International Journal of Innovative
Computing, Information and Control
Volume **18**, Number **1**, February **2022**

 ISSN 1349-4198
pp. **271–287**

SENSOR FUSION OF CAMERA AND MMW RADAR BASED ON MACHINE LEARNING FOR VEHICLES

Yi-Horng Lai[1], Yu-Wen Chen[2] and Jau-Woei Perng[2,*]

[1]School of Mechanical and Electrical Engineering
Xiamen University Tan Kah Kee College
Merchants' Economic and Technological Development Zone, Zhangzhou 363105, P. R. China
lai81@xujc.com

[2]Department of Mechanical and Electro-Mechanical Engineering
National Sun Yat-sen University
No. 70, Lienhai Rd., Kaohsiung 80424, Taiwan
ywchen1994@gmail.com; *Corresponding author: jwperng@faculty.nsysu.edu.tw

Received June 2021; revised November 2021

Abstract. *This study develops a forward collision warning system for vehicles based on sensor fusion of a camera and a millimeter wave radar. The proposed system has a parallel architecture. The algorithm of the millimeter wave radar subsystem includes density-based spatial clustering of applications with noise, particle filter, and multi-objective decision-making algorithms. The image subsystem uses the You Only Look Once v3 network and a Kalman filter to detect and track four types of objects (i.e., cars, motorcycles, bikes, and pedestrians). All radar objects are projected onto the image coordinates using a radial basis function neural network. Only the objects inside the region of interest of the on-road lane are tracked by the sensor fusion mechanism. The proposed system is evaluated in four types of weather scenarios: daytime, nighttime, rainy daytime, and rainy nighttime. The experimental results validate that the fusion strategy can effectively compensate any single-sensor failure. In the four scenarios, the average detection rate of the sensor fusion reaches 98.7%, which is higher than those of the single-sensor systems.*
Keywords: MMW radar, YOLO network, Sensor fusion, Particle filter

1. **Introduction.** At present, increasing number of advanced driver assistance system (ADAS) applications are being used as standard car equipment for protecting drivers. Obstacle and on-road lane detection are the critical issues in an ADAS application. The development of obstacle and lane detection technology is a critical task for lane departure warning (LDW), forward collision warning (FCW), pedestrian detection system (PDS), adaptive cruise control (ACC), and autonomous emergency braking (AEB) [1-6]. Three types of sensors (LiDARs, radars, and cameras) are well-known tools in collision avoidance research [7,8].

Using a single sensor has limitations in object detection. A camera is easily affected by variations in light and weather conditions. Furthermore, distance can be only approximately estimated from an image. Concurrently, detection by a millimeter wave (MMW) radar depends on the relative velocity of the object. For a LiDAR sensor, the detection distance is less than that of an MMW radar. The development of sensor fusion systems can compensate the disadvantages of individual sensors.

A series fusion architecture was proposed in [9], wherein first the position of an object was obtained using an MMW radar, and subsequently the radar coordinates were converted into image coordinates. Owing to the application of a region of interest (ROI), the

DOI: 10.24507/ijicic.18.01.271

number of images required to be processed for vehicle detection and tracking was reduced. In addition, a Kalman filter was used for radar tracking to improve the vehicle detection rate and reduce false alarms.

In [10], three cameras and the edge drawing lines (EDLines) method were used for on-road lane detection. The intersection of the left and right lanes was considered as the vanishing point. A horizontal edge was used to filter the shadow position of the vehicle. Experiments were conducted on expressways and under different weather conditions.

Based on a deep learning network, a fast region-based convolutional neural network object detection model was used in [11]. First, the featured images were extracted in the convolution layer, and candidate regions were obtained from the region recommendation network. Subsequently, the candidate regions and the feature images were placed into the ROI pooling layer for adjustment and classification. When a vehicle was detected, the horizontal position and width of the detected object were used to estimate the distance.

Pedestrian detection using a LiDAR under various driving conditions was conducted in [12]. Based on the distance, intensity, and width of the reflected LiDAR signal, the features of a pedestrian were extracted. The object information was fused into an image using a LiDAR clustering algorithm.

An MMW radar and a camera were integrated to detect and identify objects in front of vehicles [13]. First, the ROI was selected using an image system. Subsequently, the radar object information was projected onto the ROI of an image. Image recognition was based on the histogram of oriented gradient features and a radial basis function (RBF). The classification of objects was conducted using K-means and random decision forest algorithms.

Although the traditional feature-based image processing method is suitable for detecting objects, its object classification ability is poor. Generally, deep learning algorithms are superior to feature-based methods. However, for real-time computing applications, a deep learning-based recognition method requires GPU hardware.

When the electromagnetic wave of an MMW radar is transmitted, reflection and refraction occur simultaneously. The echo points for radar objects are not necessarily fixed. In addition, the electromagnetic waves are easily affected by the shadowing effect, and the echo noise does not have a Gaussian normal distribution. The density-based spatial clustering of applications with noise (DBSCAN) algorithm can set two parameters (the core point and the radius of the outer search neighborhood) and is suitable for radar data with unknown number of clusters [14]. Concurrently, based on the concept of the Monte Carlo method, the particle filter algorithm can use multiple random particles to calculate the probability distribution of objects, and it is suitable for nonlinear and nonGaussian motion tracking [15,16].

In this study, a sensor fusion system of a camera and an MMW radar is proposed. Based on a parallel architecture design, the individual sensors can detect and track an object independently. The sensor fusion subsystems can compensate the detection failure of each single sensor and improve the detection accuracy.

Although the proposed systems are based on the previous research of the authors [17], the innovations of the proposed methods include the following. The radar subsystem overcomes the drawbacks of the previous research and can track multiple radar objects. Compared with the traditional image feature extraction method used in [17], the novel image subsystem adopts a deep learning architecture. Furthermore, the image subsystem can identify on-road lanes and objects simultaneously. The detection rate of the proposed system is better than those achieved in previous studies. The contributions of this study include the following.

1) Radar subsystem: Radar noise is reduced by the DBSCAN algorithm. A particle filter is used for object tracking. Multi-object decision-making mechanisms coordinate the tracking list of the radar objects.
2) Image subsystem: The You Only Look Once v3 (YOLOv3) network is applied for object detection and recognition. A Kalman filter is used to track an object in an image.
3) On-road lane detection: First, EDLines and HSV masking are used to detect lane lines. Second, line segments and angles are classified into left and right lane lines. Finally, the least squares method is used to detect on-road lanes.
4) The MMW radar objects are projected onto the image system to obtain the fusion information. The fusion information is used to calculate the time to collision (TTC) and the post-encroachment time (PET) for collision warning.

2. **System Architecture.** This study presents a parallel architecture sensor fusion based on an MMW radar and a camera. The system includes 3 subsystems: The MMW radar detection and tracking subsystem; the image object detection and identification subsystem; the sensor fusion subsystem. Under the parallel system architecture, the radar and the image subsystem senses the object information respectively. In addition, the imaging subsystem uses HSV masking and EDLines algorithm to detect on-road lane and displays on the human-machine interface. The fusion subsystem integrates the object information, evaluates the risk of collision and provides collision warning in time. The system architecture is shown in Figure 1.

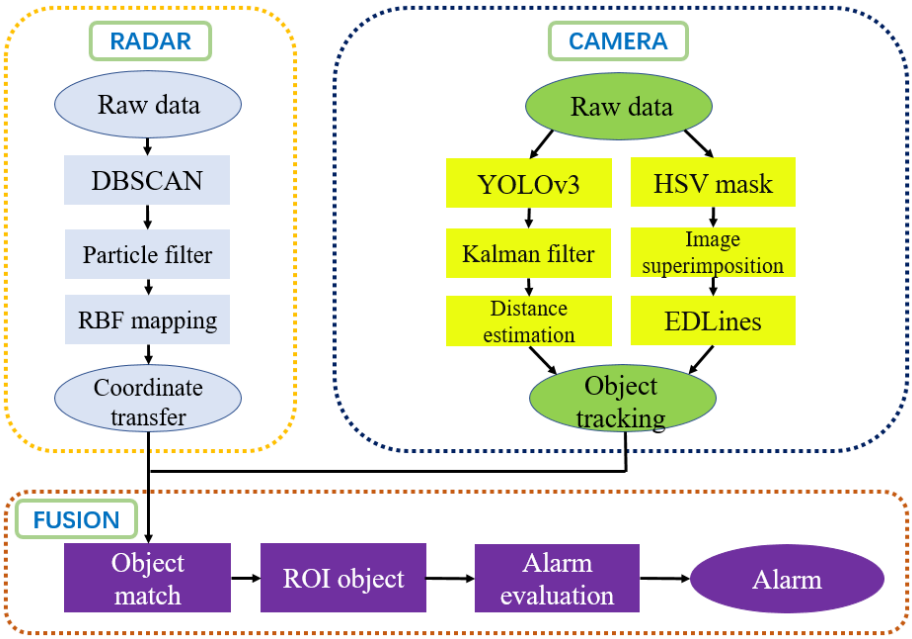


FIGURE 1. The system architecture

Owing to the computational requirements of the deep learning network, a GeForce GTX 1080 Ti GPU was used to train the YOLOv3 network. In the experiments, an Intel i7-7700K 4.2-GHz CPU was used for the data processing of the radar and image data.

The system integrates a 24-GHz short-range MMW radar (WistronNeWeb Corp. WNC, Taiwan) and a li-cam-imx224 vehicle camera (Sony, Inc., Japan). The camera is installed at the front bumper approximately 60 cm above the ground and 15 cm away from the MMW radar level (shown in Figure 2).

FIGURE 2. The sensors installation

3. System Development.

3.1. **MMW radar detection subsystem.** A flowchart of the MMW radar subsystem is shown in Figure 3. First, the DBSCAN algorithm is used to reduce the noise signals. Subsequently, the particle filter conducts the target tracking. The tracking results are projected onto the image coordinates. If a radar projection point is within the ROI of the lane line, the relative distance and speed of the object are tracked.

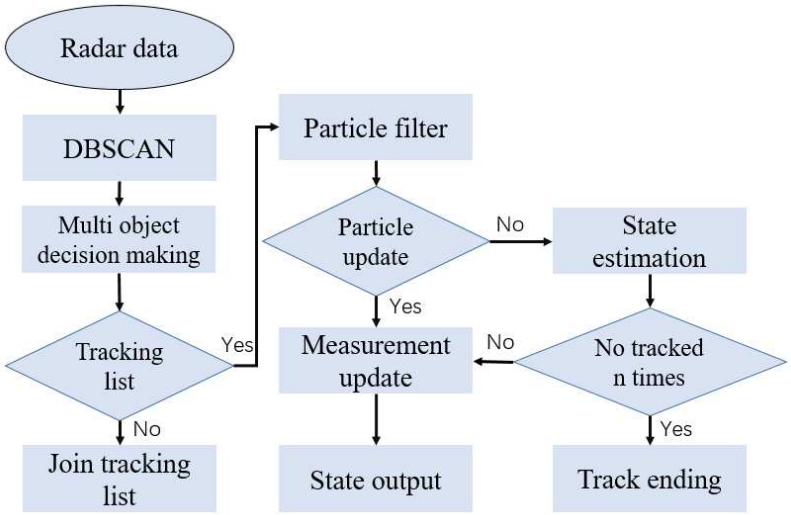


FIGURE 3. The flowchart of MMW radar detection subsystem

3.1.1. *DBSCAN algorithm.* The DBSCAN algorithm can identify high-density areas in the data distribution of an unknown space and treat relatively scattered low-density data as noise. The algorithm has two parameters: Eps and Minpts. Eps is the radius of the core point for searching the neighborhood, and Minpts is the threshold value for the minimum amount of data within the neighborhood radius.

In this study, first the original data of the MMW radar are converted from polar coordinates into rectangular coordinates in advance, following which the Euclidean distance between two data points is calculated. Because most of the radar objects have only a single point, we set Minpts = 0 for noise filtering.

3.1.2. *Particle filter algorithm.* When a motion model is nonlinear or nonGaussian, the particle filter can use random particles with corresponding weights to express the probability distribution. When the number of particles is large, the state variables are close to the true probability distribution.

In this study, we use a sequential importance sampling (SIS) particle filter. The input is the position of a measured object (x, y) of the MMW radar. The state equation used to predict this object is expressed as follows:

$$\boldsymbol{X}_k = \boldsymbol{F}\boldsymbol{X}_{k-1} + \boldsymbol{G}\boldsymbol{W}_k = \begin{bmatrix} 1 & 0 & T & 0 \\ 0 & 1 & 0 & T \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \dot{x}_{k-1} \\ \dot{y}_{k-1} \end{bmatrix} + \begin{bmatrix} \frac{T^2}{2} & 0 \\ 0 & \frac{T^2}{2} \\ T & 0 \\ 0 & T \end{bmatrix} \begin{bmatrix} \boldsymbol{W}_x \\ \boldsymbol{W}_y \end{bmatrix}, \quad (1)$$

where $\boldsymbol{X}_k = [x_k \ y_k \ \dot{x}_k \ \dot{y}_k]^T$ denotes the state vector. x_k and x_{k-1} denote the relative lateral coordinates at the current time and the previous moment, respectively. y_k and y_{k-1} are the relative longitudinal coordinates at the current time and the previous moment, respectively. T is the sampling time of the radar sensor, and $\boldsymbol{W}_k$ is the zero-mean Gaussian white noise.

The measurement area of the MMW radar is divided into $M \times N$ square blocks. The side length of a square block is 1 m^2. The measurement model of the MMW radar is expressed as follows:

$$\boldsymbol{Z}_k^{(i,j)} = \boldsymbol{h}_k^{(i,j)}(\boldsymbol{X}_k) + \boldsymbol{v}_k^{(i,j)}, \quad (2)$$

where $\boldsymbol{v}_k^{(i,j)}$ is the measured noise in the (i, j) square block. $\boldsymbol{h}_k^{(i,j)}(\boldsymbol{X}_k)$ is the signal strength of the object in the (i, j) square block.

3.2. **Image detection subsystem.** Owing to the parallel fusion architecture, the imaging subsystem has independent object detection, tracking, and distance estimation algorithms. A flowchart of the image detection subsystem is shown in Figure 4.

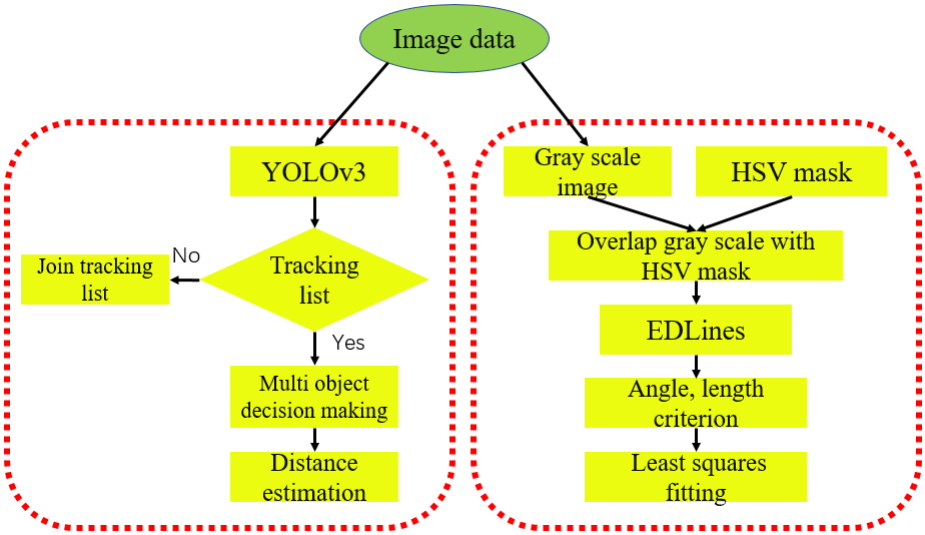


FIGURE 4. The flowchart of image detection subsystem

3.2.1. *YOLOv3 network.* The YOLOv3 network is used to detect on-road objects [18]. The types of objects include pedestrians, bicycles, motorcycles, and cars.

First, the graph of an original image (416 × 416 pixel) is reduced to $S \times S$ scale grids. When the center coordinates of an object fall into a scale grid, the grid is responsible for detecting the object. Each scale grid predicts the confidence score and category probability of the bounding box. Each bounding box contains five predictors: P_x, P_y, P_w, P_h, and

confidence score. P_x, P_y are the center coordinates of the bounding box, and P_w, P_h are the offsets from the grid position and normalized between $[0, 1]$. When there is no target in the grid, the confidence level is 0. Therefore, the confidence level is equal to the predicted intersection over union (IOU) value. The calculation is as expressed in (3).

$$confidence = Pr(object) * IOU_{predict}^{truth}. \tag{3}$$

The YOLO network uses the sum of the mean square errors as a loss function (4): coordinate error (*coorErr*), IOU error (*iouErr*), and classification error (*clsErr*).

$$loss = \sum_{i=0}^{S^2} coorErr + iouErr + clsErr. \tag{4}$$

Compared with the YOLOv2 network, two modifications were made in the YOLOv3 network to improve the position accuracy and recall rate of an object. First, the Softmax classifier was replaced with a logistic classifier to improve the classification ability of overlapping category labels. Second, a feature pyramid network multilayer prediction architecture was used to strengthen the prediction ability of small objects.

The YOLOv3 network uses the previous 52 layers of darknet-53 (without a fully connected layer). YOLOv3 has three scale feature maps for detection. The corresponding grid cell sizes are 13×13, 26×26, and 52×52. Each cell in the feature map can predict three bounding boxes. Because for the 13×13 feature map, the sizes of the proposal boxes are 116×90, 156×198, and 373×326, respectively, it is suitable for detecting large objects. Concurrently, because for the 52×52 feature map, the sizes of the proposal boxes are 10×13, 16×30, and 33×23, respectively, it is suitable for detecting small objects.

3.2.2. *Kalman filter algorithm.* A Kalman filter is used to track the YOLOv3 objects. In particular, when a YOLOv3 object is obscured by other moving vehicles, the Kalman filter can still predict the position of the object. Until YOLOv3 detects the same object again, the prediction position of the Kalman filter is updated with the new measured position of YOLOv3.

The state and observation equations of the Kalman filter are as follows:

$$\boldsymbol{X}_k = \mathbf{A}\boldsymbol{X}_{k-1} + \boldsymbol{w}_k = \begin{bmatrix} 1 & 0 & 1/30 & 0 \\ 0 & 1 & 0 & 1/30 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_{k-1} \\ y_{k-1} \\ \dot{x}_{k-1} \\ \dot{y}_{k-1} \end{bmatrix} + \boldsymbol{w}_k, \tag{5}$$

$$\boldsymbol{Z}_k = \boldsymbol{H}_k\boldsymbol{X}_k + \boldsymbol{v}_k = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x_k \\ y_k \\ \dot{x}_k \\ \dot{y}_k \end{bmatrix} + \boldsymbol{v}_k, \tag{6}$$

where $\boldsymbol{X}_k$ is the system state vector at time k, $\mathbf{A}$ is the system transition matrix, $\boldsymbol{w}_k$ is the system noise at time k (assuming the system noise covariance = 0), $\boldsymbol{Z}_k$ is the measurement vector at time k, $\boldsymbol{H}_k$ is the measurement model, and $\boldsymbol{v}_k$ is the measurement noise at time k (assuming the measurement noise covariance = 1).

3.2.3. *Image distance estimation.* The imaging subsystem has an independent object distance estimation algorithm. When an image object moves from far to near, the image coordinates of the object change accordingly, and the relationship between the two is nonlinear. A polynomial curve equation is used to calculate the distance of an object (7).

$$f(y_{im}) = P_0 y_{im}^5 + P_1 y_{im}^4 + P_2 y_{im}^3 + P_3 y_{im}^2 + P_4 y_{im} + P_5, \tag{7}$$

where y_{im} represents the position of the object in the y-axis coordinates of the image.

Using the SFTOOL toolbox of MATLAB, a total of 50 sample pairs with known distances and the ground position images of an object are utilized to determine the parameters of the polynomial curve fitting equation expressed in (8).

$$f(y_{im}) = -0.1983y_{im}^5 + 1.203y_{im}^4 - 2.337y_{im}^3 + 1.922y_{im}^2 - 2.172y_{im} + 3.958. \quad (8)$$

The proposed image distance estimation is applicable only to flat roads. When traveling uphill or downhill sections, the distance estimation error becomes large.

3.2.4. *On-road lane detection.* An on-road lane is detected using the EDLines algorithm [19,20]. The five steps of the EDLines algorithm are as follows.

Step 1: Lane-line color enhancement: The original RGB image is converted to an HSV image. A color mask is imposed on the HSV image to strengthen the location of the on-road lane.

Step 2: A low-pass filter (such as a Gaussian filter) is used to suppress the noise of grayscale images and to output smooth images.

Step 3: The gradient magnitude of a smoothed image and the direction of each pixel element are calculated.

Step 4: The peak of the gradient graph is called the anchor point. The anchor point is highly probable to be located on the edge line.

Step 5: Using the least squares method, the on-road line is completed by connecting the anchor points.

The results of the lane detection under different weather scenarios are shown in Figure 5.

(a) Daytime (b) Rainy (daytime) (c) Nighttime (d) Rainy (nighttime)

FIGURE 5. EDLines detection results in different weather scenarios

3.3. **Fusion and HMI subsystem.** The fusion and human-machine interface (HMI) subsystem is divided into two parts.

3.3.1. *Fusion algorithm.* Two-dimensional coordinates in the MMW radar are projected onto the image system. This study uses a feedforward RBF neural network architecture to transform the radar coordinates.

The camera is installed at an angle parallel to the horizon. When a target object moves from far away to nearby, the position of its center point slightly changes near the center point of the image in the vertical direction. Thus, the variation in the vertical direction (v coordinate) of the image is insignificant. Therefore, the corresponding u coordinate of the horizontal direction in the image is considered as the output of the RBF network.

The three-layer network is composed of input, hidden, and output layers. The longitudinal and lateral distances of an MMW radar object are used as the input of the RBF neural network. The corresponding u coordinate of the horizontal direction in the image is the output of the RBF network. The hidden layer has eight neurons. The RBF and the norm are the Gaussian function and the Euclidean distance, respectively.

Because an MMW radar has high reflectivity for metal objects, a metal triangular reflector was used as a coordinate alignment object. The triangular reflector was placed randomly inside the ROI of the lane (1-12 m) in front of the vehicle. A total of 280 sets of radar and image coordinates were collected as training samples for the RBF network modeling. The detailed modeling process can be found in our previous paper [17].

3.3.2. *HMI.* The radar and image subsystems use particle and Kalman filters, respectively, to track objects. Therefore, the proposed system can determine the relative distance and speed of an object. The calculation formula of the TTC is as follows:

$$\mathrm{TTC} = \frac{d}{V_R}, \tag{9}$$

where d and V_R are the relative distance and relative speed between the experimental vehicle and an object, respectively.

The PET is mainly used to examine whether the safety distance is sufficient and to determine whether the vehicle is potentially dangerous. The calculation formula of the PET is as follows:

$$\mathrm{PET} = \frac{d}{V}, \tag{10}$$

where V is the speed of the experimental vehicle.

The warning indices (TTC and PET) have the same definition as provided by the National Highway Traffic Safety Administration in 2013 [21]. Based on the TTC and PET indices, three levels of collision warnings are triggered by the fusion subsystem:

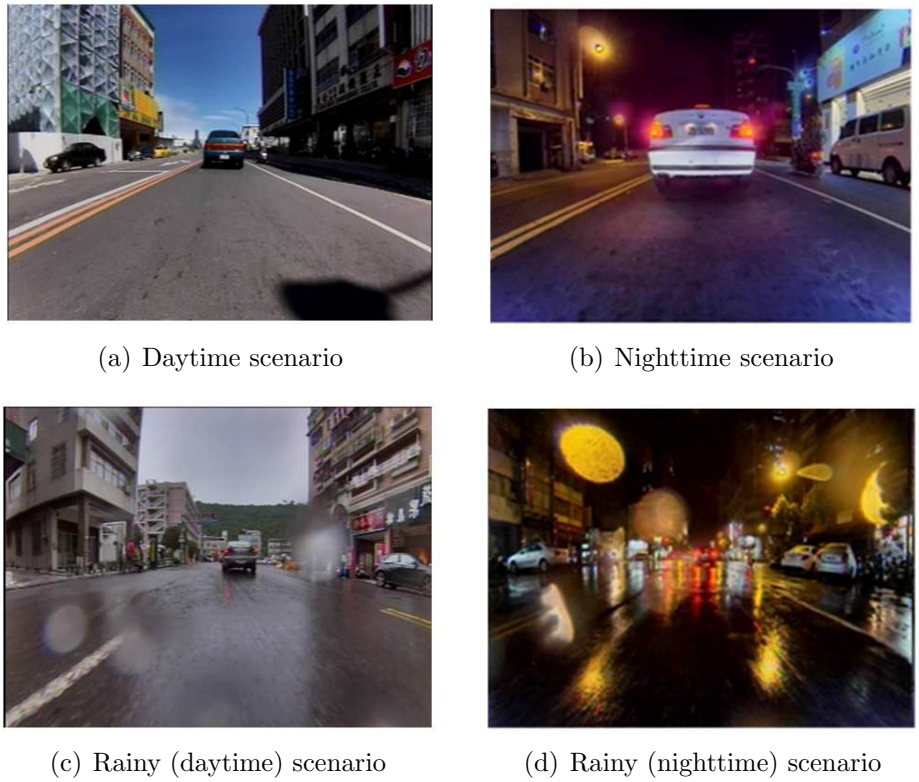
- Green level: PET $\leq$ 1.5 s
- Yellow level: PET $\leq$ 1 s
- Red level: TTC $\leq$ 2.5 s

The HMI uses a warning index that combines text and color to indicate the detection results. The text message includes three parts: identification results of four types of objects (pedestrians, bicycles, motorcycles, and cars), sensor type (R: radar, I: image, and F: fusion), and the distance of the objects in front of the car (meter). When the sensor type is radar, only the sensor type (R) and the distance of objects (meter) are displayed.

4. **Experiments.** The experimental field is mainly an urban road near National Sun Yat-sen University, Kaohsiung, Taiwan. Considering the actual on-road conditions, the weather scenarios are four types of weather conditions: daytime, nighttime, rainy daytime, and rainy nighttime.

The daytime scenarios comprise direct sunlight interference, rain, reflection of road ponding, and shadow environment. The nighttime scenarios are brake light reflection, headlight reflection, and insufficient light source environment. In a rainy scenario, raindrops may adhere to the camera lens. Under the influence of raindrops, the light could be haloed, causing an object to be invisible (shown in Figure 6).

First, all objects appearing in the ROI of the experimental image are manually counted. When an object appears in the first frame of the image, the object category is labeled manually. When the object disappears in the ROI of the image, all image frames are counted. The type and number of objects in front of the vehicle in each experimental scenario are listed in Table 1. The manually labeled objects in Table 1 are also the ground truths for determining if the radar or imaging subsystem performs accurate detection.



(a) Daytime scenario (b) Nighttime scenario

(c) Rainy (daytime) scenario (d) Rainy (nighttime) scenario

FIGURE 6. Four kinds of weather scenarios

TABLE 1. The type and quantity of the objects in scenarios

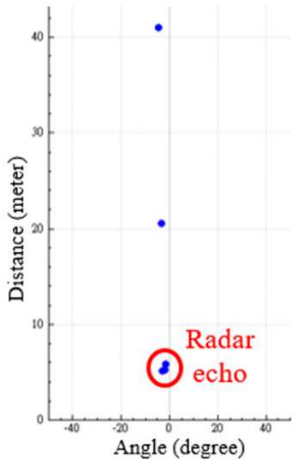
	Daytime	**Nighttime**	**Rainy daytime**	**Rainy nighttime**	**Quantity**
Car	16702	11654	9014	19090	56460
Motorcycle	9027	15457	1653	2639	28776
Bicycle	161	0	14	0	175
Pedestrian	810	3494	417	812	5533

4.1. **MMW radar detection subsystem experiment.** Compared with the camera sensor, the MMW radar subsystem is unaffected by ambient light and is suitable for all types of weather conditions. The experimental results of the radar subsystem in the rainy scenarios are shown in Figure 7 and listed in Table 2.

The related radar echoes in Figures 7(b) and 7(d), shown as red points, are the raw data of the MMW radar, and the blue points are the measurement outputs after the application of the DBSCAN and particle filter algorithms. As the number of radar objects changes with different test scenarios, the HMI system shows the object closest to the vehicle. Other objects are still tracked continuously and projected onto the image subsystem.

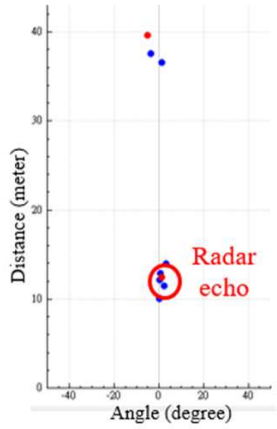
In Figure 7(a), the HMI displays warning index "R: 4.67" in red color. The red warning index represents in order a radar detecting object, distance of 4.67 m, and red level warning (TTC $\leq$ 2.5 s). In Figure 7(c), the HMI displays warning index "R: 11.29" in green color. The green warning index represents in order a radar detecting object, distance of 11.29 m, and green level warning (PET $\leq$ 1.5 s). The MMW radar subsystem can only detect objects; it does not have the ability to classify objects. In Figures 7(b) and 7(d), the radar echo of the object which displayed on the image screen is addressed with a red circle and text description.

(a) Daytime scenario



(b) Radar echo

(c) Nighttime scenario



(d) Radar echo

FIGURE 7. (color online) Experiment results of radar subsystem in rainy scenarios

TABLE 2. The experiment results of MMW radar subsystem

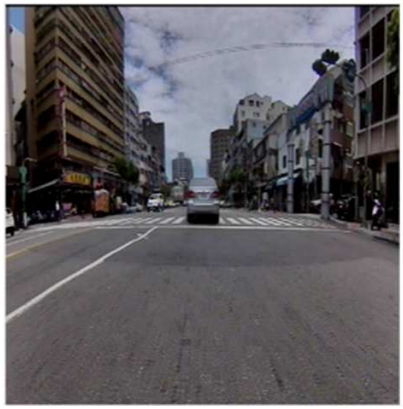
	Object	**No detection**	**Detection rate**
Car	56460	1789	0.968
Motorcycle	28776	5235	0.818
Bicycle	175	88	0.497
Pedestrian	5533	692	0.875
Average precision	90944	7804	0.914

In Table 2, radar detection rates of four types of objects are compared. It can be found that the detection performance of the radar for bicycles is poor. A possible reason is that the reflection area of a bicycle metal is small.

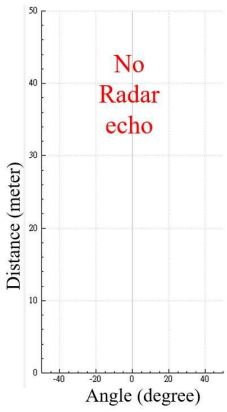
The two main failure cases of the MMW radar are as follows:

- MMW radar failure case 1: The relative speed of the object is less than 5 km/h.
- MMW radar failure case 2: Static object.

Both in Figures 8(a) and 8(b), although there is an object on the image screen (left side), there is no object echo on the radar screen (right side).

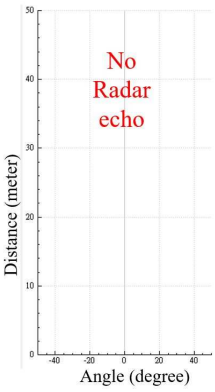


(a) Low relative speed



Radar failure case 1

(b) Static object



Radar failure case 2

FIGURE 8. MMW radar failure case

4.2. **Image detection subsystem experiment.** The experimental results of the image subsystem in the rainy scenarios are shown in Figure 9 and summarized in Table 3. In Figure 9, the objects detected by YOLOv3 are presented in a bounding box, and the object category is marked at the top of the bounding box. Table 3 shows that compared to the radar subsystem, the image subsystem has a higher detection rate for bicycles. This is because bicycles mainly appear in the daytime scenario, which is conducive for image detection.

(a) Rainy nighttime

(b) Rainy daytime

FIGURE 9. Experiment results of image subsystem in rainy scenarios

TABLE 3. The experiment results of image subsystem

	Object	**No detection**	**Detection rate**
Car	56460	291	0.995
Motorcycle	28776	1101	0.962
Bicycle	175	22	0.874
Pedestrian	5533	878	0.841
Average precision	90944	2292	0.975

The two main failure cases of the image subsystem shown in Figure 10 are

- Image failure case 1: The object distance is very long or the object is extremely small for YOLOv3, as shown in the yellow bounding box in Figure 10(a).
- Image failure case 2: Classification failure for YOLOv3. However, the classification failure will not affect the object tracking and distance estimation. One classification failure (a motorcycle is identified as a pedestrian) is shown in Figure 10(b).

(a) Image failure case 1 (b) Image failure case 2

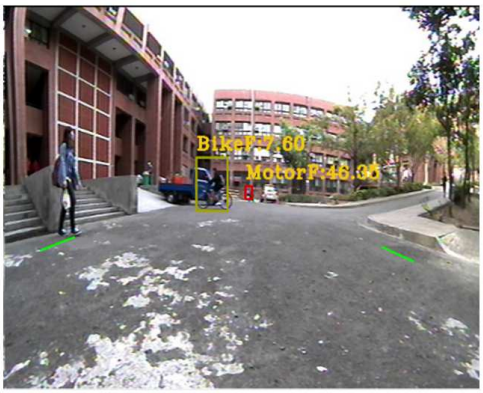
FIGURE 10. Image failure case

4.3. **Sensor fusion experiment.** In this study, a camera and an MMW radar are integrated in a sensor fusion architecture to compensate the limitations of the individual sensors. The fusion results of object detection for the four types of scenarios are shown in Figure 11 and summarized in Table 4.

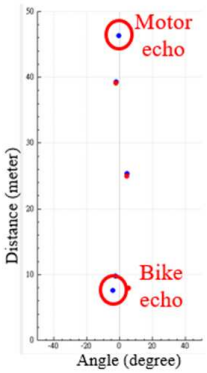
In Figure 11(a), the HMI displays nearest warning index "Bike F: 7.60" in yellow color. The yellow warning index represents in order the object category of bike, the fusion object, distance of 7.6 m, and yellow level (PET $\leq$ 1 s). Corresponding to the radar echo plot on the right side, the radar echo of the tracking object is addressed with a red circle and text description. In Figure 11(c), due to the independent object distance estimation function of the image subsystem, the fusion mechanism will search for the suitable radar echo. The radar echo of the fusion object is addressed with a red circle and text description. The proposed fusion system has the advantages of accurate radar distance and image object classification.

The advantages of the proposed system can compensate the failures of each sensor. When the MMW radar detection fails, the image subsystem automatically compensates by object tracking, and vice versa (shown in Figure 12).

In Figure 12(a), the HMI displays the warning index of the nearest object in the ROI of the image: "Car F: 8.65" in yellow color. The yellow warning index represents in order the object category of car, the fusion object, distance of 8.65 m, and yellow level (PET $\leq$

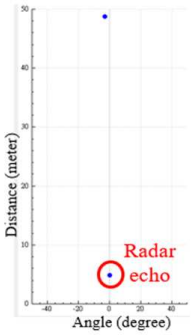


(a) Bike detection in daytime



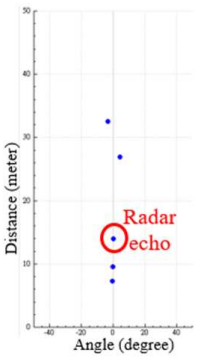
Radar echo

(b) Pedestrian detection in rainy daytime

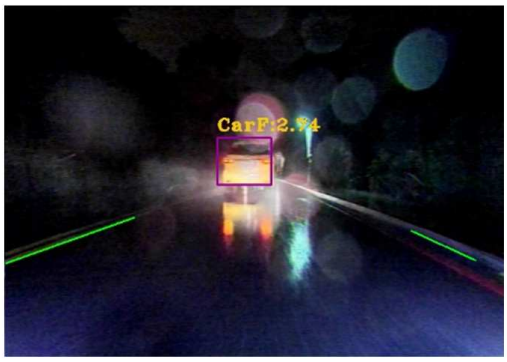


Radar echo

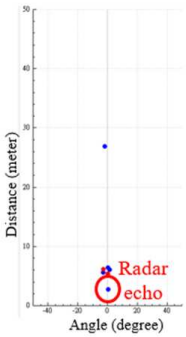
(c) Motorcycle detection at nighttime



Radar echo



(d) Car detection in rainy nighttime



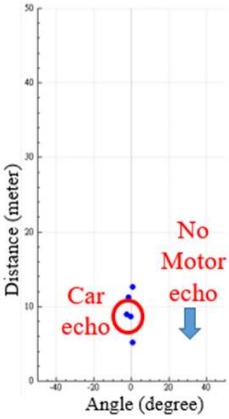
Radar echo

FIGURE 11. (color online) Experiment results of sensor fusion system in all scenarios

TABLE 4. The experiment results of sensor fusion subsystem

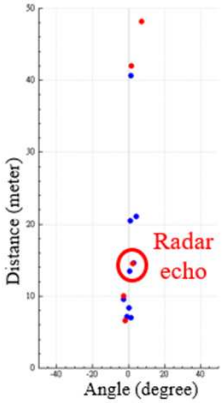
	Object	No detection	Detection rate
Car	56460	78	0.998
Motorcycle	28776	759	0.974
Bicycle	175	22	0.874
Pedestrian	5533	278	0.949
Average precision	90944	1137	0.987

(a) Motorcycle detection by image



Radar failure

(b) Motorcycle detection by MMW radar



Radar echo

FIGURE 12. (color online) Fusion system compensates each single sensor failure

1 s). Corresponding to the radar echo plot on the right side, the radar echo is addressed with a red circle and text description.

Although the motorcycle on the right side of the lane is detected by YOLOv3, because the object has the same speed as the experimental vehicle, no motorcycle echo is displayed in the corresponding radar echo plot. The warning index of the motorcycle is "Motor I: 4.80" in yellow color. The yellow warning index represents in order the object category of motorcycle, the image object, distance of 4.8 m, and yellow level (PET $\leq$ 1 s). The image subsystem is independently responsible for object distance estimation.

Based on Figure 12(b), if the image subsystem does not track an object at a certain moment but the radar still detects it, the information of the object is updated by the radar.

The comparison results of the radar, image subsystem, and fusion system are summarized in Table 5. Because the radar has a poor detection rate for bicycles, the detection of bicycles is mainly dominated by the image subsystem. The bicycle detection rate of the fusion system is the same as that of the image subsystem.

TABLE 5. The experiment results of sensor fusion subsystem

	Radar	**Image**	**Fusion**
Car	0.968	0.995	**0.998**
Motorcycle	0.818	0.962	**0.974**
Bicycle	0.497	**0.874**	**0.874**
Pedestrian	0.875	0.841	**0.949**

Based on Table 5, the proposed sensor fusion system improves the overall detection rate effectively, and its detection rate is higher than those of both single sensors in all scenarios.

The comparison results of the proposed system and some reported in the previous literature are summarized in Table 6. Compared to the previous study, the proposed system has a higher detection rate, more robust capability of object tracking, and adaptability to the environment.

TABLE 6. The comparison results between the proposed system and the previous literature

Sensor	**Object**	**Weather**	**Hardware**	**Time cost** (ms/frame)	**Detection rate**
Camera, radar [9]	Car	Daytime	Intel i7 3.4 GHz	32	92.36%
Camera [10]	Car	Daytime, rainy, fog	Intel i7 3.3 GHz	43	–
Camera [11]	Car, truck, pedestrian	Daytime	Intel i7 4 GHz, GPU Titan	30	84%
Camera, radar [13]	Car, pedestrian	Daytime	Intel i7 IPC	50	83%
Camera, radar (proposed)	Car, bicycle, motorcycle, pedestrian	Daytime, nighttime, rainy	Intel i7 2.6 GHz, GPU 1080Ti	25	98.7%

5. **Conclusions.** In this study, a collision warning system based on a parallel fusion architecture is developed using a camera and an MMW radar. The system detects and identifies obstacles in front of a vehicle. The radar detection system uses a particle filter to track the target. The image system uses the YOLOv3 network for object detection. The types of objects are pedestrians, bicycles, motorcycles, and cars. Radar object information is fused into image coordinates by an RBF neural network. The system is evaluated under four weather scenarios (daytime, nighttime, rainy daytime, and rainy nighttime). The detection rate of the sensor fusion reaches 98.7% and is higher than those of the single sensors. In the future, a controller area network bus and a 79-GHz MMW radar will be integrated into the proposed system to expand the related application scope of ADAS.

Acknowledgment. This research was funded by Xiamen University Tan Kah Kee College (no. JG2019SRF06). The authors also gratefully acknowledge the helpful comments and suggestions of the reviewers, which have improved the presentation.

REFERENCES

[1] T.-J. Lee, D.-H. Yi and D.-I. D. Cho, A monocular vision sensor-based obstacle detection algorithm for autonomous robots, *Sensors*, vol.16, no.3, 2016.

[2] T. Wang et al., Integrating millimeter wave radar with a monocular vision sensor for on-road obstacle detection applications, *Sensors*, vol.11, no.9, pp.8992-9008, 2011.

[3] A. Prioletti et al., Part-based pedestrian detection and feature-based tracking for driver assistance: Real-time, robust algorithms, and evaluation, *IEEE Transactions on Intelligent Transportation Systems*, vol.14, no.3, pp.1346-1359, 2013.

[4] H. Hassannejad et al., Detection of moving objects in roundabouts based on a monocular system, *Expert Syst.*, vol.42, no.9, pp.4167-4176, 2015.

[5] W. Song et al., Lane detection and classification for forward collision warning system based on stereo vision, *IEEE Sensors Journal*, vol.18, no.12, pp.5151-5163, 2018.

[6] S. H. Jeong et al., Technology analysis and low-cost design of automotive radar for adaptive cruise control system, *International Journal of Automotive Technology*, vol.13, no.7, pp.1133-1140, 2012.

[7] D. Kang and D. Kum, Camera and radar sensor fusion for robust vehicle localization via vehicle part localization, *IEEE Access*, vol.8, pp.75223-75236, 2020.

[8] P. Wei, L. Cagle, T. Reza, J. Ball and J. Gafford, LiDAR and camera detection fusion in a real-time industrial multi-sensor collision avoidance system, *Electronics*, vol.7, no.6, 2018.

[9] X. Wang et al., On-road vehicle detection and tracking using MMW radar and monovision fusion, *IEEE Transactions on Intelligent Transportation Systems*, vol.17, no.7, pp.2075-2084, 2016.

[10] V. Nguyen et al., A study on real-time detection method of lane and vehicle for lane change assistant system using vision system on highway, *International Journal of Engineering Science and Technology*, vol.21, no.5, pp.822-833, 2018.

[11] Y. Chen et al., Multi-task learning for dangerous object detection in autonomous driving, *Information Sciences*, vol.432, pp.559-571, 2018.

[12] T. Ogawa et al., Pedestrian detection and tracking using in-vehicle Lidar for automotive application, *2011 IEEE Intelligent Vehicles Symposium (IV)*, Baden-Baden, Germany, 2011.

[13] F. A. R. Alencar et al., Fast metric tracking by detection system: Radar blob and camera fusion, *2015 12th Latin American Robotics Symposium and 2015 3rd Brazilian Symposium on Robotics (LARS-SBR)*, Uberlandia, Brazil, 2015.

[14] A. Manjunath et al., Radar based object detection and tracking for autonomous driving, *IEEE MTT-S International Conference on Microwaves for Intelligent Mobility (ICMIM)*, Munich, Germany, 2018.

[15] S. Wang et al., Cognitive structure adaptive particle filter for radar maneuvering target tracking, *IET Radar, Sonar & Navigation*, vol.13, pp.23-30, 2019.

[16] A. Mahabub, M. I. Mahmud and M. F. Hossain, A robust system for message filtering using an ensemble machine learning supervised approach, *ICIC Express Letters, Part B: Applications*, vol.10, no.9, pp.805-811, 2019.

[17] Y.-W. Hsu et al., Developing an on-road object detection system using monovision and radar fusion, *Energies*, vol.13, no.1, pp.1-18, 2019.

[18] M. Krišto, M. Ivasic-Kos and M. Pobar, Thermal object detection in difficult weather conditions using YOLO, *IEEE Access*, vol.8, pp.125459-125476, 2020.

[19] C. Topal and C. Akinlar, Edge drawing: A combined real-time edge and segment detector, *Journal of Visual Communication and Image Representation*, vol.23, no.6, pp.862-872, 2012.

[20] C. Lee and J. Moon, Robust lane detection and tracking for real-time applications, *IEEE Transactions on Intelligent Transportation Systems*, vol.19, no.12, pp.4043-4048, 2018.

[21] N. Nadimi, H. Behbahani and H. Shahbazi, Calibration and validation of a new time-based surrogate safety measure using fuzzy inference system, *Journal of Traffic and Transportation Engineering*, vol.3, no.1, pp.51-58, 2016.

Author Biography

Yi-Horng Lai received the B.S. degree from the Chinese Naval Academy, Taiwan in 1992, and the M.S. degree in weapon system engineering from the Chung Cheng Institute of Technology, Taiwan, in 2001 and the Ph.D. degree in mechanical and electro-mechanical engineering from the National Sun Yat-sen University, Kaohsiung, Taiwan in 2017. From 2017 to 2019, he was a postdoctoral fellow in the National Sun Yat-sen University, Taiwan. He is currently an associate professor in the School of Mechanical and Electrical Engineering, Xiamen University Tan Kah Kee College, China. His research interests include control theory, systems engineering, and bio-signal processing.

Yu-Wen Chen received the B.S. degree from the Tamkang University, Taiwan in 2017, and the M.S. degree in mechanical and electro-mechanical engineering from the National Sun Yat-sen University, Kaohsiung, Taiwan, in 2019. His research interests include systems engineering and applications of control systems.

Jau-Woei Perng received the B.S. and M.S. degrees in electrical engineering from the Yuan Ze University, Chungli, Taiwan, in 1995 and 1997, respectively, and the Ph.D. degree in electrical and control engineering from the National Chiao Tung University (NCTU), Hsinchu, Taiwan, in 2003. From 2004 to 2008, he was a Research Assistant Professor with the Department of Electrical and Control Engineering, NCTU. Since 2008, he has been with the Department of Mechanical and Electromechanical Engineering, National Sun Yat-sen University, Kaohsiung, Taiwan, where he is currently a Professor. His research interests include robust control, nonlinear control, fuzzy logic control, neural networks, mobile robots, systems engineering, artificial intelligence, and intelligent vehicle control.