A NOVEL REGIONAL FUSION NETWORK FOR 3D OBJECT DETECTION BASED ON RGB IMAGES AND POINT CLOUDS

Hung-Hao Chen[1], Chia-Hung Wang[1], Hsueh-Wei Chen[1],
Pei-Yung Hsiao[2], Li-Chen Fu[1] and Yi-Feng Su[3]

[1]Department of Computer Science and Information Engineering,
National Taiwan University, Taipei, Taiwan
[2]Department of Electrical Engineering,
National University of Kaohsiung, Kaohsiung, Taiwan
[3]Research and Development Division,
Automotive Research and Testing Center (ARTC), Changhua, Taiwan

ABSTRACT

The current fusion-based methods transform LiDAR data into bird's eye view (BEV) representations or 3D voxel, leading to information loss and heavy computation cost of 3D convolution. In contrast, we directly consume raw point clouds and perform fusion between two modalities. We employ the concept of region proposal network to generate proposals from two streams, respectively. In order to make two sensors compensate the weakness of each other, we utilize the calibration parameters to project proposals from one stream onto the other. With the proposed multi-scale feature aggregation module, we are able to combine the extracted region-of-interest-level (RoI-level) features of RGB stream from different receptive fields, resulting in fertilizing feature richness. Experiments on KITTI dataset show that our proposed network outperforms other fusion-based methods with meaningful improvements as compared to 3D object detection methods under challenging setting.

KEYWORDS

Machine Learning, 3D Object Detection, Data Fusion, Autonomous Driving.

1. INTRODUCTION

Owing to the speedy development of computer vision technologies, more and more companies have started to invest in and invent intelligent vehicles. Therefore, autonomous driving has become a popular issue nowadays. The most essential property of autonomous driving is to perceive the surroundings of vehicles and provide safety for drivers. Accordingly, the key to this property is object detection. Over the past few years, there has been many successful 2D object detection approaches proposed, such as Faster R-CNN [1] and RetineNet [2]. However, 2D object detection is unable to provide sufficient ability of perceptions in comparison with 3D object detection because 2D object detection lacks the information of depth and the knowledge of orientation. The depth can hint that the distance of the object is too close, and the orientation is capable of knowing whether the object is in the same direction as the vehicle. With the help of 3D detection, the intelligent vehicles are able to make precise decisions under different situations. In order to detect on-road objects, most of the intelligent vehicles are equipped with multiple

David C. Wyld et al. (Eds): MLDS, NECO, SEMIT, IBCOM, SPPR, SCAI, CSIA, ICCSEA - 2021
pp. 145-160, 2021. CS & IT - CSCP 2021 DOI: 10.5121/csit.2021.111812

sensors such as RGB cameras and LiDARs. Thus, various 3D object detectors based on these sensors are proposed.

Some image-based approaches were presented to utilize monocular [3,4] or stereo images [5,6] to better obtain 3D information of objects. RGB images are good at providing color information and detailed contours of front view. Nevertheless, they still suffer from the limitation of insufficient depth information.

On the contrary, LiDAR-based methods were also proposed to explore the use of 3D LiDAR points. In comparison with RGB images, LiDAR points offer accurate depth information that can be leveraged to localize the objects in the 3D space. Some works [7,8,9] transformed 3D point clouds into 2D bird's eye view (BEV) images or 2D front view images and performed typical convolutional operations to obtain the latent features. Other methods [10] voxelized the 3D point clouds and applied 3D convolution on the generated voxels. However, LiDAR-based methods suffer from sparse observations especially at long range.

To compensate the disadvantages of two sensors, we present Regional Fusion network for 3D object detection (RF3D), which is a fusion-based framework that leverages both cameras and LiDARs jointly. We generate region proposals from both streams, respectively, rather than from LiDAR stream only. Thus, proposals can be projected from one stream to the other stream mutually. In this way, we can fuse the features in deeper layers for better refinement, taking advantage of two sensors and predicting accurate estimations. Furthermore, the presented multi-scale feature aggregation module makes use of different levels of RGB features to obtain low-level contents and high-level semantic meanings simultaneously. With the help of proposed regional fusion layer, the fusion between two streams of feature maps from different sensors is conducted in RoI-level, avoiding cascading redundant feature-level fusion. To verify the effectiveness of RF3D, we conduct several experiment on KITTI Vision Benchmark [11]. The experimental results manifest that our network outperforms other methods under hard difficulty in 3D detection.

In this paper, we design a Multi-Scale Feature Aggregation module (MSFA) with upsampling and downsampling layers to aggregate features from different receptive fields. For two stream fusion, we propose Regional Fusion Layer to fuse point clouds and RGB images based on the RoI estimated in the first stage. Based on above methods, we present a novel two-stream deep architecture for 3D detection, Regional Fusion Network (RF3D), that simultaneously conducts on both point clouds and RGB images in a fusion way for autonomous driving.

The rest of the paper is organized as follows. The overview including image-based, LiDAR-based and fusion-based approaches are introduced in section 2. Then, we define the problem formulation in section 3. In section 4, we present the overall architecture and details in our proposed RF3D. The experimental results on KITTI dataset are shown in section 5. In section 6, we conclude the paper and give directions for future improvement.

2. RELATED WORK

3D object detection is very necessary for intelligent transportation systems. Recently, many works on this topic have gradually emerged. After reviewing the existing approaches of 3D object detection, we categorize these approaches into three groups, namely, image-based approaches, LiDAR-based approaches and fusion-based approaches. To sum up, they are divided according to the inputs.

2.1. Image-based Approaches

Using RGB images to infer accurate 2D bounding boxes of objects is no longer difficult for many state-of-the-art methods since RGB images can provide texture and color information in the form of pixel-wise intensity. Also, there are many works that utilize RGB images to predict 3D bounding boxes of objects. MonoFENet [3] used monocular image to additionally generate the disparity map to enhance the extracted features. D4LCN [4] firstly generated the depth map using the monocular image, and then the depth-guided filtering module was utilized to fuse features of image stream and depth stream. DSGN [5] detected 3D objects on a differentiable volumetric representation that effectively encoded 3D geometric structure for 3D regular space. Disp R-CNN [6] predicted disparity only for pixels on objects of interest and learned a category-specific shape prior for more accurate disparity estimation. However, these image-based methods suffer from the inherent difficulties of estimating depth from images and as a result perform poorly in 3D localization.

2.2. LiDAR-based Approaches

Unlike RGB images, point clouds collected by LiDARs are unordered and discrete. As a result, raw point clouds cannot serve as the inputs of the convolutional layer. Pixor++ [7] and Pointpillars [9] firstly transformed the 3D point clouds into the 2D BEV images, and utilized a 2D CNN to learn the point cloud features for the 3D bounding boxes generation. VoxelNet [12] grouped the point clouds into the voxels and used a 3D CNN to learn the features of the voxels to generate the 3D bounding boxes. However, the BEV projection and voxelization process suffered from the information loss due to the data quantization. Moreover, the 3D CNN was both memory and computation inefficient. On the other hand, PointRCNN [13] directly learned point-wise features and generated 3D bounding boxes from raw point clouds and utilized ground-truth augmentation to gain significant improvements. TANet [14] jointly used channel-wise, voxel-wise, and point-wise attention to alleviate the impact of noisy points. Although depth measurements provided by LiDARs are useful for localizing the 3D bounding boxes of objects, the observations are usually sparse especially at long range.

2.3. Fusion-based Approaches

Since the fusion mechanism between RGB images and LiDAR point clouds remains an open problem nowadays, there are only few approaches that take both RGB images and LiDAR point clouds as inputs. AVOD-FPN [15] applied a 2D convolutional network on both RGB images and LiDAR BEV representations, and fused them at the intermediate region-wise convolutional feature map via feature concatenation. Frustum PointNet [16] utilized mature 2D object detection to firstly generate the 2D region proposals based on the RGB images, and lifted the proposals to the 3D frustums. Then, the points inside the 3D frustums were used to infer the 3D bounding boxes. However, the 2D object detection was the bottleneck. PointPainting [17] designed a painted version of PointRCNN [13] by appending the class score from image to each point. PI-RCNN [18] proposed an Attentive Cont-conv Fusion (PACF) module to fuse point and image features. MMF [19] used a joint model to do four tasks, and each task could benefit from other tasks. ContFuse [20] performed one-way fusion to fuse the feature maps of the RGB images to the BEV feature maps, and CrossFusion [21] utilized the spatial relationship between the BEV features and RGB features to perform two-way fusion. Both ContFuse and CrossFusion applied hierarchical feature-level fusion, which was time-consuming and redundant. Besides, none of the aforementioned methods directly use raw point clouds to perform fusion. Consequently, the information may be lost during the process of data quantization.

In this work, we aim to propose a fusion-based 3D object detection network that exploits the use of raw point clouds and RGB images. In addition, our presented network generates rough proposals from two streams, respectively, and the network fuses two inputs in RoI-level, which avoids extra computation cost on the fusion of non-interest regions. To increase the richness of RGB features, our presented multi-scale feature aggregation module further provides the RGB features with richer information from different features that are with various receptive fields.

3. PROBLEM FORMULATION

We present a deep learning network that aims to solve the task of 3D object detection consuming the inputs of RGB images and LiDAR point clouds. Firstly, an RGB image can be regarded as a set of integer pixel values I_{RGB}, where $I_{RGB} = \{v_{ij} | 1 \leq i \leq W, 1 \leq j \leq H\}$, W denotes the width and H symbolizes the height of the image. Each element v_{ij} in the image is an integer with the range of $[0, 255]$. On the contrary, a LiDAR point cloud can be represented as a set of discrete points I_{LiDAR}, where $I_{LiDAR} = \{P_s | s = 1,2, \dots, N\}$ and N stands for the number of points in a point cloud. Note that N might vary among different collected frames. Additionally, each point P_s can be parameterized into a four-dimensional tensor (x_s, y_s, z_s, r_s), where $(x_s, y_s, z_s)$is the coordinate with regard to the origin of coordinate system and $r_s$denotes the reflectiveness of the point P_s.

Given RGB images I_{RGB} and LiDAR point clouds I_{LiDAR}, our goal is to predict accurate 3D detection that contains both localization and classification information. In general, the outputs of the 3D object detection are represented as a set of 3D bounding boxes O_{box}, where $O_{box} = \{B_k | k = 1,2, \dots M\}$ and M symbolizes the number of predicted 3D bounding boxes. Furthermore, each 3D bounding box B_k is composed of an eight-dimensional tensor $(x_k, y_k, z_k, w_k, h_k, l_k, \theta_k, cls_k)$, where $(x_k, y_k, z_k)$is the localization information that denotes the center coordinate of the bounding box with respect to the coordinate of the LiDAR and (w_k, h_k, l_k)represents the size of the bounding box. In the typical 3D on-road object detection, there only exists yaw rotation along with the axis perpendicular to the ground which is denoted as θ_k. Last but not the least, the classification information is represented as cls_k, indicating the category that the bounding box belongs to.

To sum up, the entire formula for the 3D object detection task T_{det} can be denoted as

$$T_{det}(I_{RGB}, I_{LiDAR}) = O_{box} = \{B_k | k = 1,2, \dots M\} \quad (1)$$

The goal is to propose a 3D detection network that can generate accurate 3D bounding boxes O_{box}based on RGB images I_{RGB} and LiDAR point clouds I_{LiDAR}.

4. REGIONAL FUSION NETWORK

The architecture of RF3D is shown as Figure 1. Our proposed method is composed of five major components including (1) backbone for retrieving latent features, (2) mutual projection for projecting proposals from LiDAR to RGB stream and the reverse, (3) multi-scale feature aggregation module for generating rich RGB features in different scales, and (4) regional fusion layer for performing RoI-level fusion between two input sources.

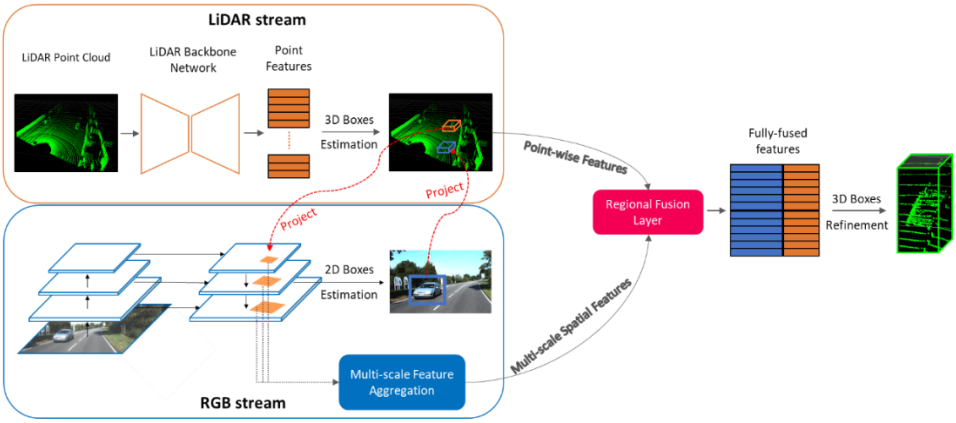

Figure 1. Overview of the proposed Regional Fusion Network

4.1. Backbone Network

The backbone networks aim to obtain discriminative features and generate 2D proposals from RGB images and 3D proposals from LiDAR point clouds, respectively. In order to perform fusion between RGB images and LiDAR point clouds, there exists two streams in our network. One stream is for RGB images and the other is for LiDAR point clouds. However, the discrete and unordered data format of point clouds is very different from pure images that we are not able to apply conventional convolutional operation on the point clouds. Consequently, we utilize separate backbone networks for RGB stream and LiDAR stream.

4.1.1. LiDAR Stream

We utilize PointNet++ [22] as the backbone network, as shown in Figure 2, for the LiDAR stream due to its capabilities of handling unordered issue and learning point-wise features of point clouds. Specifically, we employ four sets of abstraction layers as well as multi-scale grouping that are utilized to subsample original 16,384 points into regions with sizes of 4096, 1024, 256 and 64, respectively. Then, the feature propagation layer is used to obtain the point-wise features for the 3D proposal generation and fusion.

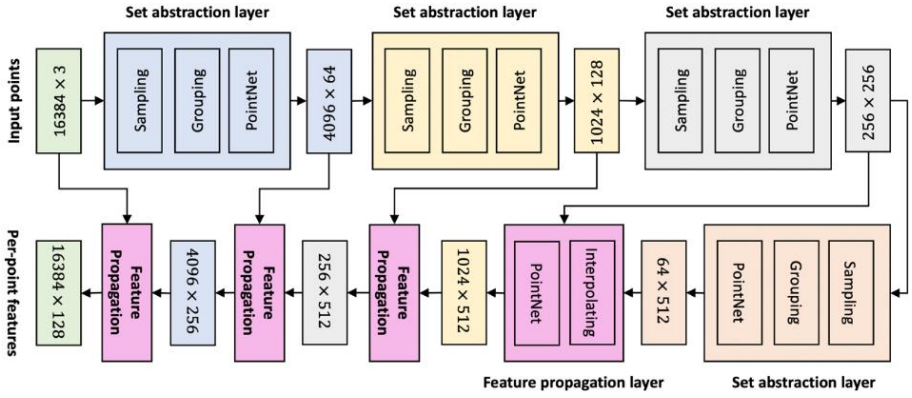

Figure 2. The LiDAR stream backbone PointNet++

4.1.2. RGB Stream

We apply ResNet-50 [23] combined with a feature pyramid network (FPN) [24] as shown in Figure 3. It augments a standard convolutional network using lateral connections and atop-down pathway so as to obtain rich multi-scale feature maps from a single resolution input image. We exploit the feature maps C_2, C_3 and C_4 of ResNet-50 having scales of $1/4$, $1/8$ and $1/16$ to build the feature pyramid. Consequently, the resultant feature pyramid is leveraged to generate 2D proposals from RGB images and provides multi-scale feature maps for multi-scale feature aggregation module.

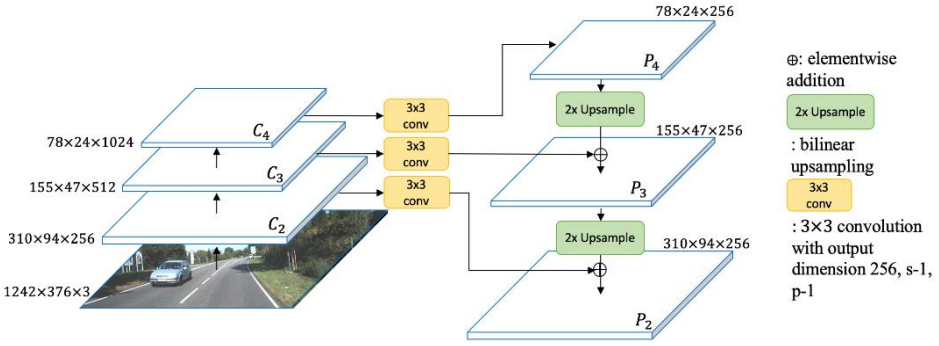

Figure 3. The RGB stream backbone ResNet-50-FPN

4.2. Mutual Projection

As aforementioned, LiDARs and RGB cameras have their own disadvantages. LiDARs possess sparse observations at long range while RGB cameras have limited usage in nighttime, cloudy and rainy weather. Some objects might be detected in one stream while they cannot be captured in the other stream. In order to perform regional fusion and make two sensors benefit each other, we have to obtain an object in both LiDAR stream and RGB stream. Therefore, we project the proposals, which are estimated in the backbone networks, from one stream onto the other. To be more specific, 2D proposals from the RGB stream are projected onto the 3D LiDAR coordinate system and 3D proposals from the LiDAR stream are projected onto the 2D image coordinate system as well, as depicted in Figure 4.

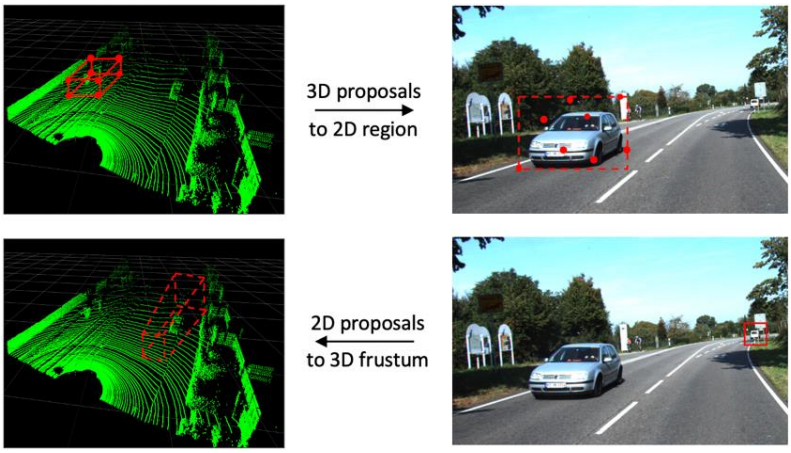

Figure 4. Illustration of mutual projection. The first row showing a 3D bounding box generated from the LiDAR stream is projected onto the RGB image. On the other hand, the second row demonstrating a 2D bounding box predicted from the RGB stream is lifted to a 3D frustum with near and far planes.

4.2.1. 3D Proposals to Images

Given a 3D proposal whose coordinates of eight corners are $\{C_n^{LiDAR} | n = 1,2,\dots,8\}$, where $C_n^{LiDAR} = (x^{LiDAR}, y^{LiDAR}, z^{LiDAR})$ and x^{LiDAR}, y^{LiDAR}, z^{LiDAR} represent the coordinates in the LiDAR coordinate system, we can utilize the calibration matrix to project each point C^{LiDAR} to the image coordinate system and generate corresponding eight points $\{(u_n^{RGB}, v_n^{RGB}) | n = 1,2,\dots,8\}$, where $C_n^{RGB} = (u_n^{RGB}, v_n^{RGB})$ and u^{RGB}, v^{RGB} symbolize the coordinates in the image coordinate system. The calibration matrix is pre-determined, and the entire projecting process can be performed through matrix multiplication.

After obtaining eight corners in the image view, we find the tightest 2D bounding box that can bound all eight corners as the corresponding projected 2D proposals. Hence, RGB features inside 2D bounding boxes are utilized to conduct RoI-level fusion in proposed regional fusion layer.

4.2.2. 2D Proposals to Point Clouds

A 2D proposal which is in the image coordinate system can be lifted to a frustum. A frustum is constructed with two planes which are near-plane and far-plane in the LiDAR coordinate system as shown in Figure 5. The near one is generated with smaller predefined depth d_{near} and the far one is obtained from predefined larger depth d_{far}. As a result, a 3D frustum is generated through connecting these two planes. However, there might be some points that do not belong to the object detected from the RGB stream. Inspired by Frustum PointNet [16], we only select the 3D points whose confidence scores generated in the backbone are greater than the predefined threshold in the frustum. Therefore, those selected points and original 2D proposals are fed into the presented regional fusion layer to conduct RoI-level fusion so as to make two sensors benefit each other.

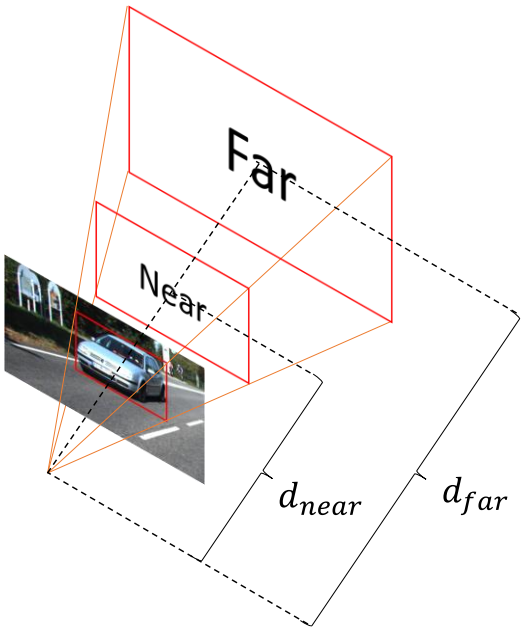

Figure 5. Back projection from the 2D proposals to the 3D space

4.3. Multi-scale Feature Aggregation Module

Features are the most demanding components for the network to generate high-quality predictions. In general, a CNN comprises a number of convolutional layers to extract discriminative features of images. In addition, convolutional layers that are located in different levels can generate various kinds of features. Low-level features are more content descriptive. Besides, the receptive field of the low-level layer is relatively small so that the information of small-size objects can be preserved well. On the other hand, deep high-level layers usually generate class-specific features having more semantic meanings. Since the receptive field of the high-level layer is large, some knowledge of small-sized objects might lose, leaving only global information. In order to perform RoI-level fusion and localize the objects precisely, we have to keep information from multiple receptive fields together as shown in Figure 6. Therefore, we rely not only on low-level features that indicate the appearances of objects but also high-level features that give the semantic meanings of objects.

Accordingly, the 2D proposals from the RGB stream and projected 2D proposals from the LiDAR stream are generated through the backbone networks. Both of them are 2D bounding boxes in nature. In order to generate high-quality RGB features that contain high-level and low-level information simultaneously, we aggregate features from multiple receptive fields. Inspired by Mask R-CNN [25], given the bounding box and multi-scale feature maps P_2, P_3 and P_4, we apply RoIAlign to extract the corresponding features and pool the feature maps into the sizes of 16×16, 8×8 and 4×4 based on different receptive fields, respectively. After that, we utilize upsampling operation as well as downsampling operation to resize the features and aggregate them together. Hence, the multi-scale spatial features containing both high-level semantic meanings and low-level geometric information are generated. Then, these features representing potential foreground objects are used to perform RoI-level with the point-wise features in the proposed regional fusion layer.

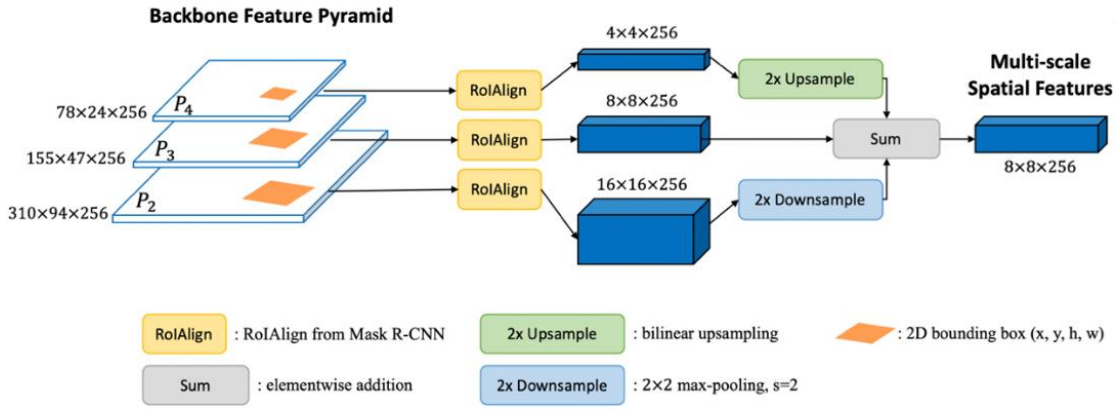

Figure 6. Architecture of multi-scale feature aggregation module

4.4. Regional Fusion Layer

Generally, RGB images provide rich color information of objects while LiDAR point clouds have fine-grained 3D structures. Each kind of data has its own superiority. In order to obtain high-quality detection results, fusion between RGB images and LiDAR point clouds is inevitable. Besides, the spatial relationship between RGB images and LiDAR point clouds is necessary to reason the fusion process. Without utilizing the spatial relationship between two sources, the fusion process may result in severe errors and lack the ability to learn representative fused

features. With a known calibration projection matrix, the projection from a point cloud to an RGB image can be completed. Each point of point clouds in the 3D space is related to a pixel in an image. This one-to-one correspondence can be utilized to fuse the data and supply each point feature with additional information from the RGB stream.

As illustrated in Figure 7, the proposed regional fusion layer leverages spatial features from the RGB stream and the point-wise features from the LiDAR stream to conduct the data fusion between two sources. For each proposal generated in the backbone network, our main purpose is to associate its point-wise features with pixel-wise RGB features so as to increase the feature richness of the LiDAR features for the box refinement. As a matter of fact, we choose to enrich LiDAR features because they are more suitable for performing 3D object detection than the RGB features.

At the first step of regional fusion layer, we apply 1×1 convolution on the spatial features and resize the spatial features along with the height and width dimension on the RGB feature map F_{RGB}, where $F_{RGB} \in \mathbb{R}^{H \times W \times C}$. The transformed resized RGB features are denoted as F'_{RGB}, where $F'_{RGB} \in \mathbb{R}^{HW \times C}$, and HW stands for the number of pixels and C represents the number of channels. Then, we apply attention mechanism to find the correspondence between RGB features F'_{RGB} and LiDAR point-wise features F_{LiDAR}, where $F_{LiDAR} \in \mathbb{R}^{N \times C}$, and N denotes the number of sampled points in the proposal and C symbolizes the channels. In our experiments, the number of sampled points in each 3D proposal is set 512. In the procedure of attention mechanism, we first calculate the attention scores M, whose formula is defined as

$$M = Softmax\left(F_{LiDAR} \times F'^{T}_{RGB}\right) \tag{2}$$

where the superscript T represents the transpose matrix and $M \in \mathbb{R}^{N \times HW}$. In addition, the softmax function is applied along each row in the matrix. As a result, each row vector in M, representing the importance scores of pixels contributing to each LiDAR point, is set as the size of $1 \times HW$. After obtaining the attention scores M, we use the matrix M to calculate the weighted summation of pixel-level RGB features with respect to each LiDAR point, whose formula is defined as

$$F_{attenRGB} = M \times F'_{RGB} \tag{3}$$

where $F_{attenRGB} \in \mathbb{R}^{N \times C}$, and $F_{attenRGB}$ serves as the additional RGB information for the point-wise features. Finally, we concatenate F_{LiDAR} and $F_{attenRGB}$ together and generate fully fused features, which can be utilized for the box refinement.

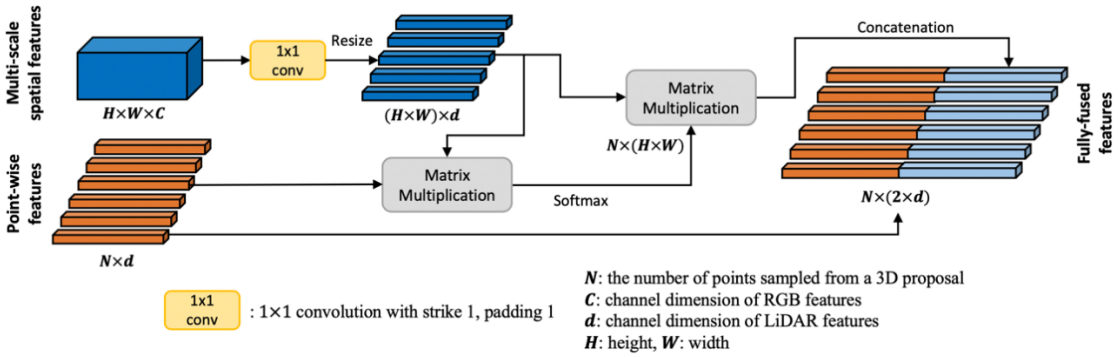

Figure 7. Operational steps of regional fusion layer

4.5. Loss Function

In our network, we use a multi-task loss to train our network. To be more specific, we define the total loss function as the summation of regression loss and classification loss. Since large regression targets are not good for training a detector, we normalize the center and the size of each ground-truth as well as anchor box. The center of each ground-truth and anchor box is normalized as

$$\Delta x = \frac{x^g - x}{w^g}, \Delta y = \frac{y^g - y}{h^g}, \Delta z = \frac{z^g - z}{l^g} \tag{4}$$

where g stands for the ground-truth. In contrast, the size of each ground-truth and anchor box is normalized as

$$\Delta w = \frac{w^g}{w}, \Delta h = \frac{h^g}{h}, \Delta l = \frac{l^g}{l} \tag{5}$$

As for the orientation of each ground-truth and anchor box, it is defined as

$$\Delta\theta = \theta^g - \theta \tag{6}$$

By normalizing the anchor box and the ground-truth, we can obtain a regression tensor T for each of them, where $T = (\Delta x, \Delta y, \Delta z, \Delta w, \Delta h, \Delta l, \Delta\theta)$. To calculate box regression loss L_{box}, we apply the common smooth L1 loss being represented as

$$L_{box}(T^g, T^a) = \sum_{\substack{j\in\{\Delta x,\Delta y,\Delta z,\\ \Delta w,\Delta h,\Delta l,\Delta\theta\}}} smooth_{L1}\left(T_j^g, T_j^a\right) \tag{7}$$

in which

$$smooth_{L1}\left(T_j^g, T_j^a\right) = \begin{cases} 0.5(T_j^g - T_j^a)^2 & , if \left|T_j^g - T_j^a\right| < 1 \\ \left|T_j^g - T_j^a\right| - 0.5 & , otherwise \end{cases} \tag{8}$$

where a denotes the anchor box. On the other hand, we utilize simply binary cross-entropy loss as our classification loss L_{cls}, which can be expressed as

$$L_{cls}(cls^g, cl^a) = -[cls^g \log(cls^a) + (1 - cls^g)\log(1 - cls^a)] \tag{9}$$

After all, the multi-task loss we use to train our model is a weighted sum of the box regression loss L_{box} and the classification loss L_{cls}, which can be expressed as

$$L = \alpha\frac{1}{N}\sum_{i=1}^{N} L_{cls}\left(cls_i^g, cls_i^a\right) + \beta\frac{1}{N_{pos}}\sum_{i\in positive} L_{box}\left(T_i^g, T_i^a\right) \tag{10}$$

where N symbolizes the total number of positive and negative samples, that is to say, $N = N_{pos} + N_{neg}$, and α,β are the hyperparameters controlling the ratio of these two losses.

5. EXPERIMENT

5.1. Experimental Data

In this paper, we choose the task of 3D object detection in KITTI Vision Benchmark to validate our proposed RF3D. In the task of 3D object detection of the benchmark, there are 7,481 training data and 7,518 testing data, and each of them comprises an RGB image, a LiDAR point cloud as well as a calibration file. There are three object categories annotated in the dataset, including car, pedestrian and cyclist. Besides, the category of car has the most sufficient training samples in the dataset. As a consequence, we choose the category of car to evaluate the testing set performance of our approaches as other methods selected. Following the KITTI setting, we accomplish evaluations on three difficulty regimes, namely easy, moderate and hard, which is decided occlusion level, truncated level and distance of the object.

5.2. Evaluation Metric

The predicted results of 3D detection are verified by submitting to KITTI official testing server. The Average Precision (AP) with 40 points is adopted as the evaluation metric for both 3D and BEV detection. In the class of car, the threshold of Intersection over Union (IoU) is set as 0.7 to determine whether the prediction belongs to true positive or false positive.

5.3. Implementation Details

For the LiDAR stream, we only preserve points belonging to the image view via calibration parameters. We subsample 16,384 points from each frame as inputs. For those frames with the number of points fewer than 16,384, we randomly choose points until retrieving 16,384 points. For the RGB stream, we resize the image to the size of 1242×376. The number of points inside 3D proposals for fusion is set 512. We do not apply data augmentation in our experiments because mismatch problems usually occur between point clouds and images.

We implement our network on single GPU GTX 1080 Ti with Pytorch [26]. Two stages of RF3D are trained separately. First stage network, which is utilized to generate proposals, is trained with batch size 8, and second stage network, which is exploited to refine 3D boxes, is trained with batch size 3. Adam [27] is used for optimization with weight decay of 0.001. The learning rate is initialized as 0.001 and decay with a factor of 0.5 at 100, 150, 180 and 200 epochs, respectively.

5.4. Experiment Result of 3D Detection Benchmark

The 3D detection results of the class car on KITTI testing dataset is shown in Table 1. The task of 3D detection is more challenging than that of BEV detection, because 3D detection requires the involvement of height information. Our RF3D outperforms other published state-of-the-art methods with respect to AP under all difficulty regimes in 3D detection except for MMF [19], which is the state-of-the-art fusion-based method in easy and moderate difficulties. In our experimental results, we observe that our network surpasses other methods by a large margin under the hard case. This situation represents that directly utilizing raw point clouds as inputs can preserve the 3D geometric information of those highly occluded or truncated objects. In addition, observing the additional information from RGB stream and the proposed regional fusion layer, the network is able to predict high-quality 3D bounding boxes.

Table 1. Comparison of results on KITTI 3D detection benchmark testing split (car), where PC denotes point clouds.

Method	Types of Input	3D AP of car (in %)		
		Easy	Moderate	Hard
Disp R-CNN [6]	Image	59.58	39.34	31.99
VoxelNet [12]	PC	81.97	65.46	62.85
PointPillars [9]	PC	79.05	74.99	68.30
TANet [14]	PC	83.81	75.38	68.32
F-PointNet [16]	PC+Image	81.20	70.39	62.19
ContFuse [20]	PC+Image	82.54	66.22	64.04
AVOD-FPN [15]	PC+Image	81.94	71.88	66.38
CrossFusion [21]	PC+Image	83.20	74.50	67.01
PointPainting [17]	PC+Image	82.11	71.70	67.08
PI-RCNN [18]	PC+Image	84.37	74.82	70.03
MMF [19]	PC+Image	86.81	76.75	68.41
ours	PC+Image	85.18	75.76	70.99

5.5. Ablation Study on Components

Since KITTI official testing server has limited submissions per month, we use the validation set to conduct our ablation studies and several experiments. We follow the rule proposed in [28] to split the training data into training set and validation set. As a consequence, there are total 3,712 training frames and 3,769 validation frames, respectively.

There are two components presented to reason the fusion between two input sources, including the multi-scale feature aggregation module and the regional fusion layer. The multi-scale feature aggregation module enriches the feature maps of RGB stream by combining feature maps from different receptive fields with upsampling and downsampling layers. The regional fusion layer utilizes the proposals from one stream and their projected proposals from the other to perform the RoI-level fusion so as to fertilize the LiDAR features with additional RGB information. Other than the proposed two modules, we also exploit the proposals generated from both streams to make two sensors compensate with each other. In order to validate the effectiveness of these methods, we conduct several ablation studies on the validation set of class car as well. The experimental results are shown in Table 2.

In the beginning, we simply utilize the LiDAR data to perform 3D object detection without any RGB images as listed in the first row. There is no fusion between two input sources. Secondly, we utilize two input sources simultaneously without the multi-scale feature aggregation module as presented in the second row. Meanwhile, we only leverage the 3D proposals generated from the LiDAR stream and their corresponding projected ones from the RGB stream to perform fusion. As a consequence, there is no 2D region proposal generated from the RGB stream. Besides, we exploit the multi-scale feature aggregation module alone to validate its effectiveness for improving the fusion as illustrated in the third row. After that, we enable the network to generate 2D and 3D proposals simultaneously and project proposals from one stream onto the other stream as shown in the fourth row. In this way, we demonstrate the importance of 2D estimations generated from the RGB images. Finally, we combine all the properties together to use as the last derived model. It is obvious that the performance is more profitable than the others. Therefore, we choose the last one as our final model and comparison with other methods.

Table 2. Ablation studies of each component on KITTI validation split of 3D detection (car), where the RF layer stands for regional fusion layer and the MSFA module indicates multi-scale feature aggregation module.

2D proposals	RF layer	MSFA module	3D AP of car (in %)		
			Easy	Moderate	Hard
✗	✗	✗	83.78	74.34	73.67
✗	✓	✗	86.12	76.89	75.54
✗	✓	✓	88.21	78.42	76.82
✓	✓	✗	87.84	78.36	77.10
✓	✓	✓	89.54	79.22	78.37

5.6. Qualitative Results

We visualize several predicted results from KITTI dataset as illustrated in Figure 8. It is observed that some objects are very difficult to be captured through only RGB images due to serious occlusion and truncation. However, with the help of 3D proposals generated from LiDAR point clouds, these highly occluded and truncated objects can be easily detected since raw point clouds do not suffer from these issues. We also find that several objects have limited points collected in point clouds, resulting in poor performance of 3D proposals generation. Since we simultaneously utilize RGB images to generate 2D proposals, it is verified that the RGB images can compensate the weaknesses of LiDAR point clouds.

Figure 8. Visualization of the prediction results of RF3D on KITTI dataset

6. CONCLUSIONS

In this paper, we propose Regional Fusion Network for 3D on-road object detection. Our network directly consumes raw point clouds as inputs to perform data fusion. To the best of our knowledge, we are the first to integrate raw point clouds and RGB images to conduct 3D object detection. We are able to compensate the weaknesses of two sensors through projecting the proposals from one stream to the other. Additionally, our proposed multi-scale feature aggregation module can combine features from different receptive fields to enrich the RGB features and improve overall detection results. Moreover, the presented regional fusion layer is able to fuse two inputs based on their corresponding RoIs and provide additional RGB information for LiDAR features. The experimental results on KITTI Vision Benchmark show that

our model outperforms other methods in 3D detection especially under challenge setting. However, in order to obtain satisfying detection results, our proposed RF3D has longer inference time. The future research emphasizes on designing an efficient and lightweight proposed RF3D to reduce inference time. Besides, data augmentation techniques for both point clouds and images can be developed to improve the performance of the presented Regional Fusion Network.

ACKNOWLEDGEMENTS

This work was in cooperation with Automotive Research and Testing Center under the grant number of 110-EC-17-A-25-1588 in the project of Department of Industrial Technology of Ministry of Economic Affairs, Taiwan ROC. This research was also supported by the Joint Research Center for AI Technology and All Vista Healthcare under Ministry of Science and Technology of Taiwan, and Center for Artificial Intelligence & Advanced Robotics, National Taiwan University, under the grant numbers of 110-2634-F-002-042 and 110-2634-F-002-016.

REFERENCES

[1] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," in *Advances in neural information processing systems*, 2015, pp. 91-99.

[2] T.-Y. Lin, P. Goyal, R. Girshick, K. He, and P. Dollár, "Focal loss for dense object detection," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2980-2988.

[3] W. Bao, B. Xu, and Z. Chen, "Monofenet: Monocular 3d object detection with feature enhancement networks," *IEEE Transactions on Image Processing*, vol. 29, pp. 2753-2765, 2019.

[4] M. Ding *et al.*, "Learning depth-guided convolutions for monocular 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2020, pp. 1000-1001.

[5] Y. Chen, S. Liu, X. Shen, and J. Jia, "Dsgn: Deep stereo geometry network for 3d object detection," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 12536-12545.

[6] J. Sun *et al.*, "Disp R-CNN: Stereo 3D Object Detection via Shape Prior Guided Instance Disparity Estimation," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2020, pp. 10548-10557.

[7] B. Yang, M. Liang, and R. Urtasun, "Hdnet: Exploiting hd maps for 3d object detection," in *Conference on Robot Learning*, 2018, pp. 146-155.

[8] B. Yang, W. Luo, and R. Urtasun, "Pixor: Real-time 3d object detection from point clouds," in *Proceedings of the IEEE conference on Computer Vision and Pattern Recognition*, 2018, pp. 7652-7660.

[9] A. H. Lang, S. Vora, H. Caesar, L. Zhou, J. Yang, and O. Beijbom, "Pointpillars: Fast encoders for object detection from point clouds," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 12697-12705.

[10] Z. Wu *et al.*, "3d shapenets: A deep representation for volumetric shapes," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2015, pp. 1912-1920.

[11] A. Geiger, P. Lenz, and R. Urtasun, "Are we ready for autonomous driving? the kitti vision benchmark suite," in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, 2012, pp. 3354-3361.

[12] Y. Zhou and O. Tuzel, "Voxelnet: End-to-end learning for point cloud based 3d object detection," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2018, pp. 4490-4499.

[13] S. Shi, X. Wang, and H. Li, "Pointrcnn: 3d object proposal generation and detection from point cloud," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 770-779.

[14] Z. Liu, X. Zhao, T. Huang, R. Hu, Y. Zhou, and X. Bai, "TANet: Robust 3D Object Detection from Point Clouds with Triple Attention," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 11677-11684, 04/03 2020.

[15] J. Ku, M. Mozifian, J. Lee, A. Harakeh, and S. L. Waslander, "Joint 3d proposal generation and object detection from view aggregation," in *2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2018, pp. 1-8.

[16] C. R. Qi, W. Liu, C. Wu, H. Su, and L. J. Guibas, "Frustum pointnets for 3d object detection from rgb-d data," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2018, pp. 918-927.

[17] S. Vora, A. H. Lang, B. Helou, and O. Beijbom, "PointPainting: Sequential Fusion for 3D Object Detection," in *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 13-19 June 2020 2020, pp. 4603-4611.

[18] L. Xie *et al.*, "Pi-rcnn: An efficient multi-sensor 3d object detector with point-based attentive cont-conv fusion module," *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, pp. 12460-12467, 2020.

[19] M. Liang, B. Yang, Y. Chen, R. Hu, and R. Urtasun, "Multi-Task Multi-Sensor Fusion for 3D Object Detection," in *2019 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 15-20 June 2019 2019, pp. 7337-7345.

[20] M. Liang, B. Yang, S. Wang, and R. Urtasun, "Deep continuous fusion for multi-sensor 3d object detection," in *Proceedings of the European Conference on Computer Vision (ECCV)*, 2018, pp. 641-656.

[21] D.-S. Hong, H.-H. Chen, P.-Y. Hsiao, L.-C. Fu, and S.-M. Siao, "CrossFusion net: Deep 3D object detection based on RGB images and point clouds in autonomous driving," *Image and Vision Computing*, vol. 100, p. 103955, 2020.

[22] C. R. Qi, L. Yi, H. Su, and L. J. Guibas, "Pointnet++: Deep hierarchical feature learning on point sets in a metric space," in *Advances in neural information processing systems*, 2017, pp. 5099-5108.

[23] K. He, X. Zhang, S. Ren, and J. Sun, "Deep residual learning for image recognition," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2016, pp. 770-778.

[24] T.-Y. Lin, P. Dollár, R. Girshick, K. He, B. Hariharan, and S. Belongie, "Feature pyramid networks for object detection," in *Proceedings of the IEEE conference on computer vision and pattern recognition*, 2017, pp. 2117-2125.

[25] K. He, G. Gkioxari, P. Dollár, and R. Girshick, "Mask r-cnn," in *Proceedings of the IEEE international conference on computer vision*, 2017, pp. 2961-2969.

[26] A. Paszke *et al.*, "Automatic differentiation in pytorch," 2017.

[27] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *Proceedings of the 3rd International Conference on Learning Representations (ICLR)*, 2014.

[28] X. Chen *et al.*, "3d object proposals for accurate object class detection," in *Advances in Neural Information Processing Systems*, 2015, pp. 424-432.

AUTHORS

Hung-Hao Chen received the B.S. degree in Department of Computer Science and Engineering from National Chen Kung University, Tainan, Taiwan in 2018 and the M.S. degree with the Department of Computer Science and Engineering in Department of Computer Science and Engineering from National Taiwan University, Taipei, Taiwan in 2020. His research interests include deep learning and computer vision.

Chia-Hung Wang received the B.S. degree in Department of Electrical Engineering, College of Electrical and Computer Science from National Taiwan University of Science and Technology, Taipei, Taiwan in 2018 and the M.S. degree with the Department of Computer Science and Engineering in Department of Computer Science and Engineering from National Taiwan University, Taipei, Taiwan in 2021. His research interests include deep learning, computer vision and sensor fusion.

Hsueh-Wei Chen received the B.S degree in Department of Computer Science and Information Engineering from National Chung Cheng University, Chiayi, Taiwan in 2020. He is currently pursuing the M.S degree with the Department of Computer Science and Engineering in Department of Computer Science and Engineering from National Taiwan University, Taipei, Taiwan. His research interests include deep learning and object detection.

Pei-Yung Hsiao received the B.S. degree in chemical engineering from Tung Hai University, in 1980 and the M.S. and Ph.D. degrees in electrical engineering from the National Taiwan University, in 1987 and 1990, respectively. In 1990, he was an Associate Professor in the Department of Computer and Information Science, National Chiao Tung University, Hsinchu, Taiwan. In 1998, he was the CEO of Aetex Biometric Corporation. He is currently a Professor in the Department of Electrical Engineering, National Univ. of Kaohsiung. His research interests and industrial experiences include VLSI/CAD, image processing, fingerprint recognition, visual detection, embedded systems, and FPGA rapid prototyping.

Li-Chen Fu received the B.S. degree from National Taiwan University in 1981, and the M.S. and Ph.D. degrees from the University of California, Berkeley, in 1985 and 1987, respectively. Since 1987, he has been on the faculty of and currently is a professor in both the Department of Electrical Engineering and Department of Computer Science & Information Engineering of National Taiwan University. He is now a senior member of both the Robotics and Automation Society and Automatic Control Society of IEEE, and he became an IEEE Fellow (F) in 2004. His areas of research interest include robotics, FMS scheduling, shop floor control, home automation, visual detection and tracking, E-commerce, and control theory & applications.

Yi-Feng Su received the M.S. degree in Electrical Engineering from National Changhua University of Education (NCUE), Changhua, Taiwan, in 2005. Presently, he works as an engineer in the Automotive Research & Testing Center (ARTC), Taiwan. His research areas include image processing, machine vision, algorithm development, and applications of embedded systems.