

FPGA Prototyping of a High-Resolution TerraSAR-X Image Processor for Iceberg Detection

Daniel Gregorek, Dominik Günzel, Jochen Rust, Steffen Paul
 Communication Electronics Group
 Institute of Electrodynamics and Microelectronics
 University of Bremen, Bremen, Germany
 {gregorek, guenzel, rust, stpa}@item.uni-bremen.de

Domenico Velotto, James Imber, Björn Tings, Anja Frost
 SAR Signal Processing
 Remote Sensing Technology Institute
 German Aerospace Center (DLR), Bremen, Germany
 {domenico.velotto, james.ember, bjoern.tings, anja.frost}@dlr.de

Abstract—The radar satellite TerraSAR-X monitors the Earth’s surface in a near-polar orbit. To provide the maritime community with up-to-date information on the presence and location of icebergs, the high-resolution radar images provided by the satellite are processed automatically by an image processing chain. A cell-averaging constant false alarm rate detector is used for iceberg detection, which has proven its usefulness for terrestrial object detection already. However, data transmission and processing require an impractical amount of time for real-time operation. In the present work, an FPGA-based hardware prototype of the detection algorithm is proposed, which accelerates image processing by a factor of ten compared to a software implementation and shows potential for further speed-up in the future.

Index Terms—FPGA, CFAR, SAR image processing

I. INTRODUCTION

The majority of icebergs present in the northern hemisphere are created when large chunks of ice break off marine glaciers in Greenland. Estimates of the number of icebergs that have their origin in Greenland range from 10,000 to 30,000 every year [1]. Icebergs are mainly carried by ocean currents but also by winds, and they may cross shipping lanes and interfere with offshore installations. In order to improve maritime safety, the German Aerospace Center (DLR) developed a new algorithm that detects and charts icebergs from images provided by the Synthetic Aperture Radar (SAR) satellite TerraSAR-X [2].

TerraSAR-X operates from a near-polar orbit at approx. 500 km altitude. Equipped with an active X-Band radar antenna, it is able to monitor oceans and frozen waters independent of cloud cover and fog, in any weather and sea state conditions, and independent of solar illumination. This capability is a decisive advantage, particular for high latitude locations, where shipping routes are infested with drifting icebergs. Different image products with varying spatial resolutions, scene sizes and polarizations are available for TerraSAR-X as listed in Tab. I. A common TerraSAR-X image taken in Stripmap mode has 500 million pixels to be processed. As an example, Fig. 1 shows a section of a TerraSAR-X Stripmap image capturing icebergs in open water.

The algorithm for automated iceberg detection [2] is based on a constant false alarm rate (CFAR) detector, which has proven its usefulness for ship detection from SAR data already [4] [5] [6] and has subsequently been applied to iceberg detection [7] [8] [9]. In addition to the standard approach, recurring patterns (i.e. waves) are surveyed to discriminate

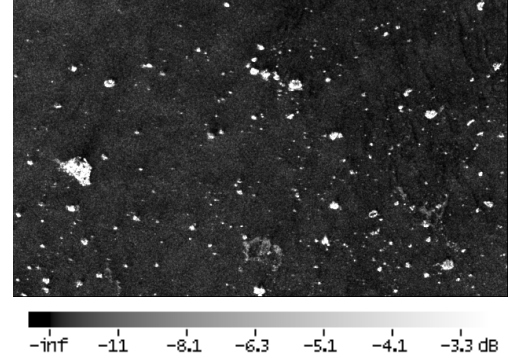


Fig. 1: SAR image example showing potential icebergs

icebergs from most false alarms that arise from rough seas and strong winds. The proposed algorithm has been implemented into the operational processing chain at the DLR ground station Neustrelitz server systems. Through the chain, the positions and size categories of icebergs (Tab. II) are output fully automatically after satellite data downlink, decryption, and L0-processing, i.e. SAR image generation from the raw data.

However, within the processing chain, the CFAR detector is one of the most latency critical operations. In order to provide the maritime community with information on the presence of icebergs as fast as possible, it is desirable to reduce the amount of downlink data and perform part of the processing on-board the satellite, including the iceberg detection.

This paper contributes an FPGA-based implementation for iceberg detection to reduce the overall computation time and the potential amount of downlink data. The FPGA implementation utilizes a highly-parallel and pipelined data path for CFAR detection.

The remainder of this paper is organized as follows: Sec. II outlines related work in the area of hardware implementations for CFAR detection. Sec. III presents our FPGA-based hardware implementation of a CFAR detector. Sec. IV evaluates the performance and discusses our results. Finally, Sec. V concludes the paper.

II. RELATED WORK

Previous hardware implementations for CFAR processing only deal with lower resolutions and smaller windows sizes

TABLE I: Specifications of common TerraSAR-X imaging modes [3]

Mode	Stripmap, single polarized, radiometrically enhanced	Stripmap, single polarized, spatially enhanced	Stripmap, dual polarized, radiometrically enhanced	Stripmap, dual polarized, spatially enhanced	ScanSAR, single polarized, radiometrically enhanced
Standard scene size	$30 \times 50 \text{ km}^2$	$30 \times 50 \text{ km}^2$	$15 \times 50 \text{ km}^2$	$15 \times 50 \text{ km}^2$	$100 \times 150 \text{ km}^2$
Resolution					
in near range (20°)	8.0 m	3.5 m	11.8 m	6.6 m	19.2 m
in far range (45°)	7.0 m	3.8 m	9.9 m	6.6 m	17.0 m
Effective no. of looks					
in near range (20°)	6.1	1.0	6.5	1.8	5.6
in far range (45°)	6.4	1.3	6.6	2.8	11.1
Polarizations available	HH or VV	HH or VV	HH+HH or HH+HV or VV+VH	HH+HH or HH+HV or VV+VH	HH or VV or HV or VH

TABLE II: Iceberg classification

Size	Height (m)	Length (m)
Growler	less than 1	less than 5
Bergy Bit	1-4	5-14
Small	5-15	15-60
Medium	16-45	61-122
Large	46-75	123-213
Very Large	Over 75	Over 213

compared to what is required for reliable iceberg detection. Cumplido et al. [10] utilize a 1D sliding windows which potentially limits accuracy for 2D image processing. A parallel approach was taken by Kyovtorov et al. [11], who accelerated the algorithm by considering multiple staggered 1D sliding windows. However, their approach exhibits a high degree of hardware redundancy and also is limited to a low resolution.

III. IMPLEMENTATION

A cell-averaging algorithm is applied to implement the CFAR detector. According to Fig. 2a, the so-called background window is applied to each pixel on the radar image, whereby a certain amount of so-called guard pixels near the tested pixel are excluded because icebergs of relevant sizes are usually larger than a single pixel. The intensities of the reference pixels in the hatched area are considered to obtain the statistics of the local background.

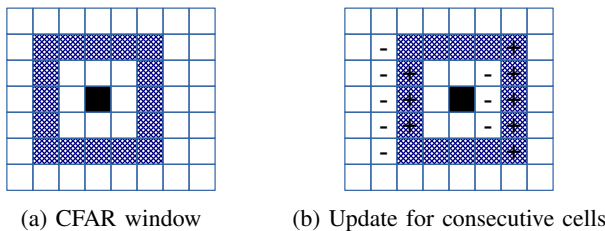


Fig. 2: Calculation of the CFAR background statistics.

For a faster calculation, it is assumed that the distribution of the background intensities is Gaussian [2]. In that case, the mean intensity μ_b and the standard deviation σ_b of the background pixels can be used to calculate a local threshold T according to:

$$T = \mu_b + k \cdot \sigma_b \quad (1)$$

If the intensity of the pixel under test itself exceeds this threshold, it is assumed that an object is detected. The CFAR-parameter k in equation (1) scales the threshold and therefore influences the false alarm rate. The parameter k should be set between 5 and 15 to achieve a good detection performance for icebergs [2]. To obtain the mean of the local background, the intensities of all the pixels have to be accumulated. Additionally, the squared intensities are accumulated for the standard deviation. As Fig. 2a shows, this is only necessary for the first target in each row of the image. For consecutive targets (Fig. 2b), the previous sums can be used by only subtracting the block of pixels marked with a minus (−) and adding the blocks with a plus (+). For practical large-sized background windows the amount of required operations and the accessed data is therefore reduced significantly.

The hardware implementation stores the image at an external memory and prefetches the required pixels into the internal BRAMs using an AXI interconnect. To maximize the throughput and minimize the image processing latency, M pixels are input simultaneously. For a high accuracy, the intensity of each pixel is stored as a 16 bit unsigned integer. The actual logic for calculating the CFAR threshold can read up to 256 reference pixels from the BRAMs in parallel. The required additions and subtractions to calculate the mean μ_b and the standard deviation σ_b for the threshold T are performed by pipelined multi-operand adder trees. We use two dedicated adder trees to separately compute the sum and the squared-sum of the background intensities. The standard deviation σ_b is obtained from the variance by using a pipelined CORDIC square root unit. Finally, a decision whether the target pixel intensity exceeds the local threshold is taken. The block diagram in Fig. 3 shows the core components of the implemented hardware detector.

IV. EVALUATION AND DISCUSSION

The implemented design is integrated on a Xilinx Zynq System on a Chip (SoC), which combines the flexibility of an ARM-based processing system (PS) with the programmable logic (PL) of an FPGA. We use the PS to transfer the image data from an SD-card to a DRAM which is connected to the PL. From the DRAM at the PL, the image data is read by the hardware CFAR detector, which performs the iceberg

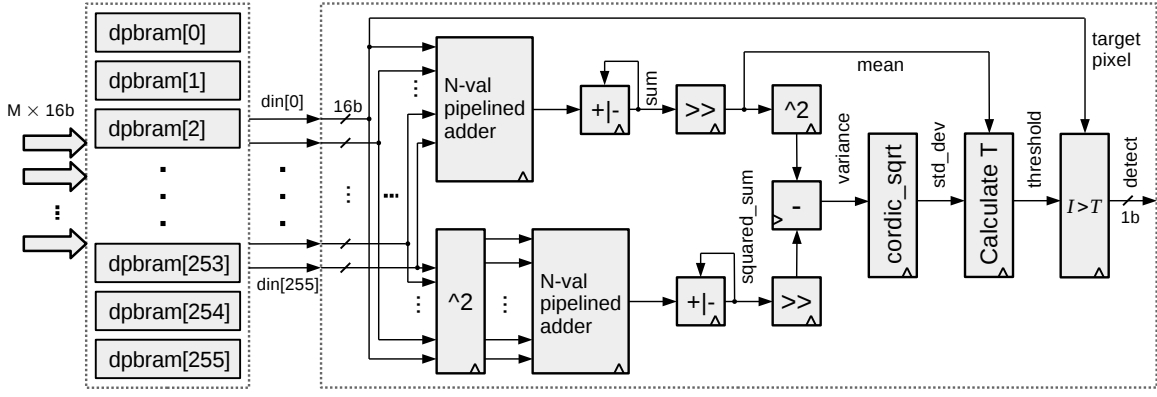


Fig. 3: Data path of the implemented hardware CFAR detector.

detection. Finally, the detection results are read from the DRAM by the PS and saved to a file at the SD-card.

On Zynq SoCs, communication between different hardware blocks is typically realized by the AMBA AXI4 protocol. Since both the ARM-processor and the hardware CFAR detector access the PL DRAM, a Xilinx AXI interconnect is required for arbitration. Additionally, the Xilinx DataMover IP core is used to facilitate accessing the DRAM with the CFAR detector. The block diagram in Fig. 4 shows the components of the implemented hardware CFAR detector and the general data flow in the system.

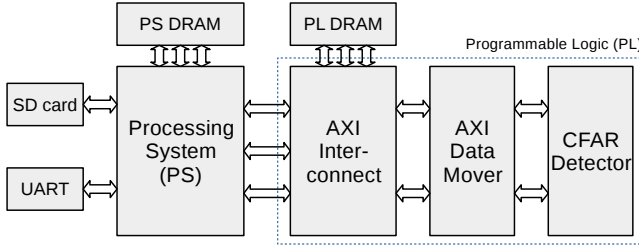


Fig. 4: Evaluation setup using a Xilinx Zynq XC7Z100.

The CFAR detector has been synthesized and implemented with a window size of 256×256 pixels and a guard window size of 128×128 pixels. The design is integrated on a Xilinx Zynq XC7Z100 SoC. The circuit requires 13.60% of the available slice lookup-tables and 3.92% of the registers on the target FPGA. 17.55% of the 755 block RAM tiles are needed for temporary data storage, while 12.82% of the DSP slices are used for accumulation and multiplication (see Tab. III). The circuit runs at a specified target clock frequency of 100 MHz.

TABLE III: Utilization of the implemented circuit on the XC7Z100 FPGA.

Site Type	Used	Available	Util (%)
Slice LUTs	37735	277400	13.60
LUT as Logic	36072	277400	13.00
LUT as Memory	1663	108200	1.54
Slice Registers	21734	554800	3.92
Registers as Flip Flop	21734	554800	3.92
Registers as Latch	0	554800	0.00
Slice Multiplexers	243	208050	0.12
Block RAM Tiles	132.5	755	17.55
RAMB36	4	755	0.53
RAMB18	257	1510	17.02
DSP48E1 Slices	259	2020	12.82

TABLE IV: Measured latency of the iceberg detection for different SAR-images. From the image dimensions and latency, the throughput D of the DataMover MM2S path for reading from the PL DDR3 can be calculated. The last column lists the throughput relative to the theoretical maximum of 800 MB/s.

Dim [px ²]	Latency [s]	MM2S D [MB/s]	% Max.
1500 x 1500	1.5	607	76
2994 x 2370	4.8	644	81
2610 x 3102	5.6	648	81
3696 x 2760	6.7	675	84
8984 x 5456	34.5	662	83
10440 x 16464	121.9	678	85
10359 x 18447	135.5	679	85

A 10440×16464 pixels large image is analyzed in 121.9 s. For the same image, the existing software algorithm requires twenty minutes when ten threads are executed in parallel. Thus, a speed-up of ten is achieved with the hardware CFAR detector in this case. Tab. IV lists the measured latency of the iceberg-detection for images with varying dimensions. In the current hardware implementation, the latency is limited by the prefetching of the pixel blocks into the BRAMs at the PL due to a bottleneck in the DataMover for accessing the AXI interconnect. By replacing the Xilinx IP core we expect a further speed-up by a factor of two. The last two columns in Tab. IV list the throughput D of the DataMover MM2S read path for transferring the background columns from the PL DDR3 DRAM to the CFAR detector. Due to overhead when issuing transfers, the theoretical maximum of 800 MB/s cannot be reached. However, It should be noted that the CFAR detector itself is not fully utilized because the throughput of the DataMover presents a bottleneck. While the latter achieves a theoretical data rate of 800 MB/s, the pipelined CFAR detector is able to process up to 10240 MB/s. For this reason, an alternative to the DataMover and a board with faster DDR4 memory may be used in the future to further speed up the detection.

The example in Fig. 5 shows the output binary masks created with both the software (left) and the hardware implementation (right). Pixels detected as objects are colored white. The evaluation of the output masks shows a high accuracy of the detection performance compared to the software implementation. Depending on the image tested, the hardware implementation detects between 88.7% and 97.5%

of the pixels that are detected by the software reference, while producing between 0.1% and 2.9% false positives.

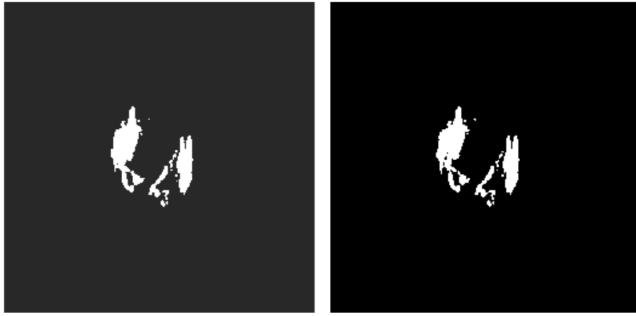


Fig. 5: Example of detected pixels in the binary masks created by the software (left) and hardware implementation (right).

Slight differences between both implementations are expected because the background and guard window sizes of the software implementation are one pixel smaller in both dimensions. Additionally, the hardware implementation uses integer numbers in contrast to floating point numbers at the software algorithm. The detection performance of the implemented prototype is sufficient because icebergs of relevant sizes occur as clusters of pixels, so that slight variations in the binary masks can be neglected.

V. CONCLUSION

In this work, a highly parallel data path of a CFAR processor for iceberg detection on high-resolution SAR images has been developed. The integration on a Xilinx Zynq XC7Z100 showed the feasibility to process hundreds of reference pixels in parallel at a clock frequency of 100 MHz. Analysis of a 10440×16464 large SAR image takes around two minutes on the hardware, which is ten times faster than a reference software implementation and potentially has several orders of magnitude lower power consumption. In comparison to the software reference, the hardware implementation has equivalent accuracy while improving computation time towards real-time capabilities.

REFERENCES

- [1] D. Diemand, "Icebergs.[w:] encyclopedia of ocean sciences," 2001.
- [2] A. Frost, R. Ressel, and S. Lehner, "Automated iceberg detection using high-resolution x-band SAR images," *Canadian Journal of Remote Sensing*, vol. 42, no. 4, pp. 354–366, 2016.
- [3] T. Fritz, M. Eineder, J. Mittermayer, B. Schättler, W. Balzer, S. Buckreiß, and R. Werninghaus, "TerraSAR-X ground segment basic product specification document," *cluster applied remote sensing, TX-GS-DD-3302*, pp. 1–103, 2008.
- [4] L. L. Scharf, *Statistical signal processing*. Addison-Wesley Reading, MA, 1991, vol. 98.
- [5] P. W. Vachon, J. Campbell, C. Bjerkelund, F. Dobson, and M. Rey, "Ship detection by the RADARSAT SAR: Validation of detection model predictions," *Canadian Journal of Remote Sensing*, vol. 23, no. 1, pp. 48–59, 1997.
- [6] S. Brusch, S. Lehner, T. Fritz, M. Soccorsi, A. Soloviev, and B. van Schie, "Ship surveillance with TerraSAR-X," *IEEE transactions on geoscience and remote sensing*, vol. 49, no. 3, pp. 1092–1103, 2010.
- [7] D. Power, J. Youden, K. Lane, C. Randell, and D. Flett, "Iceberg detection capabilities of radarsat synthetic aperture radar," *Canadian Journal of Remote Sensing*, vol. 27, no. 5, pp. 476–486, 2001.

- [8] R. Gill, "Operational detection of sea ice edges and icebergs using SAR," *Canadian journal of remote sensing*, vol. 27, no. 5, pp. 411–432, 2001.
- [9] R. Ressel, A. Frost, and S. Lehner, "Navigation assistance for ice-infested waters through automatic iceberg detection and ice classification based on TerraSAR-X imagery," *International Archives of the Photogrammetry, Remote Sensing & Spatial Information Sciences*, 2015.
- [10] R. Cumplido, C. Torres, and S. López, "On the implementation of an efficient FPGA-based CFAR processor for target detection," in *(ICEEE). 1st International Conference on Electrical and Electronics Engineering, 2004*. IEEE, 2004, pp. 214–218.
- [11] V. Kyovtorov, H. Kabakchiev, and G. Kuzmanov, "Power analysis of parallel CA-CFAR FPGA design," in *11-th International Radar Symposium*. IEEE, 2010, pp. 1–4.