RadSimReal: Bridging The Gap Between Synthetic and Real Data In Radar Object Detection With Simulation

Oded Bialer* and Yuval Haitman*
General Motors, Technical Center Israel
oded.bialer8@gmail.com, haitman@post.bgu.ac.il

1. PSF and Noise Variance Measurement

This section explains the process of measuring the radar's Point Spread Function (PSF) and noise variance, essential components for *RadSimReal* as elaborated in Section 3 of the main paper. To obtain the PSF, a radar measurement can be conducted in a scenario featuring a narrow and stationary object, like a pole, positioned in an isolated area where there are no prominent reflecting objects nearby. The pole, characterized by narrow spread in distance, azimuth angle, and Doppler frequency, can be treated as an approximation of a point reflector. If possible, using a radar corner reflector as the isolated target is preferred. A radar corner reflector is specifically designed to exhibit an exceptionally narrow spread in all dimensions [4].

To measure the PSF, multiple radar tensors that capture the same scenario as described above are collected at various time instances when both the radar and the observed object remain stationary. These tensors are then averaged to reduce the noise in the PSF measurement. Subsequently, the truncated PSF utilized in *RadSimReal* is derived by extracting a 3D segment from the averaged radar tensor, centered around the reflection point of the narrow object. The intensity of the PSF diminishes rapidly from its central point. Each dimension of the PSF is truncated at a point where its intensity significantly falls below the radar's noise variance (the noise variance in the tensor without averaging).

Fig. 1 provides a demonstration of the PSF measurement for the radar utilized in the RADDet dataset. Fig. 1(a) presents a camera image of a scene from RADDet featuring a pole that was used for the PSF measurement. Figs. 1(b), (c), and (d) display the measured PSF slices in range, Doppler, and azimuth angle, respectively, alongside corresponding slices of a PSF obtained through conventional simulation of the radar in the RADDet dataset (as detailed in Fig. 2(a)+(b) in the main paper). The figure illustrates that the measured PSF closely resembles the simulated radar's PSF.

It is worth noting that the simulation method employed to derive the reference PSF in Figs. 1 necessitates an in-depth understanding of the specific radar hardware design and processing algorithms. This information is not always disclosed by radar suppliers. In contrast, the measurement procedure outlined above enables the acquisition of the PSF through a straightforward measurement that does not require detailed knowledge of the radar design.

Next, we proceed to elucidate how to measure the radar's noise variance, a prerequisite for *RadSimReal* as detailed in Section 3 of the main paper. The noise variance can be determined by identifying a region within the radar tensor that lacks any objects. This specific portion of the radar tensor comprises only noise, allowing the calculation of noise variance by assessing the variance of the tensor cells within this region. Fig. 2 illustrates an instance of a radar image from the RADDet dataset, with red rectangles indicating sections without reflections that can be utilized for measuring the noise variance.

2. Equivalency Between Outputs of Conventional Simulation and *RadSimReal*

In Section 3 of the main paper, we claimed that the identical output tensor from the conventional simulation (Fig. 2(a)+(b) in main paper) could be achieved by convolving the radar's Point Spread Function (PSF) with the 3D reflection points in the scene (Fig. 2(a)+(c) in main paper). This section provides an explanation for the validity of this equivalence. We first provide an intuitive understanding of this equivalence through a straightforward example in Section 2.1. Subsequently, in Section 2.2, we present a formal mathematical derivation of this equivalence.

2.1. Intuitive Explanation of Simulations equivalence

We explain the equivalence between the conventional simulation and *RadSimReal* through a simplified example of a

[*] Both authors contributed equally to this work.
Both authors are with General Motors, Yuval Haitman is also with the School of Electrical and Computer Engineering in Ben Gurion University of the Negev.

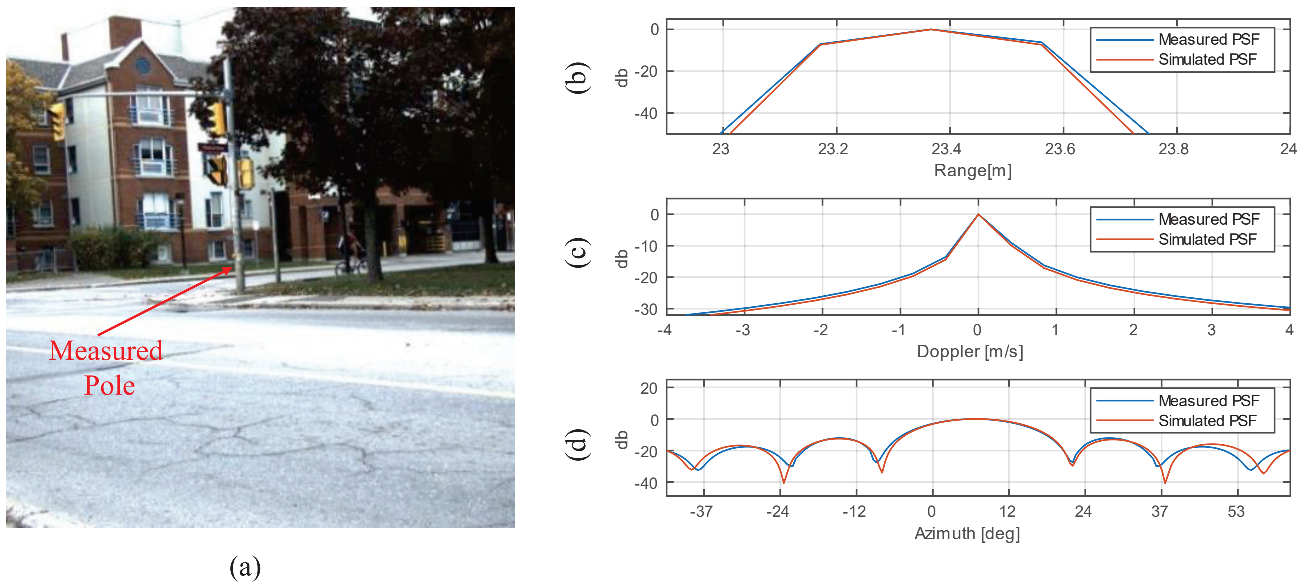

Figure 1. Measurement of the radar's PSF using a pole. (a) Image depicting the scenario and the employed pole. PSF slices in range, Doppler, and azimuth angle of the measured PSF compared with the PSF obtained through conventional radar simulation in (b), (c), and (d), respectively.

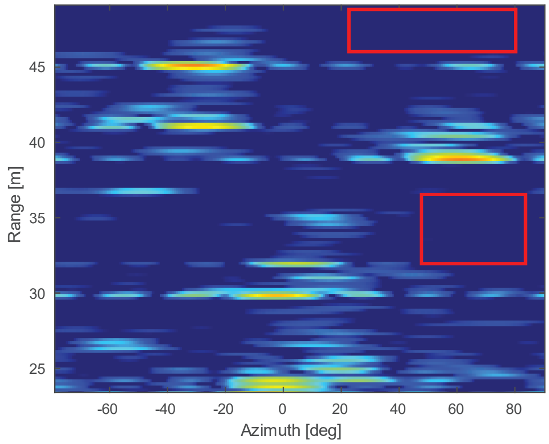

Figure 2. Measurement of radar noise variance from radar image in the RADDet dataset. Red rectangles in the radar image mark reflection-free regions utilized for noise variance measurement.

radar signal. The conventional radar simulation of this example is depicted in Fig. 3. Fig. 3(a) illustrates a transmitted radar pulse signal at time zero. In Fig. 3(b), the received signal is illustrated for a scenario involving two reflection points. It is evident that the received signal is essentially the transmitted signal but delayed by τ_1 and τ_2. These delays represent the times taken for the signal to travel from the radar to the first and second reflection points and back. Importantly, these time delays are proportional to the distances of the first and second reflection points denoted by d_1 and d_2, respectively. To obtain the received energy for each delay (distance) hypothesis, the radar employs a match filter on the received signal [3, 5]. The match filter operation is a correlation between the received signal and the transmitted signal. Fig. 3(c) displays the result of the match filter in this example. The signal propagation times, τ_1 and τ_2, are proportional to the reflection points distances. Hence the time scale in Fig. 3(c) can be converted to distance. The corresponding match filter output as a function of distance is depicted in Fig. 3(d).

Fig. 4 illustrates the simulation conducted by *RadSimReal* for the same example depicted in Fig. 3. In Fig. 4(a), Kronecker delta functions at the distances d_1 and d_2 of the two reflection points are presented. Fig. 4(b) illustrates the radar's PSF, which is the auto-correlation of the transmitted signal. Fig. 4(c) displays the output of the convolution between the delta functions in Fig. 4(a) and the PSF in Fig. 4(b). It is evident that the match filter output in Fig. 3(d) is identical to the result in Fig. 4(c). Thus, the output of the conventional simulation can be achieved through convolution between the radar's PSF and the reflection points, represented as delta functions at the distances of the reflection points. This approach is the simulation methodology employed by *RadSimReal*.

While the representation in Fig. 3 and 4 show a one-dimensional match filter applied to a simplified single transmitted radar pulse, practical automotive radars involve extending the transmitted signal and match filter across a sequence of pulses and multiple antennas. Consequently, the result is a multidimensional output tensor with dimensions

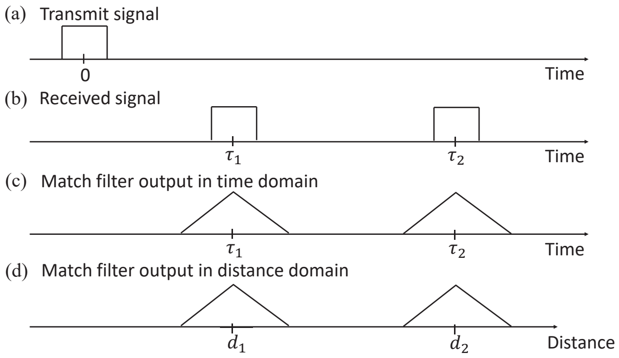

Figure 3. Illustration of the conventional simulation processes with a simplified Example. (a) Transmitted pulse. (b) Received signal from two reflections at delays τ_1, τ_2. (c) Output of receiver processing obtained by correlating the received signal with the transmitted signal (match filter). (d) Match filter output converted from time to distance scale.

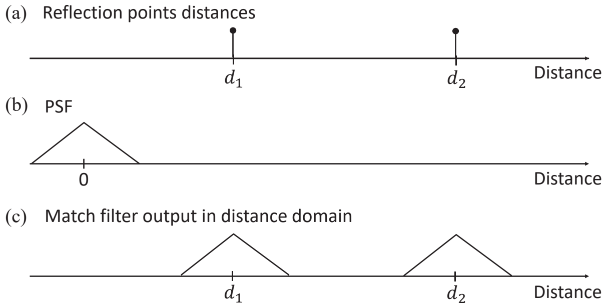
(a) Reflection points distances
d_1
d_2
Distance
(b) PSF
0
Distance
(c) Match filter output in distance domain
d_1
d_2
Distance

Figure 4. Illustration of the *RadSimReal* processes achieving an identical output as the conventional simulation in the simplified example shown in Fig. 3. (a) Representation of two reflection points using Kronecker delta functions centered at reflection points' distances d_1 and d_2. (b) The radar's PSF as a function of distance. (c) The result of convolution between the reflection points in (a) and the PSF in (b). The output in (c) is identical to the output of the conventional simulation shown in Fig. 3(d).

of range (distance), Doppler, and angle, rather than a single-dimensional output. This tensor is the one discussed in Section 3 of the main paper. Nevertheless, the processes in each dimension can be separable. Therefore, the fundamental principle remains unchanged: the output radar tensor can be calculated by convolving a 3D PSF with dimensions for range, Doppler, and angle with delta functions in the 3D space having the same dimensions. The delta functions are positioned in this space at the reflection points' range, Doppler, and angle. A mathematical derivation of this equivalence is detailed in Section 2.2.

2.2. Mathematical Derivation of the Simulations equivalence

The radar emits a periodic sequence of short signals through multiple antennas, and this signal can be characterized in three dimensions as $s(n, m, q)$, where the indices n, m, and q correspond to distinct time scales. These time scales represent the time samples of the short signal, the time samples of the short signal periods within the repetition sequence, and the signal duration along the antenna array, respectively.

The transmitted signal reflects off objects in the environment and returns to the radar with delays in each of the time scales, which are proportional to the reflection position and speed. Let τ_d^i, τ_f^i, and τ_θ^i denote the i^{th} reflection point's delays in the short signal duration, the delay in the period between the short signals, and the delay between the antennas. These delays correspond to the reflection point's distance (range), Doppler frequency (radial velocity), and angle, respectively. The received signal is an aggregation of the received signals from individual reflection points, each with its corresponding reflection intensity. The received signal samples along each of the three time dimensions can be expressed as follows:

$$r(n, m, q) = \sum_i \alpha^i s(n - \tau_d^i, m - \tau_f^i, q - \tau_\theta^i), \qquad (1)$$

where α^i represents the intensity of the i^{th} reflection point.

The radar tensor, which represents the received energy in each distance, Doppler, and angle, is obtained by applying a match filter to the received signal [3, 5]. The match filter is a 3D correlation between the received signal and the transmitted signal $s(n, m, q)$, in all three delays dimensions, which are proportional to the distance, Doppler, and angle. This correlation is expressed by:

$$y(n, m, q) = \sum_{c,u,k} r(n-k, m-u, q-c) s(k, u, c) =$$
$$\sum_{i,c,u,k} \alpha_i s(n - k - \tau_d^i, m - u - \tau_f^i, q - c - \tau_\theta^i) s(k, u, c). \qquad (2)$$

The radar's PSF is obtained by the auto-correlation of the transmitted signal given by

$$x(n, m, q) \triangleq \sum_{c,u,k} s(n-k, m-u, q-c) s(k, u, c). \qquad (3)$$

Then by substituting (3) into (2) we obtained that the match filter output can be expressed by

$$y(n, m, q) = \sum_i \alpha_i x(n - \tau_d^i, m - \tau_f^i, q - \tau_\theta^i). \qquad (4)$$

From (4), it becomes evident that the radar tensor is a superposition of the radar's PSF shifted by the delays' of the reflection points and scaled by their intensities. This relationship can equivalently be expressed as:

$$y(n,m,q) = x(n,m,q) * \sum_i \alpha_i \delta(n-\tau_d^i, m-\tau_f^i, q-\tau_\theta^i), \tag{5}$$

where the symbol $*$ denotes a convolution operation, and $\delta(n-\tau_d^i, m-\tau_f^i, q-\tau_\theta^i)$ is a 3D Kronecker delta function. This function takes a value of 1 when $n = \tau_d^i$, $m = \tau_f^i$, and $q = \tau_\theta^i$, and is zero elsewhere. Therefore, rather than obtaining the radar tensor through match filtering, as shown in (2), it can equivalently be obtained by convolving the radar's PSF, $x(n,m,q)$, with reflection points that are represented by 3D Kronecker delta functions that are shifted by the 3D delays' $(\tau_d^i, \tau_f^i, \tau_\theta^i)$ of the reflection points and scaled by their intensities (α_i), as expressed in (5). The 3D delays are directly proportional to the range, Doppler frequency, and angle of the reflections. Consequently, the dimensions of the radar tensor are eventually transformed from delays to range, Doppler, and angle. This equivalent approach is the methodology utilized in deriving the tensor in *RadSimReal*.

3. Simulation Computation Complexity

In this section, we assess the computational complexity of conventional radar simulation in comparison to *RadSimReal*. Both simulations initiate with the shared step of generating reflection points in the environment (as depicted in Fig. 2(a) in the main paper). The runtime of this phase depends on the graphics simulation engine and can be very fast, even in real-time. The significant run-time difference lies in transforming reflection points into radar images (parts (b) and (c) of Fig. 2 in the main paper), which are evaluated next.

The initial phase of the conventional simulation involves the generation of received samples per radar image frame. These samples result from aggregating the received signals from individual reflection points, leading to a computational complexity of $O(N_pN_r)$, where N_r represents the number of received samples per radar image, and N_p denotes the number of reflection points in the scenario. The number of received samples per frame, N_r, is directly proportional to the number of cells in the radar tensor, denoted as N_s. Consequently, the complexity of the first part of the conventional simulation can be expressed as $O(N_pN_s)$.

In the subsequent stage of the conventional radar simulation, signal processing algorithms are applied to the received signal to generate the radar tensor. These algorithms coherently combine received signal samples for each range, angle, and Doppler cell in the radar tensor. This process, called match filtering, is efficiently executed through a series of Fast Fourier Transform (FFT) operations in range, Doppler, and angle, resulting in an overall complexity that is lower bounded by $O(N_s \log(N_s))$.

The total complexity of the conventional simulation is derived by summing the complexities of the two aforementioned parts. This yields a complexity of $O(N_s(N_p + \log(N_s)))$. Since N_p is significantly larger than $\log(N_s)$, the complexity of the conventional simulation approach simplifies to $O(N_sN_p)$.

Moving on, we proceed to compute the computational complexity of *RadSimReal*, which involves performing convolution between the reflection points and the radar's PSF. The reflection points are sparsely distributed within the radar tensor, i.e., $N_s \gg N_p$. Consequently, the convolution between the reflection points and the PSF can be carried out as a sparse convolution. This operation entails aggregating the PSFs of reflection points, resulting in a complexity of $O(N_pN_f)$, where N_f denotes the number of cells in the radar's PSF.

Therefore, the complexity ratio between conventional simulations and *RadSimReal* is expressed as $O((N_sN_p)/(N_pN_f)) = O(N_s/N_f)$, which represents the proportion of the entire radar tensor volume to the PSF volume. As detailed in Section 3 of the main paper, our simulation truncates the PSF to preserve 99% of its energy, significantly reducing its volume. Consequently, this leads to a substantial reduction in complexity, exemplified by a ratio of 1250 for the radar utilized in the RADDet, CARRADA, and CRUW datasets.

We tested the average run time for generating radar images using the conventional physical radar simulation and our simulation. These tests were conducted for the TI radar employed in the RADDet [1, 2]. We implemented the simulations in Matlab 2020a, making use of the parallel processing toolbox. The computations were executed on a computer equipped with an Intel(R) Xeon(R) W-2235 CPU operating at 3.80GHz, alongside an Nvidia Quadro RTX 5000 GPU with 16GB of memory. The results reveal that the average run time for generating a radar images from reflection points with our simulation is 0.0105 second, and with the conventional physical simulation it takes 5.296 seconds. *RadSimReal* generates images about 500 times faster than the conventional simulation, which is on the order of the factor 1250 that was obtained from the analysis above.

4. RADDet Train-Test Set Split

The performance evaluation conducted in Section 4 of the main paper utilized a train-test set split of the RADDet dataset that differs from the split proposed in [6]. Table 1 displays the Average Precision (AP) results at IOU 0.1, 0.3, and 0.5 for the three object detection methods employed in the paper ('RADDet', 'Probabilistic', and 'U-Net'). The comparison is made between the original RADDet train-test split and our suggested split. The AP results are assessed on the test set within each respective split. The results re-

Table 1. Object detection average precision (AP) on RADDet with original train-test split vs. our split

	AP Original Split			AP Our Split		
Method	@ 0.1	@ 0.3	@ 0.5	@ 0.1	@ 0.3	@ 0.5
RADDet	93.82	88.03	68.71	83.69	72.96	47.95
Probabilistic	92.87	86.36	66.30	83.31	74.80	40.68
U-Net	94.68	89.59	71.00	84.76	83.01	55.53

Table 2. Instances in the RADDet dataset divided into distinct scenes.

Scene ID	Frame Numbers
0	$0-439, 559-724, 1549-1971$
1	$440-555, 731-1548, 1972-2571$
2	$2572-3038$
3	$3039-3437$
4	$3438-3653$
5	$3654-4073$
6	$4074-4331$
7	$4332-5018, 5623-6243$
8	$5019-5622, 6244-6608$
9	$6609-8046$
10	$8047-8634$
11	$8635-9158$
12	$9159-9437$
13	$9438-9745, 10175-10292$
14	$9746-10174$

veal that all methods achieved significantly higher AP results with the original RADDet split compared to our split. This discrepancy in results can be attributed to the fact that the test and training images in the original split [6] were derived from the same scenarios, often with small temporal gaps. Consequently, a strong correlation is established between the test and training samples, leading to overfitting of all methods on the test set.

To address the issue of overfitting, we implemented a train-test set partitioning strategy that ensures distinct scenarios between the training and testing sets. The RADDet dataset comprises 15 unique scenes, each detailed in Table 2. In our partitioning scheme, scenes 9 and 11 were designated for the test set, while the remaining scenes were used for the training set. The adjusted training set comprises a total of 17,021 cars compared to 16,755 in the original split. For the test set, we have 4,094 cars compared to 4,135 in the original split.

References

[1] AWR1843 data sheet, product information and support, https://www.ti.com/product/awr1843, . 4

[2] TI mmwave-sdk software development kit (SDK), https://www.ti.com/tool/mmwave-sdk, . 4

[3] Charles Cook. *Radar signals: An introduction to theory and application*. Elsevier, 2012. 2, 3

[4] Eugene F Knott, John F Schaeffer, and Michael T Tulley. *Radar cross section*. SciTech Publishing, 2004. 1

[5] George Turin. An introduction to matched filters. *IRE transactions on Information theory*, 6(3):311–329, 1960. 2, 3

[6] Ao Zhang, Farzan Erlik Nowruzi, and Robert Laganiere. Raddet: Range-azimuth-doppler based radar object detection for dynamic road users. In *2021 18th Conference on Robots and Vision (CRV)*, pages 95–102. IEEE, 2021. 4, 5