2016 13th International Conference on Ubiquitous Robots and Ambient Intelligence (URAI)
August 19~22, 2016 at Sofitel Xian on Renmin Square, Xian, China

Fast Multiple Objects Detection and Tracking Fusing Color Camera and 3D LIDAR for Intelligent Vehicles

Soonmin Hwang*, Namil Kim*, Yukyung Choi, Seokju Lee and In So Kweon

Robotics and Computer Vision Laboratory, Department of Electrical Engineering,
Korea Advanced Institute of Science and Technology, Daejeon 305-701, Korea
(E-mail: {smhwang, ykchoi, nikim, sjlee}@rcv.kaist.ac.kr, iskweon@kaist.ac.kr)

Abstract - For many robotics and intelligent vehicle applications, detection and tracking multiple objects (DATMO) is one of the most important components. However, most of the DATMO applications have difficulty in applying real-world applications due to high computational complexity. In this paper, we propose an efficient DATMO framework that fully employs the complementary information from the color camera and the 3D LIDAR. For high efficiency, we present a segmentation scheme by using both 2D and 3D information which gives accurate segments very quickly. In our experiments, we show that our framework can achieve the faster speed (∼4Hz) than the state-of-the-art methods reported in KITTI benchmark (>1Hz).

Keywords - DATMO, Sensor fusion

1. Introduction

With the great success in academia and industries such as Google and Tesla, the autonomous driving and advanced driver assistance system (ADAS) are attracting the public attention. Among the various modules in the autonomous driving technology or in the ADAS, the detection and tracking multiple objects (DATMO) is one of the most important component to understand a traffic environments.

For the inputs of the intelligent driving (or assistance) system, various sensors could be considered. Generally, many researchers utilize color cameras due to its cheap and versatility characteristics. Recently, some researchers have made a breakthrough in terms of object detection performance [10–12]. Especially, for pedestrian detection, the advancement is still going on in [13]. Although the color camera is a popular choice, there is another research direction to use time-of-flight (ToF) type sensors to measure the distance from sensor. The most popular one for the intelligent vehicles is the LIDAR (LIght Detection And Ranging) sensor. Similar to the human perception system, the distance from objects can be useful in this task [14–16]. Even some researches have tried to fully utilize multiple sensors, called sensor fusion approach [17]. The sensor fusion approach is a promising direction to build a very accurate recognition system due to the complementary characteristics; e.g. different wavelength of lights or independent physical quantities.

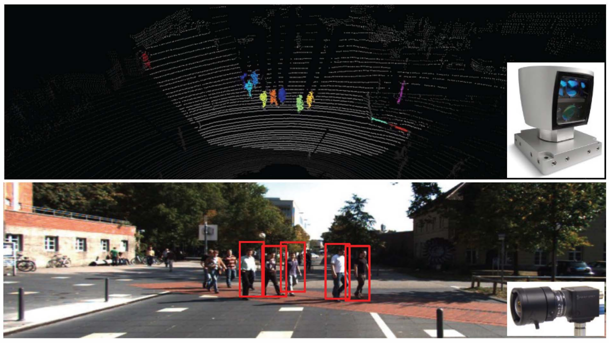


Fig. 1: The 3d points from the LIDAR (top) and a color image (bottom) are used for efficient object detection and tracking. In this paper, we consider pedestrians, cyclists, and vehicles as our main targets.

Table 1: Efficiency of the previous works.

Authors	Segmentation	Total	Publication
Gonzales et al.	·	4 sec	IV (2015) [20]
Wang et al.	·	0.5 sec *	RSS (2015) [21]
Held et al.	2.17 sec **	5 sec	RSS (2014) [24]
Moosmann et al.	2.5 sec	26.4 sec	ICRA (2013) [25]
Teichman et al.	2 sec	·	ICRA (2011) [22]
Proposed	0.04 sec	0.23 sec	·

* Low accuracy ** well-optimized

Moreover, many academic researches [15, 18, 19, 22–25] or industrial products[1],[2] have developed based on multi-sensor platforms. However, a high computational complexity is usually limited in the sensor fusion framework. As shown in Table 1, most previous works are not efficient to apply practical environments, even though we consider the fact that this comparison does not conducted on the same public dataset.

In this paper, we propose an efficient framework for object detection and tracking composed of several independent modules. In addition, we design a novel segmentation scheme including grid based projective DBSCAN followed by 2D/3D joint object proposal that lead great speed-up in our framework. Also, we adopt the sensor fusion approach to achieve better performances. We focus on the representative targets on the road, i.e. pedestrians and vehicles, however, our method could be extended for the other objects on traffic environment.

* Two authors contributed equally

[1] http://www.caranddriver.com/features/semi-autonomous-cars-compared-tesla-vs-bmw-mercedes-and-infiniti-feature-the-physiology-of-semi-autonomy-and-test-results-page-6

[2] https://www.google.com/selfdrivingcar/how/

978-1-5090-0821-6/16/$31.00 ©2016 IEEE

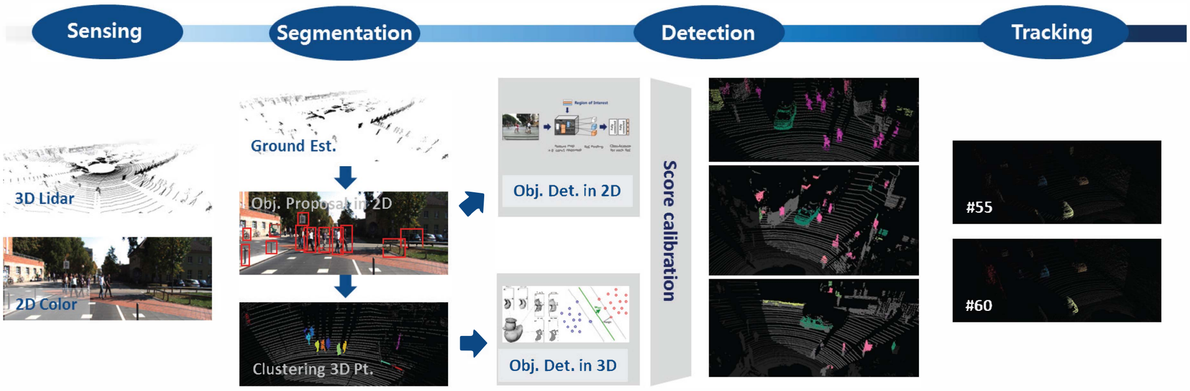


Fig. 2: Overview of the proposed framework[3]. For the inputs from camera and LIDAR, the segmentation is applied at first and then the object detection from two modalities is operated. After that, we calibrate two detection score to combine and track the object based on segment matching.

2. Proposed framework

We propose an efficient framework for detection and tracking multiple objects (DATMO). As shown in Fig 2, the proposed framework consists of four stages. First, the input data is entered from camera and 3D LIDAR. The second, the segmentation stage to prepare inputs for detection and tracking stages is followed. Then, two independent object detectors from different modalities, i.e. 3D point space and 2D image space, are applied. Finally, the tracking procedure which contains calculation of tracking features and association them is provided.

2.1 Sensing

Our framework takes inputs from two modalities. The first input is the 3D point cloud from LIDAR, especially Velodyne HDL-64E. The second is the color image which is already calibrated with the LIDAR. For efficiency, the horizontal 3d point sampling is applied as a pre-processing which is similar to run Velodyne as fast mode (20Hz). The resulting point cloud from sampling is different from Velodyne HDL-32E, because the sampled points could have 64-levels of vertical angles. In our implementation, we sampled 3d points at every one of the two horizontal points. It has little effect on the performances.

2.2 Segmentation

2.2.1 Ground plane estimation

The ground plane estimation is to handle the huge number of points from the sensor by early rejecting ground points and to make clustering in 3D space easy. As there are a lot of methods to estimate the ground plane, we focus on the efficiency with acceptable accuracy. We propose a simple iterative way to estimate the ground plane with local constraints. Our method divides entire 3D points into several groups G_i according to the distance from the sensor and estimates local planes from close plane to far plane sequentially. We make several reasonable assumptions that (1) the close points from the sensor are more likely to be the points from ground plane and that (2) the ground consists of several local planes. We employ the statistical analysis to estimate local planes. Our method starts from the close group of points G_0 to compute initial plane a_0.

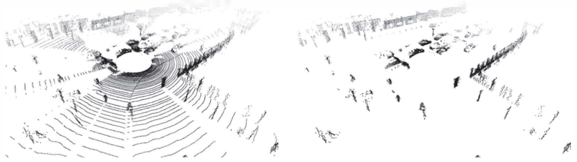

(a)Input point cloud (b)After removing ground points

Fig. 3: Result of the ground plane removal. Our simple method could effectively estimate the ground plane.

$$a_0 = \arg\min_{a_\bullet} ||a^T x_i||^2, \quad x_i \in G_0. \tag{1}$$

The points from the closest ground plane are estimated by following:

$$X_0 = \{x \mid \; |a_0^T \, x| < \delta, \quad x \in G_0\}. \tag{2}$$

For practical use, the under-clustering caused by over-estimated ground plane, i.e. clipping object, does not critically affect the detection performance. We estimate the local ground plane with soft margin using this assumption. For our implementation, we set the δ to 0.1. Then, the iterative updating is followed to estimate next local plane using current local plane prior.

$$a_t = \arg\min_{a_t} ||a^T x_t||^2, \qquad |a_t^T \cdot x_{t-1}^{far}| < \delta, \tag{3}$$

where $x_{t-1}^{far} \in X_{t-1}^{far} \subset X_{t-1}$ is the far point in previous local plane and $x_t \in G_t$ is the point in current local group. This means that the next local plane should satisfy both the conditions that it contains the outermost points in the previous local plane and that it is a solution for minimizing the plane equation. In our implementation, we simply divide the points into two groups, i.e. close plane

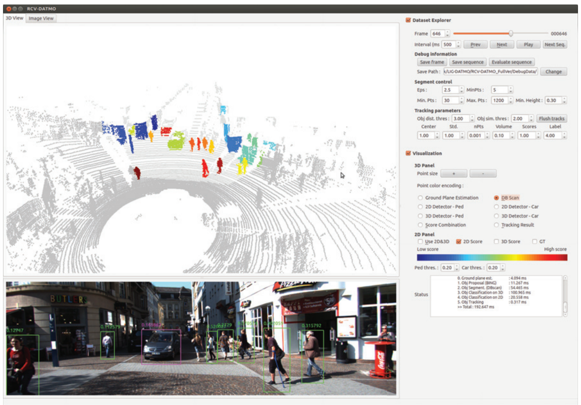
[3] The full video results can be found in following website: https://sites.google.com/site/smhwangcv/datmo_sensor_fusion

Table 2: Comparison of object proposal methods. For efficiency, the input image is resized at ×0.5 scale except BING×1.0 (First row). The performances are estimated only using top-500 regions.

Method	Avg.Time [sec]	MABO*	Recall
BING×1.0 [2]	0.45	0.4794	0.5605
BING×0.5 [2]	**0.11**	**0.4739**	**0.5279**
EdgeBox [3]	0.88	0.3081	0.2227
Geodesic [4]	0.74	**0.4965**	**0.5437**
Selective Search [5]	2.05	0.4348	0.3941

* MABO: Mean Average Best Overlap

and far plane. If we consider the steady order of the input stream from Velodyne, we can solve above equations efficiently.



☑ Dataset Explorer

Frame 646 000646

Interval (ms 500

Debug information

Save Path : k/LIG-DATMO/RCV-DATMO_FullVer/DebugData/

Segment control

Eps : 2.5 MinPts : 5

Min. Pts : 30 Max. Pts : 1200 Min. Height : 0.30

Tracking parameters

Obj dist. thres : 3.00 Obj sim. thres : 2.00

Center	Std.	nPts	Volume	Scores	Label
1.00	1.00	0.001	0.10	1.00	4.00

☑ Visualization

3D Panel

Point size

Point color encoding :

○ Ground Plane Estimation ● DB Scan
○ 2D Detector - Ped ○ 2D Detector - Car
○ 3D Detector - Ped ○ 3D Detector - Car
○ Score Combination ○ Tracking Result

2D Panel

□ Use 2D&3D ☑ 2D Score □ 3D Score □ GT

Low score High score

Ped thres. : 0.20 Car thres. : 0.20

Status

```
0. Ground plane est.          : 4.094 ms
1. Obj Proposal (BING)        : 11.267 ms
2. Obj Segment. (DBscan)      : 54.465 ms
3. Obj Classification on 3D   : 100.965 ms
4. Obj Classification on 2D   : 20.558 ms
5. Obj Tracking               : 0.317 ms
>> Total : 192.647 ms
```

Fig. 4: Result of the proposed grid-based DBSCAN method and our graphical analyses tool.

2.2.2 Object proposal in 2D image

The second step is object proposal in 2D image. This step is to separate foreground object from background. The output of this step is used for input of clustering 3d points step. In the proposed framework, as the performance of this step directly influences the entire performance, we compare several state-of-the-art object proposal methods (Table 2).

For fast object detection and tracking, this step should not put off much time and should have acceptable accuracy. As shown in Table. 2, the BING [2] offers the best trade-off if we consider both speed and accuracy.

2.2.3 Clustering 3D points

Clustering 3D points is a grouping process to form an object. It is an intermediate step from points to an encoded vector for classification. The output of clustering step, i.e. a segment, is used for inputs of 2D/3D detection. In this step, we should deal with various shapes and densities of objects. Among the numerous methods of clustering, we choose a DBSCAN (Density-Based Spatial Clustering of Applications with Noise) algorithm which could be robust due to its less-parametric characteristic. The DBSCAN has only two parameters, minimum distance to nearest point (Eps) and the number of minimum points to be a valid segment ($Minpts$). However, it has a limitation in a variable density space. Since the LIDAR sensor such as the Velodyne could see a certain face of the object, segments from the same object could have various shapes. Also, as the radiated rays to measure the distances from the LIDAR, i.e. each scanline of the LIDAR, are straight lines with constant angle between each rays, the densities of 3d points are inherently different according to the distance from the sensor.

To overcome this limitation, we propose a grid-based DBSCAN on projected 2D space. First, every point is projected on $xy - plane$ for efficiency. Due to the assumption of that the sensor keeps an upright position above the ground, this projection does not hurt the spatial relationship between objects. Then, the regular occupation grid for the projected points is computed and then the DBSCAN is applied on the occupation grid points. This grid-based quantization scheme is the essential part for handling various density. Regardless of the density, i.e. distance from the sensor, the points mapped to a single grid are considered as a single grid point. This many-for-one mapping relieves the scanned density problems. Not only this grid mapping itself is computed efficiently, but the mapping also makes the DBSCAN on the grid points efficient because the grid points have integer values.

In this step, as our grid-based DBSCAN make an exhaustive search several times for every point, it is important to reduce the number of 3D points. To keep as many 3D points from target object as possible, we use the object proposal in 2D to determine whether the 3D point is kept or not. Our clustering method is applied to remaining 3D points. Using the clustered 3D points, i.e. 3D segment, we generate more accurate region-of-interests (RoIs) by projecting them on the 2D image. This is our novel 2D/3D joint object proposal scheme.

2.3 Detection

2.3.1 Object detection in 2D image

To detect target objects in 2D image, any off-the-shelf detection algorithms could be employed. We use the *Fast R-CNN* [1] which is one of the state-of-the-art object detection algorithms without training. The *Fast R-CNN* takes an image and several regions of interest (RoIs) as inputs. Although the original authors utilize *Selective Search* [5] for their RoIs, we use more accurate RoIs from our 2D/3D joint object proposal method.

2.3.2 Object detection in 3D space

To detect objects in 3D space, we also use an off-the-shelf method to describe the segments and to classify them. For 3D features, we employ the spin image method [7, 8], which is a popular description method for 3D object recognition. This method extracts histograms locally and globally to represent a certain object. Then, the linear SVM trained on the collection of positive and negative vectors is followed. This is an efficient way to perform the detection task in 3D space.

2.3.3 Score calibration

In our framework, two independent classifiers from 2D image and 3D space are applied to the same region, i.e. the segment in 3D space. We need a score integration method to decide the final detection confidence from the two independent scores. As the two scores might have different score ranges, we should adjust the confidence score levels first. We employ the score combination method proposed in [6] to transform the scores into pseudo-probability values. We choose the logistic regression as our score calibration method.

$$p(x) = \frac{1}{1 + exp(-wx + b)}, \ \forall x \in \mathbb{R}. \tag{4}$$

The parameters w, b are determined by minimizing the negative log-likelihood objective function on validation data.

$$[w^*, b^*] = \arg\min_{w^*,b^*} \left(\sum_{k=1}^{n} L(s_k, y_k) \right), \tag{5}$$

$$L(s_k, y_k) = y_k \log(p(s_k)) + (1 - y_k) \log(1 - p(s_k)), \tag{6}$$

where L is the loss, s_k is the score for k^{th} window and y_k is the ground-truth label. We simply combine two scores from each modality using multiplication of them [6].

2.4 Tracking

Object tracking is the process of associating moving objects in the current frame and the previous frame. For the object tracking, we adopt the *segment matching based method*. Each object should be represented by corresponding entity type and distance metric should be also specified. Even though some features such as the spin images are already computed at detection step, we need to specify more discriminative feature for tracking, because it has no enough distinctiveness for the identification in terms of a tracking.

We use simple but effective features for each segment. Our feature consist of five elements as follows: the mean of the 3d points in coordinates (x-y-z, 3 dim), standard deviations (std(x)-std(y)-std(z), 3 dim), quantized color histograms (for 8-level, 8 * 3 dim), the volume size of the segment ($depth$-$width$-max($height$)-min($height$), 4 dim), and the number of 3d points (1 dim). To measure the dissimilarity between segments, we use Euclidean distance (L^2) on the feature space. We establish some rules for handling appearing/disappearing objects. At every frame, we associate a segment in the current frame to the most similar segment in previous frame. If both the physical distance and dissimilarity score are larger than α and β respectively for every segments in previous frame, we consider that the tracked object is disappeared. We do not consider ID-remapping problem. If an object is appeared again, a new ID is assigned.

3. Experiments

To evaluate the proposed framework, we conduct experiments on the popular KITTI tracking dataset [9]. Since the KITTI dataset [9] does not provide ground truth annotations for the test set, we disjointly split the train set to two subsets. We utilize the first subset for train and the other for test.

3.1 Graphical analyses tool

Since the proposed framework consists of several modules, there are many parameters to affect the entire performance even we choose less-parametric methods for each modules. We implement a graphical user interface for analyzing the effect of each module (Fig. 4). Using this tool, we could check what the fragile part is for parameter tuning, what the important part is for entire performance, and so on. Also, it could help to find better parameters.

3.2 Computation time

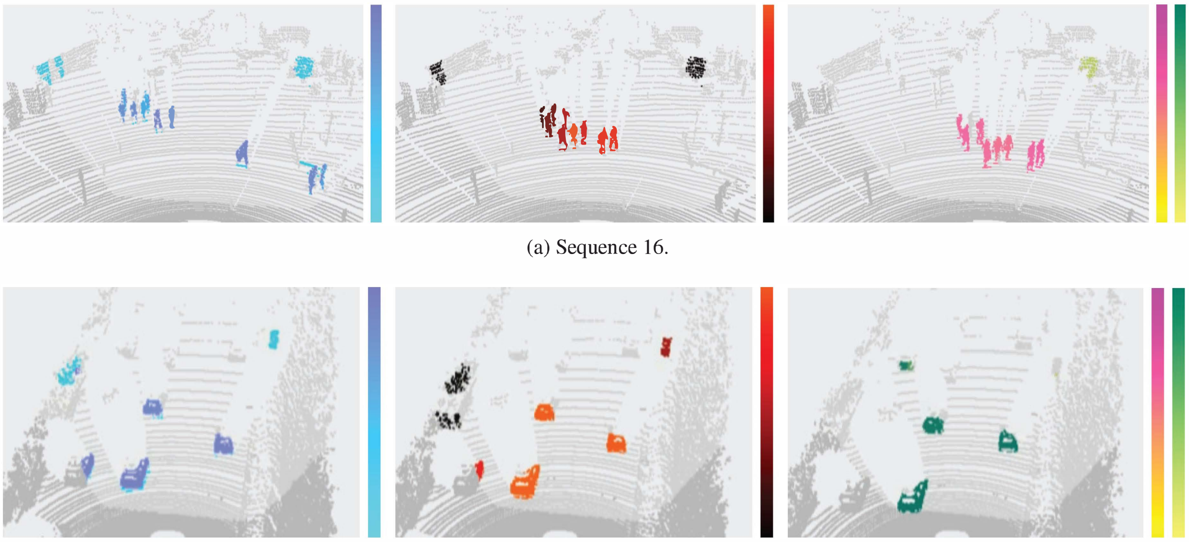
Using the above analyses tool, we estimate the computation time for each module. With our careful choices for efficient modules, the proposed framework achieves 4.5 frames per second speed. As shown in Table 3, the most time-consuming part is the object detection in 3d space (Sec. 2.3.2). In details, most of the time is spent for the calculation of the spin image feature. The spin image is a surface representation which encodes the global properties of surfaces in an object-oriented coordinate system. This representation is invariant to the rigid motion of objects, e.g. moving and rotation without deformations, due to its view-independent characteristic. The weakness is that it requires a lot of computations. For better speed, someone could replace this part to another fast description method, because there are many fast description methods for 3d recognition tasks. Thus the proposed framework has a potential to be faster.

3.3 Performance

The experimental results are summarized in Table 4. Our target objects are pedestrians (Pedestrian, Person, Cyclist) and vehicles (Car, Van, Truck, Bus). We follow the KITTI object detection criterion that more than 70% overlap is required for car and that more than only 50% overlap is required for pedestrian. For evaluating the proposed framework, we use conventional measurements such as F1-score[4], average precision (AP) and tracking error[5]. The F1-score and the average precision are used to evaluate detection performance and the tracking error is used to evaluate tracking performance.

[4] The F1-score: the harmonic mean of precision and recall

[5] The tracking error: an error ratio for object tracking



(a) Sequence 16.

(b) Sequence 20.

Fig. 5: Qualitative results. 2D detection score (Left), 3D detection score (Center), combined score (Right). Upper part of color-bar indicates high score.

Table 3: Average computation time.

Stage	Step	Time (Std.) [ms]	Ratio (%)
Segmentation	Ground Est.	4.24 (1.58)	1.83
	Obj. Proposal in 2D	10.85 (1.08)	4.68
	Clustering 3D Pt.	25.35 (11.15)	10.93
Detection	**Obj. Det. in 2D**	**143.24 (19.43)**	**61.78**
	Obj. Det. in 3D	47.09 (36.02)	20.31
Tracking	Calc. Feat. & Association	0.20 (0.05)	0.86
Total computation time		231.86 (48.56)	·

As shown in Table 4, proposed framework shows better performance in most cases when we use combined 2d and 3d data instead of only 3d data. Due to the imbalance distribution for target objects, some sequences such as seq-16 and seq-20 are used for evaluating a specific class.

Considering that the best performance for car detection is 89.72% on moderate condition, our method shows reasonable performances in terms of the trade-off between performance and accuracy. In detail, the best algorithm requires 4.5 seconds for 89.72% on moderate condition, and our method achieves 67 to 96% with 4Hz. Even it is not an exactly fair comparison, it just shows that our method could give a respectable accuracy in a short time. Also, our fusion approach leads to great improvement when the single modality data (3D or 2D) shows bad performances, e.g. Table 4-(b) Seq. 15. It means that our framework has a possibility to be more effective in tough environments.

4. Conclusion

Table 4: Qualitative evaluation on KITTI dataset.

(a) Pedestrian, Cyclists

Measurement	Seq	Targets	3D	2D	2D+3D
F1-Score	15	1289	0.4991	0.7464	**0.7660**
AP			0.2941	0.7535	**0.8123**
TrackErr.			·	·	0.2957
F1-Score	16	2299	0.9098	0.9046	**0.9141**
AP			0.8787	0.8785	**0.9101**
TrackErr.			·	·	0.2224

(b) Vehicles, Truck, Bus, Van)

Measurement	Seq	Targets	3D	2D	2D+3D
F1-Score	15	899	0.4295	0.4670	**0.6307**
AP			0.4141	0.3123	**0.6799**
TrackErr.			·	·	0.2829
F1-Score	20	6404	0.8729	0.8959	**0.9085**
AP			0.9359	0.9358	**0.9646**
TrackErr.			·	·	0.3611

In this paper, we presented an efficient detection and tracking multiple objects (DATMO) framework using color camera and 3D LIDAR. We designed a grid based projective DBSCAN and 2D/3D joint object proposal for efficiency. With the proposed fusing approach, our framework is robust for the challenging cases which are hardly recognized in single camera or 3D LIDAR. In our experiments, we demonstrated that our framework can achieve a high efficiency ($\sim$ 4Hz) with a reasonable performance in KITTI benchmark. Furthermore, our framework could be upgraded to get better performances by replacing some modules. We expect that our framework could be utilized for the real-time intelligent vehicle.

Acknowledgement

This work was supported by the Development of Autonomous Emergency Braking System for Pedestrian Protection project funded by the Ministry of Trade, Industry and Energy of Korea. (MOTIE)(No.10044775)

References

[1] Ross Girshick, ”Fast R-CNN.” IEEE International Conference on Computer Vision (ICCV), 2015.

[2] Ming-Ming Cheng, Ziming Zhang, Wen-Yan Lin, Philip Torr, ”BING: Binarized normed gradients for objectness estimation at 300fps,.” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2014.

[3] C. Lawrence Zitnick and Piotr Dollr, ”Edge boxes: Locating object proposals from edges,” European Conference on Computer Vision (ECCV), 2014.

[4] Philipp Krhenbhl and Vladlen Koltun, ”Geodesic object proposals,” European Conference on Computer Vision (ECCV), 2014.

[5] Jasper RR Uijlings, Koen E. A. van de Sande, Theo Gevers, Arnold W. M. Smeulders, ”Selective search for object recognition,” International journal of computer vision (IJCV), 2013.

[6] Philippe Xu, Franck Davoine, and Thierry Denux, ”Evidential combination of pedestrian detectors,” British Machine Vision Conference (BMVC), 2014.

[7] Andrew E. Johnson, ”Spin-images: a representation for 3-D surface matching,” Ph.D thesis, 1997.

[8] Andrew E. Johnson and Martial Hebert, ”Using spin images for efficient object recognition in cluttered 3D scenes,” IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 1999.

[9] Geiger, Andreas, Philip Lenz, and Raquel Urtasun. ”Are we ready for autonomous driving? the kitti vision benchmark suite,” IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2012.

[10] Pedro F. Felzenszwalb, Ross B. Girshick, David McAllester and Deva Ramanan, ”Object detection with discriminatively trained part-based models,” IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2010.

[11] Piotr Dollr, Ron Appel, Serge Belongie, and Pietro Perona, ”Fast feature pyramids for object detection,” IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI), 2014.

[12] Shaoqing Ren, Kaiming He, Ross Girshick, Jian Sun, ”Faster R-CNN: Towards real-time object detection with region proposal networks,” Advances in Neural Information Processing Systems (NIPS), 2015.

[13] Jianan Li, Xiaodan Liang, ShengMei Shen, Tingfa Xu, Shuicheng Yan, ”Scale-aware Fast R-CNN for Pedestrian Detection,” arXiv preprint arXiv:1510.08160 (2015).

[14] Dominic Zeng Wang and Ingmar Posner and Paul Newman, ”What could move? Finding cars, pedestrians and bicyclists in 3D laser data,” Robotics and Automation (ICRA), 2012.

[15] Attila Borcs, Balazs Nagy and Csaba Benedek, ”Fast 3-D urban object detection on streaming point clouds,” European Conference on Computer Vision Workshop (ECCVW), 2014.

[16] M. Himmelsbach, A. Muller, T. Luttel and H.J. Wunsche , ”LIDAR-based 3D object perception,” International Workshop on Cognition for Technical Systems (IWCTS), 2008.

[17] Yukyung Choi, Namil Kim, Kibak Park, Soonmin Hwang, JaeShin Yoon and In So Kweon, ”All-Day Visual Place Recognition: Benchmark Dataset and Baseline,” IEEE Conference on Computer Vision and Pattern Recognition Workshop on Visual Place Recognition in Changing Environments (CVPRW-VPRICE), 2015.

[18] Hyunggi Cho, Young-Woo Seo, B.V.K. Vijaya Kumar, and Ragunathan (Raj) Rajkuma, ”A multi-sensor fusion system for moving object detection and tracking in urban driving environments,” International Conference on Robotics and Automation (ICRA), 2014.

[19] Cristiano Premebida, Goncalo Monteiro, Urbano Nunes and Paulo Peixoto, ”A lidar and vision-based approach for pedestrian and vehicle detection and tracking,” Intelligent Transportation Systems Conference (ITSC), 2007.

[20] Alejandro Gonzlez, Gabriel Villalonga, Jiaolong Xu, David Vazquez, Jaume Amores, and Antonio M. Lopez, ”Multiview random forest of local experts combining rgb and lidar data for pedestrian detection,” Intelligent Vehicles Symposium (IV), 2015.

[21] Dominic Wang and Ingmar Posner, ”Voting for voting in online point cloud object detection.,” Proceedings of Robotics: Science and Systems (RSS), 2015.

[22] Alex Teichman, Jesse Levinson, and Sebastian Thrun, ”Towards 3D object recognition via classification of arbitrary object tracks,” International Conference on Robotics and Automation (ICRA), 2011.

[23] B. Douillard, J. Underwood, N. Kuntz, V. Vlaskine, A. Quadros, P. Morton, A. Frenkel, ”On the segmentation of 3D LIDAR point clouds,” International Conference on Robotics and Automation (ICRA), 2011.

[24] David Held, Jesse Levinson, Sebastian Thrun, Silvio Savarese, ”Combining 3d shape, color, and motion for robust anytime tracking,” Robotics: Science and Systems (RSS), 2014.

[25] Frank Moosmann and Christoph Stiller, ”Joint self-localization and tracking of generic objects in 3D range data,” International Conference on Robotics and Automation (ICRA), 2013.