Segmentation of 3D LIDAR Point Clouds," Robotics and Automation (ICRA), 2011 IEEE International Conference on IEEE, 2011: 2798-2805, 2011, doi:10.1109/ICRA.2011.5979818.
5. Barnea, S. and Filin, S., "Segmentation of Terrestrial Laser Scanning Data Using Geometry and Image Information," ISPRS Journal of Photogrammetry and Remote Sensing 76: 33-48, 2013, doi:10.1016/j.isprsjprs.2012.05.001.
6. Lin, H., Gao, J., Zhou, Y., Lu, G. et al., "Semantic Decomposition and Reconstruction of Residential Scenes from LiDAR Data," ACM Transactions on Graphics (TOG) 32 (4): 1, 2013, doi:10.1145/2461912.2461969.
7. Zhu, X., Zhao, H., Liu, Y., Zhao, Y. et al., "Segmentation and Classification of Range Image from an Intelligent Vehicle in Urban Environment," Intelligent Robots and Systems (IROS), 2010 IEEE/RSJ International Conference on IEEE 2010: 1457-62, 2010, doi:10.1109/IROS.2010.5652703.
8. Chen, X., Ma, H., Wan, J., Li, B. et al., "Multi-View 3D Object Detection Network for Autonomous Driving," arXiv preprint, 2016, arXiv:1611.07759.
9. Valentin, J. P., Sengupta, S., Warrell, J., Shahrokni, A. et al., "Mesh Based Semantic Modelling for Indoor and Outdoor Scenes," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013: 2067-74, 2013, doi:10.1109/CVPR.2013.269
10. Habermann, D., Hata, A., Wolf, D. and Osório, F. S., "Artificial Neural Nets Object Recognition for 3D Point Clouds," Intelligent Systems (BRACIS), 2013 Brazilian Conference on IEEE, 2013: 101-6, 2013, doi:10.1109/BRACIS.2013.25.
11. Maturana, D. and Scherer, S., "3D Convolutional Neural Networks for Landing Zone Detection from LiDAR," Robotics and Automation (ICRA), 2015 IEEE International Conference on IEEE, 2015: 3471-8, 2015, doi:10.1109/ICRA.2015.7139679.
12. Zhao, G., Xiao, X., Yuan, J. and Ng, G. W., "Fusion of 3DLIDAR and Camera Data for Scene Parsing," Journal of Visual Communication and Image Representation, 2014, 25(1): 165-83, 2014, doi:10.1016/j.jvcir.2013.06.008.
13. Kundu, A., Li, Y., Daellert, F., Li, F. et al., "Supplementary Material for Joint Semantic Segmentation and 3D Reconstruction from Monocular Video," *Computer Vision--ECCV* 2014, 703-8, 2014, doi:10.1007/978-3-319-10599-4_45.
14. Zhao, H., Shi, J., Qi, X., Wang, X. et al., "Pyramid Scene Parsing Network," arXiv preprint, 2016, arXiv:1612.01105.
15. Cordts, M., Omran, M., Ramos, S., Rehfeld, T. et al., "The Cityscapes Dataset for Semantic Urban Scene Understanding," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 2016: 3213-3223, 2016, doi:10.1109/cvpr.2016.350
16. Rusu, R. B. and Cousins, S., "3D Is Here: Point Cloud Library (PCL)," Robotics and automation (ICRA), 2011 IEEE International Conference on IEEE, 2011: 1-4. doi:10.1109/ICRA.2011.5980567.
17. Papon, J., Abramov, A., Schoeler, M. and Worgotter, F., "Voxel Cloud Connectivity Segmentation - Supervoxels for Point Clouds," Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2013: 2027-34, 2013, doi:10.1109/CVPR.2013.264.
18. Golparvar-Fard, M., Balali, V. and de la Garza, J. M., "Segmentation and Recognition of Highway Assets Using Image-Based 3D Point Clouds and Semantic Texton Forests," Journal of Computing in Civil Engineering, 2012, 29(1): 04014023, 2012, doi:10.1061/(ASCE)CP.1943-5487.0000283.
19. Zhang, R., Candra, S. A., Vetter, K. and Zakhor, A., "Sensor Fusion for Semantic Segmentation of Urban Scenes," Robotics and Automation (ICRA), 2015 IEEE International Conference on IEEE, 2015: 1850-57, 2015, doi:10.1109/ICRA.2015.7139439.

CONTACT INFORMATION

Dr. Kun Jiang
State Key Lab of Automotive Energy and Safety, Tsinghua University, Beijing, China
kunjiangtsinghua@163.com

Prof. Diange Yang
State Key Lab of Automotive Energy and Safety, Tsinghua University, Beijing, China
ydg@tsinghua.edu.cn

ACKNOWLEDGMENTS

This work is supported by International Science & Technology Cooperation Program of China under contract No.2016YFE0102200

DEFINITIONS/ABBREVIATIONS

LiDAR - light detection and ranging

SfM - structure from motion

CNN - convolutional neural network

SVM - support vector machine

PSPNet - pyramid scene parsing network

RANSAC - random sample and consensus

FoV - field of view

IoU - intersection over union