SoftwareX 8 (2018) 39–42

Contents lists available at ScienceDirect

SoftwareX

journal homepage: www.elsevier.com/locate/softx

Original Software Publication

RmSAT-CFAR: Fast and accurate target detection in radar images

Fatih Nar [a], Osman Erman Okman [b], Atilla Özgür [c,*], Müjdat Çetin [d,e]

[a] Computer Engineering, Konya Food and Agriculture University, Konya, Turkey
[b] Electrical and Electronics Engineering, Konya Food and Agriculture University, Konya, Turkey
[c] Electrical Engineering, Başkent University, Ankara, Turkey
[d] Department of Electrical and Computer Engineering, University of Rochester, Rochester, NY, USA
[e] Faculty of Engineering and Natural Sciences, Sabanci University, İstanbul, Turkey

ARTICLE INFO

Article history:
Received 11 January 2017
Accepted 27 September 2017

Keywords:
Integral image
Summed area table
CFAR
Target detection
Rayleigh mixture
Adaptive simulated annealing
Parallelization
OpenMP
OpenCV

ABSTRACT

As the first step of automatic image interpretation systems, automatic detection of the targets should be accurate and fast. Constant False Alarm Rate (CFAR) is the most popular target detection framework for Synthetic Aperture Radar (SAR) images. For CFAR, modeling of the clutter is crucial since the decision threshold is calculated based on this model. We have developed a new target detection approach in which clutter is modeled using a Rayleigh mixture model that fits well to a variety of high-resolution SAR imagery. For computational efficiency, we use summed area tables (SAT) for computing background statistics. The resulting approach, called RmSAT-CFAR, is a promising general-purpose SAR target detection tool. This paper describes the open-source software for RmSAT-CFAR. Details of RmSAT-CFAR is published in the study named *Fast Target Detection in Radar Images using Rayleigh Mixtures and Summed Area Tables*. In addition to Rayleigh mixture and SATs, the software also uses tiling and parallelization to obtain faster and scalable results. This software repository also contains open source implementations for following algorithms: (a) Cell Averaging CFAR (CA-CFAR), (b) Automatic Censored CFAR (AC-CFAR), and (c) Adaptive and Fast CFAR (AAF-CFAR).

Code metadata

Current code version	0.1
Permanent link to code/repository used for this code version	https://github.com/ElsevierSoftwareX/SOFTX-D-17-00008
Legal Code License	MIT license (MIT)
Code versioning system used	mercurial
Software code languages, tools, and services used	C++, OpenCV 3.1, Visual C++ 2015
Compilation requirements, operating environments & dependencies	OpenCV, Visual C++ 2015
If available Link to developer documentation/manual	None
Support email for questions	ati.ozgur@gmail.com; fatihnar@hotmail.com;

Software metadata

Current software version	0.1
Permanent link to executables of this version	https://github.com/ati-ozgur/RmSAT-CFAR/releases
Legal Software License	MIT license (MIT)
Computing platforms/Operating Systems	Microsoft Windows
Installation requirements & dependencies	Microsoft Windows
If available, link to user manual — if formally published include a reference to the publication in the reference list	Rayleigh Mixture Summed Area Table CFAR (RmSAT-CFAR) [6]
Support email for questions	ati.ozgur@gmail.com; fatihnar@hotmail.com;

DOI of original article: http://dx.doi.org/10.1016/j.dsp.2017.09.015.
* Corresponding author.
E-mail address: ati.ozgur@gmail.com (A. Özgür).

https://doi.org/10.1016/j.softx.2017.09.005

1. Introduction

Synthetic aperture radar (SAR) is a radar technology that can produce high resolution images of the earth based on data collected by airborne or space-borne platforms. SAR images are used in wide variety of applications such as geoscience and climate change research, environmental and Earth system monitoring [1] because of (a) providing better resolution compared to conventional radars, (b) being independent of weather and day–night conditions. Three military origin terms – target, clutter, and noise – widely used in SAR applications [2]. Target is the object of interest, such as a vehicle on land or a ship in the ocean. Clutter is surrounding areas such as soil, sea, or forest. Noise is the imperfections in the SAR image [2]. Target Detection is the first step in all SAR Automatic Target Recognition (SAR-ATR) pipelines where targets are differentiated from clutter considering imperfections such as imaging noise. Probability of detection, probability of false alarm and speed of this step directly affect other steps in the pipeline. These three requirements are often in conflict with each other.

Sliding window based Constant False Alarm Rate (CFAR) methods are the most popular target detection methods in the literature [2]. Some well-known CFAR methods include CA-CFAR [3], AC-CFAR [4], and AAF-CFAR [5]. Methods differ from one another mainly in terms of how they model and learn the clutter model, as well as how they solve the computational problems involved in target detection.

We have developed a new target detection approach called RmSAT-CFAR [6] that learns and uses Rayleigh mixture densities for accurate clutter modeling over a variety of scenes and clutter types. Our approach also uses summed area tables (SAT) [7] for efficient computation of certain background statistics involved in the clutter model. For further computational gains, we use tiling and parallelization. In this paper, we describe the open-source software for RmSAT-CFAR [6]. Furthermore, our repository also contains our implementations of (a) Cell Averaging CFAR (CA-CFAR) [3], (b) Automatic Censored CFAR (AC-CFAR) [4], and (c) Adaptive and Fast CFAR (AAF-CFAR) [5]. Prototype version of these implementations are written for a commercial project and used in the previous study of authors [8]. These prototype implementations are further refined and improved; thus, these mature versions are used for comparison purposes in RmSAT-CFAR study [6].

Remainder of this paper is organized as follows: Section 2 gives overview of software, Section 3 illustrates examples, Section 4 explains the impact of the study, and finally Section 5 concludes the paper.

2. Software description

The overview of RmSAT-CFAR is given in Fig. 1. The input image is divided into tiles to achieve scalability to process big sized images. Then, the clutter model is obtained adaptively in each image tile using both in-tile and global statistics of the image. Using the learned Rayleigh mixture clutter model and SAT, all pixels are tested and labeled as target or clutter. Finally, all the tiles are merged to generate the final target map. All of this work flow is parallelized to speed-up the computations.

2.1. Parallelization

One of the major areas to evaluate the performance of a target detection approach is its computational speed [2]. Parallelization is commonly used approach to improve computational speed. Open Multi-Processing (OpenMP) [9] is a parallelization architecture supported in wide variety of processors and operating systems. Thus, we utilized OpenMP in the implementation. Due to SAT and OpenMP, RmSAT-CFAR has a detection speed below 4 s on a 100 mega-pixel SAR image outperforming its current competitors. SAT-CFAR, lightweight version of RmSAT-CFAR, can process the same 100 mega-pixel image in less than 2 s by losing a little bit accuracy. These tests are performed on a PC with Intel i7 3.5 GHz CPU, 4-cores, 8 hyper-threads, and 64 bits Windows 10.

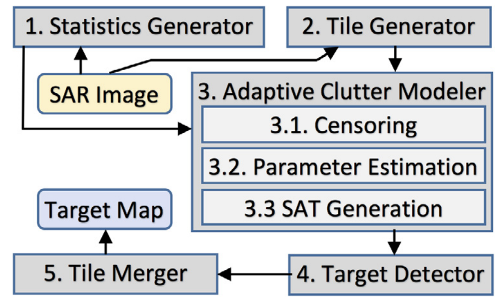


Fig. 1. Overview of RmSAT-CFAR.

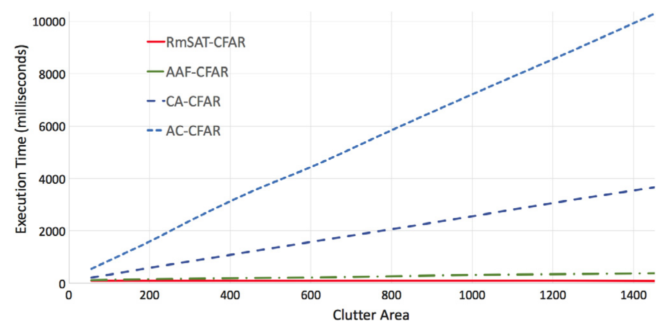


Fig. 2. Execution time (ms) versus clutter area (pixels).

Execution times of RmSAT-CFAR, CA-CFAR, AC-CFAR, and AAF-CFAR are presented in Fig. 2 which shows superiority of RmSAT-CFAR. Fig. 2 also shows that execution speed of RmSAT-CFAR is almost independent of clutter area which is important since SAR images increase in resolution; consequently increase in need to use larger guard and clutter regions.

Execution times for 320 and 1240 pixel clutter regions for all algorithms are presented for a 100 mega-pixel SAR image in Table 1, showing superiority of RmSAT-CFAR to other methods. Also, speedups from 1 thread to 8 threads shows parallelization performance, in Table 1.

2.2. Image tiles

In the first step of RmSAT-CFAR, image tiling [10] is used as a divide and conquer approach. In this approach, image is divided into equal sized sub-images, patches (see [6]). These patches are processed and then these results are merged to one overall result.

2.3. Adaptive clutter modeling

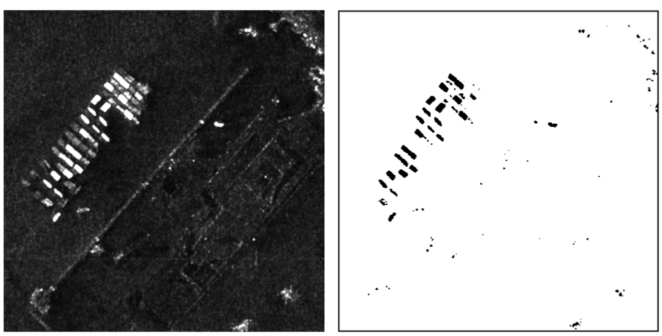
CFAR algorithms use a sliding window approach to find targets in SAR image. As the defined window slides, the clutter is modeled using the pixels inside the window with a statistical distribution which can be G^0, Gamma, Rayleigh, etc. [11]. If the pixel under test is at the end of the tail of the distribution, it is labeled as target (anomaly). In this study, the clutter is modeled by Rayleigh mixtures [12]. The reasons for this choice are given in main RmSAT-CFAR study [6]. The reflectivity of pixel under test is compared with the clutter distribution, leading to a decision on whether this is a target or a clutter pixel. To achieve scalability, to speed up

Table 1
Execution time (in seconds)

Clutter area	1 thread	2 threads	4 threads	8 threads	Speedup 8 vs 1
320 pixels					
SAT-CFAR	6.71	3.54	2.12	1.62	4.14
RmSAT-CFAR	14.63	7.91	4.29	3.90	3.75
AAF-CFAR [5]	18.95	10.06	5.97	5.30	3.58
CA-CFAR [3]	93.95	47.55	27.21	19.22	4.89
AC-CFAR [4]	255.56	128.94	69.98	46.78	5.46
1240 pixels					
SAT-CFAR	6.92	3.62	2.12	1.69	4.09
RmSAT-CFAR	14.79	8.02	4.71	4.07	3.63
AAF-CFAR [5]	34.17	19.90	11.59	10.06	3.40
CA-CFAR [3]	333.18	173.78	87.47	68.42	4.87
AC-CFAR [4]	930.54	465.94	246.45	159.46	5.84

Table 2
SAT versus Naive approach for summation over grid.

	SAT	Naive
Space complexity	$O(I_A)$	$O(1)$
Time complexity for range sum query	$O(1)$	$O(R_A)$
Time complexity for updating a value in table	$O(I_A)$	$O(1)$



(a) Sample SAR image. (b) Target map for 3(a).

Fig. 3. RmSAT-CFAR Target detection result.

this process, and to obtain a good accuracy, RmSAT-CFAR features: (1) Censoring, (2) Finding parameters of Rayleigh mixtures using Adaptive Simulated Annealing (ASA), (3) Utilization of SAT, (4) and image tiling approach where more details are given in [6].

2.3.1. Censoring mechanism

Details of this censoring mechanism can be found in [6].

2.3.2. Finding parameters of Rayleigh mixtures

We use ASA, a widely used heuristic optimization algorithm for global optimization, [13] to estimate the parameters of Rayleigh Mixture distribution. In this study, ASA parameters (see appendix of [6]), tile image, and maximum number of components in the Rayleigh mixture (M_{max}) are inputs and the number of mixture components (M_a) and corresponding interval parameters (d) are outputs of the ASA.

An independent ASA library is implemented in this study. C++ class file, AdaptiveSimulatedAnnealing, can be found in project repository, see Code metadata. To test ASA code, 19 different nonlinear cost functions are also implemented and given in NonlinearTestCostFunctions file. Definitions for these cost functions can be found in [14]. In ASA, instead of standard C++ random generation function, a fast random number generator, Mersenne Twister (MT) [15], is used. MT generates faster and better pseudo-random numbers compared to standard C++ pseudo-random generator.

2.3.3. Summed area table

Summed Area Table (SAT) [7] is a data structure and an algorithm used for efficient calculation of sum of the values in a rectangular window in an image. With a naive approach, this process has a linear computational complexity, while SAT performs the same operation in constant time. Thus, statistical moments such as mean or variance can be calculated very quickly using SAT. Since estimation for Rayleigh parameters requires sum of square of the reflectivities, SAT generates efficient computation model where target decision for each pixel becomes a constant time operation, Table 2. Details of the complexity analysis for SAT can be found in [6].

2.4. Target detector

In this step, sliding-window CFAR approach is used for target detection. For each pixel, parameters of Rayleigh mixture are calculated using SATs in an efficient manner. Then reflectivity of pixel is compared with the obtained Rayleigh mixture using CFAR to make a target decision [6].

3. Illustrative examples

Fig. 3 shows a sample target detection result of RmSAT-CFAR method. More RmSAT-CFAR results and detection performance analysis can be found in [6].

4. Impact

Since automatic target detection is the first step of SAR image analysis pipeline, such as automatic target recognition (ATR), accurate and fast target detection is crucial. The proposed method [6] is an accurate and fast target detector due to employed Rayleigh mixture model and SAT. Moreover, the algorithm is scalable and parallelizable; therefore, an efficient implementation of RmSAT-CFAR is applicable even for real time scenarios. For instance, RmSAT-CFAR allows real time target detection in strip mode airborne SAR system whilst image is formed.

In this study, RmSAT-CFAR is also compared with baseline and state-of-the-art approaches. Accordingly, we introduce a software repository that contains two baseline CFAR methods (CA-CFAR [3], AC-CFAR [4]) and two state-of-the-art methods AAF-CFAR [5] and RmSAT-CFAR [6]. To the best of the authors' knowledge, there are no open source software repositories that contains easy to use and reproducible CFAR implementations. Most of the CFAR implementations are in Matlab thus they are not suitable for speed comparisons. Since our repository provide efficient and parallelized C++ codes, future comparisons with baseline methods would be fair. Therefore, this repository and accompanying article would be a reference for further CFAR studies. In the future, other CFAR implementations will be added to this software repository.

5. Conclusions

In this paper, a novel target detection approach for radar images is presented that uses Rayleigh mixtures for background modeling. Utilization of SAT enables the estimation of the Rayleigh parameters in a computationally efficient manner. Employed tiling mechanism guarantees scalability of the method. Also, since this article introduces a open source software repository for baseline and state-of-the-art CFAR methods, it may be a reference study in the future CFAR studies.

References

[1] Moreira A, Prats-Iraola P, Younis M, Krieger G, Hajnsek I, Papathanassiou KP. A tutorial on synthetic aperture radar. IEEE Geosci Remote Sens Mag 2013;1(1).
[2] El-Darymli K, McGuire P, Power D, Moloney C. Target detection in synthetic aperture radar imagery: a state-of-the-art survey. J Appl Remote Sens 2013;7(1).
[3] Novak LM, Owirka GJ, Brower WS, Weaver AL. The automatic target-recognition system in SAIP. Linc Lab J 1997;10(2).
[4] Farrouki A, Barkat M. Automatic censoring CFAR detector based on ordered data variability for nonhomogeneous environments. IEE Proc, Radar Sonar Navig 2005;152(1):43–51.
[5] Gao G, Liu L, Zhao L, Shi G, Kuang G. An adaptive and fast CFAR algorithm based on automatic censoring for target detection in high-resolution SAR images. IEEE Trans Geosci Remote Sens 2009;47(6):1685–97.
[6] Nar F, Okman OE, Özgür A, Çetin M. Fast target detection in radar images using rayleigh mixtures and summed area tables. Digit Signal Process 2017 [Accepted for publication].
[7] Crow FC. Summed-area tables for texture Mapping. ACM SIGGRAPH Comput Graph 1984;18(3):207–12.
[8] Nar F, Demirkesen C, Okman OE, Cetin M. Region based target detection approach for synthetic aperture radar images and its parallel implementation. In: SPIE defense, security, and sensing. International Society for Optics and Photonics; 2012. p. 83940O.
[9] Chapman B, Jost G, van der Pas R. Using OpenMP portable shared memory parallel programming. The MIT Press; 2007.
[10] Patel M. A memory-constrained image-processing architecture. Dr.Dobb's J 1997.
[11] Gao G. Statistical modeling of SAR images: A survey. Sensors 2010;10(1):775–95.
[12] Siddiqui M. Statistical inference for Rayleigh distributions. J Res Natl Bur Stand D 1964;68(9).
[13] Ingber L, et al. Adaptive simulated annealing (ASA): Lessons learned. Control Cybern 1996;25:33–54.
[14] Wikipedia, Test functions for optimization, https://en.wikipedia.org/wiki/Test_functions_for_optimization. 2016.
[15] Matsumoto M, Nishimura T. Mersenne twister: A 623-dimensionally Equidistributed uniform pseudo-random number generator. ACM Trans Model Comput Simul 1998;8(1):3–30.