2014 IEEE International Conference on Robotics & Automation (ICRA)
Hong Kong Convention and Exhibition Center
May 31 - June 7, 2014. Hong Kong, China

Road Scene Segmentation via Fusing Camera and Lidar Data

Wenqi Huang[1], Xiaojin Gong*[2], Zhiyu Xiang[3]

Abstract— **This paper presents an approach for pixel-wise object segmentation for road scenes based on the integration of a color image and an aligned 3D point cloud. In light of the advantage of range information in object discovery, we first produce initial object hypotheses by clustering the sparse 3D point cloud. The image pixels registered to the clustered 3D points are taken as samples to learn each object's prior knowledge. The priors are represented by Gaussian Mixture Models (GMMs) of color and 3D location information only, requiring no high-level features. We further formulate the segmentation problem within a Conditional Random Field (CRF) framework, which incorporates the learned prior models, together with hard constraints placed on the registered pixels and pairwise spatial constraints to achieve final results. Our algorithm is validated on the challenging KITTI dataset which contains diverse complicated road scenarios. Both qualitative and quantitative evaluation results show the superiority of our algorithm.**

I. Introduction

Nowadays, a standard solution for an autonomous vehicle to perceive its navigational environment is utilizing a camera in conjunction with a ranging sensor, for instance, a Velodyne HDL lidar [1], [2]. The latter produces reliable range information, which is invariant to illumination change and is of no perspective transformation. Therefore, the joint use of these two modal sensors provides a robot with higher abilities to achieve tasks such as road and obstacle detection, road scene parsing, urban modeling, and autonomous driving.

This paper proposes an approach for pixel-wise object segmentation for road scenes based on the integration of visual and range data. By taking advantage of range information, object instances are segmented in an image without tedious off-line training. Although no semantic label is attached to each segment, we believe that such object-level segmentation is critical for various applications. For instance, as mentioned in [3], it is able to intelligently suggest locations for pedestrian, vehicle, or other object detection, so that the detection performance can be improved and the time-consuming searching procedure is avoided. Moreover, object-level segmentation is also an important pre-processing step to approaching road scene understanding.

The approach proposed in this paper distinguishes itself from other semantic segmentation techniques in a few aspects. First, our approach performs pixel-wise object segmentation mostly relying on a single image and an aligned 3D point cloud. Except a color Gaussian Mixture Model (GMM) learned from a set of hand-labeled sky regions, no other tedious hand labeling or off-line training are needed. Second, instead of extracting complicated features such as textons or Histogram of Oriented Gradients (HOG) descriptors [4], [5], we only use the color and location of each pixel as the feature. Last but not least, the integration of visual and range data is conducted in multiple stages. Initially, the 3D points are clustered to generate candidate object instances. The pixels registered to the 3D points are taken as samples to build GMMs of each object and are also used as hard constraints incorporated in a Conditional Random Field (CRF) framework to guide image segmentation. The incorporation of the registered pixels within the CRF as hard constraints guarantees good segmentation results. Our approach achieves high performance on the challenging KITTI dataset [2], which contains a variety of complex road scenarios.

II. Related Work

Semantic segmentation is one of the most challenging but important problems in computer vision. It aims to segment an image into regions that correspond to objects, with or without semantic labels attached. Traditionally, this task relies on visual information alone. Complicated features, including color, location, textons, and even HOG descriptors, are extracted [6] or learned [7] for pixels or superpixels [8]. The features are further fed into an off-line trained classifier to generate object class likelihood. Then, a random field framework is commonly used to incorporate the unary likelihood potential and pairwise or higher order spatial constraints in order to produce final segments [6], [5], [7]. As geometric data can provide complimentary information, cues from depth maps [9], [10], dense stereo maps [11], structure from motion [4], [12], or 3D layouts learned from images [13], [14] are also utilized by researchers. Most of the cues are manifested as extra features that are used in classification to improve the segmentation performance. Almost all existing approaches require tedious hand labeling work and a time-consuming training procedure. Their segmentation results highly depend on the selected training sets.

Road scene segmentation shares some commonalities with the above-introduced general semantic segmentation techniques. However, the segmentation of road scenes can be benefitted from extra cues such as vanishing points [15], [16], ground planes [17], or 3D scene layout [13]. Due to the

*This research work was supported in parts by the National Natural Science Foundation of China via grants 61001171, 61071219, 90820306, and Chinese Universities Scientific Fund.

[1]Wenqi Huang is with the Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China 11031017@zju.edu.cn

*[2]Xiaojin Gong, the corresponding author, is with Faculty of the Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China gongxj@zju.edu.cn

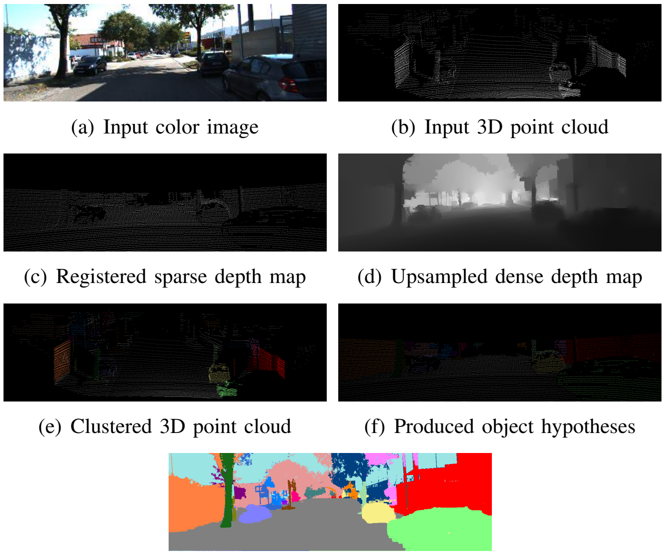
[3]Zhiyu Xiang, is with Faculty of the Department of Information Science and Electronic Engineering, Zhejiang University, Hangzhou 310027, China xiangzy@zju.edu.cn

navigational demand, a majority of researches on road scene analysis are focusing on drivable road detection [15], [17] and pedestrian or other moving obstacle detection [18], [19]. Only a small number of works study on scene segmentation and parsing. For example, Sturgess et al. [4] and Zhang et al. [20] combine appearance with structure from motion features and dense depth maps to segment road scenes into multiple classes. Alvarez et al. [13] segment images into sky, horizontal ground and vertical surface regions. With the advent of ranging sensors, particularly, the extensive use of Velodyne HDL lidars on unmanned vehicles, more and more 3D lidar datasets have become available [2], [21]. Besides the study of segmenting 3D point clouds [22], researchers have also started working on fusion based scene parsing. They often conduct segmentation on an image and a 3D point cloud individually, while paying attention to integrating two segmentation results via fusion techniques such as the fuzzy logic inference framework [23].

III. Proposed Framework

The objective of this paper is to achieve pixel-wise object segmentation when an image and an aligned 3D point cloud are given. The data are, respectively, collected by a camera and a lidar that are mounted on a vehicle. As the intrinsic and extrinsic parameters of both sensors are known, it is readily able to register the 3D point set and the image to each other. By registration, we obtain a sparse depth map, in which the registered pixels are assigned with corresponding depth values and the remaining pixels are of no depth information. For the convenience of subsequent processes, the sparse depth map is upsampled by a guided depth enhancement technique [24], which generates a dense depth map via integrating the sparse one with the color image.

In order to perform object-level segmentation in the image, we first partition the set of 3D points into different clusters according to their locations. Each cluster potentially corresponds to a geometrically meaningful object in 3D, and therefore stands for an object hypothesis. The pixels registered to the 3D points of each cluster are taken as samples, whose color and location are taken into account to build Gaussian mixture models of each object. We then formulate the segmentation problem within a CRF framework, whose energy function is constructed in terms of the learned prior models, together with hard constraints anchored by the registered pixels and pairwise spatial constraints. The final segmentation is achieved by solving the CRF problem via Graph Cuts [25]. Fig. 1 illustrates the exemplar result of each step in our framework.



(a) Input color image

(b) Input 3D point cloud

(c) Registered sparse depth map

(d) Upsampled dense depth map

(e) Clustered 3D point cloud

(f) Produced object hypotheses

(g) Segmentation result

Fig. 1. Visualization of each step in the framework of our algorithm. (Zoom in for better view.)

IV. Object Hypothesis Generation

The aim of this stage is to generate initial object hypotheses and learn their prior knowledge. Considering that geometric information is more reliable than visual cues for finding objects, we start from partitioning the 3D point cloud into clusters to obtain object hypotheses. As pointed out by Douillard et al. [22], prior ground extraction significantly improves clustering performance. Therefore, we split the clustering procedure into two steps: ground plane estimation and the remaining 3D point clustering. Once we get the clusters, the pixels registered to them are taken to build prior models of objects.

A. Ground Plane Estimation

The ground is commonly the dominant plane in most road scenes. We therefore use the Random Sample Consensus (RANSAC) algorithm to estimate it. However, in some scenarios, for instance, a narrow street with buildings on both sides, the estimated dominant plane may lie on a wall of the buildings. In order to avoid such consequence, we define a rough range of height according to where the lidar is equipped on the vehicle. Only the 3D points within the range are taken into consideration for ground plane estimation.

B. 3D Point Clustering

A variety of techniques have been developed for 3D point clustering [22], [26] and can be applied here. For the purpose of efficiency, we adopt a simple but effective one, which is based on the nearest neighbor (KNN) scheme with respect to Euclidean distance [27]. It can be implemented with a kd-tree data structure and therefore is quite efficient. This approach can produce a set of object clusters well, especially for the separated objects on the road.

Note that our clustering is performed on the original sparse 3D lidar points, instead of the denser points reconstructed from the upsampled dense map. The reason is that the upsampling techniques are prone to generate artifacts, especially on the places near object boundaries and in large invalid regions, leading to errors that might be propagated to following stages.

C. Learning Prior Models

Once the ground and other object clusters are produced, the pixels registered to them are taken as samples to learn their prior models. In our work, we only take the RGB color and 3D location of each pixel as the features. No other complex features are considered. Therefore, for each object instance, a Gaussian mixture model of the 6D feature (R, G, B, X, Y, Z) is built. It needs to be mentioned that a different means is taken for building the sky model. Since there is no way to sample the sky from lidar data, sky regions in a set of images are manually labeled to learn a color GMM for the sky.

V. CRF-based Pixel-wise Object Segmentation

Assume that N object clusters are generated. The segmentation problem is then equivalent to assigning each pixel with a label $l \in \{0, \cdots, N+1\}$, denoting the l-th object instance, 0 for the ground and $N+1$ for the sky. We formulate this problem via a pairwise conditional random field, which allows us to incorporate the learned prior models, hard constraints anchored by the sampled pixels, as well as spatial information.

Given an image $\mathbf{I}$, we construct a graph $\mathcal{G} = \langle \mathcal{V}, \mathcal{E} \rangle$, where each node $v_i \in \mathcal{V}$ represents a pixel and each edge $e_{ij} \in \mathcal{E}$ stands for a 4-connected neighboring relationship between pixels. In addition, each node is associated with a random variable l_i, i.e. a label. The object-level segmentation is achieved through the labeling problem, which minimizes the following energy function:

$$E(\mathbf{L}) = \sum_{v_i \in \mathcal{V}} \psi_i(l_i) + \lambda \sum_{e_{ij} \in \mathcal{E}} \psi_{ij}(l_i, l_j), \qquad (1)$$

where $\mathbf{L} = \{l_i \big| i = 1, \cdots, M\}$ is the labeling set and M is the total number of pixels. $\psi_i(l_i)$ and $\psi_{ij}(l_i, l_j)$ are, respectively, the unary and pairwise potentials that are defined below. λ is a weight balancing two terms. This optimization problem is efficiently solved by Graph Cuts [25].

A. Unary Potential with Hard Constraints

The unary potential evaluates the confidence for a pixel to be labeled with l_i. Commonly, it is designed in terms of the likelihoods as follows [28], [25]:

$$\psi_i(l_i) = -\ln p(\mathbf{f}_i | \Theta_{l_i}), \qquad (2)$$

where $\mathbf{f}_i = (R_i, G_i, B_i, X_i, Y_i, Z_i)$ is the feature vector associated with the pixel v_i, Θ_{l_i} denotes the parameters of the GMM of the l_i-th object, and $p(\mathbf{f}_i | \Theta_{l_i})$ is the likelihood.

The above-defined likelihood potential is sensitive when two objects share similar features. For instance, strong shadows on the ground and bushes nearby are prone to being labeled as the same object by mistake. In contrast, 3D point clustering performs better, at least the clustering results are invariant to illumination change. Therefore, we place high confidence [28] on the pixels that are registered to the clustered 3D points. Let us denote the entire set of registered pixels by $\mathcal{L}$, and the set of pixels registered to the lidar points belonging to the l-th object by $\mathcal{L}_l$. Then, the unary potential with hard constraints is defined by

$$\psi_i(l_i) = \begin{cases} \alpha & i \in \mathcal{L}_{l_i} \\ \beta & i \in \mathcal{L}/\mathcal{L}_{l_i} \\ -\ln p(\mathbf{f}_i | \Theta_{l_i}) & \text{otherwise,} \end{cases} \qquad (3)$$

where α is a small positive value and β is a large positive value, which are experimentally set to force the constraints. With these hard constraints, the labels of the registered pixels are forced to be consistent with the point clustering results.

B. Pairwise Potential

This potential explores the dependence between neighboring pixels in order to smooth out isolated labeling results. For a pixel v_i and each of its 4-connected neighboring pixels v_j, the smoothness term is defined as

$$\psi_{ij}(l_i, l_j) = exp\left(-\frac{||\mathbf{f}_i - \mathbf{f}_j||_2^2}{\sigma^2}\right) \cdot T(l_i \neq l_j), \qquad (4)$$

where $||\mathbf{f}_i - \mathbf{f}_j||_2$ is the L_2 norm of the difference between the features $\mathbf{f}_i$ and $\mathbf{f}_j$. $T()$ is an indicator, whose value is 1 when its parameter is true and 0 otherwise. This term indicates that the more similar the features are, the more likely it is that the two pixels belong to the same object.

VI. Experiments

A. Experimental Setup

In order to validate the proposed approach, we have conducted a series of experiments on the KITTI vision benchmark suite [2], which provides us with numerous color images and 3D point clouds. The data are captured by a PointGrey Elea2 video camera and a Velodyne HDL-64E 3D LIDAR that are jointly mounted on a vehicle. We test our approach mainly on the 'City' data category. It contains a variety of complex scenarios on urban roads, with the presence of vehicles, cyclists, pedestrians and other objects.

The images that we deal with are at the resolution of 1242×375 pixels. Each 3D point cloud is of $100,000$ points or so, covering a 360^o field of view (FOV). But only those falling within the camera's FOV are taken into consideration. These two modal data are registered to each other according to the sensor parameters provided on the KITTI's website. We use the nearest neighbor clustering algorithm implemented in the Point Cloud Library (PCL) [29] to generate initial object hypotheses. The produced clusters that have very small amount of faraway points are discarded for robustness. The parameters of our algorithm are empirically set as follows: $\lambda = 0.1$, $\alpha = 10$, $\beta = 200$, $\sigma = 10$, and each Gaussian mixture model has five components. In addition, both the RGB and XYZ of the feature are normalized to the range of $[0, 255]$.

B. Qualitative Evaluation

To illustrate the performance of our algorithm, a group of comparative experiments are conducted. In the comparison, we pay attention to how much improvement can be achieved by additionally using location features and placing

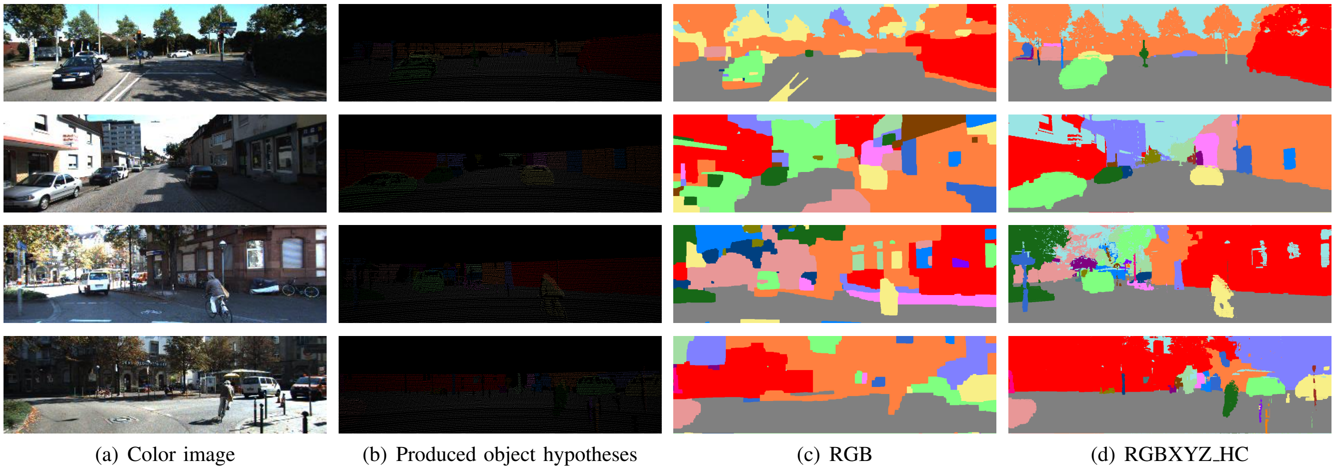

(a) Color image (b) Produced object hypotheses (c) RGB (d) RGBXYZ_HC

Fig. 2. Comparative experimental results between RGB and RGBXYZ_HC. (a) Original color images. (b) The pixels which are registered to clustered objects are marked in different colors. (Zoom in for better view.) (c) The results obtained by the algorithm only using color information. (d) The results obtained by the RGBXYZ_HC algorithm. Each color on the results represents an object instance.

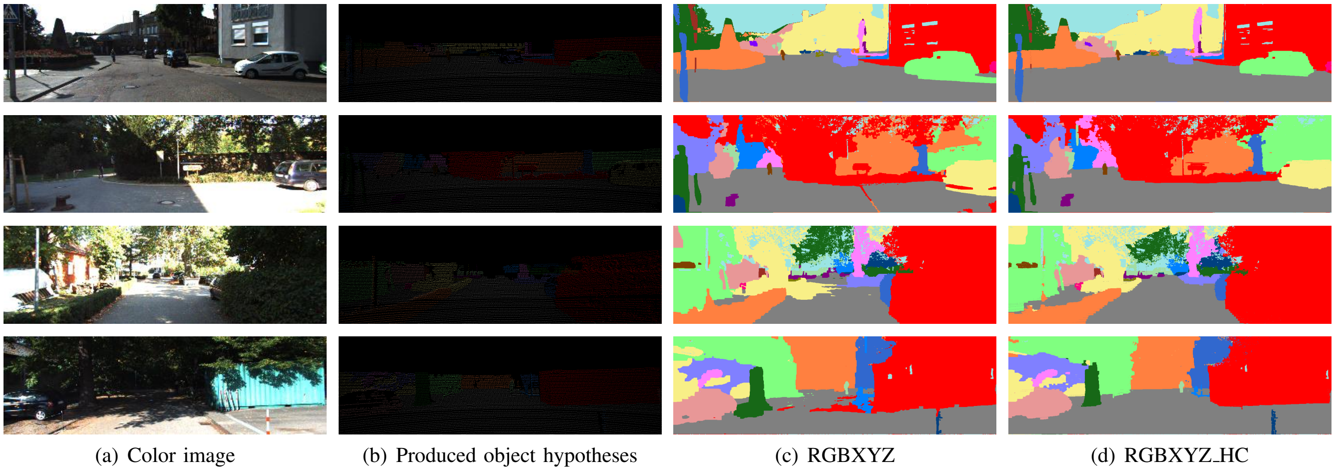

(a) Color image (b) Produced object hypotheses (c) RGBXYZ (d) RGBXYZ_HC

Fig. 3. Comparative experimental results between RGBXYZ and RGBXYZ_HC.

the hard constraints on the unary potential. According to whether location information is used and whether the hard constraints are placed or not, we denote the algorithms by RGB, RGBXYZ, RGB_HC, and RGBXYZ_HC respectively. For instance, literally, RGBXYZ_HC represents the one using both color and location features and with hard constraints, and likewise for the others. Fig. 2 - 4 present typical examples.

Fig. 2 illustrates the comparative experimental results between RGB and RGBXYZ_HC. From them we can observe that the algorithm only using color is sensitive to lane markings and shadows on the ground. It is also prone to mislabel two objects as the same one, even if they are geometrically separated. In contrast, RGBXYZ_HC performs much better in these cases.

Fig. 3 presents the comparison between RGBXYZ and RGBXYZ_HC. The RGBXYZ_HC algorithm outperforms RGBXYZ mainly on shadow or bright speckle regions on the ground. For instance, the RGBXYZ algorithm mislabels the shadow behind the car, the boundary between shadow and bright regions, the bright speckle on the ground and the shadow of trees on the ground, respectively, in the four examples. In other scenarios, these two algorithms obtain comparable results.

Fig. 4 shows the comparison of RGB_HC and RGBXYZ_HC performed on extensive scenarios, including urban roads, road intersections, narrow streets and so on. Generally speaking, both algorithms perform well on objects on the road, such as vehicles and pedestrians. In most cases, RGBXYZ_HC is superior to RGB_HC on roadside scenes. The object regions segmented by RGBXYZ_HC are more consistent in geometry.

From Fig. 2 and Fig. 4, we observe that, although depth upsampling may introduce errors, the use of dense depth information improves the segmentation results to a great extent.

C. Quantitative Evaluation

The results presented above provide us with an intuition about how well the algorithms perform. In this subsection, we quantitatively evaluate them by measuring the global

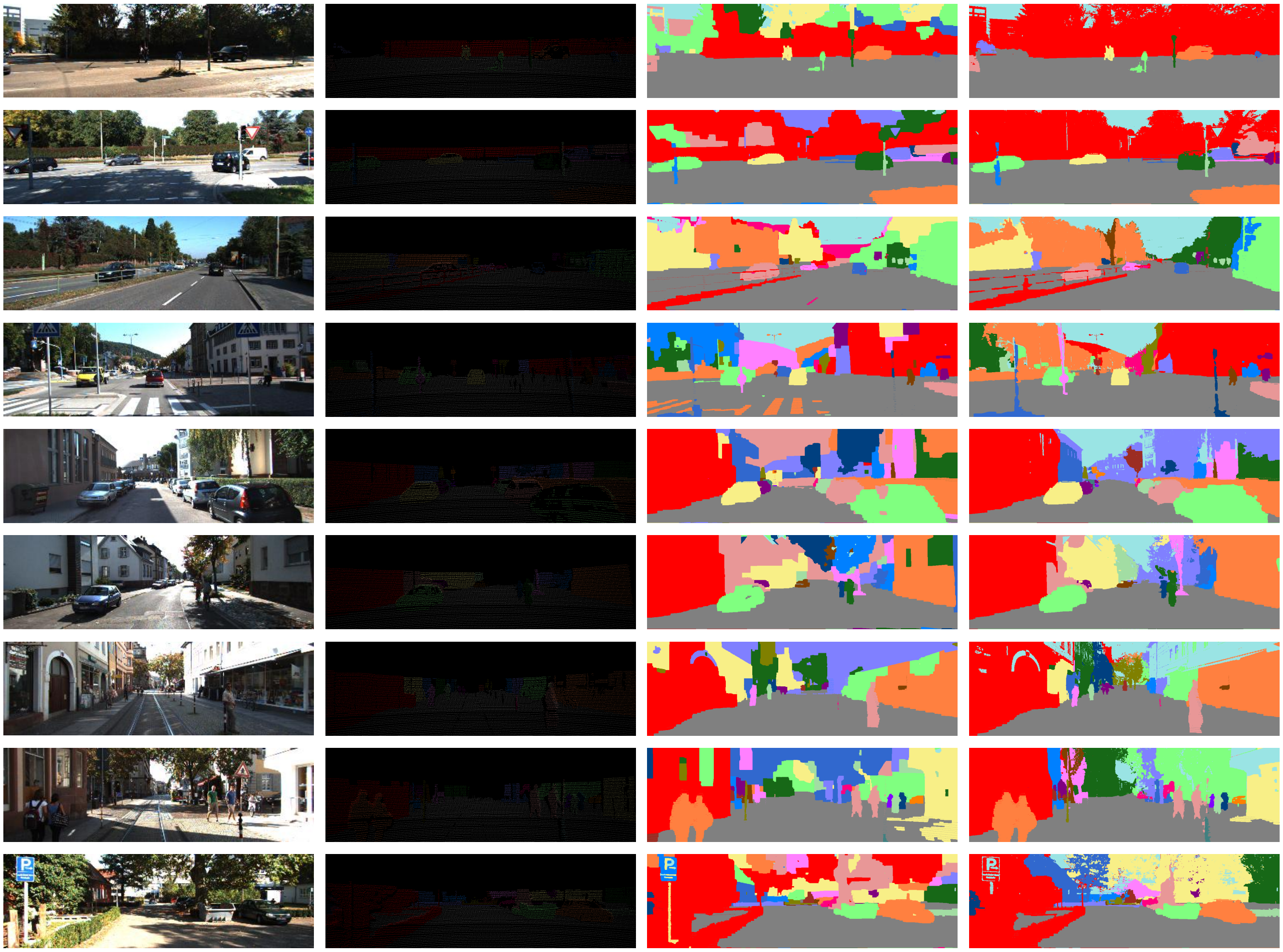

(a) Color image (b) Produced object hypotheses (c) RGB_HC (d) RGBXYZ_HC

Fig. 4. Comparative experimental results between RGB_HC and RGBXYZ_HC.

consistency error (GCE) and the local consistency error (LCE), which are two criteria proposed by Martin et al. [30] for measuring consistency between two segmentation results. These criteria are designed to be tolerant to different numbers of segments arising from different perceptual levels when observing complex scenarios.

Assume that $\mathcal{I} = \{A_1, A_2, \cdots, A_I\}$ is a reference segmentation of an image and $\mathcal{J} = \{B_1, B_2, \cdots, B_J\}$ is a segmented result obtained by an algorithm. The error between a fragment A_i and B_j is defined [30], [31] by

$$E(A_i, B_j) = \frac{|A_i \backslash B_j|}{|A_i|} \times |A_i \cap B_j|, \tag{5}$$

where $|x|$ denotes the cardinality of the set x and $\backslash$ stands for set difference. Note that this error is not symmetric.

Therefore, GCE and LCE are, respectively, defined as

$$GCE(\mathcal{I}, \mathcal{J}) = \frac{1}{M} min(\sum_i^I \sum_j^J E(A_i, B_j), \sum_i^I \sum_j^J E(B_j, A_i)), \tag{6}$$

and

$$LCE(\mathcal{I}, \mathcal{J}) = \frac{1}{M} \sum_i^I \sum_j^J min(E(A_i, B_j), E(B_j, A_i)), \tag{7}$$

in which M is the total number of pixels of an entire image. Both GCE and LCE are in the range of $[0, 1]$, where 0 signifies no error and 1 is for the worst. Moreover, $LCE \leq GCE$ for any two segmentations.

We manually label 66 images for evaluation and the ground truth is released on our website [1]. The global and local consistency errors are computed with respect to the labeled ground truth. The evaluation results are listed in Table I for each algorithm and the best performance is marked as bold. From the results we conclude that the RGBXYZ_HC algorithm performs best, followed closely by RGBXYZ. RGB gives the worst results. This evaluation is consistent with the observation that we obtained before.

Method / Criterion	RGB	RGB_HC	RGBXYZ	RGBXYZ_HC
GCE	0.382	0.283	0.202	**0.196**
LCE	0.382	0.251	0.183	**0.175**

TABLE I

QUANTITATIVE EVALUATION RESULTS FOR EACH ALGORITHM.

[1] http://mypage.zju.edu.cn/en/gongxj

D. Discussion

With the use of both color and location information, together with the incorporation of hard constraints, our algorithm achieves promising segmentation results in most scenarios. Of course, there is still room for improvement. For example, our algorithm is prone to segment two objects that are next to each other as one and segment roadside scenes into multiple objects due to its high dependence on the 3D clustering results. Therefore, a better clustering algorithm may help. Second, our algorithm is also affected by the results of depth upsampling, meaning that errors in dense depth maps are propagated to segmentation results. To circumvent such errors, we will propose to formulate both depth upsampling and object-level image segmentation as a joint optimization problem and solve them simultaneously.

In our experiments, we have not compared our algorithm with others' work yet. The reason lies in two aspects. First, to the best of our knowledge, there has been no other work developed for object-level segmentation of images while integrating sparse depth information, except that done in [23]. Second, both [23] and almost all traditional semantic image segmentation methods require sufficient manually labeled ground truth for training, which is very time consuming, and their performances highly depend on training sets.

The algorithm is implemented in Matlab and running on a desktop with an Intel Core i5 2300 and 4 GB memory. Since our implementation has not been optimized for efficiency, the whole process is about $30s$ per frame. Specifically, it takes about $5s$ for loading and registering over 1 million 3D points, $1s$ for 3D points clustering, $9s$ for generating GMMs and terms for GraphCut and GraphCut takes about $10s$ for solving 20 labels in a 1242×375 image.

VII. Conclusions and Future Work

In this paper, we have presented an approach for object-level road scene segmentation by integrating visual and range information. The approach has been validated by extensive experiments on the KITTI dataset. Both qualitative and quantitative evaluations have been made, which show that our algorithm is promising. In future, besides improving our algorithm in the aspects discussed above, we are also going to apply this work for further parsing road scenes and detecting individual or groups of objects [32].

References

[1] Velodyne, "Velodyne HDL-64E," 2013, http://velodynelidar.com/lidar/.

[2] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *Proc. CVPR*, Providence, USA, June 2012.

[3] A. Karpathy, S. Miller, and L. Fei-Fei, "Object discovery in 3d scenes via shape analysis," in *Proc. CVPR*, 2013.

[4] P. Sturgess, K. Alahari, L. Ladicky, and P. Torr, "Combining appearance and structure from motion features for road scene understanding," in *Proc. BMVC*, 2009.

[5] C. Russell, P. H. S. Torr, and P. Kohli, "Associative hierarchical crfs for object class image segmentation," in *Proc. ICCV*, 2009.

[6] J. Shotton, J. Winn, C. Rother, and A. Criminisi, "Textonboost for image understanding: Multi-class object recognition and segmentation by jointly modeling texture, layout, and context," *International Journal of Computer Vision*, vol. 81, no. 1, pp. 2–23, 2009.

[7] C. Farabet, C. Couprie, L. Najman, and Y. LeCun, "Learning hierarchical features for scene labeling," *TPAMI*, 2013.

[8] S. Gould, R. Fulton, and D. Koller, "Decomposing a scene into geometric and semantically consistent regions," in *Proc. ICCV*, 2009.

[9] S. Sengupta, E. Greveson, A. Shahrokni, and P. Torr, "Urban 3d semantic modeling using stereo vision," in *Proc. ICRA*, 2013.

[10] C. Couprie, C. Farabet, L. Najman, and Y. LeCun, "Indoor semantic segmentation using depth information," in *International Conference on Learning Representations (ICLR2013)*, 2013.

[11] L. Ladicky, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. H. Torr, "Joint optimization for object class segmentation and dense stereo reconstruction," *International Journal of Computer Vision*, vol. 100, no. 2, pp. 122–133, 2012.

[12] G. J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and recognition using structure from motion point clouds," in *Proc. ECCV*, 2008, pp. 44–57.

[13] J. M. Alvarez, T. Gevers, Y. LeCun, and A. M. Lopez, "Road scene segmentation from a single image," in *Proc. ECCV*, 2012.

[14] D. Hoiem, A. A. Efros, and M. Hebert, "Putting objects in perspective," in *Proc. CVPR*, 2006.

[15] H. Kong, J. Audibert, and J. Ponce, "Vanishing point detection for road detection," in *Proc. CVPR*, 2009.

[16] J. M. Alvarez, T. Gevers, and A. M. Lopez, "3d scene priors for road detection," in *Proc. CVPR*, 2010, pp. 57–64.

[17] W. Huang, X. Gong, and J. Liu, "Integrating visual and range data for road detection," in *Proc. ICIP*, 2013.

[18] A. Ess, B. Leibe, and L. V. Gool, "Depth and appearance for mobile scene analysis," in *Proc. ICCV*, 2007.

[19] B. Leibe, N. Cornelis, K. Cornelis, and L. V. Gool, "Dynamic 3d scene analysis from a moving vehicle," in *Proc. CVPR*, 2007, pp. 1–8.

[20] C. Zhang, L. Wang, and R. Yang, "Semantic segmentation of urban scenes using dense depth maps," in *Proc. ECCV*, 2010.

[21] G. Pandey, J. R. McBride, and R. M. Eustice, "Ford campus vision and lidar data set," *International Journal of Robotics Research*, vol. 30, no. 13, pp. 1543–1552, November 2011.

[22] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, and A. Frenkel, "On the segmentation of 3d lidar point clouds," in *Proc. ICRA*, 2011, pp. 2798–2805.

[23] G. Zhao, X. Xiao, and J. Yuan, "Fusion of velodyne and camera data for scene parsing," in *in Proc. Information Fusion*, 2012, pp. 1172–1179.

[24] J. Liu and X. Gong, "Guided depth enhancement via anisotropic diffusion," in *the Pacific-Rim Conference on Multimedia (PCM)*, 2013.

[25] Y. Boykov and M. Jolly, "Interactive graph cuts for optimal boundary & region segmentation of objects in nd images," in *Proc. ICCV*, 2001.

[26] M. Himmelsbach, A. Mueller, T. Luettel, and H.-J. Wuensche, "LIDAR-based 3D Object Perception," in *Proc. 1st International Workshop on Cognition for Technical Systems*, 2008.

[27] R. B. Rusu, "Semantic 3d object maps for everyday manipulation in human living environments," Ph.D. dissertation, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009.

[28] C. Rother, V. Kolmogorov, and A. Blake, "GrabCut - interactive foreground extraction using iterated Graph Cuts," in *SIGGRAPH*, 2004.

[29] PCL, "Euclidean cluster extraction in pcl," http://www.pointclouds.org/.

[30] D. Martin, C. Fowlkes, D. Tal, and J. Malik, "A database of human segmented natural images and its application to evaluating algorithms and measuring ecological statistics," in *Proc. ICCV*, 2001, pp. 416 – 423.

[31] M. Polak, H. Zhang, and M. Pi, "An evaluation metric for image segmentation of multiple objects," *Image and Vision Computing*, vol. 27.

[32] C. Li, D. Parikh, and T. Chen, "Automatic Discovery of Groups of Objects for Scene Understanding," in *Proc. CVPR*, 2012.