Robust Target Detection Methods:

Performance Analysis and Experimental Validation

by

Huiwen Chu

A Thesis Presented in Partial Fulfillment
of the Requirements for the Degree
Master of Science

Approved October 2020 by the
Graduate Supervisory Committee:

Daniel W. Bliss, Chair
Antonia Papandreou-Suppappola
Ahmed Alkhateeb

ARIZONA STATE UNIVERSITY

December 2020

ABSTRACT

Constant false alarm rate is one of the essential algorithms in a RADAR detection system. It allows the RADAR system to dynamically set thresholds based on the data power level to distinguish targets with interfering noise and clutters.

To have a better acknowledgment of constant false alarm rate approaches performance, three clutter models, Gamma, Weibull, and Log-normal, have been introduced to evaluate the detection's capability of each constant false alarm rate algorithm.

The order statistical constant false alarm rate approach outperforms other conventional constant false alarm rate methods, especially in clutter evolved environments. However, this method requires high power consumption due to repeat sorting.

In the automotive RADAR system, the computational complexity of algorithms is essential because this system is in real-time. Therefore, the algorithms must be fast and efficient to ensure low power consumption and processing time.

The reduced computational complexity implementations of cell-averaging and order statistic constant false alarm rate were explored. Their big O and processing time has been reduced.

DEDICATION

To my family and friends

ACKNOWLEDGMENTS

I would like to express my sincere gratitude and appreciation to Dr. Daniel W. Bliss for his professional advice and guidance. I am grateful to Prof.Bliss for giving me this opportunity to work on this project. He has always been patient when I encounter problems and guide me to solve it. The two years of graduate research with Prof.Bliss are meaningful and colorful.

I would also like to extend my sincere thanks to Prof.Antonia Papandreou-Suppappola and Prof.Ahmed Alkhateeb for being my committee members. Their professional advice guide me have more in-depth and careful thinking.

I am also grateful to Sharanya Srinivas and my colleagues for their supports and encouragements. Guide me when I'm lost and frustrated.

And thanks, most sincerely, to my mom and dad, Yuan Xiang and Pengying Chu for their love and endless tolerance and encouragement. They always affirm every choice I made and believe in me.

TABLE OF CONTENTS

Page

CHAPTER Page

LIST OF TABLES

LIST OF FIGURES

Figure Page

Chapter 1

INTRODUCTION

1.1 Background

Suppose in some statistical tests, we have two simple hypotheses, the null hypothesis and the alternative hypothesis. We can apply the Neyman-Person test and the likelihood ratio test approach to provide decision rules for deciding hypothesis. The distributions of data must be fully specified. However, in most statistical problems, the data models do not perfectly match with any distributions; also, it keeps changing over time. In order to provide accurate and reliable detection, the algorithm must dynamically and intelligently adjust the threshold level.

The constant false alarm rate(CFAR) test can provide great performances with the unknown distribution data. It applies a moving window across the data to select reference cells then calculates the threshold. This approach checks the neighbors' power to estimate noise level, and the threshold is set to limit the false alarm rate. The cell-averaging CFAR, smallest-of CFAR, and greatest-of CFAR are the most classical CFAR approaches. These methods are good at operating in the homogeneous environment. For a more complicated environment or clutter evolved background, we need to investigate more sophisticated CFAR approaches. Also, there are some combined CFAR methods like OSCA-CFAR[1], SOCA-CFAR[2], and GOCA-CFAR[2], they perform with low processing power. The performance of OSCA-CFAR in one of directions(applied CA-CFAR) will be damaged when homogeneous assumptions are violated. GOCA-CFAR has a strong masking effect problem and SOCA-CFAR easily detects noise peak as target which leads to high false alarm rate.

1.2 Motivation

In the automotive radar system, we need to have some statistical detection approaches that are able to dynamically set thresholds to identify targets against the noise background. Complicated traffic conditions will lead to data returned by radar with a lot unwanted noise. These approaches also need to maintain high performance when the environments evolved with clutters and multiple targets present. For example, in some radar applications like [3–5], there has multiple targets show up with noisy background, target detection becomes a limiting factor when the density of targets is large and there is no clear distinction between the targets and clutter. Under this circumstance, we wish to adjust the threshold level to identify targets and avoid clutter affects in constant false alarm rate algorithms.

For radar data in range-Doppler domain, we need to make sure the algorithms' performance for both directions are not deteriorates. High computations means high power consumption and more processing time. For the real-time detection system, we wish to reduce the algorithm's computational complexity. Toward this, we are investigating some efficient implementations of conventional CFAR approaches to achieve this goal. In this study, the computational complexity of algorithms is evaluated with Big O notation and central processing unit(CPU) time.

1.3 Contribution

The goal of this study is to investigate the performance of cell-averaging CFAR, and order-statistic CFAR, and how they operate in different environments, homogeneous, multiple targets, clutter evolved environment, and experimental data. First, we introduce the procedures and implementations of cell-averaging CFAR and order-statistic CFAR.

To check the performance in a higher dimension, for example, the experimental data in the range-Doppler domain, the two-dimensional CFAR data structure is also being introduced. Some classical clutter distribution models, Gamma, Weibull, and Log-normal, are added into the testing background. Here, we calculated the probability of detection and probability of false alarm rate of detection results and plotted these parameters with receiver operating curves to compare the performances.

Consider the computational complexity of these two CFAR approaches; we calculated the big O of each process. We also investigated the efficient implementations of these two algorithms. The comparison of efficient methods with conventional CFAR algorithms is given.

We tested the performance of CFAR approaches with the experimental data. To have more concise targets and get the number of targets instead of point targets, and the clustering method to group the CFAR result data has been applied.

1.4 Organization

In this study, we basically have these sections:

- Chapter2 - Introduce the detection test statistics we are using in this report to evaluate the performance of algorithms.
- Chapter3 - Introduce the procedure of cell-averaging CFAR and order-statistic CFAR, analyze the performance of these two methods in some homogeneous environments. Also, explain the two-dimensional (2D) CFAR data structure and show the test results in the 2D homogeneous Gaussian noise environment.
- Chapter4 - Introduce three clutter models, Gamma, Weibull, and Log-normal. Investigate the performance of two CFAR approaches in these clutters evolved

environments. The ROC curves of the probability of detection with the probability of false alarm rates are given.

- Chapter5 - Calculate the big O of CFAR methods. Compare the different sorting methods that affect the computational complexity of OS-CFAR. Investigate the efficient implementations of CA-CFAR and OS-CFAR.

- Chapter6 - Introduce the experimental data to do the CFAR tests. Apply clustering method density-based spatial clustering of applications with noise (DBSCAN) to group the CFAR results. Analyze the CFAR methods' performances with clustering results.

Chapter 2

DETECTION TEST STATISTIC

2.1 Constant False Alarm Rate

Constant false alarm rate (CFAR) detection refers to a common form of the adaptive algorithm used in radar systems to detect target returns against a background of noise, clutter and interference.[6] This algorithm's aim is setting a high enough threshold to limit false alarm rate within a tolerable range, but low enough to allow more targets to be detected.

If the background of targets to be detected is constant in time and space, then a fixed threshold level can be selected. The threshold level provides a specific false alarm probability, which can be determined by the probability density function of the background noise and the signal to noise ratio(SNR) of targets. However, in most radar detection systems, the background is usually mixed with a large number of clutter and interference sources, and their presence means that the noise level will vary in space and time. In this case, the constant false alarm rate algorithm can adjust the threshold according to the dynamic change of background noise level, so as to maintain the constant false alarm probability.

2.2 Performance Metrics

In this thesis, two constant false alarm rate algorithms mainly talked about are cell averaging CFAR and order statistic CFAR. In order to have a more standardized evaluation criterion to compare these two methods, the following detection performance metrics are introduced.

2.2.1 Probability of Detection and False Alarm

In detection theory, we want to identify which hypothesis is the truth:

$$\mathcal{H}_0 : \theta \in \Theta_0, null\ hypothesis \qquad \mathcal{H}_1 : \theta \in \Theta_1, alternative\ hypothesis \tag{2.1}$$

The hypothesis $\mathcal{H}_0$ corresponds to the case that target represents, and hypothesis $\mathcal{H}_1$ corresponds to the case that detection consists of noise only.

The probability of false alarm and detection are defined as follow,

$$P_{FA} = Pr\left(\mathcal{H}_1|\mathcal{H}_0\right) = \int_{\mathcal{X}} p\left(x|\theta\right) dx,\ \ for\ \theta\ in\ \Theta_0 \tag{2.2}$$

$$P_D = Pr\left(\mathcal{H}_1|\mathcal{H}_1\right) = \int_{\mathcal{X}} p\left(x|\theta\right) dx,\ \ for\ \theta\ in\ \Theta_1 \tag{2.3}$$

In constant false alarm rate results, the probability of detection(P_D) means a target being detected. The probability of false alarm(P_{FA}) is the probability of finding a spike of noise or clutter mistaken by the CFAR algorithm as a target. The goal of CFAR algorithms is to adaptively estimate the noise power and set the threshold to be high enough to limit the false alarm rate within a tolerable range. The probability of detection can help characterize the odds of detection over thousands of trail runs.

2.2.2 Receiver Operating Characteristic(ROC)

The receiver operating characteristic curve, or ROC curve, is a graphical plot that is very useful for evaluating the performance of diagnostic tests and evaluating

the accuracy of a statistical model.[7] In this thesis, ROC curves are plots of the probability of detection(P_D) versus the probability of false alarm(P_{FA}) with given different signal to noise ratio(SNR) to assess the performance of CFAR algorithms.

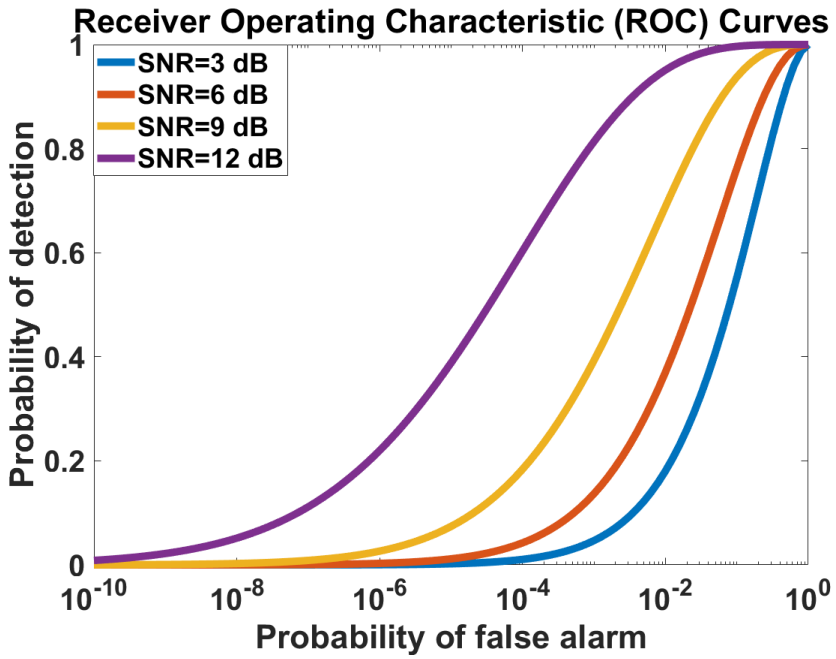

Figure 2.1: Example ROC Curves of P_D vs. P_{FA}

Fig.(2.1) shows an example of ROC curves plot between the probability of detection and probability of false alarm. The function of PD for given P_{FA} in the white Gaussian noise environment is,

$$P_D = \frac{1}{2} erfc(erfc^{-1}(2P_{FA}) - \sqrt{SNR}) \tag{2.4}$$

where $erfc$ is the complementary error function (erfc) which is defined as,

$$erfcx = 1 - erfx \tag{2.5}$$

where erf is the error function of a complex variable defined as:[8]

$$erfx = \frac{2}{\sqrt{\pi}} \int_0^x e^{-t^2} dt \tag{2.6}$$

2.3 Computation Complexity

The amount of resources required for executing a particular algorithm is the computational complexity of that algorithm. Time complexity and space complexity are two key points as considered, of which time complexity is the one which is analyzed the most. Sometimes the big O notation is denoted as the complexity of the worst case for the algorithms. The big O notation is used to classify algorithms according to how their run time or space requirements grow as the input size grows.[9] Let us set the data size as n, where the computational complexity for the method is $f(n)$.

In the real-time radar detection system, we want all methods to be achieved fast enough with low computation. For CFAR detection algorithms, the computation complexity of cell averaging CFAR is relatively low, since this method only takes a mean level of background. However, when considering the order statistic CFAR, the sorting method and repeating sorting issue can cause a huge amount of computation. Therefore, computational complexity should also be considered when comparing the performance of these two algorithms. By analyzing the computational complexity of these algorithms, we can also ensure their performance does not get too much damage to achieve more rapid and efficient algorithms.

Chapter 3

CONSTANT FALSE ALARM RATE DETECTION

In this chapter, we are introducing the procedure of two constant false alarm rate algorithms, which are Cell Averaging CFAR(CA-CFAR) and Order Statistic CFAR(OS-CFAR). This chapter includes the performance of these two methods in different scenarios. The two-dimensional data structure of the CFAR method is also shown.

The following Fig.(3.1) shows the general architecture of CFAR algorithms:

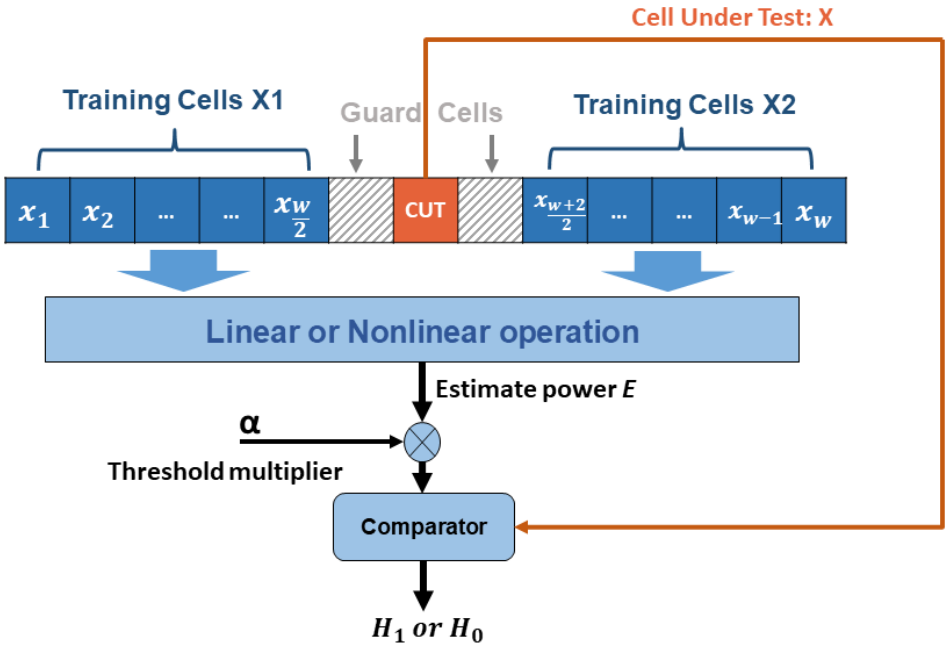

Figure 3.1: Procedure of 1D-CFAR

3.1 Cell Averaging Constant False Alarm Rate

Cell averaging CFAR(CA-CFAR) was developed early on, in 1947, by Howard Finn.[10] CA-CFAR is the most typical method in the simple detection scheme. It is accomplished with the threshold of detection by taking the mean level of the noise floor. In the homogeneous environment, CA-CFAR shows excellent performances. When enlarging the window size of training cells in CA-CFAR, the probability of detection also increases.

However, when the environment gets more complicated and introduces some clutters, the performance of CA-CFAR becomes worse.

In the Cell Averaging CFAR, the power threshold is calculated by taking the average of the training cells' power surrounding the cell under test(CUT). To avoid the self-interference in case a target is located near the CUT, there are some guard cells surrounding CUT. Let $x_1, x_2, ..., x_w$ be the reference cells surrounding the cell under test x. We calculate the threshold multiplier alpha base on the probability of false alarm and the number of training cells.

Following the procedure shows in the Fig.(3.2):

Take the average of all training cells in the window to get estimated noise level power,

$$E = \frac{1}{W} \sum_{i=1}^{W} x_i \tag{3.1}$$

Take the average as the reference power and times with threshold multiplier,

$$Z_T = \alpha * E \tag{3.2}$$

Threshold multiplier is calculated based on number of training cells and P_{FA},

$$\alpha = P_{FA}^{\frac{-1}{W}} - 1 \tag{3.3}$$

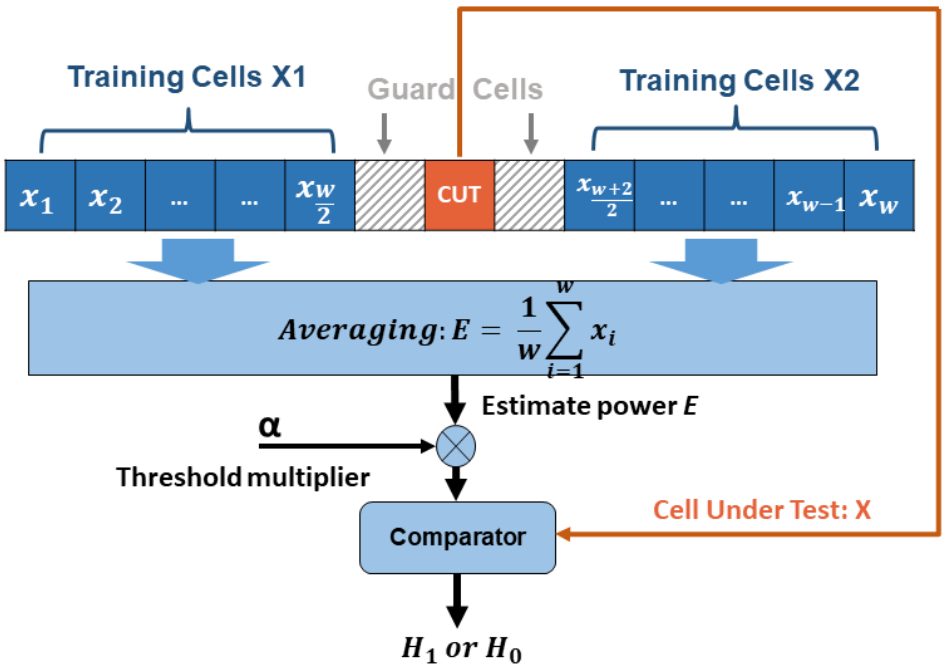

Figure 3.2: Procedure of 1D-CACFAR

Compare the cell under test with the threshold,

$$X \lessgtr Z_T \tag{3.4}$$

Move to the next test cell.

In the cell-averaging CFAR method, the P_{FA} can be derived from the equation of threshold multiplier α, and the probability of detection is given by,

$$P_D = (1 + \frac{\alpha}{1 + SNR})^{-W} \tag{3.5}$$

where SNR is the signal to noise ratio of a target.

The size of the training window and probability of false alarm(P_{FA}) is pre-selected by designers. The P_{FA} should be low enough to avoid the wrong detection, and the

size of training window N should be large enough to get efficient references. Both of these two elements will influence the performance of CA-CFAR detection. When we increase the window size, the probability of detection will also increase. As the number of cells utilized in estimating the mean level increases, the probability of detection approaches that of the classical Neyman-Person.[11] But sometimes it's not necessary to pick a very large size. This will introduce some unnecessary computations. There is a trade-off to chose the value of P_{FA}. When we pick a P_{FA} that is too small, sometimes it might cause CA-CFAR to not fully detect all the targets, which means a lower probability of detection. If we pick a P_{FA} that is too large, although we can maintain a high probability of detection, there also will be a high rate to detect noise or clutter as targets.

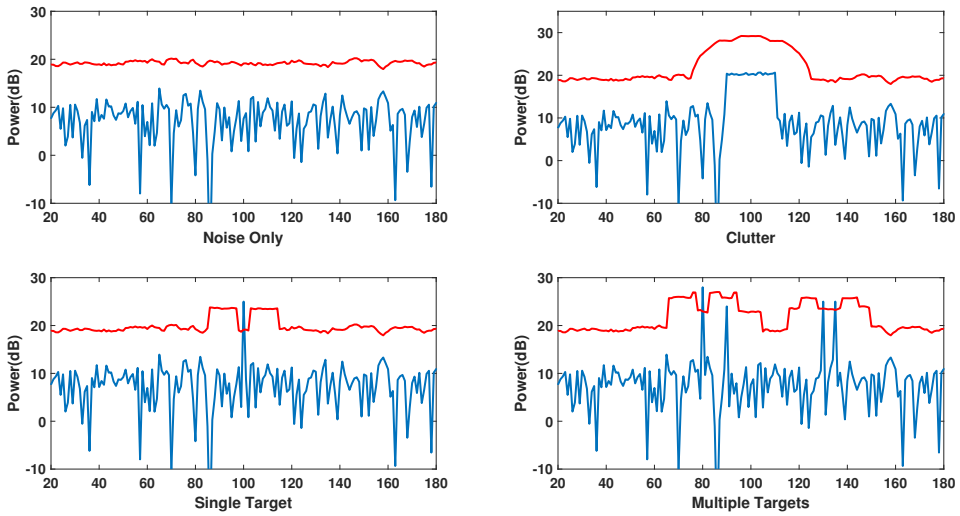

Figure 3.3: CA-CFAR(w=24, nGuard=2), $P_{FA} = 1 \times 10^{-5}$)

The number of training cells, the number of guard cells, and pre-setting P_{FA} are important parameters that can influence the performance of CA-CFAR. Fig.(3.3) shows the performance of CA-CFAR in four different scenarios. In the homogeneous noise environment, CA-CFAR provides a good detection that can be noticed

in figure(a). Also, CA-CFAR gets good results in the local clutter and single-target situation. From the figure(c), the threshold near the target is relatively high, since this caused some problems in the last situation. In the multiple targets detection scenario, the masking effect strongly influences the performance of the detection result. The closing targets cannot be detected if other targets have higher SNR.

Although CA-CFAR has great detection results in the homogeneous environment with separately spacing targets, its performance drastically dropped with closing targets, and clutter wall evolved in the background. This method is not suitable in a complicated environment and detecting multiple closely spaced targets.

3.2 Order Statistic Constant False Alarm Rate

Although Cell Averaging CFAR detection shows good performances in the homogeneous environment, it behaves very sensitively in multiple close targets. It is not efficient when the background evolves with clutters. Under these situations, Order Statistic CFAR can provide high performances in the non-homogeneous environment.

It is well known from general signal processing topics that estimation procedures are much more robust if they are based on ordered statistics.[12] In the Order Statistic CFAR, the reference cells are used to calculate the threshold picked from the sorted training cells. Due to which, OS-CFAR is not like the CA-CFAR taking the mean level of all cells in the window, which can mitigate the influences from clutter and noise. Same architecture as CA-CFAR, OS-CFAR has cells under test(CUT) surrounded with guard cells. OS-CFAR also uses the sliding window to take reference cells.

Following the procedure shows in the Fig.(3.4). Combine all the training cells together, and sort them according to increasing magnitude, resulting in the ordered

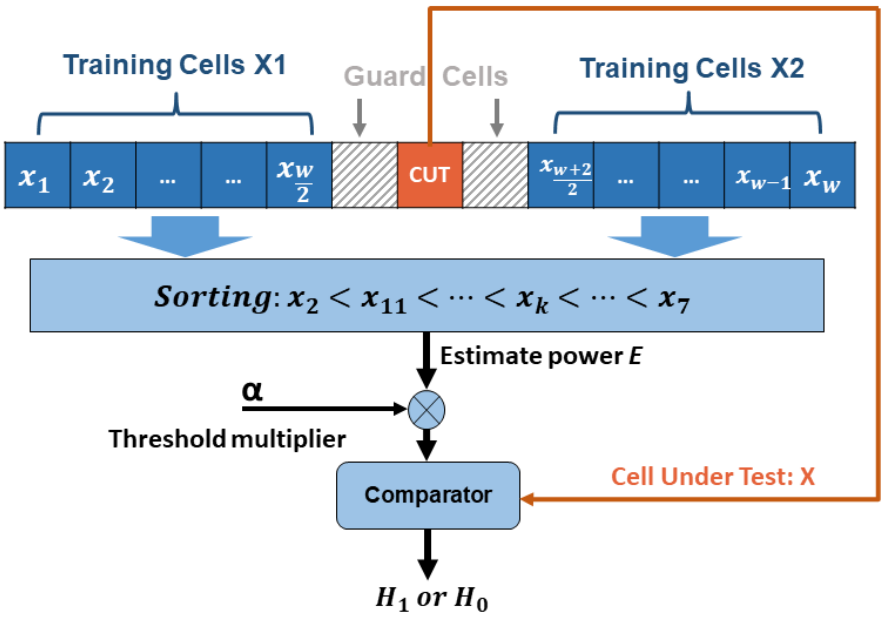

Figure 3.4: Procedure of 1D-OSCFAR

sequence,

$$x_1 \leq x_2 \leq ... \leq x_k \leq ... \leq x_w \tag{3.6}$$

Pick the reference cell by taking kth cell from sorted training cell window,

$$E = x_k \tag{3.7}$$

Calculate the decision threshold by taking the product of estimated noise level power with threshold multiplier,

$$Z_T = \alpha * E \tag{3.8}$$

Compare the magnitude of CUT with threshold,

$$X \lessgtr Z_T \tag{3.9}$$

Move to next test cell.

The threshold multiplier α can control the probability of false alarm, and the factor can be derived from the P_{FA} function. Assume the random variable x has a probability density function $p(x)$ and distribution function as $P(X)$. And the k-th selected sample has a probability density function that can be derived from $p(x)$ and $P(X)$.

$$p_k(x) = k\binom{w}{k} P\left(X\right)^{k-1} \left[1 - P(X)\right]^{w-k} p(x) \tag{3.10}$$

For Reyleigh distribution for H_0 and H_1 and square law detector,

$$p(x) = e^{-x},\ P(X) = 1 - p(x) \tag{3.11}$$

The probability of false alarm will be the average of probability crossing the threshold Z_T,

$$P_{FA} = E_x\left\{P[X > Z_T|H_0]\right\} = \int_0^\infty e^{-\alpha x_k} p_k(x) dx \tag{3.12}$$

By using Eq.(3.10) and Eq.(3.12) we can get,

$$P_{FA} = k\binom{w}{k} \frac{\Gamma\left(k-1\right)\Gamma\left(\alpha + w - k\right)}{\Gamma\left(\alpha + w\right)!} \tag{3.13}$$

$$= \prod_i^k \left[1 + \frac{\alpha}{w-i}\right]^{-1} \tag{3.14}$$

After comparison in each cell under test, the presence of a target in the current CUT when its power is over than the threshold level, the probability of detection as shown in the equation below,

$$P_D = k\binom{w}{k} \frac{\Gamma(k-1)\Gamma(\frac{\alpha}{1+S} + w - k)}{\Gamma(\frac{\alpha}{1+S} + w)} \tag{3.15}$$

$$= \prod_i^k \left[1 + \frac{\frac{\alpha}{1+S}}{w-i}\right]^{-1} \tag{3.16}$$

where S is the average of signal to noise ratio.

Same as CA-CFAR, the probability of false alarm, the rank of the reference cell, and the window size of reference cells is pre-selected by designers. The P_{FA} should be

low enough to avoid false detections, also need to keep the P_D enough high based on the relationship between P_D and P_{FA}. The rank selection is a key element to affect the performance of detection. It also depends on the SNR of targets and the model of clutter. When the SNR and clutter edges are relatively high, we should pick a higher rank to avoid the wrong detection. Usually, we pick the rank as 65% to 80% of the window size.

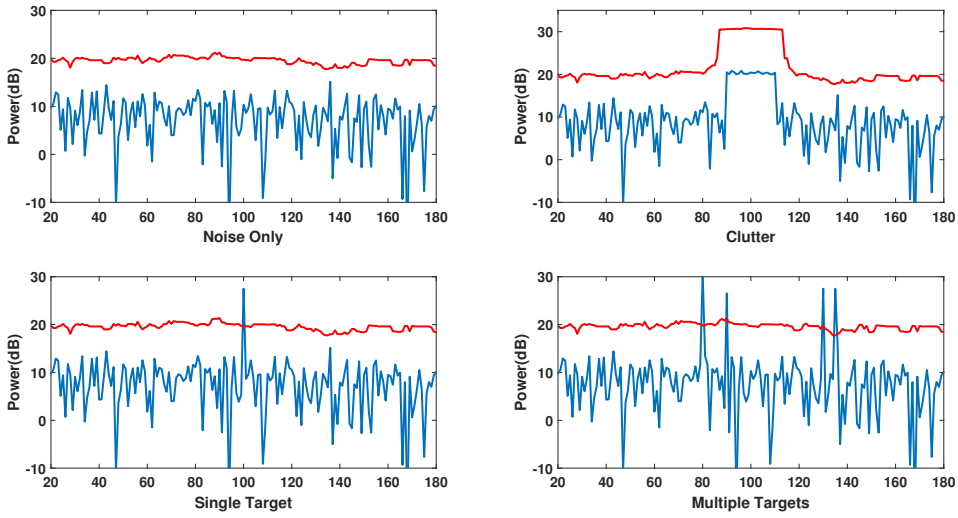

Figure 3.5: OS-CFAR(w=24, nGuard=2), P_{FA}=1×10^{-5})

In the same testing situations of CA-CFAR, OS-CFAR shows a higher capability to achieve adaptive detection. Fig.(3.5) shows the OS-CFAR threshold level in four different scenarios. In the noise only homogeneous environment, there is no significant difference with the CA-CFAR detection result. Under this circumstance, there is no need to use OS-CFAR in the homogeneous environment, because OS-CFAR has higher computation complexity but shares the same performance with CA-CFAR. The computational complexity of CFAR approaches will be discussed later.

In the next three scenarios, OS-CFAR shows its advantages. OS-CFAR calculates thresholds that have a more clear edge than CA-CFAR. Additionally, in both target

scenarios, the threshold is more or less unchanged compared with the noise only scenario. The close targets are clearly being detected.

The performance of OS-CFAR and CA-CFAR in different scenarios, basically showing the same detecting results but also confirms that OS-CFAR has a higher capability of avoiding masking effect from closing interfering targets.

3.3 Two Dimensional Constant False Alarm Rate Implementation

When considering two-dimensional signals such as the range Doppler domain, we also need to consider the two-dimensional detection method. In this section, two-dimensional constant false alarm rate detection implementation is introduced.

The following Fig.(3.6) shows the data structure of 2D-CFAR:

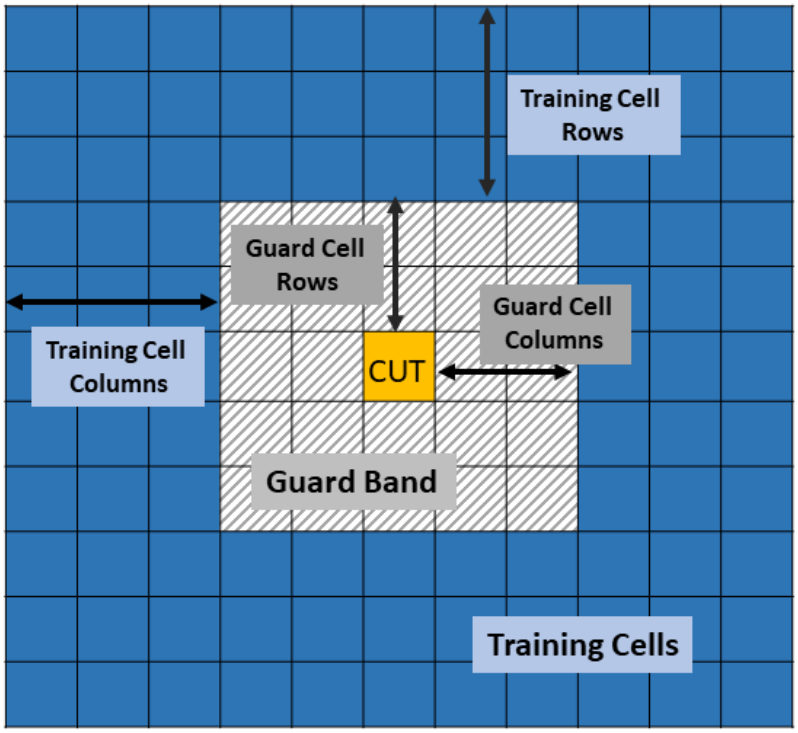

Figure 3.6: 2D-CFAR Data Structure

When testing data extends to two dimensions, the guard cell band and training cell window should also extend to two dimensions. Same as the one-dimensional CFAR method, for CA-CFAR still takes the mean of the power of all the training

cells. For the OS-CFAR, we still pick the k_{th} cell from the sorting training cells. Since the reference window turns to two-dimensional, the number of training cells is increasing. Also, the selection from both directions will influence the performance of methods.

In the 2D-OSCFAR implementation, instead of sorting the training cells surrounding CUT all together, cells are separated into four sections. Then do the sorting and picking the k_{th} cell in each section. The reference power level is by averaging the mean level of four section's k_{th} cells. The result of OS-CFAR and CA-CFAR in two-dimensional Gaussian noise background is showing,

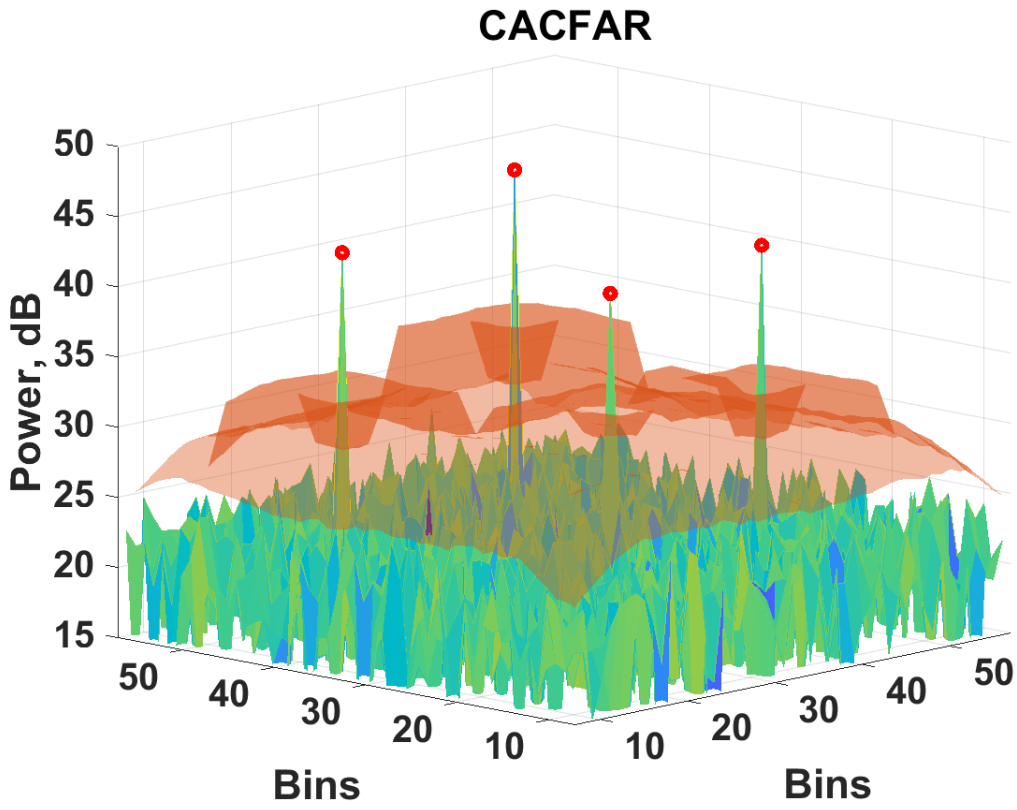

Figure 3.7: 2D-CACFAR Detected Threshold in Gaussian Noise Background with $P_{FA} = 10^{-5}$, Training Cell Window = [5,5], Guard Cell Window = [2,2]

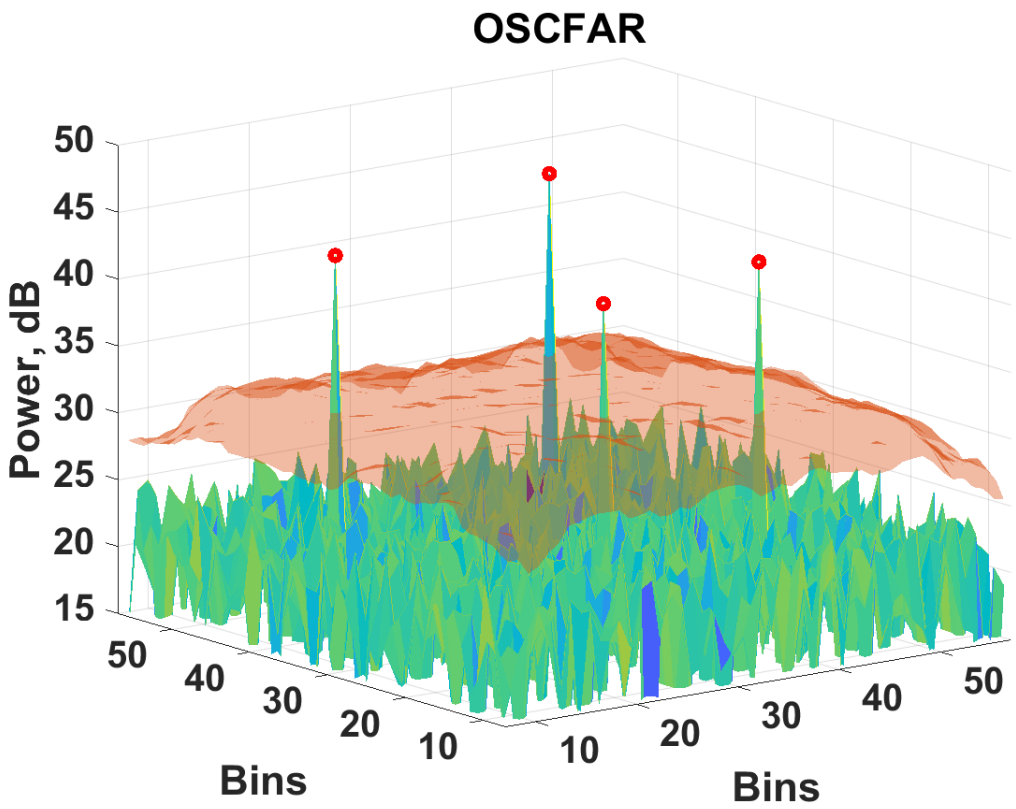

Figure 3.8: 2D-OSCFAR Detected Threshold in Gaussian Noise Background with $P_{FA} = 10^{-5}$, Training Cell Window = [5,5], Guard Cell Window = [2,2], Rank = 0.6

In Fig.(3.7) and Fig.(3.8), in the homogeneous Gaussian noise environment, CA-CFAR and OS-CFAR both show great performance in setting thresholds and clearly detected the targets. But CA-CFAR method again has masking effects in the 2D case. In this case, the detection result is not influenced by this effect. However, we cannot guarantee the performance of CA-CFAR in the case where interfering targets present in the reference window. On the other hand, the thresholds calculated from OS-CFAR are not influenced by the target's power. In the next chapter, we investigate two approaches' performances in different clutter-evolving environments and non-homogeneous environments.

Chapter 4

PERFORMANCE ANALYSIS

Radar operating in the open environment will receive return signals coming from many different sources. The surface reflects signals from the land, sea or other surfaces called clutter. CFAR algorithms should have the capability to operate in kinds of environments such as homogeneous environments, multiple targets scenarios, and clutter evolved environments. In the real life, the noise and clutter are not match with Gaussian distribution. In the proceeding chapters, three different clutter modelings participate in the performance analysis of the CFAR algorithm. The previous chapter introduced the performances of CA-CFAR and OS-CFAR in the Gaussian noise environment. To have a better performance analysis of CFAR approaches, we introduce different clutter modelings to evaluate CA-CFAR and OS-CFAR in more complicated environments.

4.1 Clutter Models

Clutter refers to radio frequency echos returned from different surfaces. Clutter signals that affect radar performance are typically categorized in terms of backscatter from the land, the sea, and the atmosphere. [13] Various probability distributions have been used to model clutter returns for radars operating in a variety of backgrounds like ground, sea, etc.[14] In this chapter, we will mainly discuss three types of popular clutter modeling: Gamma, Weibull, and Log-normal. Although these three mathematical models cannot be precisely the same as reality, they can provide a good assessment of detection methods.

4.1.1 Gamma

The Gamma distribution is widely used. It is the maximum entropy probability distribution (both with respect to a uniform base measure and with respect to a $\frac{1}{x}$ base measure) for a random variable x.[15] This distribution is a two-parameter family of continuous probability distributions.

The amplitude distribution for gamma is:

$$p\left(x_i\right) = \frac{1}{b^a \Gamma\left(a\right)} x_i^{a-1} e^{-\frac{x_i}{b}}, x_i \geq 0, a > 0, b > 0 \tag{4.1}$$

where a is a shape parameter, b is the scale parameter, and Γ is the Gamma function.

The mean of the gamma distribution is $\mu = ab$, and the standard deviation is $\sigma = ab^2$.

4.1.2 Weibull

The Weibull distribution is named after the Swedish engineer and scientist Ernst Hjalmar Waloddi Weibull(1887–1979). This distribution is widely used in engineering, medicine, and elsewhere.[16] In the radar clutter modeling, the Gaussian distribution is characterized as cloud; the Weibull distribution has been more accurately characterized as the ground clutter with two parameters: the scaling and shaping parameters.

In general, the probability density function of Weibull distribution is:

$$p\left(x_i\right) = \frac{b}{a} \frac{x_i}{a}^{b-1} e^{-\left(\frac{x}{a}\right)^b}, x_i \geq 0, a > 0, b > 0 \tag{4.2}$$

where a is the scaling parameter and b is the shape or slope parameter.

4.1.3 Log-normal

Donald McAlister in 1879 appears to have been the first to give a comprehensive view of the log-normal distribution.[17] The log-normal distribution is the statistical

distribution of the logarithmic values of the related normal distribution. This model has a remarkable feature. It has long tails which may affect the detection method performance.

In general, the probability density function of Log-normal distribution is:

$$p\left(x_i\right) = \frac{1}{x_i \sigma \sqrt{2\pi}} e^{\left\{-\frac{(logx-\mu)^2}{2\sigma^2}\right\}}, -\infty < \mu < \infty, \sigma \geq 0 \tag{4.3}$$

where μ is the mean of distribution and σ is standard deviation.

Table 4.1: Clutter Models and Equations

Clutter Type		
Distribution Model	Application	Equation
Gaussian	Clouds	$p\left(x_i\right) = \frac{1}{\sqrt{2\pi\sigma^2}} exp\left[-\left(\frac{x_i}{b}\right)^c\right]$
Gamma	Land Clutter	$p\left(x_i\right) = \frac{1}{b^a \Gamma(a)} x_i^{a-1} e^{-\frac{x_i}{b}}$
Weibull	Land Clutter	$p\left(x_i\right) = \frac{b}{a}\frac{x_i}{a}^{b-1} e^{-\left(\frac{x}{a}\right)^b}$
Log-normal	Sea Clutter	$p\left(x_i\right) = \frac{1}{x_i \sigma \sqrt{2\pi}} e^{\left\{-\frac{(logx-\mu)^2}{2\sigma^2}\right\}}$

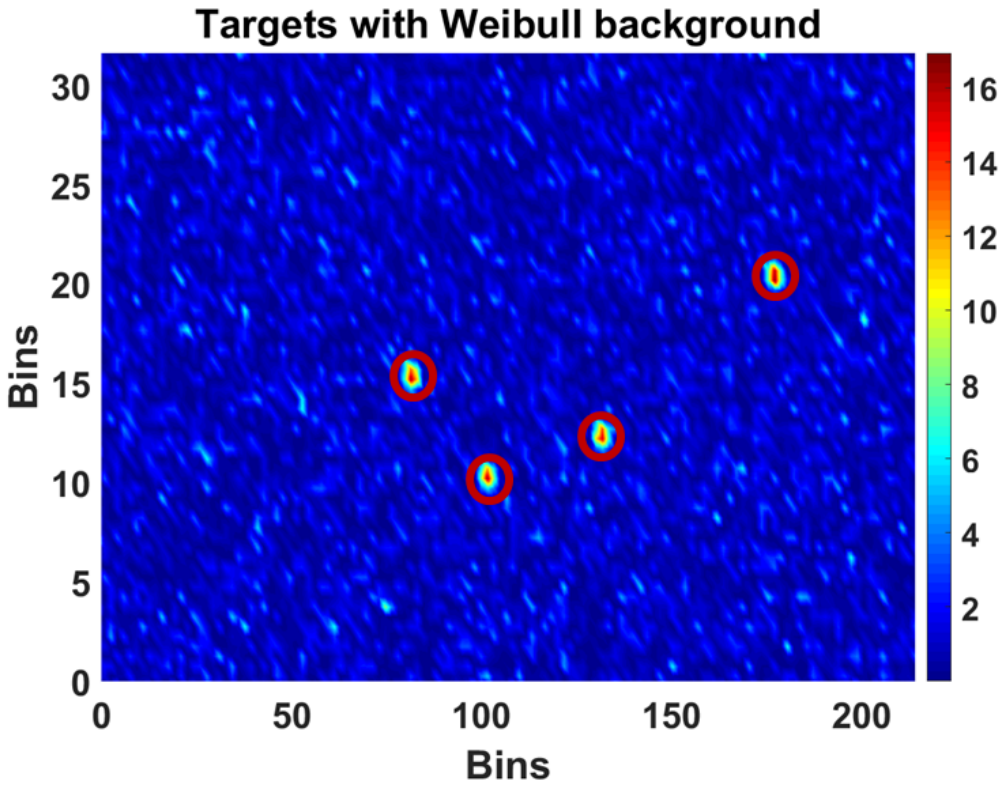

Figure 4.1: Example Case of Four-targets with Weibull Clutter Background

Fig.(4.1) shows one example case of four targets filled with the Weibull clutter background. In order to set stages of signal to clutter ratio (SNR), targets in one test are set with the same value of SNR. This scenario has been provided to test the performance of CFAR approaches.

4.2 Performance Comparison

After introducing three types of clutter modelings, we can apply these clutters into the background to compare the performance of CA-CFAR and OS-CFAR.

4.2.1 Performance Comparison in Clutter Environment

To have a better performance analysis, the probability of detection(P_D) and the probability of false alarm(P_{FA}) are being calculated under clutter noise background. Also, we apply different SNR to run Monte Carlo experiments and plotted results in receiver operating characteristic (ROC) curves. The SNR here should be the signal to clutter ratio.

The one-dimensional and two-dimensional gamma clutters being introduced into the background to test performance. For the 1D case setting, the SNR of targets are 3dB, 5dB, 7dB, 9dB, and 11dB. The shape and scale parameters of gamma are both set as 1. Also, we set both CA-CFAR and OS-CFAR with the same probability of false alarm from 10^{-6} to 1, the same size of training cell window of N=32, and the rank of OS-CFAR selected as 0.7N. In the 2D case, after we consider the environment evolved with clutter from both directions, we determine whether the size of the reference cell windows needs to be increased. We set the size of the guard cell window as [1,1], which is eight cells surrounding the cell under test (CUT), and the size of the training cells window set as [3,3], which contains a total of 82 cells. Also, because both algorithms cannot perform very well and have a high probability of false alarm, the testing stages of SNR remove the 3dB level.

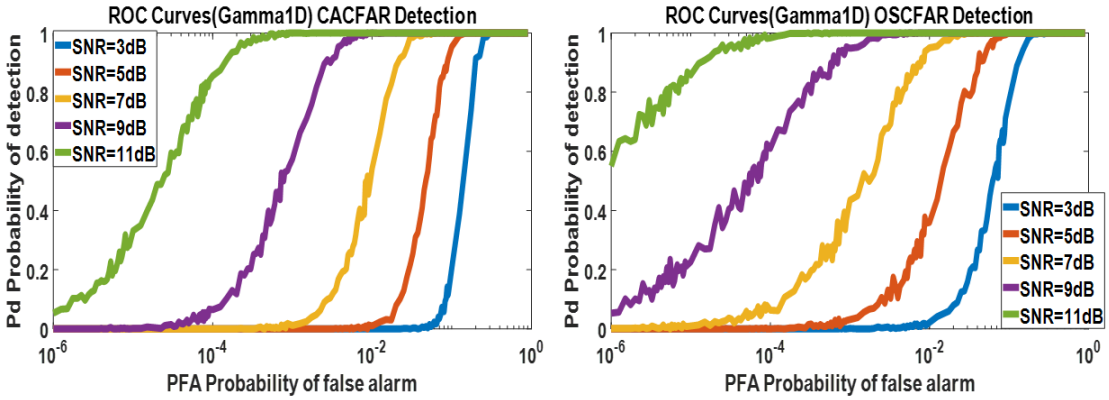

Figure 4.2: ROC Curves of CFAR Test Results in 1D Gamma Clutter Environment

Fig.(4.2) presents the contrast result of two methods in the one-dimensional gamma clutter case. SNR set as 3dB, 5dB, and 7dB, OS-CFAR slightly outperforms CA-CFAR. When SNR increasing to 9dB, the probability of detection from OS-CFAR gets great improvement. From the ROC curves of CA-CFAR, the probability of detection all present the same growth trend. However, the performance OS-CFAR gets significantly enhanced when the SNR of targets increasing.

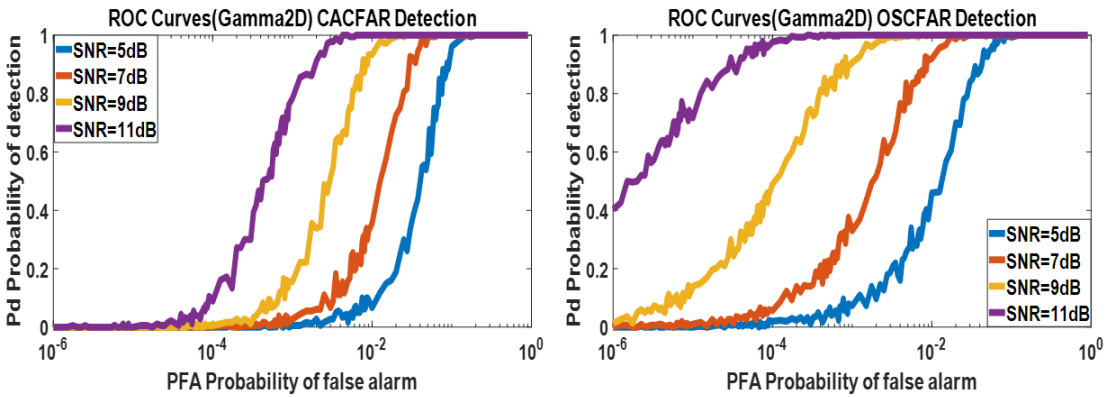

Figure 4.3: ROC Curves of CFAR Test Results in 2D Gamma Clutter Environment

On the other hand, Fig.(4.3) shows CFAR detected results in the two-dimensional gamma clutter environment. When SNR setting in lower stages, OS-CFAR keeps the high performance as the 1D scenario. However, the probability of detection from CA-CFAR decreased when compared with the 1D result, especially at 11dB, CA-CFAR start detecting targets when P_{FA} is increasing at 10^{-4}. With the same configurations, CA-CFAR lack of capability of detection in gamma clutter conditions.

When we introduced Weibull clutter modeling, the same parameter setting as the gamma case, the SNR of targets set as five stages from 3dB to 11dB. After running thousands of trials to counting the actually detected targets, the receiver operating curves are given in Fig.(4.4). CA-CFAR detection performance sharing the same increasing trend versus different SNR and CA-CFAR detection results in gamma clutter background. Its performance has slightly declined. Nevertheless, order statistic CFAR still maintain a high performance versus Weibull clutter background

and different value of SNR.

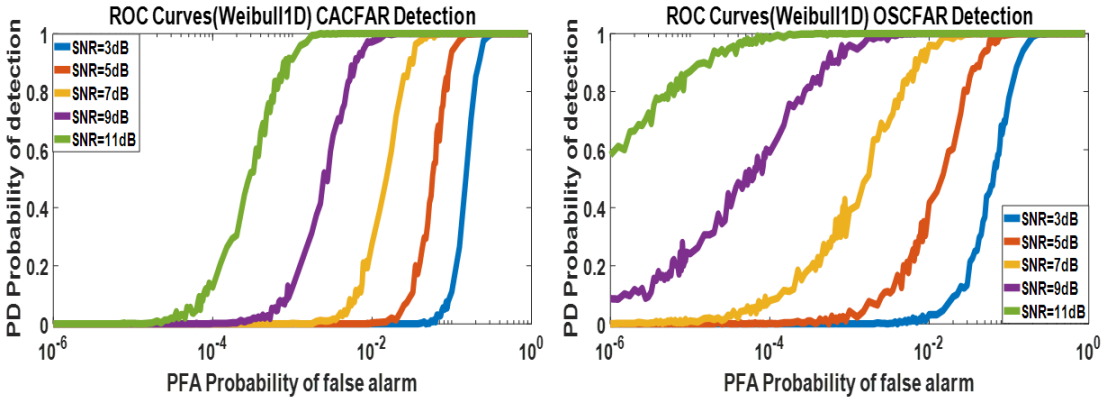

Figure 4.4: ROC Curves of CFAR Test Results in 1D Weibull Clutter Environment

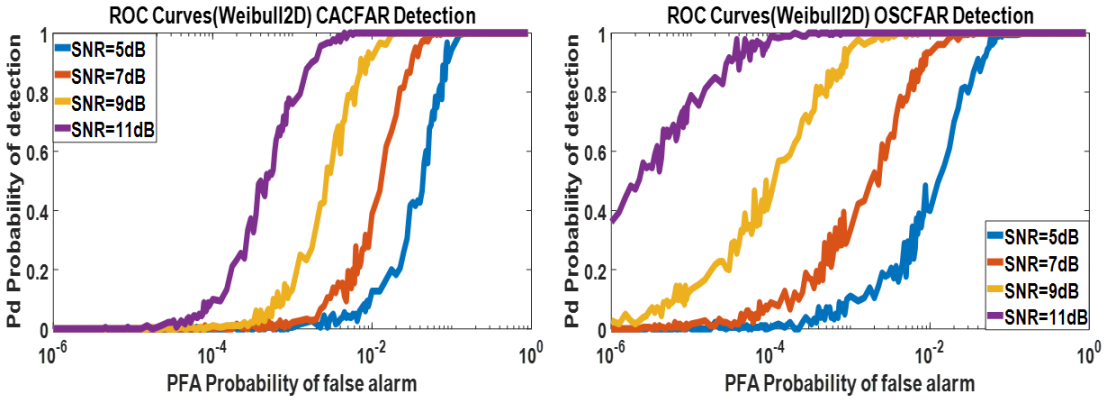

Figure 4.5: ROC Curves of CFAR Test Results in 2D Weibull Clutter Environment

In the two-dimensional Weibull clutter environment, Fig.(4.5) shows the same results in the gamma case. CA-CFAR performance gets slightly decreased while OS-CFAR still maintains well behaved. From the figure, it's obvious that the OS-CFAR detection with SNR's growth, performances are gradually improving.

Considering the log-normal clutter scenario, the same configurations in the Monte-Carlo simulation. The detection performance of CA-CFAR diminishes drastically compare with the gamma clutter background. But the performance of OS-CFAR also gets a certain degree of reduction. Although log-normal distribution cannot be presented as a compound Gaussian model, it has long tails that can seriously affect detection performance.

Fig.(4.6) and Fig.(4.7) show the result for the Log-normal environment test. In the two-dimensional scenarios, the probability of detection for CA-CFAR is decreased compared with the same SNR level results in the 1D result, which means its performance also get deteriorated in the 2D scenario. On the other hand, in the same SNR setting situation, OS-CFAR starts detecting the targets from the relatively higher PFA. However, the performance of OS-CFAR gets improving when SNR is increasing. Still, OS-CFAR performs more efficiently than CA-CFAR against with log-normal clutter environment.

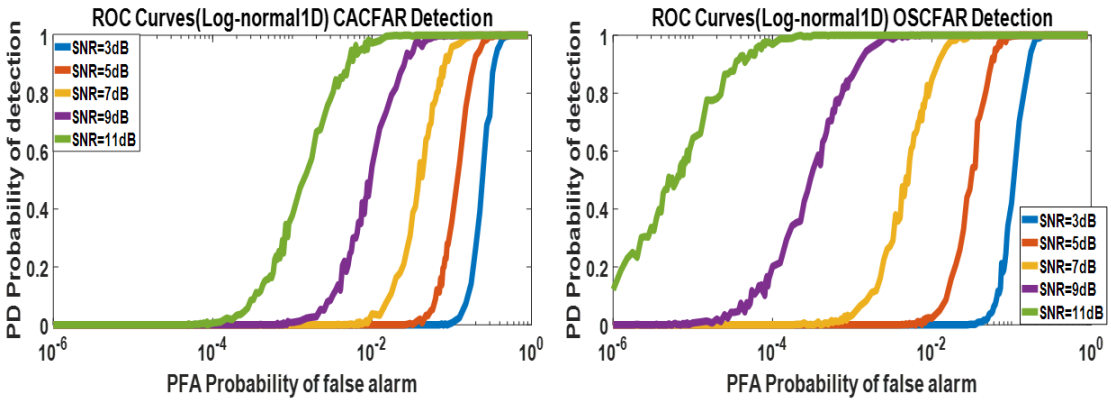

Figure 4.6: ROC Curves of CFAR Test Results in 1D Log-normal Clutter Environment

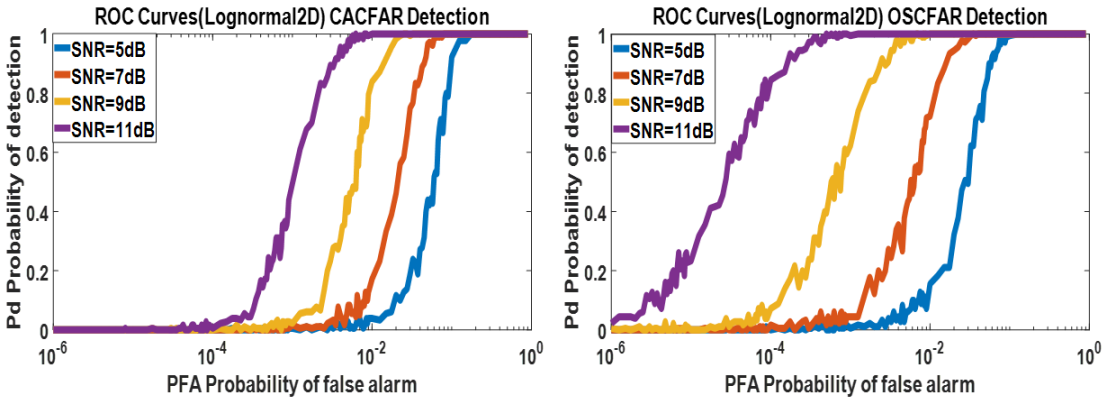

Figure 4.7: ROC Curves of CFAR Test Results in 2D Log-normal Clutter Environment

Combining with the detection results of these three different clutter environments, it is not surprisingly noticed that OS-CFAR successfully reveals its capability to detect targets against clutter and noise background. When we increase the signal to clutter ratio, the performance of OS-CFAR has also been improved. However, the performance of CA-CFAR has not changed much when SNR increases.Even though, these two methods almost share the same performance when SNR at a lower stage. But still, the CA-CFAR approach is not a good choice in clutter evolved environment, especially in the higher dimensional cases.

Chapter 5

REDUCING COMPUTATIONAL COMPLEXITY

Based on the analysis of performance comparison between OS-CFAR and CA-CFAR, it is obvious that OS-CAFR has a higher capability of detection in a more complicated environment. On the other hand, the OS-CFAR has an outstanding feature, the computational complexity is higher than other conventional CFAR methods. However, in some applications, for example, automotive radar, real-time monitoring systems, etc. The computational complexity of algorithms is an essential factor when designer selecting algorithms. High computations requires high power consumption and more processing time. Therefore, we want the computation can be reduced as much as possible.

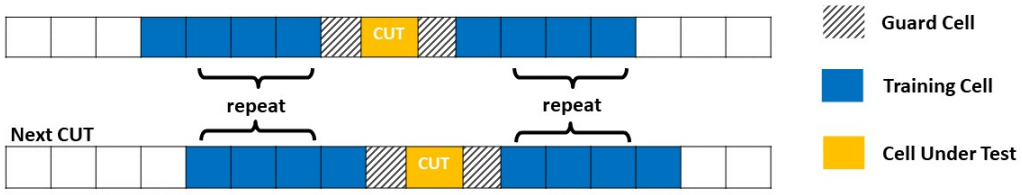

Figure 5.1: Cell Under Test Shifts with Training Cells and Guard Cells

Fig.(5.1) shows a simple case of the CFAR method. Whenever the cell under test(CUT) moves to the next position, the training cell window and guard cells will follow with the CUT. The number of training cells and guard cells will always be fixed. From the figure, we can notice that, for the next CUT, there are some training cells in the previous window that are also showing in the current one. Every time

the CFAR method is doing an operation to calculate the threshold, there are some cells that are being repeated considered. Base on this feature, we can reduce the computations by avoiding repetitions.

In this chapter, under the premise of not damaging the performance of CA-CFAR and OS-CFAR. We maintain the key characteristics of these methods, and show the implementations of reduced computational complexity algorithms in detail.

5.1 Cell Averaging Detection Statistic

The main idea of CA-CFAR is to take the mean level of all the cells in the reference window. Let set the number of detecting data as N, the number of reference cells in window set as W. Then we can get the computation complexity O_{CACFAR} as $N * W$.

Although the computation complexity of cell averaging constant false alarm rate algorithm is very low, we still can make some improvements to reduce computations. At the same time, we maintain the same performance as the conventional CA-CFAR method.

When the reference window is shifting with the cell under test, some new cells is coming into the window. Some cells in the previous window do not appear in the current one, which means every time CA-CFAR taking the mean level from the window, some cells are repeating calculating. Depending on the size of the window, the more repetitive exists when the window is bigger.

In order to avoid repeat accumulation, we can directly call the averaging level from the previous CUT and taking new cells into consideration to adjust the reference. Also, to maintain the performance of CA-CFAR, the cells excluded from the reference window still needs to be considered. The big $O_{newCACFAR}$ in one-dimensional CA-CFAR method becomes to (here we only consider the computational complexity of

computing reference value):

$$O_{newCACFAR1D} = W + 4 * (N - 1) \tag{5.1}$$

If we consider higher dimensional data, there are more new cells come into the window when CUT shift. Assume the number of new cells as n. The big O for efficient CA-CFAR is:

$$O_{newCACFAR} = W + 2n * (N - 1)\,,\ n \ll W \tag{5.2}$$

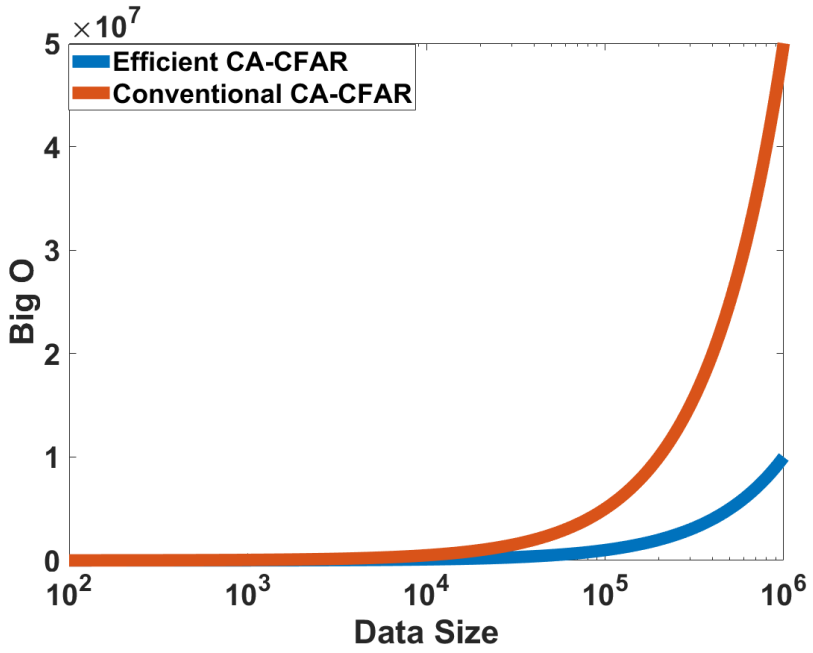

Figure 5.2: Big O of Traditional CA-CFAR and Efficient CA-CFAR, W = 50, n = 10

In Fig.(5.2), the comparison of big O for traditional CA-CFAR method and efficient approach is showing. The data size is being chose from $5 * 10^3$ to 10^6. No matter what dimension of data, only data size, number of new cells and reference cell can influence the computational complexity. From the figure, the big O of traditional method can be 5 times of efficient approach, also O_{CACFAR} is keeping a rapid growth trend.

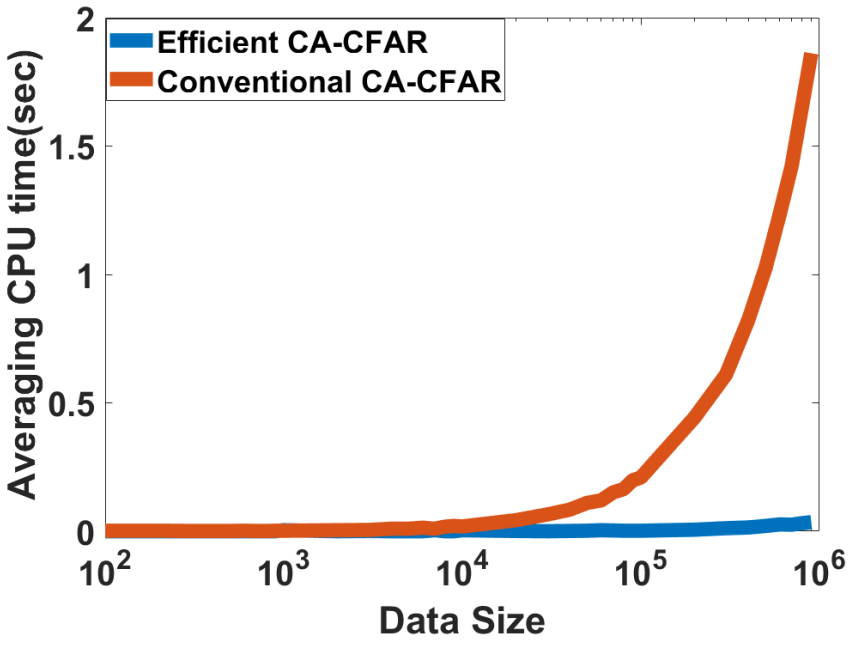

Figure 5.3: CPU Time of Conventional CA-CFAR Method and Efficient CA-CFAR Method Versus Different Testing Data Size, Training Cell Window Size W=24

After reducing the computation complexity, the central processing unit(CPU) time of two methods under the same configurations, averaging the time over 1000 Monte Carlo simulations, the results are showing in Fig.(5.3), the CPU time of conventional CA-CFAR basically no change when data size less than $4 * 10^4$, and then it increases rapidly. However, the reduced computational complexity CA-CFAR method is not influenced by the increasing data size.

Algorithm 1 Efficient CA-CFAR

Input: signal x, number of training cells W, PFA

Output: Threshold Z_T

1: Compute mean level of reference cells for first CUT, R

2: Compute threshold for first CUT, Z_T, Eq.(3.2) and Eq.(3.3)

3: **for** each cell x_i in X except first CUT **do**

4: Call mean level from previous CUT, R

5: Cells show in current window not in previous window as new

6: Cells show in previous window not in current one as old

7: R = $\frac{1}{W}(R * W + old - new)$

8: Compute threshold, Z_T, Eq.(3.2)

5.2 Order Statistic Detection Statistic

In the conventional order statistic CFAR methods, in order to select a suitable reference cell, we need to do sorting in every reference window. While the reference window is moving, followed with the cell under test, some cells in the current window have already done sorting in the previous one. So there are a lot of repetitive sortings, which results in increased computational complexity. On the other hand, the selection of the sorting method can also influence the complexity of OS-CFAR. In order to reduce the computational complexity of the algorithm, we can start by selecting the sorting method and avoid repetitions.

Some commonly used sorting method with big O are listed in table(5.1).

Table 5.1: Big O of Common Sorting Methods

Name	**Best**	**Average**	**Worst**
Quick Sort	$nlogn$	$nlogn$	n^2
Merge Sort	$nlogn$	$nlogn$	$nlogn$
Insertion Sort	n	n^2	n^2
Bubble Sort	n	n^2	n^2
Selection Sort	n^2	n^2	n^2

Assume there has a signal with N data that needs to be detected, and the number of training cells in the window set as W. A comparison sort cannot perform better than $O(nlogn)$.[18] For the lowest complexity, the big O for the OS-CFAR should be(only consider the iterations for computing reference cells):

$$O_{OSCFAR} = N * W * logW \tag{5.3}$$

The key point to implement an efficient algorithm is to avoid repetitive computations. In the one-dimensional OS-CFAR method, there have two more cells need

to consider when CUT moving to the next cell. We can save the sorting result from the previous training windows and maintain the results for all the cells surrounding the current CUT. Then next step is inserting two new cells into the previous sorting window. When the dimension of data is increasing, the number of training cells need to consider are also increasing. But the number of new training cells will never greater than the size of the training window. Here, we can set the number of new cells as n, and $n \ll W$. When considering new cells join into the already sorted cells, using the binary searching method is the most efficient way to get the right position of new cells, and the big O for that approach is $logW$.

So the efficient OS-CFAR becomes as doing a complete sorting for the first CUT, and for every CUT after the first one, only need to insert new training cells into the previously sorted data without some cells are not included in the current training window. Then the computation complexity of efficient OS-CFAR method is:

$$O_{newOSCFAR} = W * logW + (N-1) * n * logW \tag{5.4}$$

Table 5.2: Big O of Traditional CA-CFAR, OS-CFAR and Efficient CA-CFAR and OS-CFAR Approaches

Big O	**CA-CFAR**	**OS-CFAR**
Traditional	*N*W*	*N*W*logW*
Efficient	*W+2n*(N-1)*	*W*logW+n*(N-1)*logW*

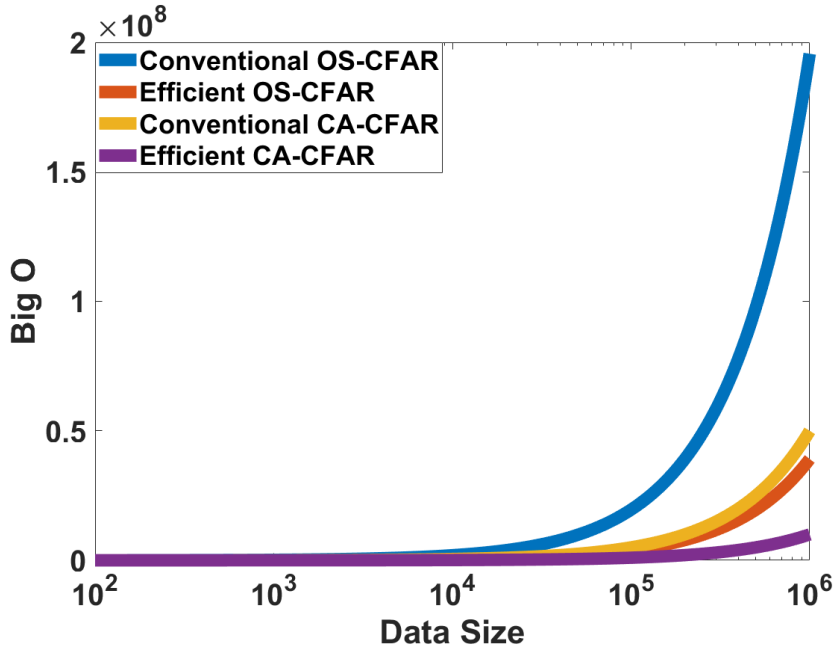

Figure 5.4: The Big O of Traditional CA-CFAR, OS-CFAR Compare with Efficient CA-CFAR and OS-CFAR, W = 50, n = 10

The result for big O of efficient CA-CFAR and OS-CFAR with traditional CA-CFAR and OS-CFAR is present in Fig(5.4). For traditional OS-CFAR, its computational complexity is higher than other methods due to repeat sorting. After we improved the OS-CFAR approach, the computation can reach the same level as the traditional CA-CFAR. We all know the CA-CFAR method has very low computational complexity, which means big O of new OS-CFAR has significant improvement. Still, the efficient approach of CA-CFAR has the lowest computations.

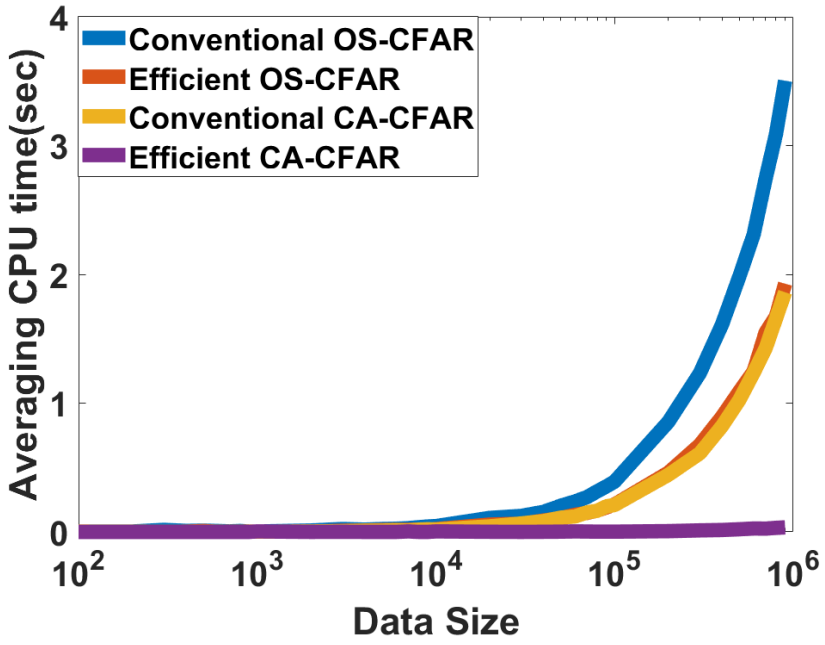

Figure 5.5: CPU Time of Conventional CA-CFAR and OS-CFAR Methods, Efficient CA-CFAR and OS-CFAR Methods Versus Different Testing Data Length, W=24

Since the computational complexity has been reduced, the CPU time should also be decreased. Fig.(5.5) presents the CPU time change based on different data sizes through four CFAR algorithms. After we run thousands of Monte Carlo simulations, it is obviously noticing that the efficient CA-CFAR and OS-CFAR method both shows great improvements on CPU time. For the efficient OS-CFAR method, its CPU time is reaching the same level as conventional CA-CFAR, by which means the computational complexity of the new method has been improved a lot.

Algorithm 2 Efficient OS-CFAR

Input: signal X, number of training cells W, PFA, rank k

Output: Threshold Z_T

1: Sort training cells in first CUT window, sort

2: Select reference cell from sorted cells, R = x_k

3: Compute threshold for first CUT, Z_T, Eq.(3.8) and Eq.(3.14)

4: **for** each cell x_i in X except first CUT **do**

5: Call sorted training cell from previous CUT, sort

6: Save cells in current CUT window and maintain sequence, sort1

7: Use binary search find right position for new cells in current window not in previous one

8: R = x_k

9: Compute threshold, Z_T, Eq.(3.8)

Chapter 6

EXPERIMENTAL DATA

In the previous chapters, the performance of CFAR methods in being considering the probability of detection and false alarm rate. Since the CFAR method processing data is cell by cell, then we only can check the results point by point. However, in real scenarios, most targets have their body shapes, and their speed is changing continuously. The point target results from the CFAR approach cannot provide efficient information on the exact number of targets. On the other hand, one CFAR method detecting more points can't prove this method is more useful than others. For example, assume one target has a huge size is being detected by OS-CFAR and CA-CFAR method. From the detection results, OS-CFAR detected more points than CA-CFAR, but in fact, no matter how many points are being detected by these two methods, they share the same performances.

Under this circumstance, we can apply the clustering method to group the CFAR results. The clustering results are more concise and have a better estimation for number of targets and its size.

6.1 Clustering Algorithm

Density-based spatial clustering of applications with noise (DBSCAN) is a density-based clustering algorithm designed to discover clusters and noise in data.[19] It's minimum density level estimation based on the threshold for the number of neighbors and $minpts$(minimum number of neighbors required for a core point) within a distance.[20]

Based on the definition of density, we can divide the data points into three sets:

- First, core point. A point is considered a core point when the number of surrounding neighbors exceeds $minpts$.
- Second, border point. A point considered as border point when the number of surrounding neighbors less than $minpts$.
- Third, noise point(labeled with -1). The Noise point is neither the core point nor the border point, and do not belong to any cluster.

DBSCAN algorithm can compute clusters base on the above three types of points. DBSCAN searches for clusters by checking the N neighbors of each point in the data set. If the N neighborhood of point p contains more than $minpts$ points, it creates a cluster with p as the core point. Then, DBSCAN iteratively aggregates points that are directly density-reachable from these core points. This process may involve the merging of some density-reachable clusters. When no new ones are added to any clusters, the iteration process ends. Here, we call points are directly density-reachable to core points when point p distance to core point less than radius ϵ. Assume we have a point link $p->l->m->n->corepoint$, the distance between any two adjacent points less than radius ϵ(they are density-reachable), the point p is density-reachable to the core point.

Algorithm 3 DBSCAN Algorithm

Input: database X, radius ϵ, $minpts$

Output: $label$: label for database in each cluster

for each point p in database X **do**
    if $label(p)$ is defined **then**
        Continue to next point
    NeighborPts N: regionQuery(p, ϵ)
    if sizeof(N)$<$ $minpts$ **then**
        $label(p) = -1$, noise
    else
        C = next cluster
        ExpandCluster$(p, C, N, \epsilon, minpts)$

ExpandCluster$(p, C, N, \epsilon, minpts)$
$label(p) = C$
for each point q in N **do**
    if $label(q) = -1$ **then**
        $label(q) = c$
    if $label(q)$ is defined **then**
        Continue to next point
    NeighborPts N_p: regionQuery(q, ϵ)
    if $sizeof(N) > minpts$ **then**
        $N = N \bigcup N_p$, combine neighbors into one cluster

regionQuery(p, ϵ)
return all the neighbors of p within the radius ϵ and p

6.2 CFAR Detection and DBSCAN Analysis

Here, some range-Doppler radar data from Metawave company allow us to do the CFAR detection test and clustering analysis.[21] These data are received by four antenna. Each frame data includes 256 *range bins* $*$ 128 *Doppler bins*, which has max range 330.0096m and velocity from $-36.9984m/s$ to $+36.9984m/s$.

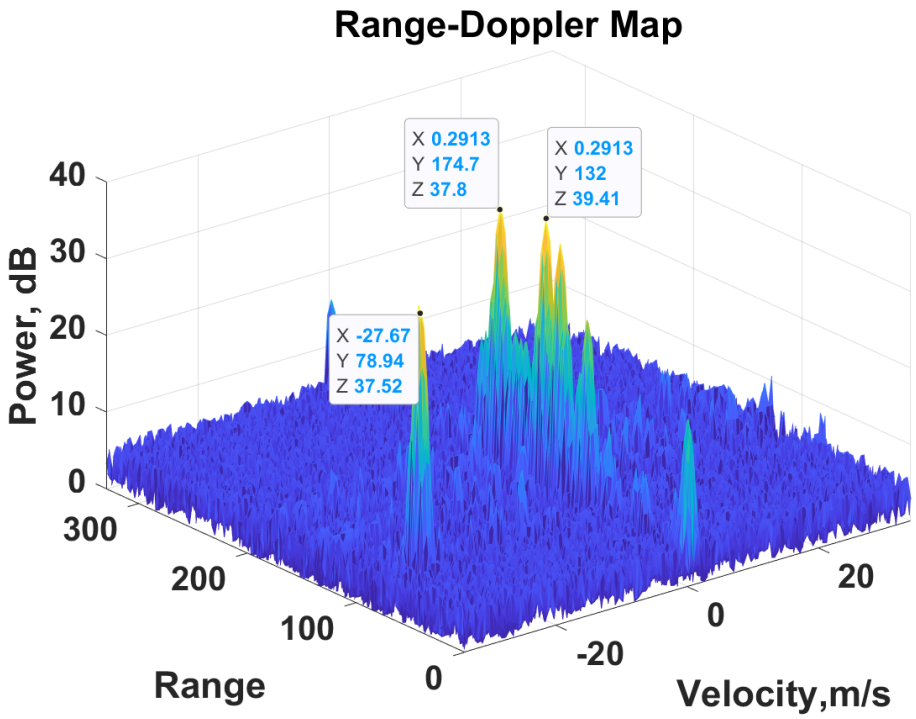

Figure 6.1: Example of One Frame Range-Doppler Data

Fig.(6.1) shows one frame range-Doppler data. From the figure, we can infer that there are some static objects and one moving object has relative velocity as 27.67m/s. By using the CFAR detection and cluster analysis, we can get more concise information of moving and static objects.

After applied the CFAR methods, we can get the range and velocity information of 'targets', the results are showing in Fig.(6.2) and Fig.(6.3). These two approaches

set with same parameters: guard cell window as [2,2], training cell window as [5,10] and P_{FA} as 10^{-5}. Compare these two figures, CA-CFAR threshold level is higher than OS-CFAR, and it shows strong mask effect where multiplier high peaks appear, some small peaks are not being detected. On the other hand, the OS-CFAR has a relatively smooth threshold level, this approach didn't influenced by high peaks. We can get the same conclusion as the previous discussion, OS-CFAR approach seems be dominate in setting detection statistics.

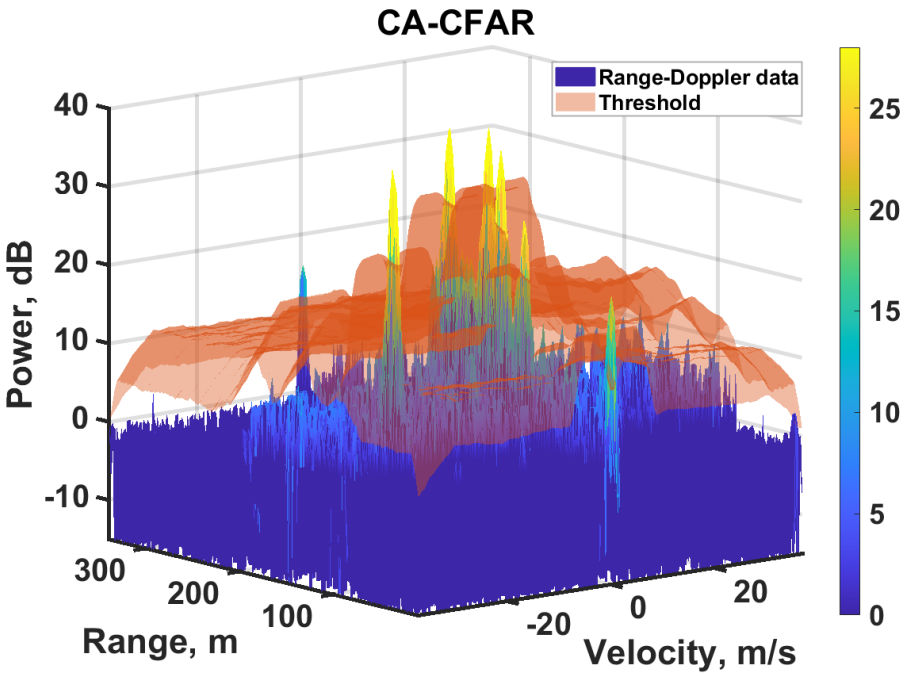

Figure 6.2: CA-CFAR Threshold Level in Range-Doppler Map

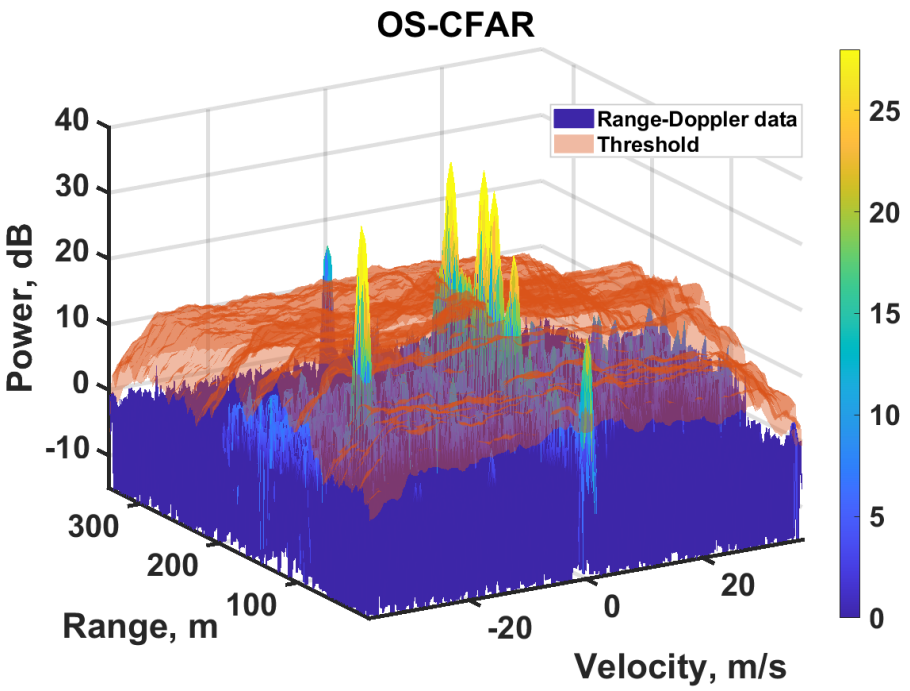

Figure 6.3: OS-CFAR Threshold Level in Range-Doppler Map

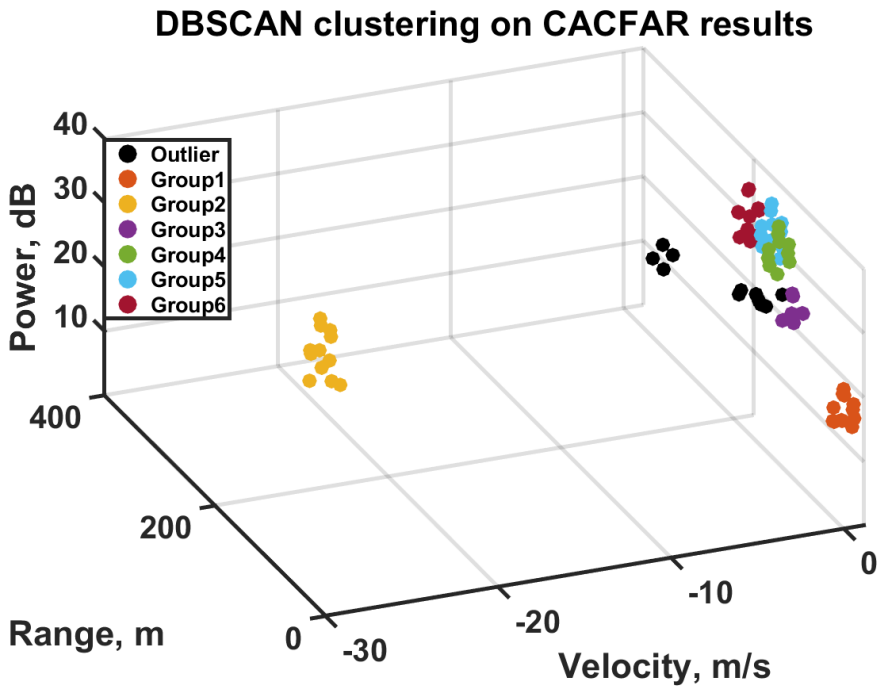

Figure 6.4: DBSCAN Clustering in CA-CFAR Results

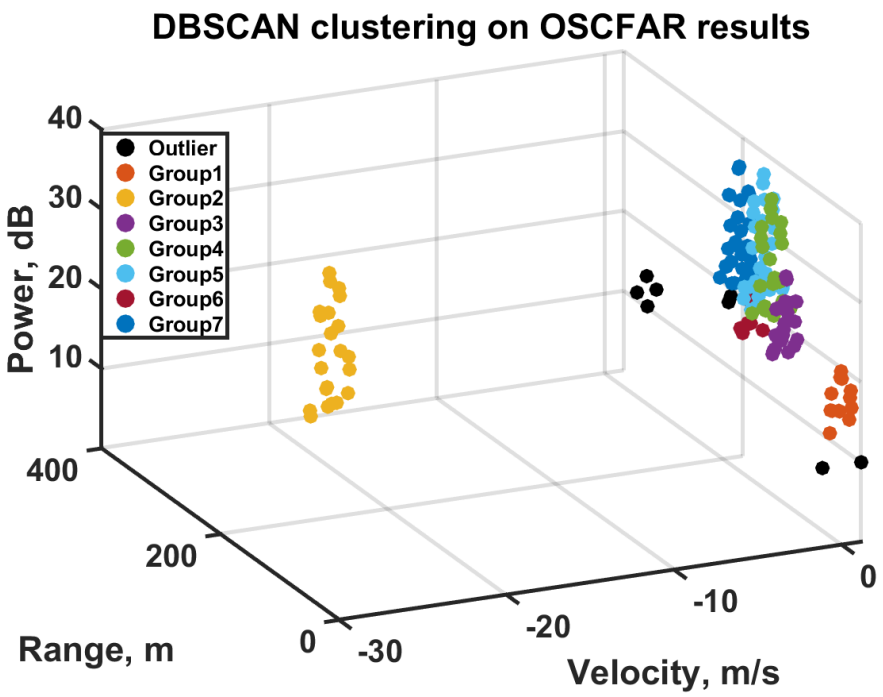

Figure 6.5: DBSCAN Clustering in OS-CFAR Results

To get the size and quantity information of targets and identify moving targets with all static object, we can continue applying clustering approach base on the CFAR results. Here we choose the *minpts* as 6 and radius ϵ as 4.5. The clustering results are given in Fig.(6.4) and Fig.(6.5). The clustering result from OS-CFAR has more number of groups than result from CA-CFAR, both results have some noise points labeled with -1. The database from OS-CFAR is larger than CA-CFAR, but also there are more points at the edge are labeled as noise. Compare the clustering result on these two database, there are one clustering groups missing in CA-CFAR results. OS-CFAR results have one cluster labeled 6 which has been identified as noise points in CA-CFAR results due to insufficient neighbors. Both data sets provide enough detected data points of moving targets to identify. Base on the comparison on these two methods, OS-CFAR results provide more information than CA-CFAR.

Based on the clustering results, CA-CFAR and OS-CFAR almost get share same performances. In the end, the number and the size of true targets are the most important to detection problems, we need to investigate more in the future with more experimental data.

Chapter 7

FUTURE WORK

In order to applied the algorithms into applications, we still need to tune the parameters until we find the most efficient sets of numbers can fit for most scenarios. In order to have more reliable performance estimation and comparison, I need to build the frequency modulated continuous wave Radar platform to provide more experimental data with known truth. In the future, we wish to investigate some intelligence algorithms like combining different CFAR methods, also we need to guarantee the low computational complexity. The current efficient OS-CFAR method still has a lot of room for improvements. We wish to explore some new sorting methods for OS-CFAR to avoid the repeat sorting problem. We are interested in CASH-CFAR algorithm and will study its performance in different environments and compared with OS-CFAR and CA-CFAR. Also, the computational complexity of clustering methods needs to be investigated, and we might need to find a more reliable and efficient clustering method.

Chapter 8

CONCLUSION

In the course of our research, we have studied two types of constant false alarm rate(CFAR) algorithms, cell averaging CFAR(CA-CFAR), and order statistic CFAR(OS-CFAR). Each approach performs very well in the homogeneous environment with Gaussian noise. CA-CFAR suffers major performative drop-off in the clutter evolved environments and data extends to a higher dimension. This approach has serious a masking effect which causes the missing targets and performance damage.

The OS-CFAR approach performs great in clutter evolved environments, it gets high probability of detection with low false alarm rate. The adjustment of the number of training cells and select rank can affect the performance a lot. However, this approach requires high power assumption due to repeat sorting in every sliding window.

The efficient implementations of CA-CFAR and OS-CFAR both achieve a quiet improvement on computational complexity, their big O and central processing unit(CPU) time has been decreased a lot and also performances are not deteriorated.

Considering the results from CFAR methods alone, the performance of OS-CFAR is much better than CA-CFAR no matter in which environment. But after applying the density-based spatial clustering of applications with noise (DBSCAN) methods to CFAR results from experimental data, results from OS-CFAR has more identify groups than CA-CFAR, but the performance of these methods still need to investigate with more experimental data.

Although CA-CFAR and OS-CFAR are not the best algorithms in detection statistics, and both two algorithms has defects in some scenarios. Compare with CA-CFAR, OS-CFAR approach is suitable for more complex environments.

REFERENCES

[1] Junwei Yan, Xiang Li, and Zhenhai Shao, "Intelligent and fast two-dimensional cfar procedure," in *2015 IEEE International Conference on Communication Problem-Solving (ICCP)*, Oct 2015, pp. 461–463.

[2] A. Jalil, H. Yousaf, and M. I. Baig, "Analysis of cfar techniques," in *2016 13th International Bhurban Conference on Applied Sciences and Technology (IBCAST)*, 2016, pp. 654–659.

[3] S. Srinivas, Y. Rong, and D. W. Bliss, "Uwb radar cardiac activity sensing: Novel arctangent demodulator for direct-rf receivers," in *2020 IEEE International Radar Conference (RADAR)*. IEEE, 2020, pp. 984–989.

[4] Y. Rong, S. Srinivas, A. Venkataramani, and D. W. Bliss, "Uwb radar vibrometry: An rf microphone," in *2019 53rd Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2019, pp. 1066–1070.

[5] Y. R. Rong, S. Srinivas, H. Chu, H. Yu, K. Liu, and D. W. Bliss, "Respiration and cardiac activity sensing using 3-d cameras," in *2020 54th Asilomar Conference on Signals, Systems, and Computers*. IEEE, 2020.

[6] L. Scharf and C. Demeure, "Statistical signal processing : detection, estimation, and time series analysis," 1991.

[7] K. H. Zou, A. J. O'Malley, and L. Mauri, "Receiver-operating characteristic analysis for evaluating diagnostic tests and predictive models," *Circulation*, vol. 115, no. 5, pp. 654–657, 2007.

[8] L. Andrews, *Special Functions of Mathematics for Engineers*, ser. Oxford science publications. SPIE Optical Engineering Press, 1998. [Online]. Available: https://books.google.com/books?id=qVBxQgAACAAJ

[9] A. Mohr, "Quantum computing in complexity theory and theory of computation," *Carbondale, IL*, 2014.

[10] H. Finn, "Adaptive detection mode with threshold control as a function of spatially sampled clutter level estimates," 1968.

[11] C. Kim and H. S. Lee, "Analysis of the generalized order statistics constant false alarm rate detector," *ETRI Journal*, vol. 16, 04 1994.

[12] H. Rohling, "Ordered statistic cfar technique - an overview," in *2011 12th International Radar Symposium (IRS)*, 2011, pp. 631–638.

[13] M. S. Greco and S. Watts, "Chapter 11 - radar clutter modeling and analysis," in *Academic Press Library in Signal Processing: Volume 2*, ser. Academic Press Library in Signal Processing, N. D. Sidiropoulos, F. Gini, R. Chellappa, and S. Theodoridis, Eds. Elsevier, 2014, vol. 2, pp. 513 – 594. [Online]. Available: http://www.sciencedirect.com/science/article/pii/B9780123965004000119

[14] K. Siddiq and M. Irshad, "Analysis of the cell averaging cfar in weibull background using a distribution approximation," in *2009 2nd International Conference on Computer, Control and Communication*, 2009, pp. 1–5.

[15] S. Park and A. Bera, "Maximum entropy autoregressive conditional heteroskedasticity model," *Journal of Econometrics*, vol. 150, pp. 219–230, Jun. 2009.

[16] A. Kızılersü, M. Kreer, and A. W. Thomas, "The weibull distribution," *Significance*, vol. 15, no. 2, pp. 10–11, 2018. [Online]. Available: https://rss.onlinelibrary.wiley.com/doi/abs/10.1111/j.1740-9713.2018.01123.x

[17] D. McAlister and F. Galton, "Xiii. the law of the geometric mean," *Proceedings of the Royal Society of London*, vol. 29, no. 196-199, pp. 367–376, 1879. [Online]. Available: https://royalsocietypublishing.org/doi/abs/10.1098/rspl.1879.0061

[18] T. Cormen, T. Cormen, C. Leiserson, I. Books24x7, M. Press, M. I. of Technology, R. Rivest, M.-H. P. Company, and C. Stein, *Introduction To Algorithms*, ser. Introduction to Algorithms. MIT Press, 2001. [Online]. Available: https://books.google.com/books?id=NLngYyWFl_YC

[19] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise." E. Simoudis, J. Han, and U. M. Fayyad, Eds. AAAI Press, 1996, pp. 226–231. [Online]. Available: http://dblp.uni-trier.de/db/conf/kdd/kdd96.html#EsterKSX96

[20] E. Schubert, J. Sander, M. Ester, H. Kriegel, and X. Xu, "Dbscan revisited, revisited: Why and how you should (still) use dbscan," *ACM Trans. Database Syst.*, vol. 42, pp. 19:1–19:21, 2017.

[21] [Online]. Available: https://www.metawave.co/