Two-Dimensional Beamforming Automotive Radar with Orthogonal Linear Arrays

Shaogang Wang[1], Honglei Chen[2], Vishal M. Patel[3] and Athina Petropulu[1]

[1]Department of Electrical and Computer Engineering, Rutgers, The State University of New Jersey
[2]The MathWorks Inc.
[3]Johns Hopkins University

***Abstract*—We propose a new automotive radar architecture that achieves high resolution in range, range rate, azimuth and elevation angles, while requiring smaller antenna aperture and fewer reception channels as compared to conventional digital beamforming planar arrays. This is achieved by leveraging two orthogonally-placed digital beamforming arrays using the frequency modulation continuous waveform. The high resolution range-Doppler images generated by azimuth and elevation beams of the two arrays isolate each physical scatterer, thus, the azimuth and elevation angles can be precisely measured. To match the measurements of an object from azimuth and elevation beams, a deep learning based beam matching method is proposed, which converts the beam matching problem into an image patch matching problem in the range-Doppler domain. Furthermore, a new radar resource management algorithm is proposed, which schedules radar jobs by their time urgency as well as beam locations. Jobs falls into the same beams are scheduled together to optimally use the radar time resource and also reduces the computation introduced by the beam matching procedure. The advantage of the proposed radar is demonstrated by simulations.**

***Index Terms*—Automotive radar, deep learning, radar resource management**

I. Introduction

Automotive radars play an increasingly important role in self-driving vehicle applications. As compared with other sensors, such as lidar and camera, radars are less affected by adverse weather conditions. In addition, radars provide 4-dimensional (4-D) measurements, i.e., range, range rate, azimuth and elevation angles, which contain rich information of targets; this is very important in perception applications. However, while the state-of-the-art automotive radars (see description in the following), achieve high range and range rate resolution, their angular resolution is lower as compared to lidars. This is one of the main limitations of the current automotive radars [1].

The basic architecture of modern beamforming automotive radars [2] is composed of a transmit antenna, and a reception uniform linear array (ULA) with digitized channels. The transmit antenna forms a broad beam patten, which covers a large FOV, while the reception array forms multiple beams simultaneously, covering the same FOV of the transit beam patten. The angular resolution is determined by the beamwidth of the reception array, which is fundamentally limited by the antenna aperture. The larger the aperture, the narrower the beamwidth. Also, for non-ambiguous angular measurements in a large FOV, the array elements must be closely spaced. The largest FOV of non-ambiguous angle measurement that a ULA can achieve is $\pm \arcsin(\frac{\lambda}{2d})$, $\frac{\lambda}{2d} \in (0, 1]$, where λ and d are the wavelength and element-wise spacing, respectively. Hence, the half-wavelength spacing, i.e., $d = 0.5\lambda$ results into the largest FOV, i.e, $\pm 90^\circ$ [2]. The higher the angular resolution and the larger the FOV, the greater the number of channels are required. For instance, to achieve 3° resolution within an FOV of $\pm 60^\circ$ requires approximately 40 channels for a $77GHz$ radar. It is even harder to have both high angular resolution in azimuth and elevation while covering a large FOV. In that case, a planar array of $N_a N_e$ channels is required, where N_a, N_e are the number of channels in azimuth and elevation, respectively; this is not realistic for the automotive radar application, where the hardware and cost are highly constrained.

One way to achieve simultaneous azimuth and elevation angle measurements and also savings in the number of channels is to leverage two linear, orthogonally placed arrays [3]–[5]. Compared to the planar array that uses $N_a N_e$ reception channels, the two orthogonal linear arrays only have $N_a + N_e$ channels. However, to form thin fan beams either in azimuth or in elevation for a large FOV, the number of channels should still be large. To reduce the number of channels, in [4], the ESPRIT super-resolution algorithm is adopted for an orthogonal-array system which only has 4 channels in each array. However, ESPRIT requires higher SNR and is less robust to noise as compared to conventional beamforming. Moreover, [4] does not address the association problem related to the array architecture. Specifically, the measurements of the two arrays should be associated to the same scatterers to form a joint measurement. This is not a trivial task when the number of scatterers to associate is large. For high resolution radar, each target is represented as a extended target that is composed of a large number of scatterers. Thus, there are a large volume of scatterers existing in both azimuth and elevation beams and it is challenging to match the azimuth and elevation measurement for each scatterer. In [5], a geometric matching based association algorithm is proposed to match the measurements from azimuth and elevation beams; such method constructs two sets of 3-dimensional locations of a scatterer in the Cartesian coordinate from the range and angular measurements of the two arrays, then, the matching is

determined by the Euclidean distance of the two locations. Such matching method requires that only a few scatterers existing at a same range ring, otherwise, the pairwise matching is computationally infeasible. This is not applicable for high resolution automotive radar in a dense environment. Hence, reducing the number of channels and addressing the measurement association in a dense environment with low computational complexity in the orthogonal array architecture remain an open problem.

In this work, to address the aforementioned problem, we propose a new 2-dimensional (2-D) beamforming automotive radar, which, based on a small number of reception channels achieves high resolution measurement in range, rang rate, azimuth and elevation angles in a dense environment. The proposed radar employs two orthogonally-placed, collocated ULAs with N_a and N_e channels in azimuth and elevation, respectively. Unlike the orthogonal arrays in [3] and [4] that achieve high resolution in the angular domain directly either via a large aperture or resorting to high-resolution algorithm such as ESPRIT, here, we take a two-step approach to resolve angles indirectly, which lends itself to a much smaller aperture and more robust angular measurement. The two-step approach is following. First, a coarse angular measurement is obtained by forming N_a and N_e fat beams in azimuth and elevation, respectively. Next, within each beam, a finer angular resolution is achieved by leveraging the high-resolution range-Doppler images (RDIs) generated by the frequency modulation continuous waveform (FMCW). Owning to the large-bandwidth, long-time coherent processing FMCW, each pixel of the RDIs is highly likely to represent a single scatterer from an object. Subsequently, the azimuth and elevation angle of scatterers can be precisely measured using the monopulse technique [6], which is more robust to noise as compared with the ESPRIT algorithm applied in [4].

The high resolution RDI not only helps to resolve angles, but also leads to a novel approach of measurement association, which is described in the following. To match the azimuth and elevation beams that cover the same objects, we propose a deep learning based approach. The main idea here is that the range-Doppler signature (patten) of the same object at the same time, corresponding to the two beams generated by the two arrays, should be similar. Hence, the beam matching problem is converted into an image patch matching problem in the range-Doppler domain. State-of-the-art image patch matching leverages the convolutional neural networks (CNN) [7]; specifically, the so-called "Siamese-net" [8] is employed to identify the similar or dissimilar image pattens. The Siamese-net is composed of a pair of identical, shared-weights CNN. During the training, the network is fed into pairs of matched and non-matched image patches, which contain similar and dissimilar pattens, respectively. The two CNNs convert the input image patches to feature vectors; the Euclidean distance of the two feature vectors depends on the similarity of the pattens contained in the image pairs. After training, such neuron-network can generalize to unseen pattens, which is excellent for the beam matching application. Compared to the point-to-point association approach of [5] that works in a sparse scenario or within a narrow beam, the proposed patten-based matching approach works for matching two broad beams that contain dense points.

The beam matching processing could be computationally heavy in a dense environment. In the worst case, each search frame would generate $N_a N_e P$ matching pairs, where P is the number of patches in each RDI. In order to reduce the computation and optimally utilize the radar resource of the proposed radar architecture, we introduce a radar resource management method for the proposed radar architecture. Radar resource management techniques are widely used in phased array radars with narrow beamwidth, whose FOV is covered by different beams in a time sharing manner. Such time-sharing scheme is scheduled by a resource manager. A popular scheduling algorithm implemented in real-life radar systems is the time-balanced algorithm [9]; the algorithm considers the urgency of each radar job and guarantees that the more urgent a job is, the higher priority it obtains. Based on the time-balanced algorithm, we propose a new scheduling algorithm, which adapts to the proposed radar architecture. Specifically, in addition to considering the urgency of jobs, the proposed algorithm also considers the beam positions of the tracking jobs; the tracking jobs which fall into the same beam are grouped and executed together; this achieves significant savings of radar time in dense environment. Moreover, since the tracking jobs are already associated with the corresponding azimuth and elevation beams, no beam matching computation is required; this enables computational saving. The advantage of the proposed radar is demonstrated by simulations.

II. System Model

A. Target model

We model an extended target as a collection of independent scatterers lying on a 3-D grid of a cuboid, i.e., $\mathbf{\Omega} \triangleq \{-L/2, -L/2+\Delta L, \cdots, L/2\} \times \{-W/2, -W/2+\Delta W, \cdots, W/2\} \times \{-H/2, -H/2+\Delta H, \cdots, H/2\}$, where L, W, H are the length, width and height of the target, respectively, and $\Delta L, \Delta W, \Delta H$ denotes the spacing of grid points along the length, width and height dimensions, respectively. The reflection coefficient of each scatterer is $\rho \triangleq p_s \rho_s e^{j\phi}$, where p_s, ρ_s and ϕ are random variables, which are distributed according to Bernoulli, Gaussian and Uniform distributions, respectively. Such model reflects that each scatterer generates the return with a probability of p for each look of the radar. A target whose geometric center is located at $\mathbf{x}_t = [x_t, y_t, z_t]^T \in \mathbb{R}^3$ can be expressed as

$$T(\mathbf{x} - \mathbf{x}_t) = \sum_{\mathbf{x}_0 \in \mathbf{\Omega}} \rho(\mathbf{x}_0) \delta(\mathbf{x} - \mathbf{x}_t - \mathbf{x}_0), \tag{1}$$

where $\mathbf{x} = [x, y, z]^T, \mathbf{x}_0 = [x_0, y_0, z_0]^T, \mathbf{x}, \mathbf{x}_0 \in \mathbb{R}^3$; $\delta(\cdot)$ is the unit impulse function. We also assume that the velocity of the target center is $\mathbf{v} = [v_x, v_y, v_z]^T \in \mathbb{R}^3$. Assuming that the radar location is at the origin and its velocity is 0, the radial velocity of each target scatterer with respect to the radar is

$$v(\mathbf{x}_0) = \mathbf{v}^T (\mathbf{x}_t + \mathbf{x}_0) / \|\mathbf{x}_t + \mathbf{x}_0\|, \ \mathbf{x}_0 \in \mathbf{\Omega}, \tag{2}$$

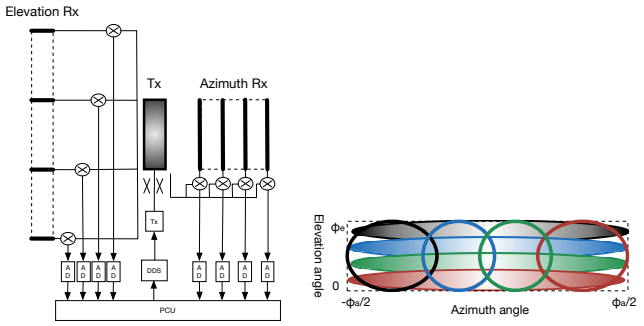
where $(\cdot)^T$ denotes for the transpose, and $\|\cdot\|$ is the l_2 norm.

The range, azimuth and elevation angle of each scatterer are calculated as follows

$$\begin{aligned} r(\mathbf{x}_0) &= \|\mathbf{x_t} + \mathbf{x_0}\|, \\ \phi_a(\mathbf{x}_0) &= \arctan(\frac{y_t + y_0}{x_t + x_0}), \\ \phi_e(\mathbf{x}_0) &= \arcsin\left((z_t + z_0)/r(\mathbf{x}_0)\right), \mathbf{x}_0 \in \mathbf{\Omega}. \end{aligned} \tag{3}$$

B. Radar architecture

The proposed radar architecture is illustrated in Fig. 1 (a). The radar contains a transmission antenna and a pair of reception antenna arrays, which are two orthogonally placed ULAs. The two arrays are named as azimuth and elevation array, and have N_a, N_e elements, respectively. The transmit antenna forms a wide beam patten, which covers the FOV of the radar, while the azimuth and elevation arrays form N_a, N_e beams in azimuth and elevation, respectively. Each reception channel is mixed with a coupled signal from the transmitter to de-chirp the received FMCW signal. The digitized received signal is processed by the processing and control unit (PCU). The arrays of the radar are to cover a large FOV in azimuth, e.g., $\pm 60°$ and a smaller FOV in elevation, e.g., $30°$. The beam patten corresponding to such design is illustrated in Fig. 1 (b).



(a) (b)

Fig. 1. The proposed radar architecture. (a) Radar architecture. (b) Beam positions. The radar forms N_a, N_e ($N_a = N_e = 4$) beams in azimuth and elevation, respective.

C. Signal model

Based on the target model and the radar architecture, for a single extended target, after demodulation, the received signal for the uth, $u \in [N_a]$ azimuth channel of the pth, $p \in [P]$ pulse can be expressed as [2]

$$s_u(t) = \sum_{\mathbf{x}_0 \in \Omega} \rho(\mathbf{x}_0) e^{j2\pi(f_r(\mathbf{x}_0)t + pTf_d(\mathbf{x}_0) + id_a \sin(\phi_a)/\lambda)}, \tag{4}$$

where $t \in (0, T)$, and T is the pulse repetition interval. $f_r(\mathbf{x}_0)$ is the frequency related to $r(\mathbf{x}_0)$; $f_d = 2v(\mathbf{x}_0)/\lambda$ is the Doppler frequency, and d_a is the element-wise spacing of the azimuth array. We use $[X]$ to denote the integer set of $\{0, \ldots, X-1\}$.

In general, when multiple targets are present, the received signal can be expressed as the superposition of the signal generated by each target and noise. The signal received from the elevation array takes a similar form. Classic signal processing extracts target parameters, i.e., range, range rate and angle by applying discrete Fourier transform (DFT) to the digitized signal in each dimension [2], however, the closely spaced scatterers are not resolvable in the angular domain. Also, how to associate the measurements of the two arrays is challenging for a large number of scatterers. In the next section, we propose the unique signal processing method to address those problems.

III. Signal and Data Processing

A. Overview

The overview of the signal and data processing of the proposed radar is shown in Fig. 2. Upon reception of the signal from the two arrays, digital beamforming is applied to form azimuth and elevation beams as shown in Fig. 1 (b). Next, the RDI is computed for each beam using a 2-D DFT. Subsequently, detection is applied on the RDIs generated from each azimuth beam. A successful detection triggers the beam matching procedure, which will be discussed in detail in Section III-B. The matched azimuth and elevation angles are subsequently measured using monopulse. Next, the 4-D measurements of targets are forwarded to a multi-object, extended target tracker [10], which proposes and maintains tracking tasks. The resource manager is adopted to schedule searching and tracking jobs; this is discussed in detail in Section III-C.

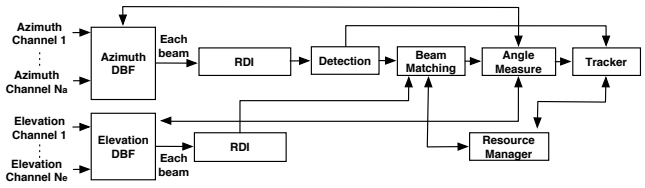


Fig. 2. Overview of signal and data processing.

B. Beam matching via deep learning

Matching the azimuth and elevation beams associated to the same targets is not trivial especially in a dense environment. In [5], the beam matching is applied on the scatterer-basis, i.e., the association is between each range-angle pairs of measurements from the azimuth and elevation arrays. This approach is not computational feasible in scenarios where the number of measurements is large.

The same target has a similar patten in the RDI of the corresponding azimuth and elevation beams, hence, it is possible to match the beams based on pattens rather than measurements of each scatterer. While similar, the range-Doppler pattens of a target in different beams are not exactly the same, due to different FOV of the azimuth and elevation beam, and different SNR of different channels. We take a deep learning based approach to address those challenges. Specifically, we adopt

CNN to match targets from different azimuth and elevation channels based on range-Doppler patches generated from different azimuth and elevation beams. We call this as Beam Matching Net (BMN). The overall structure of the BMN is shown in Fig. 3 (a).

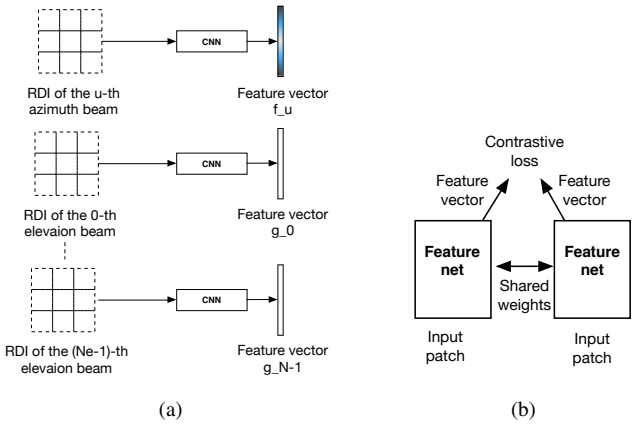


Fig. 3. The beam matching net. (a) Beam matching net. (b) Siamese net.

The goal of BMN is to match targets detected in any azimuth beam to those detected in the elevation beams. To achieve this, the RDI of the uth, $u \in [N_a]$ azimuth beam is divided into $p \times q$ patches; the size of each patch is $R/p \times D/q$, where $R \times D$ is the size of the RDI. Detection is applied to each patch. The detection can be implemented by comparing the energy of a patch to a predefined threshold. If targets are detected in the (i,k)th, $i \in [p], k \in [q]$ patch of the uth azimuth beam, such image patch is forwarded to a CNN to extract a feature vector denoted as $\mathbf{f}^u_{i,k}$. Meanwhile, the (i,k)th image patch of each elevation channel is forwarded to the same CNN, which outputs N_e feature vectors, i.e., $\mathbf{g}^v_{i,k}, v \in [N_e]$. Next, the Euclidean distance between $\mathbf{f}^u_{i,k}$ and $\mathbf{g}^v_{i,k}, v \in [N_e]$ is computed; the elevation channel corresponding to the minimum distance is the matched channel, i.e.,

$$v_m = \text{argmin}_{v \in [N_e]} = \|\mathbf{f}^u_{i,k} - \mathbf{g}^v_{i,k}\|. \tag{5}$$

The design of the CNN in BMN is based on the LeNet architecture [11], which is composed of cascade of multiple convolutional layers followed by pooling layers. Several fully connected layers are appended, which outputs a feature vector. To adapt LeNet in our application, we change the input layer as $R/p \times D/q$ and the output layer as a 64-dimensional vector.

The proposed BMN needs to identify the same and different range-Doppler pattens. In computer vision problems, such patten matching problem is often addressed by the Siamese-net (see Fig. 3 (b)) [12]. The Siamese net is composed of a pair of identical weight-sharing feature extraction CNNs (feature net). The input of the feature nets are pairs of image patches, containing identical or different pattens. In training, the contrastive loss [12] is used to enforce the feature net to output similar features for identical pattens and different features for different pattens. After training, the Siamese net can generalize to unseen pattens.

C. Resource management

The beam matching processing could be computational heavy in a dense environment. In the worst case, each frame covering the whole FOV would generate $N_a N_e P$ matching pairs, where P is number of patches in each RDI. In order to reduce the computation of the proposed radar architecture and optimally utilize the radar resource, we introduce the radar resource management to optimally schedule radar jobs in the processing.

The proposed scheduling algorithm shown in Fig. 4 is an extension of the time-balanced scheduling algorithm, which is adapted to the proposed radar architecture. Time-balanced scheduling algorithm prioritizes the radar jobs based on their urgency, i.e., the more urgent a job, the higher priority it obtains. The urgency of a radar job is characterized by the time-balance, i.e., $t_b, t_b \in \mathbb{Z}$. When a job is proposed, such job is not time-ready, and t_b is set to be a negative value; as the time elapses, t_b increases. If $t_b \geq 0$, such job is time-ready and requires to execute as soon as possible. Here, in addition to considering the urgency of jobs, the track locations are also taken into account. Specifically, the tracks that fall into the same azimuth and elevation beams are grouped together; those tracks can be updated by a single tracking job, which saves radar time as compared to updating each track individually. Moreover, since tracking jobs are already associated to specific azimuth and elevation beams, no beam matching procedure is required; this results into a significant computational saving. To save radar time and computation, it is preferred to group as many tracks as possible into the smallest number of beams. Thus, the scheduler would prefer to delay the scheduling of the time-ready tasks and gather as many time-ready tasks as possible. However, the delay of scheduling of tracking tasks would cause a lower updating rate. To save radar time and computations while guaranteeing the update rate of tracking jobs, we propose the following cost function

$$f = \alpha \sum_{i \in [J]} t_i + B/J, \tag{6}$$

where $\alpha \in (0,1)$ is the weight of the cost introduced by the time balance of J tracking jobs; the time balance of the ith job is denoted as t_i. The cost of the grouping is denoted by B/J, where B is the number of groups (beams). In the best case, all the tracking jobs can be grouped into a single group; the cost of the grouping in such scenario is $1/J$, while in the worst case, all tracking jobs fall into individual non-overlapping beams, and the cost of grouping is 1.

The scheduling algorithm shown in Fig. 4 is executed at a fixed rate; such rate is the same as the radar frame update rate, e.g. $20Hz$. For each iteration, the procedure is summarized as follows:

1) The t_b of each tracking task is increased by 1.
2) If there exists tracking tasks whose t_b is greater or equal to 0, then the resource manager asks the tracker to predict the jobs' beam locations at the execution time. Otherwise, the resource manager generates a search job.

3) Group all the tracking jobs that are requesting to execute into B beams; each beam is an intersection of a azimuth beam and a elevation beam.
4) Calculate the cost function based on (6). If the cost is greater than the threshold C, then schedule those tracking jobs. Otherwise, enter the next iteration.

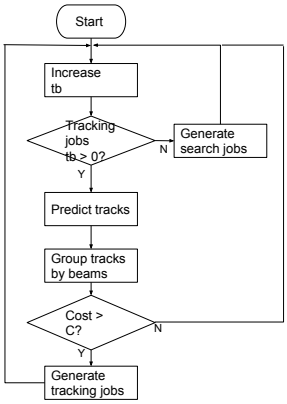


Fig. 4. The proposed resource management algorithm.

IV. Simulation

In this section, we verify the proposed radar architecture via simulations. The parameters of the simulated radar are show in Table I. The FOVs corresponding to the azimuth and elevation arrays are $\pm 60^\circ$ and $\pm 15^\circ$, respectively.

TABLE I
Radar Parameters

Parameter	Symbol	Value
Center frequency	f_c	$76GHz$
Pulse bandwidth	b_w	$1GHz$
Pulse repetition time	T	$90us$
Number of range bins	N_r	512
Number of PRI	P	256
Maximum range	R_{max}	$100m$
Number of azimuth antenna elements	N_a	4
Number of elevation antenna elements	N_e	4
Element-wise spacing of azimuth array	d_a	0.58λ
Element-wise spacing of elevation array	d_e	1.93λ

A. RDI patch matching

We generate RDI patches corresponding to targets of various shape, position and velocity at different SNR. The size of each patch is 32×32. Those patches are grouped into pairs of similar and dissimilar pattens. A pair of similar pattens corresponds to the same target of different SNR. A training sample contains a pair of patches and a label; the label is 0 and 1 for the pairs of similar and dissimilar pattens, respectively. Examples of similar and dissimilar pairs are shown in Fig. 5.

We generated 5000 pairs of training samples and divided them into a training set and an evaluation set, which contains 4000 and 1000 samples, respectively. The training set contains 7 different pattens, while the testing set contains 3 different

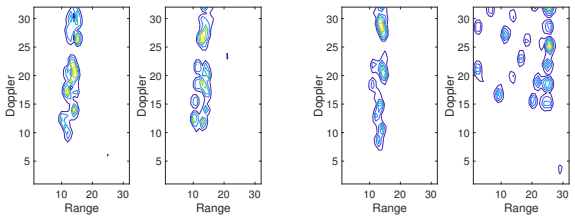


Fig. 5. Examples of similar and dissimilar range-Doppler patches. (a) Similar patch. (b) Dissimilar patch.

pattens which do not appear in the training set. We then use the TensorFlow to train and test the model. After training, we extract the 64-dimensional feature vectors from each RDI patch in the training and the test set and project them into 2-D vectors using principle component analysis. Fig. 6 shows the visualization of the clustering of the 2-D feature vectors in the training and testing set. One can see that the feature vectors of the 7 different pattens are clustered, hence the Euclidean distance within a same patten is small. Moreover, even though the pattens in the testing set do not appear in the training set, they are still clustered in the feature space, which shows the generalization of the CNN. Such simulation is for the proof of concept. To obtain a better performance in practical systems, a much larger training set containing much more different pattens is required.

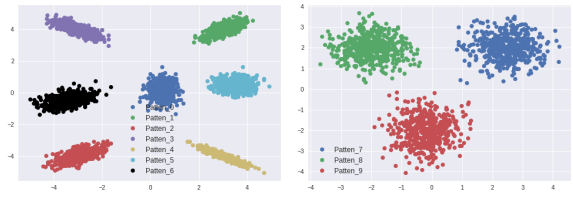


Fig. 6. Clustering of feature vectors from the train and test set. (a) Train. (b) Test.

B. Target reconstruction

We simulate two targets whose locations in the spherical coordinate, (r, ϕ_a, ϕ_e), are $(5m, 20^\circ, 5^\circ)$, $(20m, 25^\circ, 4^\circ)$, respectively, while their velocities (v_x, v_y, v_z) are $(30, 5, 1)m/s$, $(15, 5, 1)m/s$, respectively. The size of each target is $(L, W, H) = (2, 2, 1)m$ and $\Delta L = \Delta W = \Delta H = 0.5m$. The azimuth-range and elevation-range measurements after beam matching are shown in Figs. 7 (a), (c), and the positions for each scatterers in the Cartesian coordinate are shown in Figs. (b), (d). Compared with that of the elevation array, the reconstructed positions in the Cartesian coordinate from the azimuth array is closer to the ground truth. This is due to that the Doppler spreading for each scatterer is more prominent in azimuth than in elevation; as a result, the azimuth angel measurement of each scatterer is more precise than that of the elevation angle. The 3-D reconstruction is shown in Fig. 8.

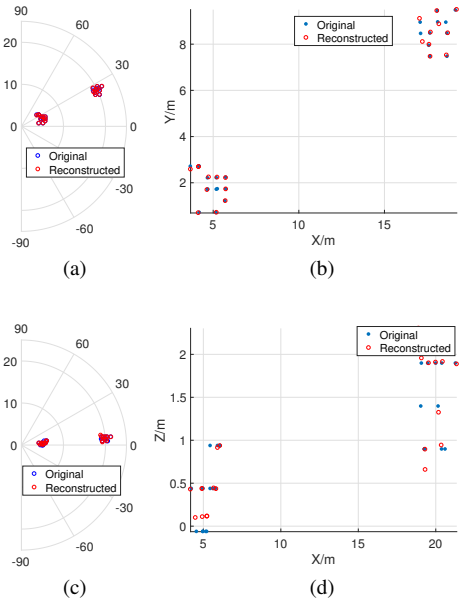


Fig. 7. Measurement and reconstruction from individual array. (a) Azimuth-range measurement from the azimuth array. (b) Reconstruction in X-Y. (c) Elevation-range measurement from the elevation array. (d) Reconstruction in X-Z.

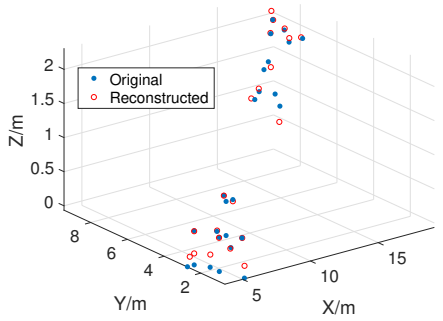


Fig. 8. Reconstruction in the X-Y-Z coordinate.

C. Resource management

In this section, we show the performance of the proposed resource management algorithm (see Fig. 4) and compare to the conventional time-balanced algorithm by simulation. In conventional time balance algorithm, the resource manager schedules one job a time. In an automotive radar scene, because the targets are much closer, and the beamwidths are wider, each search job can create many track jobs in the queue. This means that it may take a long time to clear the queue even without any new tasks come in. On the other hand, in the proposed scheduling algorithm, we propose to combine the jobs that can fall in the same beam. This way, the queue can be cleared much faster, i.e, compared to the time balance algorithm, more tracking jobs can be scheduled by the proposed algorithm in the same time frame. Fig. 9 shows that with 37 tracking jobs in queue, the proposed algorithm can finish scheduling these tracking jobs in 9 intervals compared to the 37 intervals required by the conventional time balance algorithm.

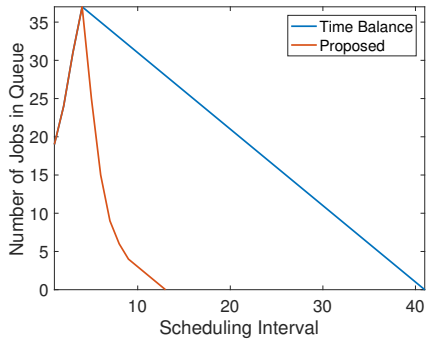


Fig. 9. Comparing of resource management algorithms.

V. Conclusion

We have proposed a new automotive radar for perception applications in self-driving scenarios. Such radar achieves high resolution measurements in range, range rate, azimuth and elevation angles of extended targets by leveraging two orthogonally-placed digital beamforming linear arrays of a few channels. The deep learning based beam matching method and a resource management algorithm have been developed for the proposed radar architecture to address the beam association and the related computational challenges.

References

[1] M. Murad, I. Bilik, M. Friesen, J. Nickolaou, J. Salinger, K. Geary, and J. Colburn, "Requirements for next generation automotive radars," in *Radar Conference (RADAR), 2013 IEEE*, pp. 1–6, IEEE, 2013.

[2] S. M. Patole, M. Torlak, D. Wang, and M. Ali, "Automotive radars: A review of signal processing techniques," *IEEE Signal Processing Magazine*, vol. 34, no. 2, pp. 22–35, 2017.

[3] F. Lalezari, "Orthogonal linear transmit receive array radar," Aug. 21 2012. US Patent 8,248,298.

[4] K. Shirakawa, S. Kobashi, Y. Kurono, M. Shono, and O. Isaji, "3d-scan millimeter-wave radar for automotive application," *Fujitsu Ten Tech. J*, vol. 38, pp. 3–7, 2013.

[5] R. E. Pavek and J. M. Willey, "Low cost 3d radar imaging and 3d association method from low count linear arrays for all weather autonomous vehicle navigation," Aug. 22 2017. US Patent 9,739,881.

[6] Y. Seliktar, *Space-time adaptive monopulse processing*. PhD thesis, School of Electrical and Computer Engineering, Georgia Institute of Technology, 1998.

[7] S. Zagoruyko and N. Komodakis, "Learning to compare image patches via convolutional neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pp. 4353–4361, 2015.

[8] R. Hadsell, S. Chopra, and Y. LeCun, "Dimensionality reduction by learning an invariant mapping," in *null*, pp. 1735–1742, IEEE, 2006.

[9] J. M. Butler, *Tracking and control in multi-function radar*. PhD thesis, University of London, 1998.

[10] K. Granstrom, M. Baum, and S. Reuter, "Extended object tracking: Introduction, overview and applications," *arXiv preprint arXiv:1604.00970*, 2016.

[11] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, "Gradient-based learning applied to document recognition," *Proceedings of the IEEE*, vol. 86, no. 11, pp. 2278–2324, 1998.

[12] J. Bromley, I. Guyon, Y. LeCun, E. Säckinger, and R. Shah, "Signature verification using a" siamese" time delay neural network," in *Advances in neural information processing systems*, pp. 737–744, 1994.