2D/3D Sensor Exploitation and Fusion for Enhanced Object Detection

Jiejun Xu
HRL Laboratories LLC
jxu@hrl.com

Kyungnam Kim
HRL Laboratories LLC
kkim@hrl.com

Zhiqi Zhang
HRL Laboratories LLC
zhangzhiqi.lg@gmail.com

Hai-wen Chen
HRL Laboratories LLC
Hwc1234@gmail.com

Yuri Owechko
HRL Laboratories LLC
yowechko@hrl.com

Abstract

This paper describes a method for object (e.g., vehicles, pedestrians) detection and recognition using a combination of 2D and 3D sensor data. Detection of individual data modalities is carried out in parallel, and then combined using a fusion scheme to deliver the final results. Specifically, we first apply deformable part based object detection in the 2D image domain to obtain initial estimates of candidate object regions. Meanwhile, 3D blobs (i.e., clusters of 3D points) containing potential objects are extracted from the corresponding input point cloud in an unsupervised manner. A novel morphological feature set Morph166 is proposed to characterize each of these 3D blobs, and only blobs matched to predefined object models are kept. Based on the individual detections from the aligned 2D and 3D data, we further develop a fusion scheme to boost object detection and recognition confidence. Experimental results with the proposed method show good performance.

1. Introduction

In this work, we propose a sensor fusion method for enhanced object detection and recognition in outdoor urban environments. The input consists of a 2D image captured with an EO (electro-optical) sensor and a 3D point cloud captured by a Lidar sensor such as the Velodyne-64 (See Figure 1). We assume the sensors are pre-calibrated, and the 2D and 3D data are aligned. This means for each point of the 3D point cloud, there is a corresponding point within the 2D image based on rigid body transformation. Given EO images with appearance information such as color, texture, and gradient information, and 3D point clouds with accurate depth (distance) information, the main goal is to leverage both for improved object detection and recognition. Our method can be used for a variety of different ground objects such as pedestrians, cyclists, cars, trucks, or buses, but we chose to detect car objects, which are widely available in many public datasets.

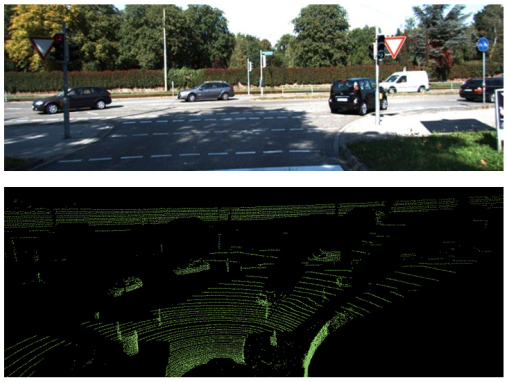

Figure 1. Sensor inputs to our system: A 2D image (top) and the corresponding 3D point cloud (bottom). Both of the data are part of the Kitti dataset.

Many 2D and 3D fusion methods have been proposed for the task of object detection in the past. In general, most existing techniques fall under two categories: *indoor* and *outdoor*. The former primarily focuses on utilizing small RGB-D(e.g., Kinect, Xtion Pro) or range sensors in conjunction with 2D cameras to improve object detection in a close-range indoor setting. The latter typically uses larger and more sophisticated 3D Lidar sensor along with 2D sensors to improve detections in a wide-range outdoor setting.

For indoor object detection, Bo et al. [2] introduced a generic approach based on hierarchical kernel descriptors to unify both 2D and 3D features to improve detection accuracy. Bar-Hillel et al. [1] proposed an integrated system to fuse image intensity and range information at multiple levels for improved object classification. Specifically, high-level fusion at the classifier level as well as low-level fusion of local descriptors were jointly explored. Collet et al. [5] developed a framework to perform indoor scene

segmentation that preserves physical objects using both 2D appearance and 3D shape data. In addition, a novel mid-level fusion technique based on the concept of *regionlet* was proposed. Lai et al. [13] introduced a detection-based approach to fuse the HoG (Histograms of Oriented Gradients) [6] features on both 2D and depth images to achieve accurate scene labeling and improve the robustness of object detection. Spinello et al. [21] proposed an adaptive hierarchical fusion approach to address the multi-modal object detection problem. For each modality, a weight function is computed using Gaussian Process that reflects the confidence of the respective detection. Other related techniques can be found in [3, 20]. In addition, a large-scale RGB-D benchmark dataset [14] consisting of a variety of indoor objects is also available for experiment on fusion-based detection.

Object detection in outdoor settings is often related to the problem of urban scene parsing, where the goal is to identify objects such as vehicle and pedestrians. Guo et al. [10] proposed a hierarchical road understanding system based on sensor fusion (i.e., Velodyne and monocular cameras) for intelligent vehicles. Their system consisted of a set of parallel modules running simultaneously to perform planning, object identification and tracking. Häselich et al. [12] presented an approach to fuse data from a Lidar sensor and three cameras with a Markov random field for terrain classification. The result of the system is an annotated 2D class grid for an autonomous system to navigate in unstructured environments. A similar fusion-based terrain classification system was developed in [15]. Munoz et al. [17] addressed the problem of outdoor scene understanding with multiple modalities when there is not a unique correspondence between data points across modalities. They proposed to treat different modalities as class objects and introduced a joint inference procedure that couples the predictions among all of the modalities. Zhou et al. [23] proposed a method to fuse laser point cloud and visual images at the data level using a reconstruction algorithm. They specifically addressed the problem of false depth assignment for visual image and incorrect colorization for laser points which result from different sensor viewpoints. Zhao et al. [22] proposed a fuzzy logic inference framework with MRF (Markov Random Field) based temporal fusion for scene parsing. Their method not only incorporates data from multiple sensors, but also from external scene knowledge. Finally, some of the 2D and 3D fusion techniques have already been integrated to real-world driving systems such as [19] and [4] for object detection.

Inspired by the success of prior work, we address the problem of object detection in outdoor urban environments with a 2D/3D fusion-based approach. The three major steps in our approach are:

1) Object detection within 2D images: Perform object (over) detection on 2D EO images using DPMs (Discriminatively Trained Deformable Part-based Models) [7] to generate an initial estimate of object candidate regions.

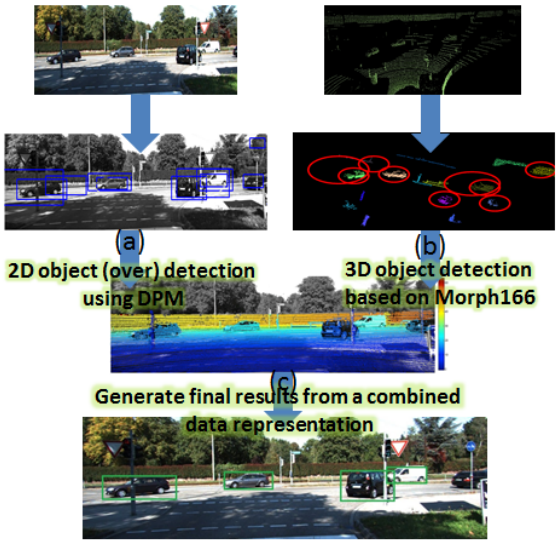

Figure 2. The overview of the object detection using 2D and 3D fusion method

2) Object detection within 3D point clouds: Extract 3D blobs from the input point cloud in an unsupervised manner through clustering. A novel morphological feature set *Morph166* is proposed to characterized each of the 3D blobs. Blobs matched to predefined objects are kept.

3) Fusion of 2D and 3D detection results: Each detected object (from both 2D and 3D input) is associated with a confidence score indicating the likelihood of the object. Detections from both modalities are projected to a common data space and subsequently combined to generate the final detections based on the fused confidence scores. The overall object detection pipeline of the proposed method is shown in Figure 2.

The rest of the paper is organized as follows. Section 2 shows the 2D object detection step with DPMs. Section 3 shows the details of object detection within a 3D point cloud. Specifically we introduce a novel morphological feature set to capture the characteristics of 3D blobs. Section 4 describes a fusion scheme to combine results from both 2D and 3D domains to boost object detection accuracy. Finally Section 6 concludes the paper.

2. 2D Object detection with DPMs

Discriminatively-trained Deformable Part-based Models(DPMs) were first introduced by Felzenszwalb et al. [7] and have shown remarkably good results for category-level object detection. Basically the method enriches the Dalal-Triggs model [6] by using a star-structured model defined

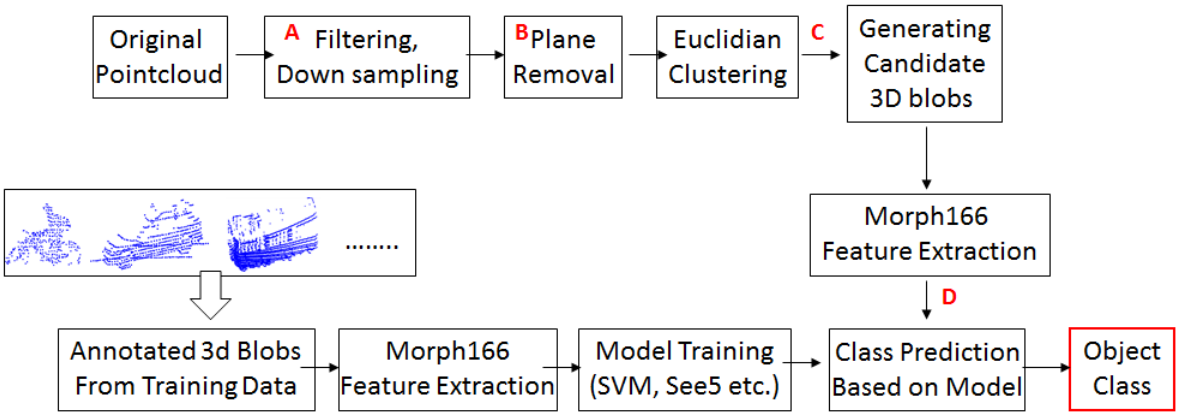

Figure 4. Block diagram of the 3D processing pipeline.

jointly by a "root" filter (analogous to the Dalal-Triggs filter), multiple higher resolution part filters, and a spatial model for the location of each part relative to the root. The DPM detector first finds a global match for the object using the root filter, and then uses its part filters and spatial models to fine-tune the result. In our experiment, we use the pre-trained DPM model (for car) from the Kitti [9] Data sets in order to have a fair comparison. The DPM model is trained by a Latent SVM.

For each frame from the EO sensor, we perform over detection, which means we keep many more 2D object detection boxes than the number of expected objects. As shown in Figure 3, the rectangular bounding boxes are the 2D object detection boxes obtained from the DPM detection. The red bounding box is the detection box with the highest confidence score, followed by the green and blue detection boxes. However, a lot of false positives (in black boxes) are retained in this step. This is to ensure as many objects (i.e., cars) are detected as possible. Most of the irrelevant detections will be filtered out later during the 2D and 3D fusion steps.

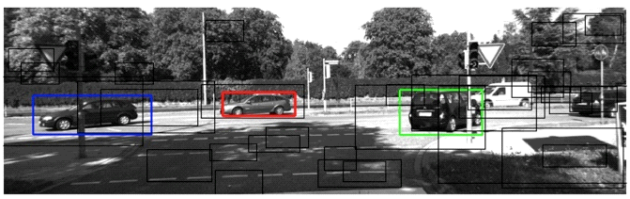

Figure 3. DPM object over detection results. The Red box has the highest detection score, the green box and the blue box has the second and third highest detection score.

3. 3D Object detection with Morph166

Given a point cloud acquired by a Lidar sensor, the 3D detection pipeline starts with down-sampling the point cloud to yield a more compact capture of the scene. The ground plane is then estimated, and 3D blobs above ground are extracted through clustering. Subsequently, morphology-based features are extracted from these 3D blobs. Finally, these blobs are classified according to a set of pre-defined classes. The overall step of 3D processing is shown in Figure 4.

A. Downsampling: In order to filter and downsample the point cloud, a typical voxelized grid approach is taken. A 3D voxel grid is essentially a set of fixed-width 3D boxes in space over the input point cloud data. In each voxel, all the points will be approximated by their centroid. A 3D voxel grid can be created efficiently with a hierarchical Octree [16] data structure. Each Octree node has either eight children or no children. The root node describes a cubic bounding box which contains all points. At every tree level, this space is further subdivided by a fixed factor, which results in an increased voxel resolution. In this work, we utilize the VoxelGrid functionality implemented in the Point Cloud Library[1] (PCL). The size of each voxel is fixed at 0.1 meter. A significant portion of the points are removed by the end of this step.

B. Ground plane removal: In this step, we extract the ground surface from the downsampled point cloud. The ground surface is important as it serves as the key reference for various geometrical estimates. This step is essentially done by fitting a planar model to the point cloud and finding the ones with sufficient number of points. To speed up the search process, Random Sample Consensus (RANSAC) [8] algorithms is used to generate plane model hypotheses. The Point Cloud Library provides a convenient

[1]Point Cloud Library, `http://www.pointclouds.org/documentation/tutorials/planar_segmentation.php`

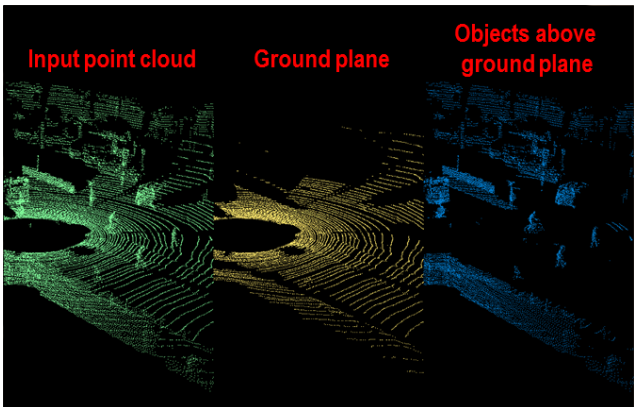

Figure 5. Visual example of removing ground surface from an input point cloud. Objects above the ground plane are kept an passed to the next step of the 3D processing pipeline.

implementation to extract the planes in their parameter form $ax+by+cz+d=0$. Planes are extracted according to their size in a sequential order. At each iteration, the set of points (inliers) aligned with the model hypotheses is selected as the support for the planar model, and they are archived and removed from the point cloud. The remaining points will be used to identify the next best plane. This process continues to detect planes and remove points until the the size of the point cloud reaches a certain threshold. Note that, for each detected plane, an additional step is taken to project all inlier points to the plane such that they lie in a perfect plane model. This makes subsequent computation more efficient and less error prone.

Given the extracted planes, the next step is to identify and remove the one that corresponds to the ground surface. Recall that the normal of a plane can be computed using the planar model coefficient directly. Let us assume two planes $\mathbf{n_0}$ and $\mathbf{n_1}$:

$$\begin{aligned} \mathbf{n_0} &= < a_0,\ b_0,\ c_0 > \\ \mathbf{n_1} &= < a_1,\ b_1,\ c_1 > . \end{aligned} \tag{1}$$

The angle θ between two planes is related to the normals of the planes as follows:

$$\mathbf{n_0} \bullet \mathbf{n_1} = ||\mathbf{n_0}||\ ||\mathbf{n_1}|| \cos\theta. \tag{2}$$

Given the plane $(y=0)$ with normal $\mathbf{n_0} = < 0, 1, 0 >$, the angle between the plane $\mathbf{n_1} = < a_1, b_1, c_1 >$ and $\mathbf{n_0}$ is computed as:

$$\theta = \arccos\left(\frac{b_1}{\sqrt{a_1^2 + b_1^2 + c_1^2}}\right) 180/\pi. \tag{3}$$

Thus, the ground surface can be identified by computing the angles between all planes with respect to $\mathbf{n_0}$, and keeping the ones which are parallel to $\mathbf{n_0}$. In our implementation, we allow a $\pm 5^\circ$ to compensate for possible sensor movements. An example of ground plane removal is shown in Figure 5. Subsequent steps will operate on points/objects above the ground surface.

C. 3D blob extraction: Given the point cloud above ground, clustering is used to divide the cloud into smaller parts in order to generate candidate object blobs for recognition. Most of the existing clustering methods rely on spatial decomposition techniques that find subdivisions and boundaries to allow the data to be grouped together based on a measure of "proximity. However, these methods are useful only for applications requiring equal spatial subdivisions. For situation where clusters can have very different sizes, a more complex algorithm is needed. Specifically, the algorithm needs to understand what an object point cluster is and what differentiates it from another point cluster. Here we define a cluster as follows.

Let $O_i = p_i \in P$ be a distinct point cluster from $O_j = p_j \in P$ if $\|p_i - p_j\| > d_{th}$, where d_{th} is a maximum imposed distance threshold. The above equation states that if the minimum distance between a set of points O_i and another set O_j is larger than a given distance value, then the points in O_i are set to belong to one point cluster and the ones in O_j to another distinct point cluster [18]. From an implementation point of view, it is important to have a notion of how this minimal distance between the two sets can be estimated. A solution is to make use of approximate nearest-neighbors queries via kd-tree representations. This allows for fast generation of clusters in an unsupervised manner. After initial clusters are extracted, an additional filtering step is performed to remove overly small / large 3D clusters. The ones which survive the filtering step are considered 3D candidate object blobs, and are passed to the next step in the pipeline for feature extraction and classification. Figure 6 shows the candidate 3D blobs generated after the clustering and filtering step.

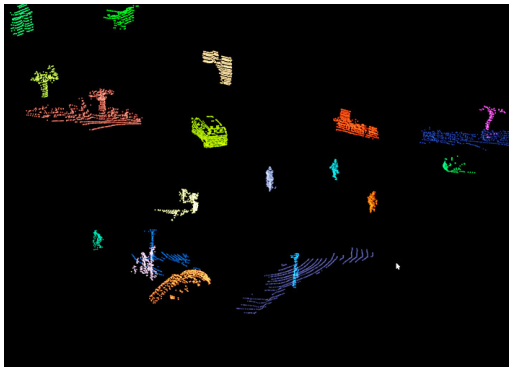

Figure 6. An example of 3D blobs obtained with the clustering-based approach applied to the point cloud without ground plane. Colors are mapped to segmented blob IDs.

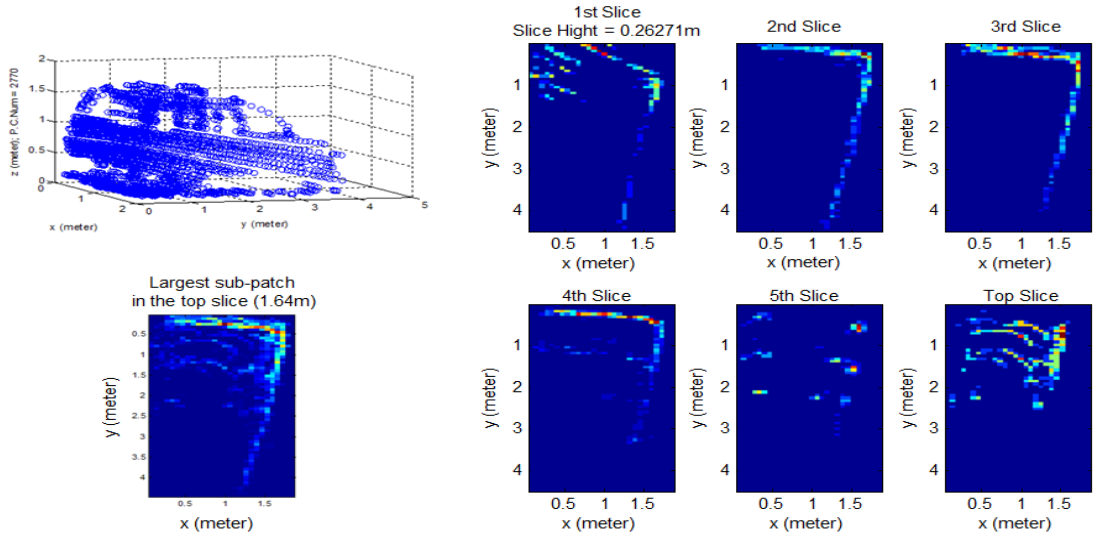

Figure 7. An examples of projecting various slices of a 3D blob into 2D images

D. 3D blob classification: It is challenging to extract robust features from a 3D object for recognition. The main reason is that the point cloud data are irregularly scattered in the 3D space, as opposed to the regularly and uniformly sampled 2D images. The point density is also reduced for objects further from the sensor. To address this issue, we propose a novel morphological feature set named *Morph166* to characterize each 3D blob. The basic idea is to project the 3D blob into multiple horizontal 2D image slices at various heights. The 2D slices contain sufficient 3D shape information of the object if slices are sampled within a close range (similar to CT/MRI scanned slices). The 2D image slices are regularly spaced images, and thus all the available image processing techniques can be applied to process these image slices, such as spatial filtering, view-invariant feature extraction, and other operations. Furthermore, the 2D image resolution is adaptively selected depending on the 3D point cloud density to avoid forming ill-conditioned images where the point cloud data are very sparse. In general, our adaptive sampling scheme allows us to deal with point cloud data with as few as 50-100 points per object. In this work, each 3D blob is decomposed into six slices, and seven morphological features along with other moment invariant features are extracted from each slice, resulting in a 166 dimensional feature vector. The seven morphological features extracted are as follows.

Pixel number: the actual number of pixels in the slice.

Bounding box: the smallest rectangle containing the pixel region in the slice.

Centroid: the center of mass of the pixel region.

Major-Axis-Length: a scalar specifying the length (in pixels) of the major axis of the ellipse that has the same normalized second central moments as the pixel region.

Minor-Axis-Length: a scalar specifying the length (in pixels) of the minor axis of the ellipse that has the same normalized second central moments as the pixel region.

Eccentricity: specifies the eccentricity of the ellipse that has the same second-moments as the pixel region.

Extent: specifies the ratio of pixels in the region to pixels in the total bounding box.

In summary, the first 10 element of the *Morph166* feature vector consist of the Centroid and Orientation difference between slices. The next 7 elements consist of the aforementioned morphological features extracted from the largest sub-patch in the top slice. Immediately following are 42 elements corresponding to the seven morphological features extracted from each of the six 2D slices. Figure 7 shows examples of 2D slices obtained for a 3D blob of a car object. Finally, the rest of the 107 elements in the *Morph166* feature are listed below. Please refer to [11] for details about the moment invariants.

- height (z), length (x), width (y)
- x_mean, y_mean, z_mean
- x_std, y_std, z_std
- Seven moment invariants for all slices
- Pixel number difference ratio between slices
- Pixel area size (in meter) difference between slices
- Bounding box area size difference ratio between slices
- Patch centroid (x, y) position difference between slices
- Patch length-to-width ratio difference between slices
- Moment invariants difference between slices
- Sub-patch number in all slices
- Largest sub-patch area to full area ratio in all slices
- 2nd largest sub-patch area to full area ratio in all slices
- Area ratio between the top two sub-patches in all slices

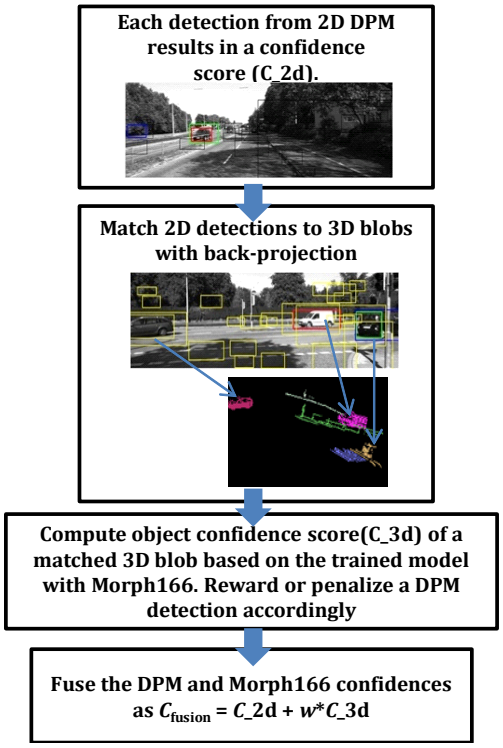

Figure 8. Illustration of 2D/3D fusion process.

Given the computed feature associated with each 3D blob, a standard supervised learning paradigm is adapted in our system for 3D object classification. For each class of object (e.g., car), a SVM model is trained in a one-against-all fashion. When a new 3D candidate blob is generated, it will be fed to the models trained for different classes. The one model giving the highest confidence score will be identified, and the 3D blob will be classified as the corresponding class.

4. 2D/3D Fusion for Improved Detection

We propose a simple fusion technique based on linear combination of the confidence scores of 2D and 3D detections. The overall 2D/3D fusion steps are illustrated in Figure 8. We start with identifying the correspondences between 2D and 3D detections. Recall that our input contains synchronized 2D and 3D data, thus it is straight forward to project detection results between 2D and 3D space. For each 2D detection bounding box, we search through all the rectangular projections (see yellow boxes in Figure 8) of the detected 3D blobs onto the 2D image space. The 3D blob projection that has the maximal overlap with the 2D bounding box is considered the matched correspondence. The overlap between the 2D bounding box and 3D blob projection is calculated as follows

$$overlap = \frac{R_{rect} \cap R_{blob\_proj}}{R_{rect} \cup R_{blob\_proj}}, \tag{4}$$

where R_{rect} is the area of the 2D rectangular bounding box resulted from DPM detection, and R_{blob_proj} is the area of the 3D blob projection.

If a correspondence is found between a pair of detections with 2D confidence score as C_{2D} and 3D score as C_{3D}, we set the new fusion score as $C_{fusion} = C_{2D} + \omega C_{3D}$. If a correspondence is not found, then we penalize 2D detection by setting $C_{fusion} = C_{2D} - \alpha$. The value of ω and α are determined empirically, and set to 0.55 and 0.4 respectively.

5. Experiment

In order to evaluate the proposed fusion-based object detection approach, we selected 6 different sequences from the standard Kitti benchmark dataset[2] for our experiment. This dataset provides full sequences of Velodyne scans in different urban settings. For each 3D scan, a pair of synchronized 2D images are provided. In our experiment, 3D blobs are first extracted from all sequences in an unsupervised manner as diescribed in Section 3. Blobs from the first half of the sequences are reserved for training, and blobs from the second half are reserved for testing. 2D images of the corresponding sequences are used for the same train/test division. We compute the Precision-Recall curves and the AP (average precision) scores for 2D only detection, 3D only detection, and fusion-based detection. Detections are considered true or false positives based on the area of overlap with ground truth bounding boxes. To be considered a correct detection, the area of overlap α_0 between the bounding box B_b of the detected object and the ground truth bounding box B_{gt} must exceed 50% according to the formular:

$$\alpha_0 = \frac{area(B_d \cap B_{gt})}{area(B_d \cup B_{gt})} \tag{5}$$

Note that for 3D detection, we only evaluate objects which are within 25 meters of the sensor. This is because points beyond that range are too sparse for reliable object classification. Other parameters are tuned to the best performance for each method. The overall experimental result is plotted in Figure 9. The proposed fusion-based method outperforms both 2D and 3D only detections substantially.

6. Conclusion

In this work, we propose a 2D/3D fusion-based object detection and recognition method for outdoor urban environment. 2D detections are obtained by state-of-the-art

[2] http://www.cvlibs.net/datasets/kitti/raw_data.php. Sequences begin with "2011_09_26_drive".

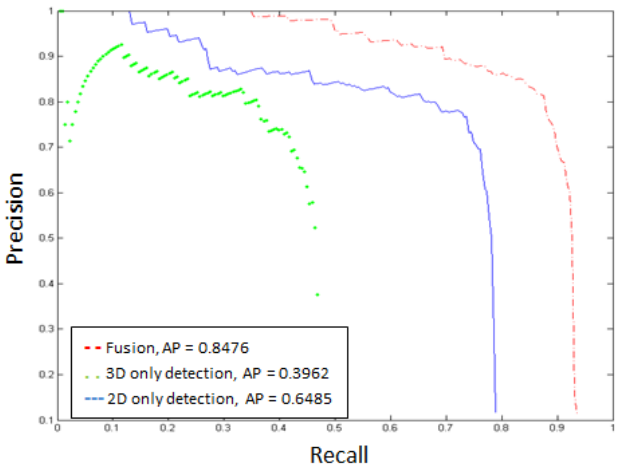

Figure 9. Precision-Recall curves and AP scores for the detection of car over 6 Kitti sequences.

DPMs detector, and 3D detections are obtained through classification of 3D blobs extracted from the scene point cloud. In particular, a novel morphological feature set *Morph166* is proposed to characterize each 3D blob. Detections from individual modalities are then combined and reinforced each other to boost the overall detection and recognition accuracy. The effectiveness of the proposed method is demonstrated with 6 outdoor sequences on the standard Kitti benchmark dataset.

References

[1] A. Bar-Hillel, D. Hanukaev, and D. Levi. Fusing visual and range imaging for object class recognition. In *Proceedings of the 2011 International Conference on Computer Vision*, ICCV '11, pages 65–72. IEEE Computer Society, 2011.

[2] L. Bo, K. Lai, X. Ren, and D. Fox. Object recognition with hierarchical kernel descriptors. In *Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition*, CVPR '11, pages 1729–1736, Washington, DC, USA, 2011. IEEE Computer Society.

[3] L. Bo, X. Ren, and D. Fox. Depth kernel descriptors for object recognition. In *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[4] H. Cho, K. B. V. Seo, Young-Woo, and R. R. (Ragunathan). A multi-sensor fusion system for moving object detection and tracking in urban driving environments. In *International Conference on Robotics and Automation*, ICRA'14. IEEE, 2014.

[5] A. Collet, S. S. Srinivasa, and M. Hebert. Structure discovery in multi-modal data: A region-based approach. In *ICRA*, pages 5695–5702, 2011.

[6] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. *Computer Vision and Pattern Recognition, 2005. CVPR 2005. IEEE Computer Society Conference on*, 1:886–893, 2005.

[7] P. F. Felzenszwalb, R. B. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. Pattern Anal. Mach. Intell.*, 32(9):1627–1645, Sept. 2010.

[8] M. A. Fischler and R. C. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 1981.

[9] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The kitti dataset. *International Journal of Robotics Research (IJRR)*, 2013.

[10] C. Guo, S. Mita, and D. McAllester. Hierarchical road understanding for intelligent vehicles based on sensor fusion. In *Intelligent Transportation Systems (ITSC)*. IEEE, 2011.

[11] R. Haralick and L. Shapiro. *Computer and Robot Vision*, volume 1. Addison-Wesley, 1992.

[12] M. Haselich, M. Arends, D. Lang, and D. Paulus. Terrain classification with markov random fields on fused camera and 3d laser range data. In A. J. Lilienthal, editor, *ECMR*, pages 153–158, 2011.

[13] K. Lai, L. Bo, X. Ren, and D. Fox. Detection-based object labeling in 3d scenes. In *ICRA*, pages 1330–1337, 2012.

[14] K. Lai, L. Bo, X. Ren, and D. Fox. Rgb-d object recognition: Features, algorithms, and a large scale benchmark. In A. Fossati, J. Gall, H. Grabner, X. Ren, and K. Konolige, editors, *Consumer Depth Cameras for Computer Vision: Research Topics and Applications*, pages 167–192. Springer, 2013.

[15] S. Laible, Y. N. Khan, K. Bohlmann, and A. Zell. 3d lidar- and camera-based terrain classification under different lighting conditions. In P. Levi, O. Zweigle, K. Huermann, and B. Eckstein, editors, *AMS*, Informatik Aktuell, pages 21–29. Springer, 2012.

[16] D. Meagher. Geometric modeling using octree encoding. *Computer Graphics and Image Processing*, 1982.

[17] D. Munoz, J. A. Bagnell, and M. Hebert. Co-inference for multi-modal scene analysis. In *Proceedings of the 12th European Conference on Computer Vision - Volume Part VI*, ECCV'12, pages 668–681, Berlin, Heidelberg, 2012. Springer-Verlag.

[18] R. B. Rusu. *Semantic 3D Object Maps for Everyday Manipulation in Human Living Environments*. PhD thesis, Computer Science department, Technische Universitaet Muenchen, Germany, October 2009.

[19] M. Skutek, T. Eisenbach, and W. Fischer. A fusion architecture for object detection using replaceable sensors. In *SAE Technical Paper*. IEEE, 2009.

[20] L. Spinello and K. O. Arras. People detection in rgb-d data. In *Proc. of The International Conference on Intelligent Robots and Systems (IROS)*, 2011.

[21] L. Spinello and K. O. Arras. Leveraging rgb-d data: Adaptive fusion and domain adaptation for object detection. In *ICRA*, pages 4469–4474, 2012.

[22] G. Zhao, X. Xiao, J. Yuan, and G. W. Ng. Fusion of 3d-lidar and camera data for scene parsing. *J. Vis. Comun. Image Represent.*, 25(1):165–183, Jan. 2014.

[23] L. Zhou. Fusing laser point cloud and visual image at data level using a new reconstruction algorithm. In *Intelligent Vehicles Symposium*, pages 1356–1361. IEEE, 2013.