

Semantic Parsing of Street Scene Images Using 3D LiDAR Point Cloud

Pouria Babahajiani Tampere University of Technology Tampere, Finland pouria.babahajiani@tut.fi Lixin Fan Nokia Research Center Tampere, Finland fanlixin@ieee.org Moncef Gabbouj Tampere University of Technology Tampere, Finland moncef.gabbouj@tut.fi

Abstract

In this paper we propose a novel street scene semantic parsing framework, which takes advantage of 3D point clouds captured by a high-definition LiDAR laser scanner. Local 3D geometrical features extracted from subsets of point clouds are classified by trained boosted decision trees and then corresponding image segments are labeled with semantic classes e.g. buildings, road, sky etc. In contrast to existing image-based scene parsing approaches, the proposed 3D LiDAR point cloud based approach is robust to varying imaging conditions such as lighting and urban structures. The proposed method is evaluated both quantitatively and qualitatively on three challenging NAVTEQ True databases and robust scene parsing results are reported.

1. Introduction

Automatic urban scene parsing refers to the process of segmentation and classifying of objects of interest into predefined semantic labels such as "building", "tree" or "road" etc. This task is often done with a fixed number of object categories, each of which requires a training model for classifying image segments (i.e. superpixels). While many techniques for 2D object recognition have been proposed, the accuracy of these systems is to some extent unsatisfactory because 2D image cues are sensitive to varying imaging conditions such as lighting, shadow etc.

In this work, we propose a novel automatic scene parsing approach which takes advantage of 3D geometrical features derived from Light Detection And Ranging (LiDAR) point clouds. Mobile Terrestrial LiDAR (MTL) provides accurate, high-resolution 3D information (e.g. longitude, latitude, altitude) as well as reflectance properties of urban environment (see such an example mobile mapping system in figure 2). Since such 3D information is invariant to lighting and shadow, as a result, significantly more accurate parsing results are achieved by using 3D cues.

1.1. Related Work

Automatic scene parsing is a traditional computer vision problem. Many successful techniques have used single 2D image appearance information such as color, texture and shape [10,12,13]. By using just spatial cues such as surface orientation and vanishing points extracted from single images, Hoiem et al. [7] achieved considerably more robust results. A drawback of single image feature extraction techniques is that they are sensitive to different image capturing conditions such as lighting, camera viewpoint and scene structure. Recently, many efforts have been made to employ 3D scene features derived from single 2D images to achieve more accurate object recognition [2,5,16,17]. Especially, when the input data is a video sequence, 3D cues can be extracted using Structure From Motion (SFM) techniques [18]. Brostow et al. [2], proposed to employ sparse structure from motion point clouds to recover 3D depth information. And in [17] the authors used dense (per-pixel) depth map information recovered in a camera pose independent manner. The SFM technique adopted in these system [2,17], however, is known to be fragile in outdoor environment because of the difficulty in obtaining correct correspondence in cases of sparse texture or occlusion in the images.

With the advancement of LiDAR sensors and Global Positioning Systems (GPS), large-scale, accurate and dense point cloud can be created and used for 3D scene parsing purpose. There has been a considerable amount of research in registering 2D images with 3D point clouds [8,14,15]. Furthermore, there are methods designed for registering point cloud to image using LiDAR *intensity* [1].

1.2. Overview of the Proposed Framework

Figure 1 shows the overview of the proposed scene parsing framework, in which images and LiDAR Point Cloud (PC) are the inputs of the processing pipeline and parsing results are image segments assigned with different class labels. The proposed parsing pipeline starts from aligning 3D LiDAR point cloud with 2D images. Input images are segmented into superpixels to reduce



Figure 1: Overview of the proposed framework

computational complexity and to maintain sharp class boundaries. Each SuperPixel (SP) in 2D image is associated with a collection of LiDAR points, which is assumed to form a planar patch in 3D world. 3D features such as *height above camera*, *surface planarity* and *reflectance strength* are extracted for each patch. Using a trained classifier based on boosted decision trees, each 3D feature vector is then assigned with a semantic label such as "sky", "road", "building" etc. The offline training of the classifier is based on a set of 3D features, which are associated with manually labeled SPs in training images. Main contributions of this work are the following:

1) We demonstrate that using 3D LiDAR point clouds for street view scene parsing improves parsing accuracies under challenging conditions such as varying lighting and urban structures. The improvement is achieved by circumventing error-prone 2D feature extraction and matching steps.

2) We propose a novel method to register 3D point cloud to 2D image plane, and by doing so, occluded points from behind the buildings are removed in an efficient manner (see Section 2 for details).

3) We propose to use a novel LiDAR point reflectance property, i.e. *intensity* feature for semantic scene parsing, and demonstrate that combining both LiDAR intensity feature and geometric features leads to more robust classification results. Consequently, classifiers trained in one type of city and weather condition is now possible to be applied to a different scene structure with high accuracy (See section 4).

2. Generating 2D-3D Association

Given a 3D points cloud and one 2D image with known viewing camera pose, the association module described in this section aims to establish correspondences between collections of 3D points and groups of 2D image pixels. In particular, every collection of 3D points is assumed to be sampled from a visible planar 3D object i.e. patch and corresponding 2D projections are confined within a homogenous region i.e. superpixels (SPs) of the image. While the 3D-2D projection between patches and SPs is

straightforward for known geometrical configurations, it still remains a challenging task to deal with outlier 3D points in a computationally efficient manner. We will illustrate in Section 2.2 a novel and simple outlier removal method, but let us first briefly review in Section 2.1 the SP segmentation technique adopted in our processing pipeline.

2.1. Segmenting Images into Superpixels

Without any prior knowledge about how image pixels should be grouped into semantic regions, one commonly used data driven approach segments the input image into homogeneous regions i.e. superpixels based on simple cues such as pixel colors and/or filter responses. The use of SPs improves the computational efficiency and increases the chance to preserve sharp boundaries between different segments.

In our implementation, we adopt the geometric-flow based technique of Levinshtein [9] to segment images into SPs with roughly the same size. Sharp image edges are also well preserved by this method. For input images with dimensionality of 2032×2032 pixels, we set the initial number of SPs as 2500 for each image. See the first image in figure 3 as the example of SP segmentation results



Figure 2: Data Collection vehicle 'NAVTEQ True' [14]



Figure 3: 2D-3D association overview.

2.2. LiDAR Point cloud to Superpixel

We first review how to project a 3D point on 2D image plane with known viewing camera pose, and then illustrate a method that associates a collection of 3D points with any given SP on 2D image.

Given a viewing camera pose i.e. position and orientation, represented, respectively, by *T* a 3×1 translation vector and *R* a 3×3 rotation matrix, and a 3D point M=[X,Y,Z]^t, expressed in a Euclidean world coordinate system, then the 2D image projection m_{p=} [u, v]^t of the point M is given by

$$\widetilde{m_{p}} = K [R|T] \ \widetilde{M} = C \ \widetilde{M} \tag{1}$$

Where K is an upper triangular 3×3 matrix

$$K = \begin{bmatrix} f_x & 0 & x_0 \\ 0 & f_y & y_0 \\ 0 & 0 & 1 \end{bmatrix}$$
(2)

where f_x and f_y are the focal length in the x and y directions respectively, x_0 and y_0 are the offsets with respect to the image axes, and $\widetilde{m_p} = [u,v,1]^t$ and $\widetilde{M} = [X,Y,Z,1]^t$ are the homogeneous coordinates of m_p and M.

3D Light Detection And Ranging (LiDAR) point clouds are often measured in a *geographic coordinate system* (i.e. *longitude, latitude, altitude),* therefore, projecting a 3D LiDAR point on 2D image plane involves two more transformation steps, namely Geo-to-ECEF and ECEF-to-NED. After these two transformations, 3D point in NED coordinate aligns to image plane by equation (2). Figure 3 illustrates an overview of these transformations.

Using the projection step in equation (2) and necessary transformation steps, we are able to identify those 3D points that are projected within a specific SP. Since we assume there is only one dominant 3D patch that associates with the given SP, so outlier 3D points that are far from the patch should be removed. In order to successfully remove outlier points, a plane-fitting method can be used e.g. as in [15]. However, such an outlier removal methods have to be repeatedly applied to every SP and turns out to be too computationally demanding for our application. In this paper, we instead propose a novel and simple method to remove outlier points for all SPs in one pass. The proposed method takes advantage of priori knowledge about urban scene environment and assumes that there are building facades along both sides of the street. While this assumption appears to be oversimplified, the method actually performs quite well with urban scenes in three different US and European cities as demonstrated in the experimental results.

The essence of the method is to fit two hyperbolic curves to 3D points represented in a camera centered twodimensional Z-u plane (see Figure 4 top image). 3D points that are far from camera center and behind these two hyperbolic curves are deeded outliers and are removed. However, points with depth less than 50 meters (see red line) are kept because they play important roles to label road or other near objects.

The derivation of hyperbolic curves in this Z-u plane is due to the normalization of homogeneous coordinates or simply:

$$v = \frac{f_y Y}{Z} + y_0 \qquad \qquad u = \frac{f_x X}{Z} + x_0$$

In this case the street width X is assumed constant, u is inversely related to the depth Z, and the collection of aligned points in the 3D world lies between two hyperbolic lines (black lines in figure 4)

3. Semantic Parsing of Street Scene Images

After the associating of the LiDAR point cloud with the image SPs, the processing pipeline proceeds to extracting 3D features for different patches. At the offline training phase, these 3D features are used to train a boosted decision tree classifier. Detailed processing steps are elaborated below and the results are presented in section 4.

3.1. Feature Extraction

For each 3D patch, seven features are extracted to train the classifier. Among these seven features, *Height above* ground, surface normal, planarity, density and intensity are five camera pose independent features whereas horizontal distance to camera path and depth to camera center are two features which are defined according to the camera location.

Height above ground: Given a collection of 3D points with known geographic coordinates, the median height of all points is considered as the height feature of the patch. The height information is independent of camera pose and is calculated by measuring the distance between points and the road ground. In contrast to 3D point clouds reconstructed with SFM technique, the advantage of LiDAR point cloud is that we know the exact measure of points height and it is not necessary to use RANSAC method to estimate the ground plane, e.g. such as in [17].

Horizontal distance to camera: Following [17], we compute the horizontal distance of the each patch to the camera as second geographical feature.

Depth to camera: Depth information helps to distinguish objects, because in that case we can estimate the 3D spatial location of each patch. Although these three features represent geographical cues of each patch, they are not completely independent from vehicle location and as seen later when we change scene structure (training in one city and testing in another), the classifier performance is lower than when training and testing are performed on the same scene.

Surface normal: Similar to [17], we also extract surface normal for each patch. But following [11], we adopt a more accurate method to compute the surface normal by fitting a plane to the 3D points in each patch. In our implementation we used RANSAC algorithm to remove outliers which may correspond to very "close" objects such as a pedestrian or vehicle [6].

Planarity: Patch planarity is defined as the average square distance of all 3D points from the best fitted plane computed by RANSAC algorithm. This feature is useful for distinguishing planar objects such as buildings form non planar ones such as trees.

Density: Some objects such as road and sky, have lower density of point cloud as compared to others such as trees and vegetation. Therefore, the number of 3D points in a patch is used as a strong cue to distinguish different classes.

Intensity: LiDAR systems provide not only positioning information but also reflectance property, referred to as intensity, of laser scanned objects. This intensity feature is used in our system, in combination with other features, to



Figure 4: Removing occluded points. The top image shows 3D LiDAR point cloud in NED system. The occluded points in the one bystreet are shown in a green circle. The Bottom image illustrates camera view of scene, occluded points in the bystreet located in the red square (which corresponding to red line in top image) will be deleted.

classify 3D points. More specifically, the Median intensity of points in each patch is used to train the classifier.

3.2. Classifier

The Boosted decision tree [3] has demonstrated superior classification accuracy and robustness in many multi-class classification tasks. Acting as weaker learners, decision trees automatically select features that are relevant to the given classification problem. Given different weights of training samples, multiple trees are trained to minimize average classification errors. Subsequently, boosting is done by logistic regression version of Adaboost to achieve higher accuracy with multiple trees combined together.

In our experiments, we boost 20 decision trees each of which has 10 leaf nodes. This parameter setting is similar to those in [7], but with slightly more leaf nodes since we have more classes to label. The number of training samples depends on different experimental settings, which are elaborated in Section 4.

4. Experimental Results

Extensive classification experiments have been performed using point clouds generated with stereo vision techniques [2,17]. Since no labeled dataset consisting of corresponding LiDAR point cloud was available, we

		TT 1. 1.0					
Drive Country	Finland	United States	France				
(City)	(Helsinki)	(Chicago)	(Paris)				
Approx. Lat, Long	60.1°, 24.9°	41.9°, -87.6°	48.8°, 2.4°				
Size of Data (GB)	2.8	4.1	3.4				
Number of Images	50	110	40				
Rate (frame/meter)	1/10	1/15	1/10				
Temperature	18.5°c	34°c	5°c				
Weather Condition	0		0000				
	Sunny	Partly Cloudy	Rainy				



Figure 5: Dataset properties

created and used labeled dataset of driving sequences from 'NAVTEQ True' for all experiments presented in this paper. The dataset includes 200 high quality cubic images and corresponding accurate LiDAR point clouds collected from three different US and European cities. We selected challenging NAVTEQ drives (see figure 5) to evaluate our algorithm in different weather conditions (cloudiness, temperature and daytime) and city landscapes (shape of the buildings vegetation and vehicles). 10 semantic object classes are defined to label the database: *building, tree, sky, car, sign-symbol, pedestrian, road, fence, sidewalk* and *water*.

Note that some of these classes e.g. building, road and sky are common objects in street view images while others such as water, fences etc. occur less frequently. The statistics of occurrences of each class are summarized in Figure 5 as well.

4.1. Data Acquisition

LiDAR data is collected by NAVTEQ True using mobile mapping system consistsing of [14]: a 360 degree LiDAR sensor (Velodyne HDL-64E), a panoramic camera (Ladybug 3), six high-resolution cameras, Inertial Measurement and Unit (IMU), GPS, Distance Measurement Instrument (DMI) (see Figure 2). The LiDAR sensor consists of 64 lasers mounted on upper and lower blocks with 32 lasers in each side and the entire unit spins, and generates over 1.5 million points per second. The panoramic camera covers more than 80 percent of a full sphere with six high quality 1600×1200 Sony CCD sensors, and provides up to 12 MP images at 15 fps. The vehicle is driven at the posted speed limit and the sensors are calibrated and synchronized to produce a coupled collection of high quality geo-referenced (i.e. latitude, longitude and altitude) data. The perspective camera image is generated by rendering the spherical panorama with a view port of 2032×2032 pixels.

4.2. Evaluation with NAVTEQ True Database

We train boosted decision tree classifiers with sample 3D features extracted from the training images. Subsequently, we test the performance of the trained

classifier using separated test images. The same tests are applied to three different urban areas. The accuracy of each test is computed by comparing the ground truth with the scene paring results. We report *global* accuracy as the percentage of superpixels correctly classified, *per-class* accuracy (the normalized diagonal of the confusion matrix) and *class average* which represents the average value of per-class accuracies. Since in each experiment, dataset randomly have been divided to two groups of training and testing categories we repeated each experiment five times and the average of resulted experiment represented as the final accuracy.

Direct training and testing: We randomly split each city dataset into two groups in such a way that 70 percent of the images are used for training the classifier and the remaining 30 percent for testing. Table 1 shows the confusion matrixes for different experiments in three cities. As can be seen, some classes in Chicago and Helsinki experiments have not been labeled because there are no sufficient samples for those classes. Our algorithm performs well on most per class accuracies, with the highest accuracy 99% achieved for the sky in Chicago and the lowest 32% for sign-symbol in Paris. The average of the global accuracy for three direct experiments is about 88 %.

Mixed training and testing: The whole 200 images collected from three cities are randomly mixed and then split into 150 for training and 50 for testing. The mixed classification confusion matrix is shown in table 2. It should be noted here that some of the classes have a distinctive geometry and can be classified more easily (e.g., sky and road) whereas others have similar geometrical features (e.g., Fence and building).

Mixing images from different cities poses serious challenges to the parsing pipeline, which is reflected by the decrease in the class average accuracy (down to 59%).

Nevertheless, it seems our system generalizes well to different city scenes and the comparable global accuracy 88% is still maintained.

Cross training and testing: The idea of cross training and testing is to challenge the system with training and testing images taken from different cities. As expected, our method works well when training in Helsinki and

uris	ky	ding	ad	ree	ar	walk	n-S	strian	ater																
Pa	s	Bui	Я	т	0	Side	Sig	Pede	W.	cago	ky	lding	oad	ree	ar	walk	n-S	nce		1	1	1	1	1	
Sky	75	4	21	0	0	0	0	0	0	Chi	s	Bui	ä	Т	Ŭ	Side	Sig	Fe			5				¥
Building	5	90	0	3	0	1	0	0	1										sin	<u>k</u>	ibi	oad	ree	a.	ewa
Road	1	0	01	0	2	6	0	0	0	Sky	99	1	0	0	0	0	0	0	He	So .	Bui	В	Т	0	Side
Roud	1	U	91	U	2	U	U	U	0	Building	12	84	0	2	0	1	0	1			_				•1
Tree	5	2	0	88	0	5	0	0	0	Road	1	0	97	0	0	1	0	0	Sky	95	4	0	1	0	0
Car	2	3	55	0	33	7	0	0	0			v	"	U	v	•	U	U	Duilding				-		
		-		_			-	_	-	Tree	10	32	0	57	0	0	0	1	Building	4	88	U	7	0	1
Sidewalk	2	1	3	1	1	91	0	0	1	Car	5	10	24	0	46	13	0	2	Road	1	0	96	0	2	1
Sign- S	5	18	14	10	0	25	32	0	6	Sidewalk	3	13	7	0	10	67	0	0	Tree	1	25	0	74	0	0
Pedestrian	16	24	0	4	0	0	0	47	9	Sign- S	5	14	0	6	0	34	41	0	Car	10	4	10	0	64	12
Water	48	5	0	3	0	3	0	0	41	Fence	7	40	0	1	4	1	0	47	Sidewalk	2	15	0	0	26	58

Table 1: Confusion matrices for direct classification in Paris, Chicago and Helsinki

mixed	Sky	Building	Road	aərT	Car	Sidewalk	Sign- S	Fence	Pedestrian	Water
Sky	96	2	0	2	0	0	0	0	0	0
Building	4	90	0	3	0	2	0	1	0	0
Road	2	0	96	0	1	1	0	0	0	0
Tree	6	17	0	74	0	3	0	0	0	0
Car	5	11	35	1	35	11	0	2	0	0
Sidewalk	2	4	12	1	4	77	0	0	0	0
Sign- S	8	2	5	4	3	60	17	0	0	1
Fence	5	37	0	3	4	1	0	49	0	1
Pedestrian	10	34	1	3	3	21	0	0	22	6
Water	48	6	1	5	1	5	0	1	0	33

Experiments \Results	Global Accuracy	Class average
		Accuracy
Direct (Helsinki)	86 %	79 %
Direct (Chicago)	93 %	67 %
Direct (Paris)	85 %	65 %
Mixed	88 %	59 %
Cross (Helsinki-Chicago)	79 %	52 %
Cross (Chicago-Helsinki)	69 %	42 %
Cross (Helsinki-Paris)	59 %	36 %
Cross (Paris-Helsinki)	64 %	41 %
Cross (Chicago-Paris)	61 %	37 %
Cross (Paris-Chicago)	68 %	45 %

 Table 2: Scene parsing statistical results, Right table shows confusion matrix for mixed classification.

 Right table compares global and class average accuracy in whole different experiments.

testing in Chicago (79 % global and 52 % class average accuracy) and vice versa (69% global, 42% class average). Comparing to other cross experiments, Chicago and Helsinki cross experiments represent best parsing accuracy because as discussed earlier there are more similar classes compared to Paris which contains major water in its scene.

Applying SP based segmentation to relatively small classes such as pedestrian and sign-symbol often leads to insufficient number of training samples, and hence, low classification accuracies. The plot in Figure 6 illustrates the qualitative comparison between per class accuracy according to their distribution in our datasets. It should be noted that sky, building, road and tree were well recognized in the street scene (all are over 70%). On the other hand, cars and pedestrian have less than 10% accuracies because these classes occur very rarely in the test images. One possible remedy is to obtain the bounding boxes of these objects with a more suitable technique, e.g. part-based object detector [4].

Our system takes advantage of geographical and intensity statistics information of LiDAR point clouds, which is not available for existing methods e.g. in [2,17].







Figure 6: Top plot compares the accuracy of mixed classification based on distribution of existing data. Bottom bar graph shows the impact of intensity feature in mixed training-testing experiment.



Figure 7: Scene parsing qualitative results. (Left to right): test image, ground truth, parsing result in direct classification, in Helsinki, Chicago and Paris respectively from top to bottom

The bar chart in figure 6 shows that using intensity feature improves classification accuracies, to various extents, for objects e.g. building, car, and signs-symbol and pedestrian. There also seems a discernible increase in its effectiveness as objects become closer to the laser scanner.

5. Conclusions

We have proposed a novel framework for semantic parsing of street view images based on 3D features derived from terrestrial LiDAR point cloud. During the offline training phase, geometrical features associated with 3D patches are extracted and are used to train boosted decision trees classifier. For new input images, the same geometrical features are extracted and semantics labels are assigned to corresponding image regions. One noticeable advantage of our method is the robustness to different lighting condition, shadows and city landscape. Furthermore, by using intensity information from LiDAR data the robustness of classifier is increased for certain object classes. Future work will focus on the combination of neighboring patches features, to improve the robustness and accuracy of the classification algorithms.

References

[1] S. Becker, and N. Haala, "Combined Feature Extraction for Facade Reconstruction," ISPRSWorkshop on Laser Scanning, 2007.

[2] G.J. Brostow, J. Shotton, J. Fauqueur, and R. Cipolla, "Segmentation and Recognition Using Structure from Motion Point Clouds," ECCV, 2008.

[3] M. Collins, R. Schapire, and Y. Singer, "Logistic regression, adaboost and bregman distances," Machine Learning, vol. 48, no. 1-3, 2002.

[4] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan, "Object Detection with Discriminatively Trained

Part Based Models," Pattern Analysis and Machine Intelligence, IEEE Transactions on, Vol. 32, No. 9. (September 2010), pp. 1627-1645

[5] G. Floros, and B. Leibe, "Joint 2D-3D Temporally Consistent Semantic Segmentation of Street Scenes," CVPR, 2012.

[6] M. A. Fischler, R. C. Bolles, "Random Sample Consensus: a Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," Communications of the ACM, vol. 24, no. 6, pp. 381-395.

[7] D. Hoiem, A. A. Efros, and M. Hebert, "Recovering Surface Layout from an Image," IJCV, vol. 75, no. 1, 2007.

[8] L. Liu, and I. Stamos, "Automatic 3d to 2d Registration for the Photorealistic Rendering of Urban Scenes," In Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05), vol. 2, p. 137–143, 2005.

[9] A. Levinshtein, A. Stere, K. N. Kutulakos, D. J. Fleet, S. J. Dickinson, and K. Siddiqi, "TurboPixels: Fast Superpixels Using Geometric Flows," IEEE Trans. on Pattern Analysis and Machine Intelligence, vol. 31, no. 12, p. 2290–2297, 2009.

[10] C. Liu, J. Yuen, and A. Torralba, "Nonparametric Scene Parsing via Label Transfer," IEEE Trans. Pattern Anal. Mach. Intell., vol. 33, no. 12, pp. 2368-2382, 2011.

[11] R. B. Rusu, "Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments,"Articial Intelligence (KI - Kuenstliche Intelligenz), 2010.

[12] E. Saber, A. M. Tekalp, R. Eschbach, and K. Knox, "Automatic Image Annotation Using Adaptive Color Classification", CVGIP: Graphical Models and Image Processing, vol. 58, pp. 115-126, 1996

[13] M. Turtinen, and M. Pietikainen, "Contextual analysis of textured scene images," 17th British Machine Vision Conference, BMVC, 2006.

[14] R. Wang, F.P. Ferrie, and J. Macfarlane, "Automatic Registration of Mobile LiDAR and Spherical Panoramas," CVPR Workshops, pp. 33-40, 2012.

[15] R. Wang, J. Bach, J. Macfarlane, and F.P. Ferrie, "A New Upsampling Method for Mobile LiDAR Data," WACV, pp. 17-24, 2012.

[16] J. Xiao, and L. Quan, "Multiple View Semantic Segmentation for Street View Images," in Proc. of Intl. Conf. on Computer Vision, 2009.

[17] C. Zhang, L. Wang, R. Yang, "Semantic Segmentation of Urban Scenes Using Dense Depth Maps," in European Conference on Computer Vision, 2010.

[18] G. Zhang, J. Jia, T.T. Wong, and H. Bao, "Consistent Depth Maps Recovery from a Video Sequence," IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), vol. 31, no. 6, pp. 974-988, 2009.