A NOVEL APPROACH OF RADAR NOISE REMOVAL VIA SENSOR FUSION AND NEURAL NETWORKS

By

Ruihao Yao

Senior Thesis in Computer Engineering

University of Illinois at Urbana-Champaign

Advisor: Sanjay Patel

May 2020

Abstract

Millimeter-Wave (mmWave) radar sensors are getting more popular for their increasing sensing capabilities especially in automotive fields. However, no matter how ideal the radar signal processing stack is, radar still suffers from internal and external noises which degrades its potential sensing solutions. There are certainly many traditional approaches to reduce radar noises like matched filter, but this paper will present a novel approach of noise removal via sensor fusion and artificial neural networks.

Subject Keywords: Millimeter-Wave; Imaging Radar; Noise Removal; Kalman Filter; Sensor Fusion

Contents

1. Introduction

The increasing popularity gained by radar in recent years is driven in part by the growth of automotive applications that use radar sensing. Under extreme weather conditions like heavy snow and rain, radar shows significant advantages against other sensors like LIDAR and cameras. However, noise is a big factor that impairs the sensing result of a radar and could cause serious problems. There are different sources of noise and the two most significant ones are sensor noises and multipath reflections. The sensor noise is going to cause inaccuracies in measurement especially in the azimuth and elevation dimension and multipath reflections are going to cause ghost targets which are phantoms of actual targets. Ghosts are serious problems in radar detection and that will lead to radar false positives and mis-classifications which will seriously inhibit the detection accuracies of radar.

As we see, since ghosts could cause serious problems in radar detections, removing ghosts should be solved. In this paper, I will present a novel approach which utilizes sensor fusion and machine learning. The classification results are based on the outputs of Extended Kalman Filter which are individual trackers. On top of that, an Intel depth camera is used to label the trackers as either actual detection or ghost. Then an LSTM is used to learn the moving patterns of ghosts and actual detections. Finally the radar is able to only use the trained weights to classify whether a tracker is an actual detection or a ghost. The results look promising and are presented in Section 4.

2. Matched Filter: Traditional Way of Noise Reduction

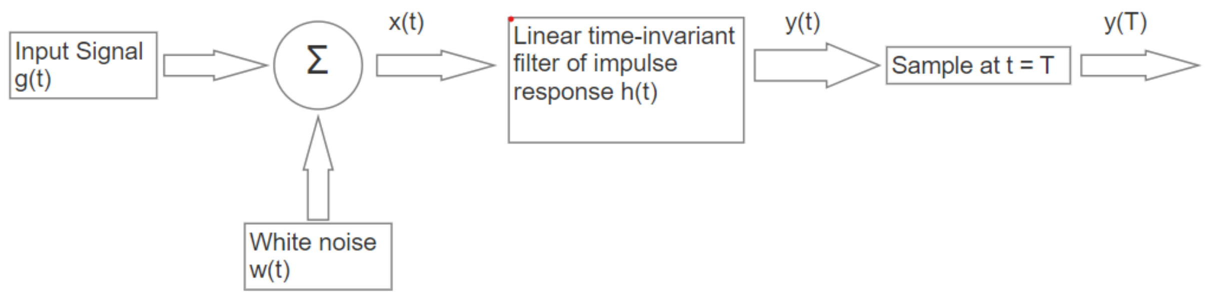

Figure 1. 1D model of Matched Filter.

One of the most common state-of-the-art methods for radar noise removal is called matched filter. The architecture of Matched Filter is represented in Figure1. It is an optimal linear filter designed to maximize the signal SNR in the presence of additive stochastic noise. The input signal, denoted as x(t), mixed by a pulse signal g(t) and additive noise w(t), as denoted in equation 2.1.

$$x(t) = g(t) + w(t) \qquad 2.1$$

In this scenario, it is safe to assume w(t) is white noise of zero mean and power spectral density No/2. To achieve the goal of matched filter, the filter needs to minimize the effect of noise at the output of filter. Because the filter is linear time-invariant, we have filter output y(t) expressed as two components, as denoted in equation 2.2.

$$y(t) = g_0(t) + n(t) \qquad 2.2$$

For Gaussian white noise, we will have the frequency response of the filter represented by equation 2.3.

$$H(f) = \frac{2K}{N_0} S^*(f) e^{-jwt_0} \qquad 2.3$$

In the specific case of white noise, we could rewrite the impulse response as equation 2.4.

$$h(t) = \ Cs(t_0 - t) \qquad 2.4$$

In equation 2.4, C is an arbitrary real positive number, t0 is the time of peak signal output and s(t) is input signal waveshape.

$$h(t) = F^{-1}[H(f)]$$

$$= \frac{2K}{N_0}\left[\int_{-\infty}^{+\infty} S(f)e^{j2\pi f(t_0 - t)}\, df\right]^{*} \qquad 2.5$$

$$= \frac{2K}{N_0}[s(t_o - t)]^{*}$$

From the above sketch of proofs, we can easily see that the impulse response is the time-reversal of the input signal wavelet such that when we convolve the input signal and filter impulse response we will effectively get a peak impulse at t0. And since this filter is applied to radar which is a digital system, the output of the system is essentially bit 1 and bit 0. Then we will append a decision device at the end of Figure 1 to compare the sampling values of the output of matched filter with a threshold set in advance. To measure the bit error rate, we just have to measure two kinds of errors, namely ,that when 1 is interpreted by the system while bit 0 was actually sent and bit 0 is interpreted by the system while bit 1 was actually sent.

3. Approach Architecture

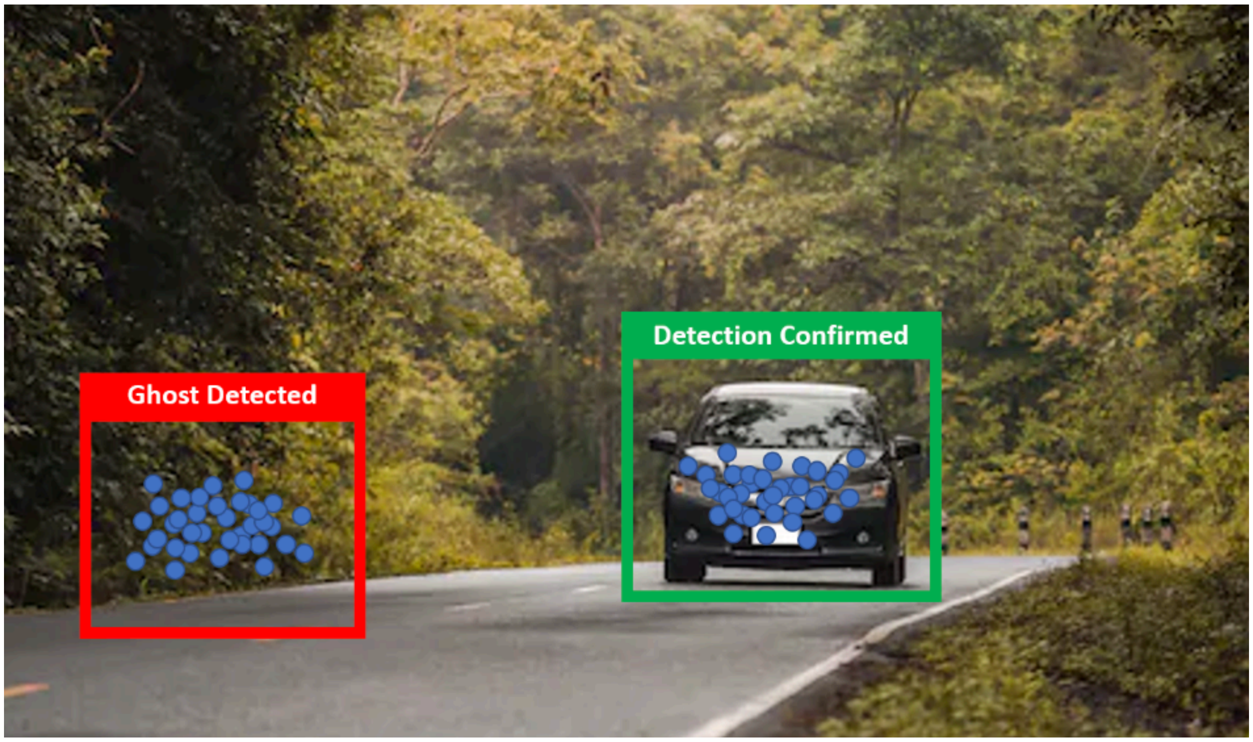

Figure 2. Ideal Output of Noised Removed Point Cloud

A good way to visualize the noise level of radar output is through the point cloud, which is a tuple consisting of [range, doppler, azimuth, elevation, SNR]. However, in TI AWR1642 radar series, elevation is not supported but we still want to keep elevation in because more advanced radars will certainly take elevation into account. An advantage of point cloud data structure is that it could be converted to Cartesian coordinates and then plotted in 2D.

$$x = range * sin(azimuth) \tag{3.1}$$

$$y = range * cos(azimuth) \tag{3.2}$$

However, point cloud data in many cases doesn't make sense either because with the effect of heavily cluttered environment and multipath propagation, it's very hard to identify which part of the point cloud represents actual target and which part of represents a ghost. Any kind of clustering algorithms don't work well because the results of clustering depend largely on the algorithm input parameters and input data. And more importantly, it's hard to establish temporal relations among frames because clustering algorithms don't have past memories. Extended Kalman Filter is built exclusively for tracking and has unique track ID for each track among all frames. However, the problem is that since point cloud is noisy, EKF is likely to produce ghost tracks(false positives). It's hard to find an universal set of EKF parameters that only detect actual targets because this will make the EKF too conservative so that false negatives will happen. False negatives are bad because in this case, EKF is not tracking what it's supposed to track. The following architecture will be introduced to effectively identify those false positives.

3.1 OpenRadar Basic Digital Signal Processing

At the very beginning of the pipeline, I applied OpenRadar(https://github.com/PreSenseRadar/OpenRadar) which is developed by a team in University of Illinois called PreSense(https://www.presenseradar.com) as part of the Alchemy(https://www.alchemyfoundry.com) project in IBM C3SR research (https://www.c3sr.com) . OpenRadar serves to poll raw ADC data out of TI AWR1642 series radar, perform range, doppler and azimuth processing and produce point cloud [range, doppler, azimuth, elevation, SNR] for each frame. The DSP details won't be listed here and are all available on OpenRadar Github repository.

3.2 Extended Kalman Filter

Extended Kalman Filter is another big feature that adds on top of OpenRadar that serves to make sense of point cloud data. It's a tracking algorithm that works like Hidden Markov Model behavior-wise. It consists of following several stages.

- Prediction:

This stage utilizes physical law of motion to perform predictions over last updated filter state. There are mainly 2 models of EKF in this scenario, constant velocity model and constant acceleration model. The prediction is done over matrix multiplication and updated results are called predicted state and will be used in later stages.

- Allocation:

At this stage, EKF utilizes clustering algorithms to analyze point clouds to determine which part of the point cloud radar data should be used for the tracking purposes. This part serves to perform an initial filtering that filters out some obvious noises.

- Association:

This stage is the most troublesome one in the entire EKF. It serves to associate the previously allocated point clouds to each active trackers. Each point could only be associated once and a lot of research is still ongoing about the data association. In general, an example of naive data association algorithm is Nearest Neighbor. However, some advanced data association algorithms take covariance matrix and group dispersion matrix into account.

- Track Management:

This stage serves like a finite state machine that manages active trackers in the current frame. It manages HIT and MISS events separately and determine whether to promote the current tracker status, degrade the current tracker status or defer the decision for further consideration.

- Update:

This stage is the final stage of the pipeline. It takes kalman gain into account and balance between the current measurements and previous predictions. If we trust the previous predictions more, then we risk accumulating the errors before we realize that they are already too big. If we trust measurement more, then we risk from the sensor noises. Therefore, we want to balance between the two.

3.3 Intel 3D Depth Camera labelling

Using 3D camera's depth information to help labelling the EKF output will be a good choice since it'll quite difficult to align radar sensor and depth camera together to make sure they always look at the same region. There will always be some drifts and we need complicated trigonometries to align them. Rather than doing all of this, we could just place radar and camera on top of each other and use depth information for labelling purposes since depth information and radar range information align pretty well. Thus, we get labelled training data.

3.4 LSTM for Training the Model

Among the models for training, I used LSTM network. There are 2 reasons for this. First, since what we want the network to learn are the different trajectories of actual targets and ghosts over time. Thus, establishing temporal relationships between consecutive frames is crucial. CNN is not specialized for this kind of learning. Secondly, the number of tracks in one single frame is unpredictable and so we may need multiple output labels in one frame. CNN is also not specialized for this kind of training task. The original data shape of EKF output is (number of frames, number of features). To get it into correct LSTM shape, I transformed the data shape into (number of frames - number of lookbacks, number of lookbacks, number of features).

3.5 Evaluation and Prediction

At evaluation and prediction stage, I also used LSTM since it is used in the training stage. Since EKF keeps track of all of its active tracks within its internal data structure, whenever EKF spawns a new active track with a new track ID, I initialized a new LSTM with trained model to predict and deleted those LSTMs which correspond to a terminated track.

4. Result Analysis

4.1 Neural Network Testing Phase Analysis

The training environment was set up in the Alchemy Lab which is located at the 3rd floor of ECEB. The devices include Intel depth camera mounted with TI AWR1642 radar. The training environment is rather noisy with a lot of ghosts appearing on radar side. The resulting testing accuracy is given below as table 1 with testing environment ordered by Alchemy lab, undergraduate lounge, classroom and ECEB atrium.

Table 1. Accuracies under 4 test environments

Testing Environment	Alchemy lab	undergraduate lounge	classroom	ECEB atrium
Accuracy(%)	98.15	88.34	88.12	94.88

4.2 Performance Analysis between Matched Filter and Neural Network Approach

There's a fundamental difference between matched filter and neural network approach which is the level they operate on. Matched filter works at the level when radar receives a signal which is even before the raw ADC data processing stack. Operating at this level often requires access to hardware like antenna and may cause some difficulties for commercialized products because those products usually don't provide interfaces for users to reach the hardware. But if we could access the hardware, then we will convolve the original signal with filter's impulse response so that the raw ADC data is optimized with maximal SNR. Then we can apply OpenRadar DSP stack on top of that to get noise-free EKF results. The neural network & sensor fusion approach operates at the level where all radar data has been processed into standard radar data format. It doesn't require any kind of access to the hardware. All of the tunable parameters lie in OpenRadar DSP stack and neural network itself. Users could tune the neural network parameters to achieve best performances of different use cases they define.

5. Case Studies

I would like to propose multiple situations where this approach of noise identification & removal would help a lot.

5.1 Autonomous Cars

When the radar is placed on the car as part of the autonomous driving components, tracking algorithm is inevitable as raw point cloud data exhibits a high probability of false alarm due to sensor measurement noises and multipath reflection. The noise identification pipeline showed in this paper will help the autonomous driving system filter only the tracks that correspond to the actual targets with a tolerable possibility of false alarm. This way, the autonomous driving system won't be too sensitive.

5.2 Indoor Tracking

Indoor environment typically generates a lot of noises on radar side because the indoor building surface is not very smooth that it causes a lot of multipath propagation and corner reflection. The current indoor positioning system typically uses WiFi and bluetooth technology. However, since they also rely on EM waves, a lot of problems like channel fading and multipath propagation will happen. The system introduced in this paper could greatly help improve the accuracy of the currently existing indoor positioning technologies.

6. Conclusion

The noise identification method introduced in this paper is a novel one that doesn't leverage on complicated method of reconstructing the original signal. Instead, it replies on the key difference between trajectories of ghost targets and actual targets. If trained well enough, the neural network is able to filter out some of the false positives of EKF output. There could still be improvements on top of this method such as mixing training data from different environments, improving the architecture of the neural network.

References

[1] Michael Goossens, Frank Mittelbach, and Alexander Samarin. *The LaTEX Companion*. Addison-Wesley, Reading, Massachussets, 1993.

[2] Arjun Gupta, Dashiell Kosaka, Edwin Pan, Jingning Tang, Ruihao Yao and Sanjay Patel. OpenRadar: A Toolkit for Prototyping mmWave Radar Applications, 2019; arXiv: 1912.12395.

[3] Arjun Gupta, Dashiell Kosaka, Edwin Pan, Jingning Tang, Ruihao Yao and Sanjay Patel. *OpenRadar*. https://github.com/presense/openradar

[4] John C. Bancroft *Introduction to Matched Filters.* https://crewes.org/ForOurSponsors/ResearchReports/2002/2002-46.pdf

[5] Ralf C. Staudemeyer and Eric Rothstein Morris. Understanding LSTM -- a tutorial into Long Short-Term Memory Recurrent Neural Networks, 2019; arXiv:1909.09586.