SAR TARGET DETECTION ALGORITHMS ON LINEAR SIMD ARRAYS

William Phillips and Rama Chellappa

Department of Electrical Engineering and Center for Automation Research
University of Maryland at College Park
College Park, MD 20742
(wphillip@cfar.umd.edu)

ABSTRACT

Constant False Alarm Rate (CFAR) detection in Synthetic Aperture Radar (SAR) is the first step in most ATR and image exploitation systems. In this paper several CFAR algorithms and their implementation on a 1-D SIMD array processor are investigated. We primarily focus on CFAR algorithms using the Weibull clutter model, but algorithms assuming K-distributed clutter should have similar implementations and runtimes. We show that high resolution SAR requires reference windows much larger than those used in traditional search radars, which permits fast moment based estimation instead of the computationally intensive maximum likelihood parameter estimates. We also extend a fast median filtering algorithm to the order statistic and censored CFAR algorithms. The running times of the CFAR algorithms are listed along with detection results using SAR imagery from the Northrop-Grumman TESAR sensor onboard the Predator unmanned aerial vehicle.

1. INTRODUCTION

SAR sensors are continuing to increase in popularity due to their all-weather and nighttime operation. In many SAR image exploitation systems, bright pixel detection using a CFAR detector is a critical first step. Listed below are some examples.

Automatic Target Recognition The first stage in a typical ATR system[7] is the prescreener stage. In this stage all potential targets should be detected. Subsequent stages of the ATR system eliminate natural and manmade clutter and then classify targets among some number of classes.

SAR Compression Image compression systems for SAR that compress target areas with high resolution and clutter at a lower resolution[8] are being investigated. Various "cuers" that identify the target regions are possible, ranging from a full ATR system to a much faster and simpler CFAR detector. In this application, the detector would be required to reside onboard an airborne sensor within severe size and power constraints. This application is becoming increasingly important as new wide area surveillance systems are developed that generate imagery at rates much greater than existing data links.

Site Model Construction When constructing site models from SAR[6], bright pixel detection is important for the identification of all manmade objects. For example, buildings typically have a strong linear streak facing the sensor with a shadow region down range from the sensor. Building detection algorithms can utilize the streak in the CFAR output along with shadow segmentation to identify buildings.

As these examples show, detection algorithms play an important role in most SAR image exploitation systems. As SAR systems mature, more processing will need to be embedded in the sensor. ATR algorithms in smart weapons or image compression systems onboard extended range unmanned aerial vehicles(UAVs) must reside in a very small physical space, dissipate very low amounts of power, and process imagery in real time at the sensor's data rate. This paper directly addresses these issues by implementing several different CFAR algorithms on a high performance parallel computer that is very efficient in terms of physical size and power.

2. PROCESSOR SELECTION

Our research addresses the requirements of an embedded computer architecture and algorithms able to effectively detect targets in SAR imagery in real time within the size and cost constraints imposed by the sensor's airframe. The choice of an embedded processing architecture is based on the following attributes: (1)physical size and power, (2)flexibility, (3) cost, and (4)processor throughput. Various designs suggested for Radar CFAR detectors include fixed function VLSI systolic arrays[4] which generally optimize size and power and programmable parallel processors[2] offering greater flexibility and low cost when applied to multiple applications. When flexibility is needed to support a variety of algorithms in different types of systems, flexibility is the most desirable attribute, and the choice is between SIMD and MIMD processors. A generally accepted fact is that the highly parallel, fine grained algorithms used for automatic target detection and other low level computer vision tasks fit well into the SIMD domain.

The final choice for the embedded target detection processing architecture is the interconnection topology. We feel that a one dimensional SIMD array is a nearly optimal

This work was supported in part by the Northrop-Grumman 1996 Advanced Study Award and an AASERT award from the Air Force Office of Scientific Research under the contract F49620-96-1-0264

Copyright 1997 IEEE

choice for size constrained embedded target detection processing. A detailed examination of the advantages of SIMD linear arrays in image processing was recently published by Hammerstrom[1].

3. CFAR ALGORITHMS

Traditional radar CFAR detectors are based on a Gaussian assumption for clutter statistics which leads to a Rayleigh distribution for the clutter in the magnitude image. High resolution SAR imagery exhibits a spikiness that is not a good match for the Rayleigh distribution[5]. Candidates for the clutter model of high resolution SAR are the two parameter Weibull, K, and Lognormal distributions. We use the Weibull distribution as our clutter model, which includes the Rayleigh distribution as a special case. The Weibull pdf is given by

$$p_x(x) = \frac{C}{B}\left(\frac{x}{B}\right)^{C-1} \exp\left[-\left(\frac{x}{B}\right)^{C}\right]; \; x \geq 0 \qquad (1)$$

where B is the scale parameter and C is the shape parameter. The Weibull pdf reduces to the common Rayleigh pdf when $C = 2$. The Weibull moment ratio

$$\frac{E[x^2]}{E^2[x]} = \frac{\Gamma(1+\frac{2}{C})}{\left[\Gamma(1+\frac{1}{C})\right]^2} \qquad (2)$$

is independent of B and can be used for estimating C.

CFAR Target detection based on a Weibull assumption can be carried out in a variety of ways, including the cell averaged CFAR(CACFAR) which uses the average of the clutter samples to form the adaptive threshold, and the order statistic CFAR(OSCFAR) which uses an order statistic of the clutter samples to compute the threshold. Each type of CFAR can be used with fixed or estimated shape parameter, C. In this paper, we consider several possibilities: (1) CACFAR with estimated and fixed C, (2) OSCFAR with estimated and fixed C, and (3) a censored CACFAR with estimated C. The reference clutter samples of the CFAR algorithms are taken from a hollow window surrounding the pixel under test with the inner window size set to be greater than the desired target.

The adaptive threshold for the Weibull CACFAR can be expressed[5] as

$$T_x = B[-\ln(P_{FA})]^{\frac{1}{C}} = \frac{E[x][-\ln(P_{FA})]^{\frac{1}{C}}}{\Gamma(1+\frac{1}{C})} \qquad (3)$$

where P_{FA} is the probability of false alarm. When the parameter C is fixed, the threshold computation is simply a constant times the estimate of the mean of the clutter samples. When C is to be estimated, the problem becomes much more difficult. Note that when a small number of samples are used, an additional factor is present in the threshold equation to achieve the desired P_{FA}. The estimation techniques discussed here are the maximum likelihood estimator(MLE) and the Weber-Haykin order statistic estimator discussed in [5] and its references, and a moment ratio estimator based on (2). The moment based estimator has been examined and is generally deemed unacceptable when using a small sample set of clutter. We have found that target

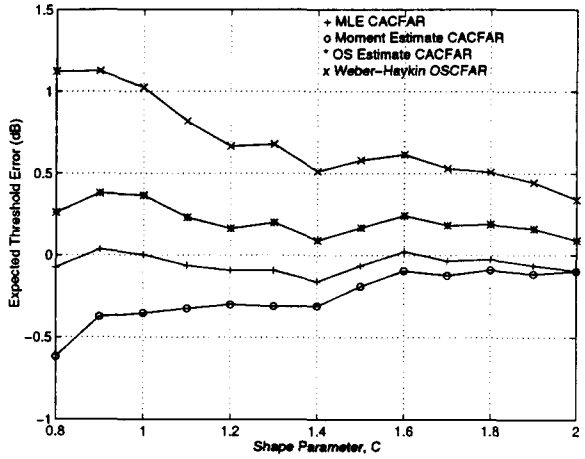

Figure 1: Threshold error of four Weibull CFAR algorithms relative to a perfect estimator. The results were obtained using 100 reference cells and 700 Monte Carlo trials.

detection in high resolution SAR imagery requires a large reference window to provide target isolation, which allows the use of a large number of reference cells without a noticable increase in processing time. When over 100 clutter samples are used, the moment based estimator is almost indistinguishable from the MLE as shown on Figure 1, which shows the threshold error in dB for several estimators given 100 reference cells. The MLE requires complex, iterative computations that are a poor match for the simple fixed point arithmetic units of the linear processor arrays we are targeting, and would greatly increase the running time of the algorithm. Real time ML estimation is not feasible using current processing architectures and the moment based estimator can be used with less than a 1 dB threshold error. Our algorithm computes the moment ratio of (2), and then uses the binary search algorithm to look up the threshold multiplier, $\frac{[-\ln(P_{FA})]^{\frac{1}{C}}}{\Gamma(1+\frac{1}{C})}$, in a table of moment ratios verses threshold multipliers.

The OSCFAR threshold is

$$T_x = x_k\left(\frac{x_j}{x_i}\right)^{\beta} \qquad (4)$$

where x_k is the k^{th} order statistic of the reference window samples. The parameter β can be precomputed from i, j, the number of reference cells, and P_{FA}[5]. This threshold requires three order statistics to be computed. Setting $i = k$ requires only two order statistics and is usually referred to as the Weber-Haykin algorithm.

Our parallel OSCFAR algorithms are based on the sliding histogram technique for fast median filtering presented in [3]. As the histogram moves, the required order statistic can be computed by simply updating the histogram and then walking up the histogram from the lowest bin until the required rank is found. The technique of tracking the number of samples less than the order statistic used in [3] can offer improvements when the size of the window is small compared to the number of histogram bins, but may not

Copyright 1997 IEEE

be parallelized correctly depending on the level of autonomy of the processing elements and the efficiency of the compiler. By tracking the number of samples less than the order statistic, we have observed up to a 15% improvement with some window and histogram sizes in our serial algorithms. If the number of samples is not small compared to the histogram size, tracking the order statistic may actually take longer than simply walking the histogram from the first bin. The constant factors in the histogram updating appear larger than the gain achieved by a faster search of the histogram. We have also considered order statistic trees[2] and a few other fast median algorithms[4]. Although some of these algorithms offer improved asymptotic performance, their higher constant factors render them ineffective for practical image and window sizes. Mapping onto a SIMD structure also requires an analysis of the processor autonomy.

The sliding histogram is also effective for the censored CACFAR algorithm. In the censored CACFAR, we use only the lowest L ranked samples of the reference window in the CACFAR algorithm. The censoring partially eliminates the masking effect of interfering targets in the reference window. We use a histogram technique where we compute running first and second moments as the histogram is traversed to find the desired ranks.

4. RUNNING TIMES ON THE SIMD ARRAY

We have implemented the CFAR detectors described in the previous section on a 256 processor CNAPS1[1] linear array shown in Figure 3. The CNAPS software was written in the data parallel C language, CNAPS-C, developed by Adaptive Solutions, Inc. No optimization was available with the CNAPS-C compiler. Square images with borders padded to produce results that are a multiple of the processor size were tested. The running times include loading the image into the processor array from the file DRAM on the CNAPS1 VME board and storing the results back to the DRAM and are listed in Table 1. For each algorithm we also show the speedup relative to a POWER2 thin node of an IBM SP2 using the most aggressive optimization available. An example image from the Northrop-Grumman TESAR sensor and the resulting CFAR images are shown in Figure 2.

When comparing the runtimes of the CNAPS1 to the POWER2 runtimes, some details need consideration. The CNAPS1 processor available for benchmarking is somewhat obsolete: it runs at only 20 MHz, has no optimizing compiler, and has no facility for parallel I/O. The POWER2 is a modern superscaler CPU running at 66.7 MHz with 2 floating point and 2 fixed point pipelined arithmetic units. An optimizing compiler tuned to the POWER2 architecture is available which has delivered speedups in the range of 3 to 6 on our CFAR algorithms. Given these factors, we would estimate that the single processor performance of the POWER2 node is roughly 40 times more powerful than a single CNAPS1 processing node when running compiled code. This factor of 40 is evident in the runtime table where speedups in the range of 3.6 to 5.8 were observed when using a 256 PN CNAPS1 system.

[1] CNAPS, CNAPS1, and CNAPS-C are trademarks of Adaptive Solutions, Inc., Beaverton, OR.

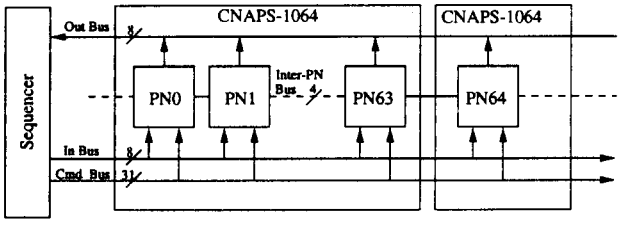

Figure 3: Block diagram of a basic CNAPS system. A detailed architectural description can be found in [1]. Our experiments use the 256 PN VME module.

5. CONCLUSIONS

The SIMD array has been shown to be an effective processing architecture for size constrained embedded CFAR processing. Testing on an obsolete CNAPS1 linear array obtained speedups in the expected range. A modern linear array such as the next generation CNAPS processor, the Motorola VCOMP, or the NEC ISP providing high clock speeds, some parallel I/O, and an optimizing compiler could be effectively used in an embedded processing system where airborne CFAR detection processing is needed at a high speed.

6. REFERENCES

[1] D. Hammerstrom and D. Lulich. Image processing using one-dimensional processor arrays. *Proc. IEEE*, 84(7):1005–1018, July 1996.

[2] D. Helman and J. JaJa. Efficient image processing algorithms on the scan line array processor. *IEEE Trans. on Pattern Anal. and Machine Intell.*, 17(1):47–56, January 1995.

[3] T. Huang, G. Yang, and G. Tang. A fast two-dimensional median filtering algorithm. *IEEE Trans. ASSP*, ASSP-27(1):13–18, February 1979.

[4] J. Hwang and J. Ritcey. Systolic architectures for radar CFAR detectors. *IEEE Trans. on Signal Processing*, 39:2286–2295, 1991.

[5] S. Kuttikkad and R. Chellappa. Non-Gaussian CFAR techniques for target detection in high resolution SAR images. *IEEE Proc. Int. Conf. Image Processing*, pages 910–914, 1994.

[6] S. Kuttikkad, R. Chellappa, and L. M. Novak. Building wide area 2-D site models from single- and multipass single-polarization SAR data. In *Proc. SPIE: Algorithms for Synthetic Aperture Radar Imagery III*, pages 34–44, Orlando, FL, April 1996.

[7] L. M. Novak, G. J. Owirka, and C. M. Netishen. Performance of a high-resolution polarimetric SAR automatic target recognition system. *The Lincoln Laboratory J.*, 6(1):11–24, 1993.

[8] D.L. Rodkey, S.P. Welby, and L. Hostetler. Clipping service: ATR-based SAR image compression. In *Proc. SPIE: Algorithms for Synthetic Aperture Radar Imagery III*, pages 390–396, Orlando, FL, April 1996.

Copyright 1997 IEEE

Detector Type	Image Size				Speedup
	256 x 256	512 x 512	768 x 768	1024 x 1024	
Rayleigh CACFAR	.0091	.035	.089	.174	3.6
Weibull CACFAR Moment Estimate	.029	.091	.227	.431	5.1
Rayleigh OS-CFAR	.099	.392	.941	1.83	5.5
Weibull OSCFAR WH Algorithm	.155	.627	1.47	2.77	5.8
Weibull Censored CACFAR	.207	.795	1.98	3.76	5.1

Table 1: Running Time in seconds for various image sizes on a 256 processor CNAPS1 systems. Note that the image sizes are the result sizes. The input image is larger by $k_1 - 1$ pixels, where k_1 is the outer window size . In this example we use a window of size 25 by 21 for a total of 184 reference cells. The speedup over a single POWER2 node of an IBM SP2 is listed for the 512 x 512 image size.

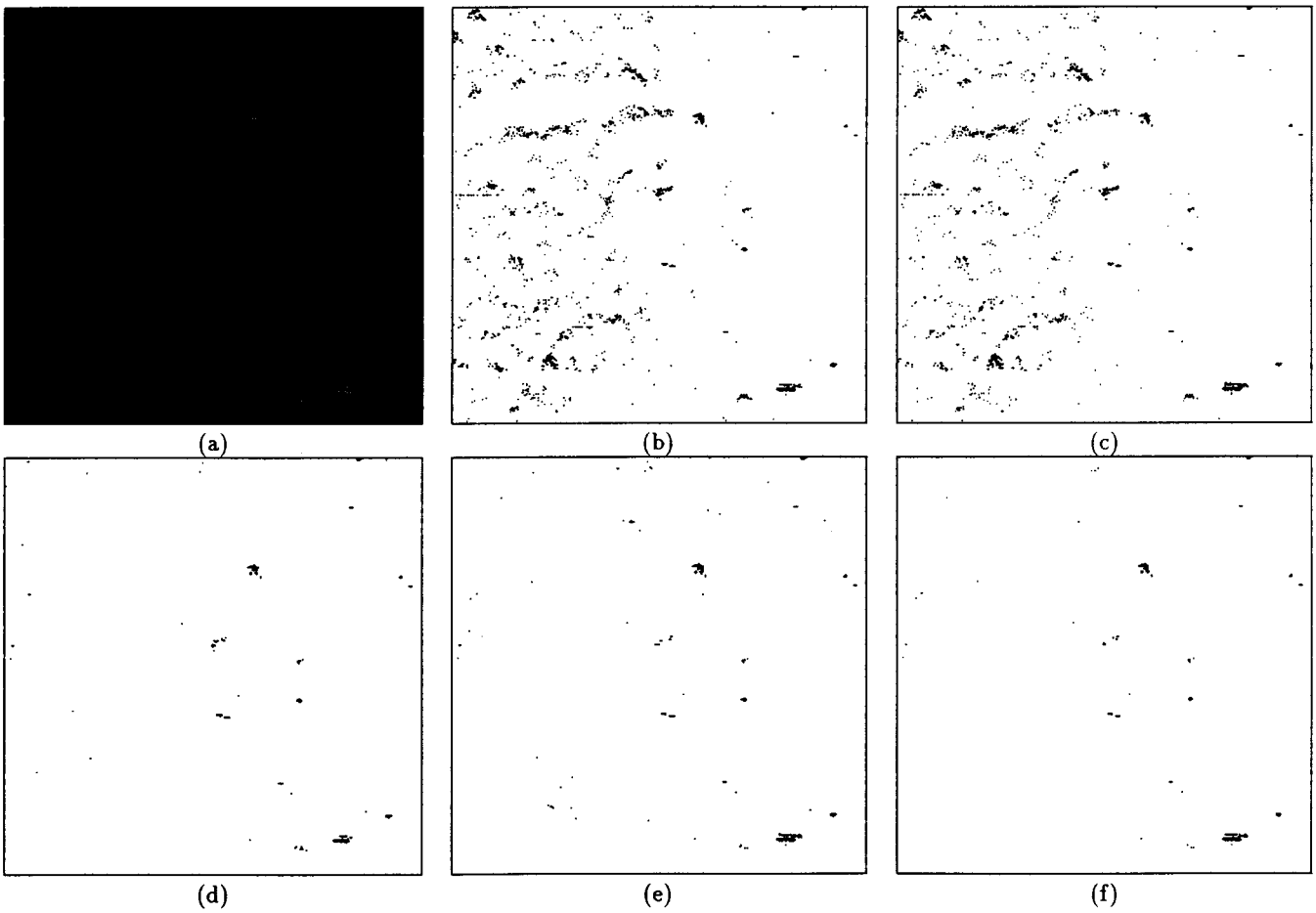

Figure 2: (a)Original TESAR image downsampled by two, and detected images using (b) Rayleigh CACFAR, (c) Rayleigh OSCFAR, (d) Weibull CACFAR with moment based shape estimation, (e) censored Weibull CACFAR using the lowest 95 % of the reference cells, and (f) Weibull OSCFAR. The algorithms used $P_{FA} = 10^{-4}$.

Copyright 1997 IEEE