Fusion Point Pruning for Optimized 2D Object Detection with Radar-Camera Fusion

Lukas Stäcker[1,2], Philipp Heidenreich[1], Jason Rambach[3], and Didier Stricker[2,3]

[1]Stellantis, Opel Automobile GmbH, Germany
[2]Technische Universität Kaiserslautern, Germany
[3]German Research Center for Artificial Intelligence, Germany

Abstract

Object detection is one of the most important perception tasks for advanced driver assistant systems and autonomous driving. Due to its complementary features and moderate cost, radar-camera fusion is of particular interest in the automotive industry but comes with the challenge of how to optimally fuse the heterogeneous data sources. To solve this for 2D object detection, we propose two new techniques to project the radar detections onto the image plane, exploiting additional uncertainty information. We also introduce a new technique called fusion point pruning, which automatically finds the best fusion points of radar and image features in the neural network architecture. These new approaches combined surpass the state of the art in 2D object detection performance for radar-camera fusion models, evaluated with the nuScenes dataset. We further find that the utilization of radar-camera fusion is especially beneficial for night scenes.

1. Introduction

To further advance the field of advanced driver assistant systems and autonomous driving, a robust perception of the environment is needed. Since single sensor setups typically have limitations, current research aims at leveraging the strengths of diverse sensor combinations. Radar-camera fusion is a particularly interesting sensor combination for the automotive industry due to its moderate cost, complementary environmental information and possibility for unobtrusive sensor integration.

One of the main challenges of radar-camera fusion is to handle the heterogeneity of the data sources. While the camera provides raw sensor information in the form of RGB images, the radar typically provides a pre-processed list of detections, containing measurements of the distance, relative velocity, azimuth angle, and radar cross-section (RCS). Note that the information from the radar is sparse when compared with the dense camera images. A common approach for fusing the different representations is to project the radar detections onto the image plane to construct additional channels with parameters of the radar detections. The radar detections can be represented in the form of a small pixel area around the projected image location [3], or a vertical line of fixed height [19] to reflect the elevation uncertainty of the radar measurement. In this paper, we suggest to additionally incorporate the azimuth uncertainty when constructing the radar channels. In particular, we assume that the measured azimuth angle follows a Gaussian distribution around the actual measurement, where the standard deviation corresponds to the accuracy value from the sensor's technical data sheet. Thereby, we are able to effectively use the additional uncertainty information and create visually denser radar channels. An example of the obtained representation is shown in Fig. 1.

Figure 1. Visualization of the camera image, augmented with the uncertainty weighted RCS channel values highlighted in red.

When designing a neural network to process both radar and camera channels, it is challenging to determine the ideal

method for how and when to fuse the branches of radar and camera. In the literature, fusion strategies are often categorized into early, deep and late fusion [5], [10], [4]. When choosing one of these fusion strategies, the main concerns are that fusing too early may not be ideal considering the heterogeneity of the data sources, while fusing too late may not enable the network to take advantage of potential synergies. A recent work from Nobis et al. [19] aims to solve this problem by designing a network architecture that concatenates radar and camera features at multiple early, deep and late stages in the network, to which we refer as fusion points. Their idea is to enable the network to adjust its weights for each fusion point during training and thereby to implicitly find the ideal fusion points. However, this approach does not lead to significant performance improvements compared to an image-only baseline, which shows that the network is not able to fully leverage the advantages of the added radar modality. In contrast to [19], we show that, surprisingly, too many fusion points can limit the performance of the network. We introduce a technique that is inspired by network pruning and automatically selects the ideal fusion points during training, effectively optimizing the network architecture and performance.

To summarize, the main contributions of this paper are:

- Two new projection techniques to create dense radar channels by incorporating elevation and azimuth uncertainty.
- A fusion point pruning (FPP) technique to automatically optimize the network architecture.
- An overall radar-camera fusion pipeline that achieves state-of-the-art performance in 2D object detection evaluated using the nuScenes [2] dataset.

2. Related work

2.1. Image-only object detection

The use of deep neural networks for object detection in images has lead to a significant increase in detection accuracy. The early methods are characterized by a two-stage approach: First, a region proposal network (RPN) is used to identify regions of interest (ROIs) in the image. Second, an image classification network is used to classify the object in each ROI. An exemplary algorithm using this approach is Faster R-CNN [21]. To reduce the computational cost of these approaches, one-stage object detectors were introduced. They eliminate the need for a RPN by replacing the ROIs with pre-defined anchor boxes. Examples of this technique include SSD [16] and RetinaNet [15]. In this paper, we build our architecture based on a modified variant of the RetinaNet architecture.

2.2. Radar-camera fusion object detection

Most of the radar-camera fusion approaches for 2D object detection use a modified image-only detection network that is enhanced by the integration of the radar.

For two-stage approaches, the radar is often used to find ROIs in the image, which can then be used for further image processing. In [6], the radar determines square ROIs in the image, which are then classified by a custom CNN with a contrastive loss function. More recently, Nabati & Qi [17] show that the RPN can be replaced by a radar-based region proposal to reduce computational cost and obtain better results on the nuScenes dataset [2]. One of their key ideas is to generate multiple anchor boxes per radar detection, which have different sizes, scales and alignments with the radar point either in the center, to the left, right or bottom of the box. In [18], the authors further develop this idea by generating 3D proposals for each radar detection, mapping them to the image and refining them using radar and image features to generate 2D radar proposals. These are then fused with 2D image proposals created by a RPN and used as input for the detection stage of the algorithm. A limitation of these sequential approaches is that the performance at each stage is limited by the corresponding sensor, rather than exploiting the synergies of multiple sensor inputs used at the same stage.

To enhance one-stage approaches, most algorithms first create additional image channels by projecting the radar points onto the image. The idea is to enable the network to learn how to make the best use of the complementary inputs. John & Mita [10] introduce the RVNet, a 2D object detection algorithm with two input and two output branches. The input branches are for the radar and camera input, respectively. The radar detections are transformed to the image plane to form a three channel radar image using the measured depth and the lateral and longitudinal velocity. The two output branches are used to detect small and large obstacles. The authors also develop the SO-Net [11], which extends the RVNet by adding another output branch for semantic segmentation, making it a multi-task learning algorithm. Chadwick et al. [3] show that the radar is especially useful for the detection of distant, small vehicles. They project the radar detections as small circles to reflect their uncertainty and use the range and the range-rate as channels of the radar image. In contrast, Nobis et al. [19] argue that the radar detections should rather be projected as a vertical line, since the radar is mainly uncertain in terms of the object's elevation. They assume a fixed height for each object and use the distance and RCS as the radar channels. They also introduce a new training technique BlackIn, which blacks out the camera inputs at a fraction of the training steps, to encourage the network to rely more heavily on radar data. In this paper, we improve upon above mentioned techniques by incorporating azimuth uncertainty in

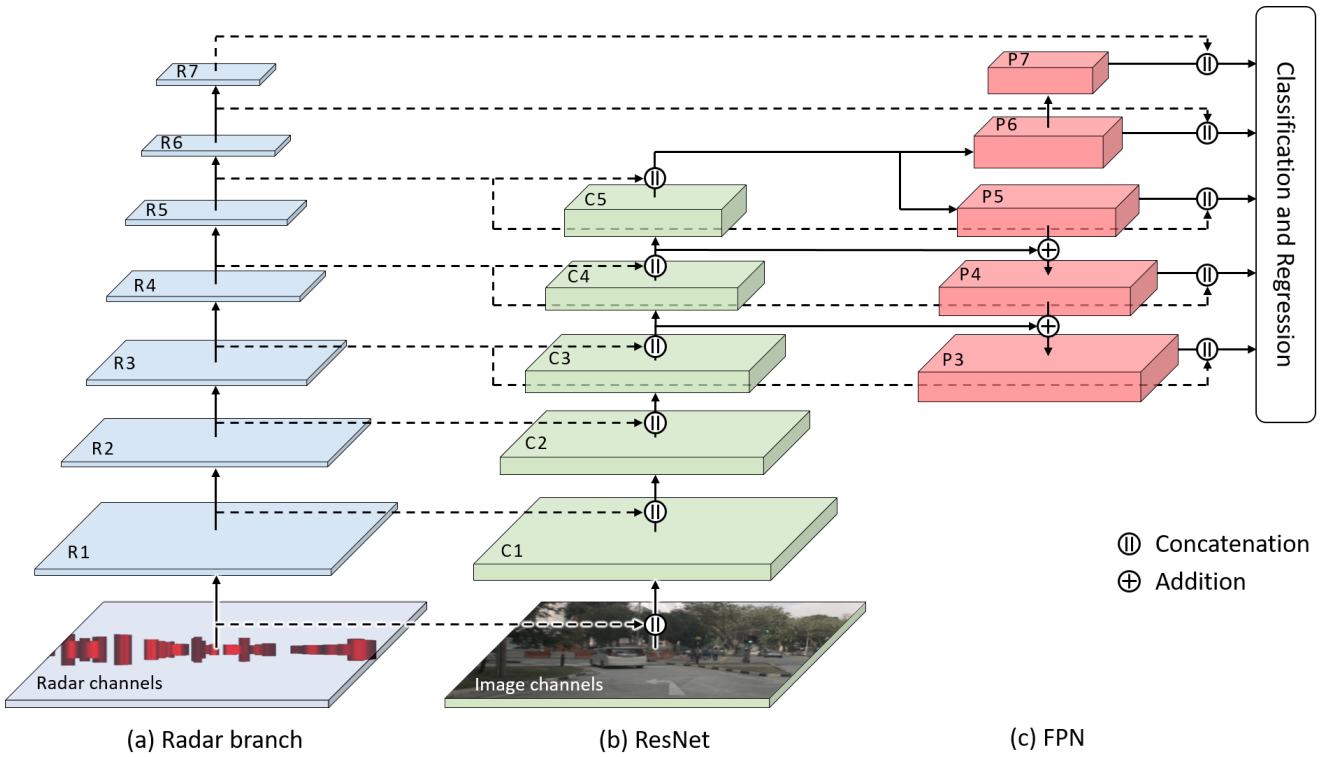

Figure 2. Model architecture showing the fusion points as concatenations of radar and image features. Dashed lines indicate that the connections are optional depending on the fusion point hyper-parameters.

the radar projection and introducing a method to automatically find the best fusion points in the network architecture.

2.3. Network pruning

Neural network pruning has been researched since the late 1980s [13], [9]. Its goal is to remove less relevant parts of the network to reduce computational cost and storage space while maintaining most of its performance. A recent overview of network pruning is given in [1].

Most network pruning techniques follow a simple high-level algorithm: The network is first trained to convergence with its original architecture. Then, its structural elements are assigned with an impact score that determines their importance. The least important structural elements are removed and the network is fine-tuned with its new architecture. The last steps can be repeated iteratively until a desired performance trade-off is achieved [7]. The impact score is mostly calculated using the magnitude of weights, but sometimes other metrics like gradients, contributions to layer activations or special importance coefficients are used [1]. Activation based impact scores were first introduced in [22] as the contribution of a single weight to the activation of a single neuron. Later works generalize this idea to the contribution of each channel in a layer to the convolutional filter activation [20], more closely related to the technique proposed in this paper. In contrast to the above works, we consider a pruning of fusion points, which deals with the challenge of calculating an impact score in the presence of heterogeneous data sources of different magnitude.

3. Methodology

3.1. Radar input generation

We project each radar detection onto the image plane using the measured azimuth angle and range, as well as the camera's extrinsic and intrinsic calibration parameters. The corresponding entries in the resulting radar channels can be filled with information from the detections like the measured distance, relative velocity, or RCS. To reflect the measurement uncertainty of the radar, we model the elevation and azimuth uncertainty in our projection technique. We follow [19] in the generation of vertical lines assuming a height of 3 m for each detection to represent the elevation uncertainty. Additionally, we introduce new techniques to further reflect the detection azimuth uncertainty. We propose the following two techniques which have been experimentally evaluated in this paper.

3.1.1 Uncertainty channel

The first method is an additional radar channel that contains information on the azimuth uncertainty of each detection.

We assume a Gaussian distribution for the measurement of the azimuth angle and calculate the density values using the measured azimuth angle as mean and the accuracy from the technical data sheet as standard deviation. Instead of a vertical line with a width of one pixel, we now generate a visually denser radar input where the corresponding entries of each detection are horizontally spread across several pixels, according to the Gaussian density curve. In the case of overlapping entries, we keep the highest value.

3.1.2 Uncertainty weighted RCS channel

The second method makes use of the idea of the first method but further enriches the radar input channel by its RCS information. To this end, we calculate the uncertainty channel as described in the previous section and multiply the density values of each detection with its measured RCS value. Since the RCS is a measure that represents object reflectivity and is often used for object size classification, we expect this to be a meaningful weighting. A visualization of the camera image, augmented with the uncertainty weighted RCS channel values highlighted in red, is given in Fig. 1.

3.2. Model architecture

The model architecture is shown in Fig. 2. It is based on a RetinaNet [15] architecture with a ResNet [8] backbone, a subsequent Feature Pyramid Net (FPN) [14] and the classification and regression detection heads. Additionally, we use a radar branch with max pooling layers to adjust the shape of the radar inputs according to the different stages in the network, similar to the CRF-Net introduced by [19]. The radar features are fed into the network on several different fusion points at early, deep and late stages of the network. The fusion operation is a concatenation of the respective feature tensors. The network is thereby expected to learn the ideal stages for the radar fusion.

In contrast to the CRF-Net [19], we add flexibility to the network architecture by introducing binary hyper-parameters to enable or disable each of the fusion points. We can use these hyper-parameters to experimentally examine the effectiveness of each fusion point. Also, we can dynamically change the network architecture during the training. We use pre-trained weights from ImageNet [12] for each layer that is identical with the RetinaNet architecture. The layers after each fusion point have additional input channels, whose weights are randomly initialized and get discarded when the corresponding fusion point is eliminated.

3.3. Fusion point pruning (FPP)

We propose a novel network pruning technique to improve the capability of the network to find the best fusion points. We note that this technique is not limited to radar-camera fusion as shown in this paper and can be applied to other fusion problems as well. After several epochs of training, we evaluate the relative importance of each fusion point to decide if it should be kept in the architecture.

A 2D convolution can be mathematically described using 3D tensors for input $\mathbf{I}$ and activation map $\mathbf{A}$ with width w, height h, as well as input and activation map channels c_I and c_A, respectively, and a 4D tensor for the kernel weights $\mathbf{K}$ with uneven kernel size k. For a stride of $s = 1$ and same padding $p = \frac{k-1}{2}$, an element of $\mathbf{A}$ at row m, column n and channel a can be calculated by:

$$\mathbf{A}[m,n,a] = \sum_{i=1}^{c_I}\sum_{u=1}^{k}\sum_{v=1}^{k}\mathbf{K}[u,v,i,a] \times \mathbf{I}[m+u-\frac{k+1}{2}, n+v-\frac{k+1}{2}, i] \quad (1)$$

For regular network pruning, the importance of neuron connections is usually determined by examining the magnitude of their corresponding weights [9], [1]. However, since we are dealing with heterogeneous data sources of different magnitude, this does not appear meaningful to us. Instead, we calculate the relative importance of the input channels at our fusion points using their impact on the next layer's activation map $\mathbf{A}$. To this end, we separately calculate the activations $\mathbf{A}_i$ for each of the input channels i in the first layer after a fusion point:

$$\mathbf{A}_i[m,n,a] = \sum_{u=1}^{k}\sum_{v=1}^{k}\mathbf{K}[u,v,i,a] \times \mathbf{I}[m+u-\frac{k+1}{2}, n+v-\frac{k+1}{2}, i] \quad (2)$$

We then take the magnitude in terms of the L1-norm of each channel's activation and normalize it to determine the relative impact_i of each input layer i in the feature tensor at a fusion point:

$$\mathrm{impact}_i = \frac{\|\mathbf{A}_i\|_1}{\sum_{j=1}^{c_I}\|\mathbf{A}_j\|_1} \quad (3)$$

We sum up the relative impacts of the radar channels that were concatenated at each fusion point to measure the importance of fusion taking place at this point in the network. We iteratively eliminate the least effective fusion points and their corresponding weights in the following layer after a predefined number of training epochs and save the corresponding model checkpoint. This is repeated until a single fusion point remains. Finally, we use the model checkpoints to evaluate the performance and select the best performing model with the optimal set of fusion points.

Table 1. Quantitative evaluation

	Camera	Radar	Car	Truck	Person	Bus	Bicycle	Motorcycle	mAP	wmAP	Night wmAP	Runtime‡
Faster R-CNN* [21]	✓		51.46	33.26	27.06	47.73	24.27	25.93	34.95	43.78	—	—
RRPN* [17]	✓	✓	41.80	**44.70**	17.10	**57.20**	21.40	**30.50**	35.45	—	—	—
Nabati & Qi 2020* [18]	✓	✓	52.31	34.45	27.59	48.30	25.00	25.97	35.60	44.49	—	—
RetinaNet [15]	✓		55.69	31.89	37.10	—	21.80	27.27	34.75	45.65	43.24	36.4ms
CRF-Net [19]	✓	✓	53.71	33.72	36.50	—	18.62	22.09	32.93	44.91 (43.95†)	46.42	37.6ms
Ours - UC	✓	✓	55.40	34.15	37.23	—	23.27	28.14	35.64	45.99	49.56	37.1ms
Ours - UwRCS	✓	✓	55.46	35.26	37.63	—	25.02	27.43	36.16	46.33	48.11	36.8ms
Ours - UC + FPP	✓	✓	55.69	35.42	37.36	—	23.42	28.78	36.14	46.43	48.56	37.0ms
Ours - UwRCS + FPP	✓	✓	**55.94**	35.60	**37.77**	—	**25.74**	28.84	**36.78**	**46.73**	**50.53**	36.7ms

* The results in these rows were calculated by Nabati & Qi [18] and use a slightly different class mapping, which leads to the mAP metric being less comparable.
† Nobis et al. [19] only report this wmAP based on a slightly different class mapping. We therefore additionally list results that we calculated ourselves with the CRF-Net using our class mapping.
‡ Average inference runtime per frame was calculated on NVIDIA GeForce RTX 2080.

4. Experimental Results

We use the nuScenes dataset [2] to train and evaluate our network. To the best of our knowledge, this is currently one of the largest datasets for autonomous driving and also the only one with series-production automotive radar sensors. The nuScenes dataset contains 3D bounding box annotations of 27 classes. As in [19] and [18], we obtain 2D bounding box annotations by projecting the 3D bounding boxes onto the image plane, and apply a similar class mapping to obtain five high-level classes: car, person, bicycle, motorcycle and truck. We train and evaluate the algorithms using the front camera and radar with the official train and val scene splits containing 28130 training and 6019 validation frames. Additionally, we evaluate them using a subset of the validation scenes consisting of 608 frames that were taken at night. The networks are trained using the Adam optimizer with an initial learning rate of 10^{-5} that is reduced by a factor of 10 when the optimization reaches a plateau.

4.1. Quantitative Results

Experimental setup. The most common metric to evaluate object detection performance is the mean average precision (mAP), which is calculated as the mean of the average precision (AP) across each class c of the dataset. Working with the nuScenes dataset, many researchers choose individual class mappings to reduce the 27 classes to a set of selected classes C. A downside of the mAP is its dependency on the class mapping, where even small changes can lead to substantial differences in the calculated mAP. Hence, in recent work [19], [18], the weighted mean average precision (wmAP) is additionally reported, which is calculated as a weighted mean of the AP for each class, using the amount of objects per class N_c as weights and divided by the total amount of objects N. This metric is less dependent on the underlying class mapping. In turn, the classes with more objects tend to dominate the results of the wmAP. We calculate the mAP and wmAP using an intersection-over-union (IoU) threshold of 0.5 as:

$$\mathrm{mAP} = \frac{1}{C}\sum_{c=1}^{C}\mathrm{AP}_c \tag{4}$$

$$\mathrm{wmAP} = \frac{1}{N}\sum_{c=1}^{C}N_c\mathrm{AP}_c \tag{5}$$

We report both the mAP and the wmAP, as well as class APs to ensure the best comparison possible in Table 1. Additionally, we provide the night scene wmAP, which highlights the benefits of radar-camera fusion in adverse conditions for the camera. We list the results of two image-only networks, RetinaNet [15] and Faster R-CNN [21], which are used as baselines for three radar-camera fusion algorithms, RRPN [17], Nabati & Qi 2020 [18] and CRF-Net [19], as well as our own results with the uncertainty channel (UC) and uncertainty weighted RCS channel (UwRCS), with or without FPP. The results from Faster R-CNN, RRPN and Nabati & Qi 2020 were calculated in [18] and use a slightly different class mapping. Their mapping includes a Bus class that we chose to merge with the Truck class in our class mapping. We compute the remaining results with the same training and evaluation settings and based on the same RetinaNet [15] as core architecture to ensure comparability. In our experiments, we choose to work with a small ResNet-18 backbone due to the real-time requirements in automotive systems. However, the results should be applicable to larger backbones as well. In the CRF-Net paper, Nobis et al. [19] only report the wmAP based on a different class mapping, which we additionally list. They further list results which were produced by filtering data based on ground truth information that is not available in a real-world scenario, which we therefore discard.

Discussion. Looking at the results, we see that all of our own methods outperform the other methods in terms of mAP and wmAP, leading to a new state of the art in radar-camera fusion for 2D object detection. Our best performing model is the UwRCS channel in combination with FPP. It

Figure 3. Qualitative comparison of detection results. Red: Car, Green: Truck, Blue: Pedestrian.

should be noted that the mAP and wmAP of all methods are in a relatively close range. The best mAP is 3.85 above the lowest, which is a relative increase of 11.7%, while the relative increase in terms of the wmAP is even less. We can observe that the Nabati & Qi 2020 [18] model only slightly outperforms its image-only baseline Faster R-CNN, while the CRF-Net [19] does not outperform its image-only baseline RetinaNet [15] in our evaluation. From our own experiments, we find that this is due to the many fusion points in the CRF-Net architecture, which can lead to a suboptimal detection performance.

For our own methods, we started experimenting with the UC and UwRCS techniques and found that we significantly outperform the original CRF-Net [19] when reducing the amount of fusion points in the network architecture experimentally. To reduce the amount of experiments needed to find the optimal architecture, we developed the FPP technique. It determined that the best fusion point for the UC + FPP technique is the concatenation of the $R3$ radar feature with the $C3$ ResNet feature, while for UwRCS + FPP it is best to have two fusion points at $R3$ with $C3$ and at $R4$ with $C4$, which are all located in the centre of the neural network. The fusion points late in the network, after the FPN, have been eliminated first in our experiments, indicating that a regression level fusion is not ideal to exploit the complementary features. The fusion points early in the network, including the concatenation of the radar and image input channels, are also eliminated, indicating that it is beneficial to separately extract some semantic information before fusing the feature tensors. The UwRCS + FPP technique further increases the detection performance and performs consistently well across all classes. It has the highest AP for Car, Person and Bicycle and second highest for Truck and Motorcycle, where it is only outperformed by the RRPN [17]. In fact, the RRPN [17] performs particularly well for these classes but equally poorly for the other classes. This method seems to have some particular strengths and weaknesses that differ from all other methods. It should be noted, that the wmAP of RRPN was not reported by Nabati & Qi [18] but should be low due to the weak performance of the Car and Person classes, which have the highest number of annotated objects.

To further underline the potential of radar-camera fusion in adverse conditions for the camera, we study the performance for scenes that were recorded at night. We can observe a more significant increase in detection performance of all fusion methods with respect to the RetinaNet [15] baseline, with up to 16.8% relative increase in night wmAP for the UwRCS + FPP technique. Note that the night wmAP can be higher than the overall wmAP due to mostly cars present at night which have high AP.

The inference times of RetinaNet [15], CRF-Net [19] and our models are very similar because the radar branch is inexpensive relative to the rest of the network. The runtime slightly increases with the amount of fusion points in the network, so the FPP can help to reduce it.

4.2. Qualitative Results

In Fig. 3, we show some qualitative results of our proposed method with UwRCS and FPP, compared with the RetinaNet [15] image-only baseline. The color of the bounding boxes relates to the different classes. In the left column, we have a very crowded urban scene with multiple cars, trucks and pedestrians. We can see that the performance is relatively similar and most of the objects in the scene are correctly detected. The most notable difference is the detection of the car and the construction worker below the road sign on the right side, which both are partially occluded by a road barrier and thus missed by the RetinaNet [15]. The same happens for the car that is to the right of the truck on the left side of the image. In these circumstances, the radar can help to detect such occluded objects. Another example of this observation is shown in the center column, where the radar can help to detect a vehicle that is occluded by bushes and would have been missed by the RetinaNet [15]. In the right column, we have a night scene with difficult lighting conditions. There is only one car on the right of the image which is hardly visible in the camera image and thus missed by the RetinaNet [15]. However, our optimized fusion network is able to detect and localize the car quite well.

5. Conclusion

In this paper, we presented two novel approaches to optimize radar-camera fusion. First, we presented a new projection technique that takes the radar's elevation and azimuth uncertainty into account and creates a visually denser radar input. Second, we developed the FPP technique that automatically selects the best fusion points in the network architecture, effectively solving the problem of when to fuse camera and radar data. We note that this technique can be applied to other fusion problems as well. Our contributions lead to an improved state-of-the-art performance in 2D object detection, as shown in the results. While the improvements compared to image-only algorithms like RetinaNet [15] are still relatively small, we can demonstrate the potential of radar-camera fusion for night scenes, where the camera is less reliable. A possible drawback for the performance of the fusion models is the quality of the radar data in nuScenes. The radar pointclouds are less dense than what can be expected from current automotive radars, so the full potential of radar can not be demonstrated using this dataset. In future work, we will examine fusion in 3D object detection, where the radar might provide an even larger advantage than on the image plane.

References

[1] Davis Blalock, Jose Javier Gonzalez Ortiz, Jonathan Frankle, and John Guttag. What is the state of neural network pruning? *arXiv preprint arXiv:2003.03033*, 2020.

[2] H. Caesar, V. Bankiti, A. H. Lang, S. Vora, V. E. Liong, Q. Xu, A. Krishnan, Y. Pan, G. Baldan, and O. Beijbom. nuScenes: A multimodal dataset for autonomous driving. In *2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 11618–11628, 2020.

[3] S. Chadwick, W. Maddern, and P. Newman. Distant vehicle detection using radar and vision. In *2019 International Conference on Robotics and Automation (ICRA)*, pages 8311–8317, 2019.

[4] Shuo Chang, Yifan Zhang, Fan Zhang, Xiaotong Zhao, Sai Huang, Zhiyong Feng, and Zhiqing Wei. Spatial attention fusion for obstacle detection using MmWave radar and vision sensor. *Sensors*, 20(4):956, 2020.

[5] D. Feng, C. Haase-Schütz, L. Rosenbaum, H. Hertlein, C. Gläser, F. Timm, W. Wiesbeck, and K. Dietmayer. Deep multi-modal object detection and semantic segmentation for autonomous driving: Datasets, methods, and challenges. *IEEE Transactions on Intelligent Transportation Systems*, pages 1–20, 2020.

[6] F. Gaisser and P. P. Jonker. Road user detection with convolutional neural networks: An application to the autonomous shuttle WEpod. In *2017 Fifteenth IAPR International Conference on Machine Vision Applications (MVA)*, pages 101–104, 2017.

[7] Song Han, Huizi Mao, and William J. Dally. Deep compression: Compressing deep neural network with pruning, trained quantization and Huffmancoding. In *4th International Conference on Learning Representations (ICLR)*, 2016.

[8] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[9] Steven A Janowsky. Pruning versus clipping in neural networks. *Physical Review A*, 39(12):6600, 1989.

[10] Vijay John and Seiichi Mita. RVNet: Deep sensor fusion of monocular camera and radar for image-based obstacle detection in challenging environments. In *Pacific-Rim Symposium on Image and Video Technology*, pages 351–364. Springer, 2019.

[11] V. John, M. K. Nithilan, S. Mita, H. Tehrani, R. S. Sudheesh, and P. P. Lalu. SO-Net: Joint semantic segmentation and obstacle detection using deep fusion of monocular camera and radar. In *Pacific-Rim Symposium on Image and Video Technology*, pages 138–148. Springer, 2019.

[12] Alex Krizhevsky, Ilya Sutskever, and Geoffrey E Hinton. ImageNet classification with deep convolutional neural networks. *Advances in Neural Information Processing Systems*, 25:1097–1105, 2012.

[13] Yann LeCun, John S Denker, Sara A Solla, Richard E Howard, and Lawrence D Jackel. Optimal brain damage. In *NIPs*, volume 2, pages 598–605. Citeseer, 1989.

[14] Tsung-Yi Lin, Piotr Dollar, Ross Girshick, Kaiming He, Bharath Hariharan, and Serge Belongie. Feature pyramid networks for object detection. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 2117–2125, 2017.

[15] Tsung-Yi Lin, Priya Goyal, Ross Girshick, Kaiming He, and Piotr Dollar. Focal loss for dense object detection. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2980–2988, 2017.

[16] Wei Liu, Dragomir Anguelov, Dumitru Erhan, Christian Szegedy, Scott Reed, Cheng-Yang Fu, and Alexander C Berg. SSD: Single shot multibox detector. In *European Conference on Computer Vision*, pages 21–37. Springer, 2016.

[17] Ramin Nabati and Hairong Qi. RRPN: Radar region proposal network for object detection in autonomous vehicles. In *2019 IEEE International Conference on Image Processing (ICIP)*, pages 3093–3097, 2019.

[18] Ramin Nabati and Hairong Qi. Radar-camera sensor fusion for joint object detection and distance estimation in autonomous vehicles. *arXiv preprint arXiv:2009.08428*, 2020.

[19] F. Nobis, M. Geisslinger, M. Weber, J. Betz, and M. Lienkamp. A deep learning-based radar and camera sensor fusion architecture for object detection. In *2019 Sensor Data Fusion: Trends, Solutions, Applications (SDF)*, pages 1–7, 2019.

[20] A. Polyak and L. Wolf. Channel-level acceleration of deep face representations. *IEEE Access*, 3:2163–2175, 2015.

[21] S. Ren, K. He, R. Girshick, and J. Sun. Faster R-CNN: Towards real-time object detection with region proposal networks. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 39(6):1137–1149, 2017.

[22] Georg Thimm and Emile Fiesler. Evaluating pruning methods. In *International Symposium on Artificial Neural Ntworks*, pages 20–25. Citeseer, 1995.