2018 IEEE Intelligent Vehicles Symposium (IV)
Changshu, Suzhou, China, June 26-30, 2018

Robust Camera Lidar Sensor Fusion Via Deep Gated Information Fusion Network

Jaekyum Kim[1], Jaehyung Choi[2], Yechol Kim[1], Junho Koh[1], Chung Choo Chung[1] and Jun Won Choi[1*]

Abstract— **In this paper, we introduce a new deep learning architecture for camera and Lidar sensor fusion. The proposed scheme performs 2D object detection using the RGB camera image and the depth, height, and intensity images generated by projecting the 3D Lidar point cloud into camera image plane. The proposed object detector consists of two convolutional neural networks (CNNs) that process the RGB and Lidar images separately as well as the fusion network that combines the feature maps produced at the intermediate layers of the CNNs. We aim to develop a robust object detector that maintains good object detection accuracy even when the quality of the sensor signals is degraded for object detection. Towards this end, we devise the *gated fusion unit* (GFU) that adjusts the contribution of the feature maps generated by two CNN structures via gating mechanism. Using the GFU, the proposed object detector can fuse the high level feature maps drawn from two modalities with appropriate weights to achieve robust performance. Experiments conducted on the challenging KITTI benchmark show that the proposed camera and Lidar fusion network outperforms the conventional sensor fusion methods even when either of the camera and Lidar sensor signals is corrupted by missing data, occlusion, noise, and illumination change.**

I. INTRODUCTION

Autonomous vehicle is equipped with various types of sensors such as camera, Lidar, radar, and ultrasonic sensors. The data collected by these sensors is used to identify surrounding objects and understand traffic situation in highly dynamic environments. In particular, detection of various dynamic traffic participants such as car, bike, pedestrian, and cyclist is a critical component of the perception for safe autonomous driving. Recently, remarkable improvement in the accuracy of object detection has been achieved as a machine learning model called *convolutional neural network* (CNN) is applied to detect the objects from the camera images. The CNN is capable of finding high level features that generalize well for various environments by training the complex neural network model with the massive amount of labeled images. Such CNN is employed to perform various complicated perception tasks such as object detection. Thus far, several CNN architectures for object detection have been proposed, including the region-CNN (R-CNN) [1], faster R-CNN [2], single shot detector (SSD) [3], and you only look once (YOLO) [4], [5]. These methods calculate the score for the bounding box candidate and the object class based on the feature map produced by the CNN and the whole architecture is trained in an end-to-end fashion. Though these methods offer significant performance gain over the conventional object detectors based on the hand-crafted features, the camera only-based methods often fail to reach the detection accuracy close to 100% especially when the camera is severely hampered by harsh environments such as intense light, shadow, reflection, malfunctioning, and so on.

One viable option to achieve reliable object detection against such challenging situations is *sensor fusion* which takes advantage of the redundant information underlying in the data collected from the multiple sensors. The use of different types of sensors provides the rich and diverse knowledge on the surroundings, thus allowing for better perception. Thus far, various sensor fusion techniques have been proposed. The widely adopted approach is *late fusion method*, which processes different sensor data independently and combine the results (perhaps probabilistically) after processing is done [6]. While this structure is simple to implement, it does not fully exploit high-level dependency between different modalities. Recently, CNN-based sensor fusion techniques have been proposed, which combine the intermediate features found by the separate CNN architectures for multiple sensor modalities [7], [8]. Since the information fusion is performed at the abstract feature levels, this approach, called *intermediate fusion*, can effectively find the joint data representation, yielding significant performance gain for various perception tasks. While CNN-based sensor fusion is capable of finding good joint representation from the multi-modal sensor data, it often fails to offer robust performance especially when some sensor data is corrupted in unfavorable situations such as occlusion, illumination change, failed operation and so on. Once the network is trained, the network parameter is fixed and the corrupted sensor data could harm the joint representation produced by the sensor fusion, consequently leading to severe performance loss. To address this issue, it is necessary develop a robust sensor fusion method which can take full advantage of the complimentary attribute of the sensor fusion.

In this paper, we propose the robust CNN-based sensor

[*] Corresponding Author is Jun Won Choi.

[*] This work was supported by Institute for Information & communications Technology Promotion (IITP) grant funded by the Korea government (MSIT) (No. R7117-16-0164, Development of wide area driving environment awareness and cooperative driving technology which are based on V2X wireless communication) and the Technology Innovation Program (10083646) funded By the Ministry of Trade, Industry & Energy (MOTIE, Korea)

[1] Jaekyum Kim, Yechol Kim, Junho Koh, Chung Choo Chung, and Jun Won Choi are with Dept. of Electrical Engineering, Hanyang University, Seoul 04763, Korea {jkkim, yckim, jhkoh}@spo.hanyang.ac.kr, {cchung, junwchoi}@hanyang.ac.kr

[2] Jaehyung Choi is with the Phantom AI, Inc., USA. jaehyung@phantom.ai

978-1-5386-4452-2/18/$31.00 ©2018 IEEE

fusion architecture referred to as *deep gated information fusion network* (DGFN). We specifically develop the camera-Lidar sensor fusion method performing 2D object detection. First, we project the Lidar 3D point cloud data into the camera image plane and generate the multi-channel Lidar image. The camera and Lidar images are separately fed to two CNN core networks to produce the intermediate feature maps used for sensor fusion. Our DGFN concatenates the intermediate feature maps and applies additional convolution layers to find the joint data representation. In order to facilitate robust sensor fusion, we devise the *gated fusion unit* (GFU) which adjusts the contribution of the feature maps from each modality adaptively based on their quality. The operation of GFU is based on the gating mechanism which weights the feature maps using the *weight map* in efforts to keep the information delivered by the reliable features while mitigating the effect of degraded modality. Note that this gating operation is analogous to that used in long short-term memory (LSTM) in that it controls the information flow for data fusion in a data-dependent manner. Note that the principle of the GFU can be readily applied to any sensor fusion methods combining CNN features. We evaluate the proposed DGFN on the KITTI object detection benchmark [9]. Our experiments show that the proposed algorithm offers significant performance improvement over the existing object detectors in the scenarios where either of two sensor measurements is corrupted by missing data, occlusion, noise, and illumination change.

II. RELATED WORK

In this section, we briefly review the CNN-based object detection methods. Then, we introduce the existing deep learning-based sensor fusion approaches presented in the literature.

A. Camera-based Object Detection

Recently, CNN has been used for object detection and has led to remarkable performance improvement. Since the region-CNN (R-CNN) [1] has first shown to achieve the unprecedented object detection performance using CNN, a variety of CNN-based object detectors have been proposed. The state-of-the-art object detectors can be categorized into two groups: two-stage detectors and single-stage detectors. The two-stage detectors employ the two separate networks; 1) the region proposal network for finding the bounding box containing the object and 2) the object classifier network for identifying the class of the object in the bounding box. Such two-stage detectors include the fast R-CNN [10] and faster R-CNN [2]. Unfortunately, the computational complexity of these two-stage object detectors is high to meet stringent real-time constraint for autonomous vehicles. Thus, single-stage detectors have been proposed, which infer the information on the bounding box and the object class in one shot through the single network. Owing to fast processing speed, these single-stage detectors have been popularly used for many practical applications and the well-known single-stage detectors include SSD [3], YOLO [4], and YOLO2 [5].

B. Deep Learning-based Sensor Fusion

The purpose of sensor fusion is to exploit the inter-relationship between the multi-modal data with different distribution. Basically, sensor fusion can be performed at the different stages of feature extraction [11]. *Early fusion* extracts the shared information on data by jointly processing the raw data measurements acquired by multiple sensors. Due to significant difference in the distribution of the multi-sensor signals, it is not easy to find good joint representation directly from the raw data. The late fusion methods combine the information at the final stage of feature extraction. Unfortunately, this approach does not fully exploit high-level dependency between the multi-modal signals. Recently, it has shown in [12]–[15] that leveraging the capability of the deep learning to find high-level data representation, the *intermediate fusion* can effectively find the joint data representation by combining the features extracted at the intermediate layers of deep neural network.

The deep learning-based sensor fusion has also been proposed in the context of autonomous driving [7], [8], [16]. In [7], the Lidar point cloud data is transformed into the multi-view images including cylindrical and bird's eye views and the CNN-based fusion network is applied to learn the joint feature from both the RGB camera image and multi-view Lidar images. In [8], the authors proposed the *point-fusion network* which predicts the corner location of the 3D bounding box based on the Lidar 3D points. In [16], the authors proposed the camera Lidar fusion method that uses a new layer called non-homogeneous pooling layer that transforms features between the bird's eye view map and the front view map.

III. PROPOSED DEEP GATED INFORMATION FUSION NETWORK (DGFN)

In this section, we present the overall structure of the proposed DGFN method and explain the key principle of the GFU.

A. Overall System Description

The overall structure of the proposed DGFN is described in Fig. 1. The camera and Lidar images are passed through two separate CNNs to produce the feature maps for each sensor data. Note that each CNN structure used in our DGFN is similar to that used for SSD. (The VGG network [17] is used for the first 15 layers and 8 extra convolutional layers are added.) In order to fuse the information extracted from two CNN pipelines, we collect the feature maps at the layers of conv4_3, conv7 (FC7), conv8_2, conv9_2, conv10_2, and conv11_2 layers[1] and combine them through the GFU. The GFU produces the joint feature maps, which are used to perform the bounding box regression and object classification as done in the SSD. The detailed operation of the GFU will be explained later.

[1] We follow the notations of the SSD in [3].

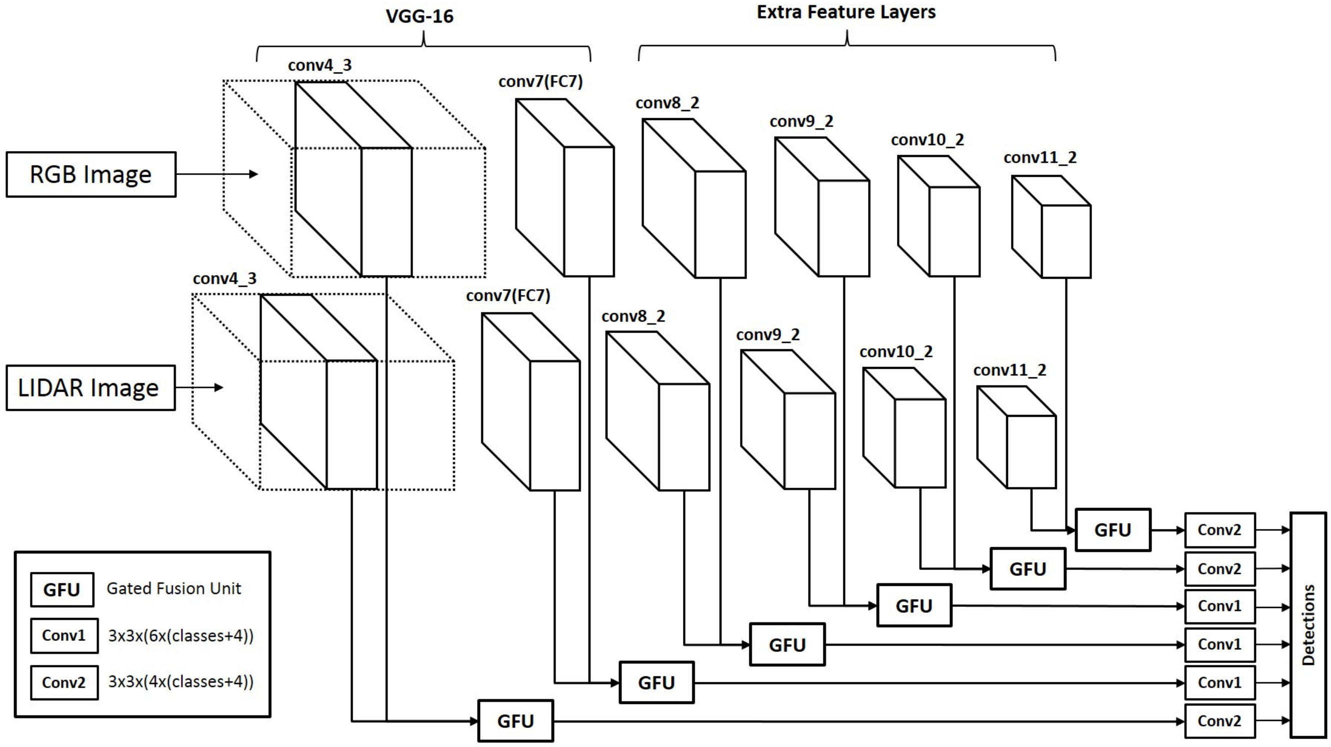


Fig. 1. Overall structure of the proposed object detector

B. LIDAR Front-view Representation

The Lidar 3D point cloud data contains the 3D coordinate (x, y, z) and the reflectivity r measured for each reflected laser pulse. Note that (x, y, z) represents the coordinates in the front, side, and top directions. Since there are numerous data points in the 3D point cloud, it is not straightforward to process them using the deep neural network. In order to leverage the capability of the CNN to process two dimensional grid data, we convert the 3D point cloud data into the 2D images. Specifically, we map the 3D coordinate (x, y, z) of Lidar data into the 2D coordinate (X, Y) on camera plane using

$$\begin{bmatrix} X \\ Y \end{bmatrix} = calib\_matrix \cdot \begin{bmatrix} x \\ y \\ z \end{bmatrix}. \tag{1}$$

where $calib\_matrix$ is the matrix for coordinate transformation. Note that we quantize (X, Y) to the nearest integer and limit the maximum range of (X, Y) by that of camera plane. For the given 2D coordinate (X, Y), we create three channel image by encoding the values of x, z, and r to the pixel values. This creates the image with the depth, height, and intensity (DHI) channels. The pixel values for the DHI channels are obtained by

$$val_d = 255 \cdot (1 - \min[x/max\_x, 1]) \tag{2}$$
$$val_h = 255 \cdot (1 - \min[z/max\_z, 1]) \tag{3}$$
$$val_i = 255 \cdot (1 - \min[r/max\_r, 1]). \tag{4}$$

Note that $x \in [0, max\_x]$, $z \in [0, max\_z]$, and $r \in [0, max\_r]$ are mapped to the pixel values between $[0, 255]$ in a linear scale. For example, we normally set max_x, max_z, and max_r to 80 meter, 6 meter, and 0.7.

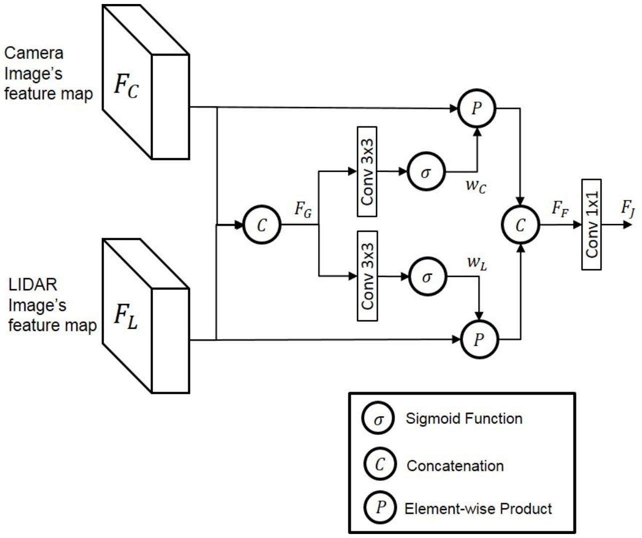


Fig. 2. The proposed gated fusion unit

C. Gated Fusion Unit (GFU)

The proposed GFU has a role of combining the feature maps produced by two CNN pipelines. In GFU, the high-level feature maps obtained by multiple CNNs are selectively weighted to accomplish robust sensor fusion. The structure of the GFU is depicted in Fig. 2. We let $\mathbf{F}_L$ and $\mathbf{F}_C$ be the $M \times N \times K$ feature maps obtained by two CNNs corresponding to the camera and the Lidar images, respectively. All $M \times N \times 1$ feature maps contained in $\mathbf{F}_L$ and $\mathbf{F}_C$ are element-wise multiplied by the $M \times N \times 1$ weight maps $\mathbf{w}_L$ and $\mathbf{w}_C$, respectively. In order to calculate the GFU weight maps $\mathbf{w}_L$ and $\mathbf{w}_C$, we concatenate the input feature maps, $\mathbf{F}_L$ and $\mathbf{F}_C$ and apply two $3 \times 3 \times 1$ kernels $\mathbf{C}_L$ and $\mathbf{C}_C$ followed by the sigmoid function. Depending on the input feature maps $\mathbf{F}_L$ and $\mathbf{F}_C$, the GFU produces the weight maps $\mathbf{w}_L$ and $\mathbf{w}_C$ whose elements have a value between 0 and 1. Note that the GFU weights are multiplied to each pixel of the feature maps independently, which means that the gating operation is performed pixel-wise. Finally, the weighted feature maps are concatenated and passed through $1 \times 1 \times K$ kernel to produce the final joint feature maps $\mathbf{F}_J$. We summarize the operation of the GFU in the following equations

$$\mathbf{F}_G = \mathbf{F}_L \boxplus \mathbf{F}_C \tag{5}$$

$$\mathbf{w}_L = \sigma(\mathbf{C}_L * \mathbf{F}_G + \mathbf{b}_L) \tag{6}$$

$$\mathbf{w}_C = \sigma(\mathbf{C}_C * \mathbf{F}_G + \mathbf{b}_C) \tag{7}$$

$$\mathbf{F}_F(i) = (\mathbf{F}_L(i) \odot \mathbf{w}_L) \boxplus (\mathbf{F}_C(i) \odot \mathbf{w}_C), \quad i = 1, ..., K, \tag{8}$$

$$\mathbf{F}_J = ReLU(\mathbf{C}_J * \mathbf{F}_F + \mathbf{b}_F) \tag{9}$$

where

- $\sigma(x) \triangleq \frac{1}{1+e^{-x}}$: sigmoid function (element-wise)
- $x * y$: convolutional layer
- $x \odot y$: element-wise product
- $x \boxplus y$: concatenation
- $\mathbf{F}(i)$: ith feature map of $\mathbf{F}$
- $\mathbf{b}_F, \mathbf{b}_C, \mathbf{b}_L$: biases of the convolutional layers.

IV. EXPERIMENTS

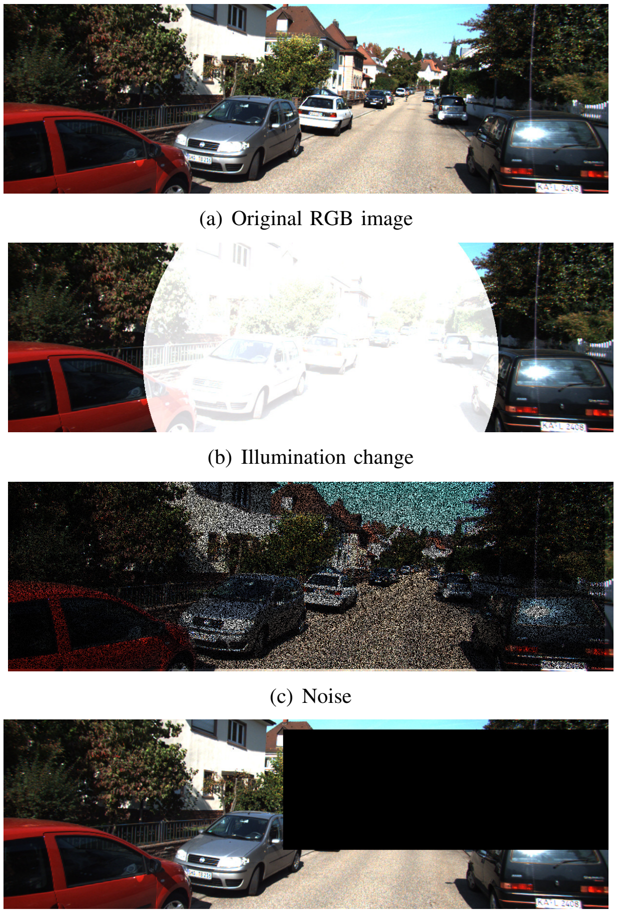
In this section, we evaluate the performance of the proposed DGFN on the KITTI object detection benchmark [9].

A. Experiment Setup

1) Dataset: The KITTI dataset consists of 7481 training data samples and 7518 testing data samples. Both camera images and 3D point cloud data are available and the object is labeled only on the camera images. Since the labels of the test images are not publicly available, we split the labeled data into the training set and the validation set by half, following the method in [18]. We consider three object classes, i.e., car, pedestrian, and cyclist and evaluate the object detectors on the task with three difficulty levels (easy, moderate, hard) as proposed in the KITTI Benchmark.

2) Training: In order to train the proposed DGFN for the degraded sensor data, we conduct the data augmentation. We modify one of camera and Lidar images using the following operations

- Blank Data: we feed all zero image to CNN in place of either camera image or Lidar image.
- Random occlusion: we occlude the object using the black box whose size and location are randomly chosen.
- Additive random noise: we add the random Gaussian noise only to the camera image where noise variance is randomly chosen within the certain range.
- Abrupt illumination change: we brighten the camera image in the rounded local region where the center and radius of the region and the brightness are randomly chosen.


(a) Original RGB image

(b) Illumination change

(c) Noise

(d) Occlusion

Fig. 3. Examples of modifications applied to the camera image

The examples of the modifications applied to the camera images are illustrated in Fig. 3. During training, for every weight update, we try one of above five modifications (including *no action*) to the training images with equal probability. We generate the extended test data set by adding the modified test images to the original data set.

We use the pretrained model of the VGG16 network for two CNN pipelines. We adopt many training strategies used in SSD, such as matching strategy, hard negative mining, and loss function. We employ stochastic gradient descent (SGD) with the mini-batch size of 2. A total of $240,000$ back propagation iterations are performed without early stopping. The initial learning rate is set to 0.0003. We set the weight decay to 0.0005 for L2 regularization and the momentum to the parameter 0.9.

TABLE I

AVERAGE PREDICTION OF THE PROPOSED ALGORITHM VERSUS THE BASELINE ALGORITHM

Test Input	Proposed (with GFU)			Baseline (without GFU)		
	Easy	Moderate	Hard	Easy	Moderate	Hard
Full	93.95	86.70	78.05	89.86	82.21	72.21
Lidar + RGB (normal)	98.69	90.31	82.16	93.61	87.01	77.52
Lidar + RGB (blank)	88.86	78.12	69.68	86.56	74.30	64.71
Lidar (blank) + RGB	97.39	90.29	81.84	91.88	88.10	78.68
Lidar + RGB (occlusion)	89.88	88.12	79.03	88.12	78.52	68.85
Lidar (occlusion) + RGB	97.72	90.23	81.94	92.75	87.10	77.67
Lidar + RGB (noise)	89.33	80.15	71.12	86.75	75.13	65.71
Lidar + RGB (illumination)	95.82	89.71	80.58	89.37	85.31	75.87

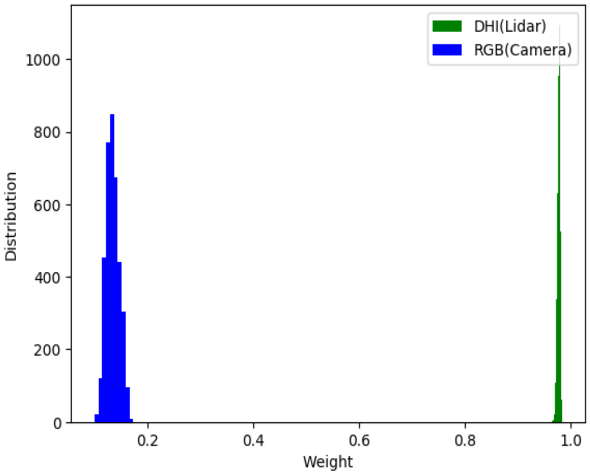
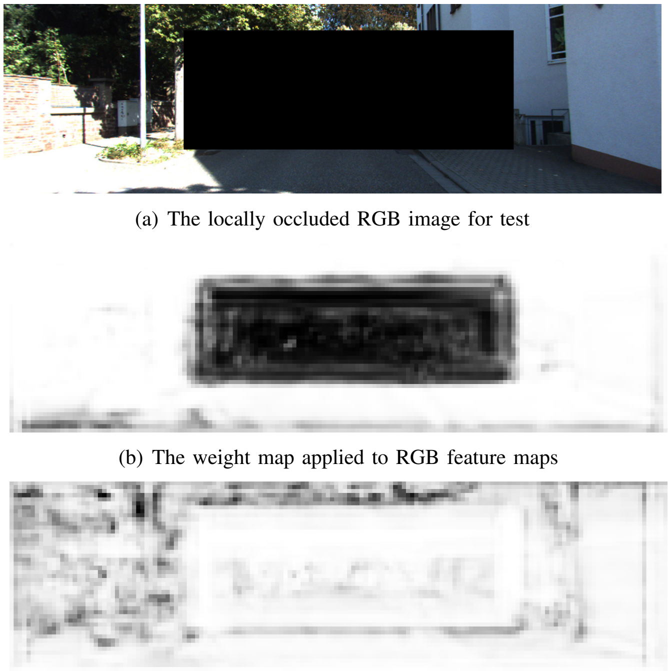


Fig. 4. The histogram of the averaged GFU weight at conv4_3 layer



(a) The locally occluded RGB image for test

(b) The weight map applied to RGB feature maps

(c) The weight map applied to the Lidar feature maps

Fig. 5. The visualization of the GFU weight maps at conv4_3 layer

TABLE II

DETECTION PERFORMANCE (AP) OF SEVERAL OBJECT DETECTORS (*: TRAINED BY US)

Method	Data	Easy	Moderate	Hard
SSD* [3]	Mono	90.27	87.87	79.29
SSD* [3]	Lidar	89.11	77.92	68.87
3DOP [18]	Stereo	94.49	89.65	80.97
Mono3D [19]	Mono	95.75	90.01	80.66
Deep Manta [20]	Mono	97.58	**90.89**	**82.72**
MV3D [7]	Lidar+Mono	95.01	87.59	79.90
Our DGFN	Lidar+Mono	**98.69**	90.31	82.16

B. Experiment Results

In order to verify the benefit of the gating operation, we first compare our object detector with the baseline model without the GFU. Note that the baseline model uses the GFU weight maps whose elements are all fixed to one. We train both models with the same augmented training data set. We evaluate the performance of two models with 1) the full test data set, 2) original test data (no action), and the modified data with 3) blanking, 4) occlusion, 5) noise and 6) illumination change. Note that we use the same amount of test examples for each case. Table I provides the average precisions (AP) achieved by both models. We observe that the proposed object detector significantly outperforms the baseline detector for all cases considered. In particular, the proposed method achieves better detection accuracy for various corruption patterns, which shows the robustness of our method. Interestingly, the proposed scheme also outperforms the baseline algorithm even when the normal test data is used without any data modification. To investigate this issue, we inspect the feature maps produced by both CNNs. We find that in order to cope with various kinds of degradation in the training data, the baseline algorithm learns somewhat similar features from both camera and Lidar images, failing to use the diversity of underlying structures in two modalities. On the contrary, two CNN features learned by the proposed model look distinct owing to the model flexibility provided by the gating operation.

Next, we look into the behavior of the gating operation in details. Fig. 4 shows the histogram of the GFU weights (averaged over the whole weight map at the conv4_3 layer) for the case where the camera sensor is completely blanked. Note that the weights multiplied to the camera side features are close to zero to reduce the contribution from the camera features while the weights to the Lidar side are close to one. In Fig. 5, we visualize the GFU weight maps learned by the GFU for the case where the RGB image is locally occluded by the black box. We find that the GFU weights in the camera side are small only within the locally occluded region while they are high in the rest of area. The GFU weights for the Lidar side are relatively high for the whole region. This shows our gating mechanism controls the amount of information combined for sensor fusion depending on the quality of the features for each local region of the feature maps.

In Table II, we compare the performance of our object detector with that of other CNN-based object detectors. For fair comparison, we use the same evaluation method as that in [7], [18]–[20]. The baseline algorithms we consider include SSD [3], 3DOP [18], Mono3D [19], Deep Manta [20], and MV3D [7]. Note that we use only the normal KITTI data without any data modification for fair comparison. We separately apply the regular SSD to both camera and DHI images generated by the same preprocessing step we used. We observe that the performance of the proposed DGFN is better or on par with the baseline algorithms for all difficulty levels. This shows that the proposed sensor fusion method achieves the competitive performance while exhibiting the robust behavior in adverse environments.

V. CONCLUSIONS

In this paper, we proposed the camera Lidar fusion-based object detector which offers robust performance against the degraded quality of sensor data. Two CNNs were employed to process the RGB image and three channel LIDAR image obtained by transformation of the 3D point cloud data. The feature maps produced at the intermediate levels of two CNNs are combined to produce the joint data representation. To facilitate the robust sensor fusion, the amount of information extracted from each sensor data is regulated by the gated fusion unit which produces the weights used to combine the feature maps based on their quality. Our experiments conducted on the KITTI object detection benchmark show that the gating operation offers significant performance gain over the baseline algorithm for various types of degradation and the proposed object detector achieves the comparable detection performance over the existing object detectors.

REFERENCES

[1] R. Girshick, J. Donahue, T. Darrell, and J. Malik, “Rich feature hierarchies for accurate object detection and semantic segmentation,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recog. (CVPR)*, 2014, pp. 580–587.

[2] S. Ren, K. He, R. Girshick, and J. Sun, “Faster r-cnn: Towards real-time object detection with region proposal networks,” in *Adv. in Neural Information Proc. Syst.*, 2015, pp. 91–99.

[3] W. Liu, D. Anguelov, D. Erhan, C. Szegedy, S. Reed, C.-Y. Fu, and A. C. Berg, “Ssd: single shot multibox detector,” in *Proc. European Conf. on Computer Vision (ECCV)*, 2016, pp. 21–37.

[4] J. Redmon, S. Divvala, R. Girshick, and A. Farhadi, “You only look once: unified, real-time object detection,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recog. (CVPR)*, 2016, pp. 779–788.

[5] J. Redmon and A. Farhadi, “Yolo9000: better, faster, stronger,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recog. (CVPR)*, 2017, pp. 7263–7271.

[6] C. Premebida, J. Carreira, J. Batista, and U. Nunes, “Pedestrian detection combining rgb and dense lidar data,” in *Proc. IEEE/RSJ Inter. Conf. on Intel. Robots and Systems (IROS)*, 2014, pp. 4112–4117.

[7] X. Chen, H. Ma, J. Wan, B. Li, and T. Xia, “Multi-view 3d object detection network for autonomous driving,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recog. (CVPR)*, 2017, pp. 1907–1915.

[8] D. Xu, D. Anguelov, and A. Jain, “Pointfusion: Deep sensor fusion for 3d bounding box estimation,” *arXiv preprint arXiv:1711.10871*, 2017.

[9] A. Geiger, P. Lenz, and R. Urtasun, “Are we ready for autonomous driving? the kitti vision benchmark suite,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recog. (CVPR)*, 2012, pp. 3354–3361.

[10] R. Girshick, “Fast r-cnn,” in *Proc. IEEE Inter. Conf. on Computer Vision (ICCV)*, 2015, pp. 1440–1448.

[11] T. Baltrušaitis, C. Ahuja, and L.-P. Morency, “Multimodal machine learning: A survey and taxonomy,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2018.

[12] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng, “Multimodal deep learning,” in *Proc. Inter. Conf. on Machine Learning (ICML)*, 2011.

[13] N. Srivastava and R. Salakhutdinov, “Multimodal learning with deep boltzmann machines,” *Journal of Machine Learning Research*, vol. 15, pp. 2949–2980, 2014.

[14] J. Wang, Z. Wei, T. Zhang, and W. Zeng, “Deeply-fused nets,” *arXiv preprint arXiv:1605.07716*, 2016.

[15] S. Gupta, R. Girshick, P. Arbeláez, and J. Malik, “Learning rich features from rgb-d images for object detection and segmentation,” in *European Conf. on Computer Vision (ECCV)*, 2014, pp. 345–360.

[16] Z. Wang, W. Zhan, and M. Tomizuka, “Fusing bird view lidar point cloud and front view camera image for deep object detection,” *arXiv preprint arXiv:1711.06703*, 2017.

[17] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *CoRR*, vol. abs/1409.1556, 2014.

[18] X. Chen, K. Kundu, Y. Zhu, A. G. Berneshawi, H. Ma, S. Fidler, and R. Urtasun, “3d object proposals for accurate object class detection,” in *Advances in Neural Information Processing Systems*, 2015, pp. 424–432.

[19] X. Chen, K. Kundu, Z. Zhang, H. Ma, S. Fidler, and R. Urtasun, “Monocular 3d object detection for autonomous driving,” in *Proc. IEEE Conf. on Computer Vision and Pattern Recog. (CVPR)*, 2016, pp. 2147–2156.

[20] F. Chabot, M. Chaouch, J. Rabarisoa, C. Teulière, and T. Chateau, “Deep manta: A coarse-to-fine many-task network for joint 2d and 3d vehicle analysis from monocular image,” in *Proc. IEEE Conf. Computer Vission Pattern Recog.(CVPR)*, 2017, pp. 2040–2049.