11-2010

Power vs. Performance Evaluation of Synthetic Aperture Radar Image-Formation Algorithms and Implementations for Embedded HEC Environments (Ongoing Study)

Ricardo Portillo
The University of Texas at El Paso, raportil@miners.utep.edu

Sarala Arunagiri
The University of Texas at El Paso, sarunagiri@utep.edu

Patricia J. Teller
The University of Texas at El Paso, pteller@utep.edu

Comments:
Technical Report: UTEP-CS-10-48

Power vs. Performance Evaluation of Synthetic Aperture Radar Image-Formation Algorithms and Implementations for Embedded HEC Environments (Ongoing Study)

Ricardo Portillo, Sarala Arunagiri, and Patricia J. Teller
{raportil@miners.utep.edu; sarunagiri@utep.edu; pteller@utep.edu}
The University of Texas at El Paso
Department of Computer Science

Abstract

The continuing miniaturization and parallelization of processing hardware has facilitated the development of mobile and field-deployable systems that can accommodate terascale processing within once prohibitively small size and weight constraints. Unfortunately, the added computational capability of these small systems often comes at the cost of larger power demands, an already strained resource in these embedded systems. This study explores the power issues in a specific type of field-deployable system, Mobile Radar. Specifically, we focus on a computationally intensive phase of Synthetic Aperture Radar, Image Formation (IF), and evaluate performance tradeoffs in terms of time-to-solution, output image quality, and power consumption for two different implementations, single- and double-precision, of two different IF algorithms, one frequency-domain based and the other time-domain based. Preliminary results show that in some CPU-based instances single-precision IF leads to significant reductions in time-to-solution and, thus, total energy consumption (over 50%) with negligible but possibly acceptable SAR image output degradation. In the near future, this ongoing study will reevaluate these results, i.e., SAR IF power consumption vs. performance tradeoffs with more sophisticated IF workloads and output quality metrics, finer-grain performance and power measurement methodologies, and more computationally powerful embedded HEC devices, i.e., GPGPUs.

1. Introduction

For the past decade, power consumption has played a gradually increasing role in the design of run-time management systems for High-End Computing (HEC) environments. In particular, power is becoming the limiting factor for large HEC environments, such as computing clusters, and small HEC environments, such as embedded many-core systems [Amarasinghe, et al. 2009].

In the case of large HEC systems, power constraints arise from the cumulative effect of scaling to hundreds of thousands of processors, which require massive power and cooling infrastructures, as well as often prohibitively high utility costs. In the case of small embedded HEC environments, power constraints arise from scarcity, i.e., they often depend on self-contained power supplies that must operate within strict size and weight constraints and, thus, limit their capacity. This study focuses on these smaller embedded HEC environments.

Although embedded systems, in general, have always dealt with power as a rationed resource, embedded HEC systems have additional power demands because of their larger computational capacity. For example, the continuing miniaturization of processing elements has enabled terascale devices, such as GPGPUs, to fit into once prohibitively small enclosures. Although these specialized

terascale devices are more power efficient than their high-end general purpose counterparts, e.g., multi-core processors, their overall power draw can be twice that of a regular CPU, thus, further increasing the power demands on an already stretched resource.

Nevertheless, for embedded systems where functional capability is constrained by computational capacity, the advent of these small HEC devices greatly motivates the search for power management solutions that can circumvent, or at least ameliorate, their added power demands, thus enabling them to function in realistic power-limited scenarios.

One example where embedded HEC systems can lead to added functional capability is in the area of Mobile Radar systems. Previously, weight and space constraints greatly limited the amount of processing hardware and, thus, computational intensity of radar processing software. This often led to compromises, where less compute-intensive algorithms were used at the expense of radar imaging quality and flexibility. This has changed in recent years as feasibly small and powerful processing units now can execute applications based on more compute-intensive algorithms, such as backprojection-based Synthetic Aperture Radar (SAR) applications, within realistic time-to-solution requirements. However, Mobile Radar systems are largely self-contained and must run for long periods of time without any connection to an external power source. Thus, although radar processing power is no longer constrained by size and weight as in the past, power remains a limiting factor.

This study represents a first step towards the exploration of adaptive power management in such systems. It explores the potential effectiveness of application-initiated resource management techniques that may enable radar-processing software to use power more efficiently by switching between implementations that are functionally equivalent but that differ in terms of power vs. performance tradeoffs. Specifically, we define the goal of this study as follows:

Study Goal

Investigate the power consumption vs. performance, i.e., watt and energy usage vs. radar image fidelity and image formation time, of single-precision and double-precision implementations of two SAR Image Formation techniques, Fourier-based (frequency-domain based) and backprojection-based (time-domain based) techniques, which differ in computational intensity and potential radar imaging capability.

The remainder of this report is organized as follows. Section 2 describes Synthetic Aperture Radar and, in particular, the two main Image Formation (IF) techniques used in most SAR implementations. Section 3 describes the two SAR workloads, each containing a representative IF algorithm, which we use in this study along with their respective input datasets. Section 4 describes power management techniques in general and the specific techniques that are explored in this study. Section 5 describes our experimental methodology, including descriptions of the experimental environment, power measuring methodology, and performance evaluation metrics. Section 6 describes preliminary results, while Section 7 delineates pending investigations for this study. Finally, Section 8 wraps up the paper by presenting preliminary conclusions.

2. Synthetic Aperture Radar and Image Formation

Mobile Radar (MR) systems are used when the environment of interest is remote and/or non-static, e.g., in reconnaissance operations. Thus, an MR system requires an encapsulated radar system that can travel to the desired location. The required mobility of MR devices limits the size and weight of the system, including the size of the radar antennas and, thus, limits the antennas' sensing capability. To circumvent this limitation, MR implementations often use Synthetic Aperture Radar (SAR), which enables a system with relatively small radar antennas to acquire images comparable to those produced by prohibitively larger sensing equipment. SAR accomplishes this by reconstructing a large radar image from sets of sensor data, i.e., one set per antenna, which span the area of interest.

Although SAR can produce large, high-resolution radar images from manageably small sensors, a major drawback is that it requires extra processing power to construct the image from the composite set of sensor data – this is known as Image Formation (IF). Often, this makes IF the most compute-intensive phase of a SAR workload. For example, the mathematically ideal method for IF is the Matched Filtering approach, which has $O(N^4)$ complexity for a 2D image [Willis & Griffiths 2007]. Since the size and weight constraints of MR devices also limit the size and, thus, the processing capability of associated computing hardware, the choice of IF algorithm to be used in MR workloads historically has been limited by this processing capability.

The processing bottleneck of the IF phase of SAR has led to many algorithms that reduce the complexity of image formation while attempting to maintain image quality. These IF algorithms can be divided into two main categories: frequency- and time-domain approaches [Rahman 2010].

Frequency-domain IF methods, such as Range Migration, also known as ω-K [Milman 1993], reduce IF complexity by using Fast Fourier Transforms (FFT) to produce a frequency representation of sensor data that is much easier to process. Although this approach reduces IF time complexity from $O(N^4)$ to a more reasonable $O(N^2 \log_2 N)$ [Na, et al. 2006], it requires the radar sensor to capture data along a linear track and, thus, the output quality can be degraded by nonlinear mobility; this linear-tracking restriction also can pose a problem for ultra-wide-beam radar systems [Ulander, et al. 2003].

Unlike frequency-domain IF, *time-domain IF* methods, such as filtered-backprojection [Munson 1983], do not have linear-tracking restrictions on sensor data capture, however, they are more computationally demanding, i.e., on the order of $O(N^3)$ for a 2D image [Na, et al. 2006]. Although backprojection methods exist that run faster, i.e., on the order of $O(N^2 \log_2 N)$, the performance benefit comes at the cost of lower image quality [Basu & Bresler 2000]. In fact, these optimized backprojection methods can produce lower quality images compared to equally-fast frequency-domain methods [Hunter, et al. 2003]. (Note that previous work has extended the theoretical comparisons between frequency- and time-domain IF methods with empirical observations [e.g., Hunter, et al. 2003; Na, et al. 2006; Vu, et al. 2008]).

Thus, if we want to take advantage of the less restrictive sensor data specifications of the backprojection algorithm while maintaining image quality levels comparable to linear-tracking frequency-domain IF, it is apparent, in general, that the hardware should be powerful enough to execute $O(N^3)$ algorithms within realistic time-to-solution constraints for realistic problem sizes. Fortunately, some computational engines can achieve this task relatively well. For example, FPGA- and ASICS-based MR processors for Unmanned Aerial Vehicles such as the SRC-7 [SRC], output respectable performance numbers for SAR backprojection benchmarks [Pointer 2008]. However, such specialized devices require time-consuming and expensive design and/or implementation effort, and their re-programmability is either non-trivial or not possible. With respect to more general-purpose hardware, SAR backprojection algorithms are highly parallelizable and, thus, ideal for small massively parallel commodity terascale devices such General Purpose GPUs, i.e., GPGPUs, which are less expensive and more flexible than FPGA technologies [Neophytou, et al. 2007], but much more power demanding.

Accordingly, we are in the process of setting up an experimental environment in which to evaluate SAR backprojection workloads executed on GPGPUs in terms of time-to-solution, output image quality, and power consumption. In the meantime, this preliminary study, based on CPU, rather than GPGPU, measurements, evaluates these performance tradeoffs for two different implementations, single- and double-precision, of two different IF algorithms, one frequency-domain based and the other time-domain based. Although our future focus is on backprojection (time-domain based) algorithms, we include a frequency-domain based algorithm for comparison. The following section describes these SAR workloads and input datasets in detail.

3. Target SAR Workloads and Datasets

The two SAR workloads and associated input data sets are described below. The first is Scalable Synthetic Compact Application #3, SSCA3, a frequency-domain SAR workload, while the second is Unoptimized Filtered Backprojection, UFB, a time-domain SAR workload.

3.1 Frequency-domain SAR Workload – SSCA3

The Scalable Synthetic Compact Application #3 (SSCA3) workload [Bader, et al. 2006] belongs to a suite of benchmarks developed by DARPA's High Productivity Computing Systems (HPCS) project. The official release of SSCA3 is a sequential MATLAB implementation that represents a SAR workload consisting of four phases of execution:

1. **Data Generation:** Creates a synthetic SAR sensor dataset from the spotlight measurement of a simulated, uniform, lattice of point reflectors. The problem size of the dataset is determined by SSCA3's input parameters.
2. **Image Formation:** Takes the generated synthetic SAR dataset and performs ω-K Image Formation to construct the SAR image.
3. **Target Insertion:** Inserts target templates randomly across the constructed SAR image.

4. **Target Detection:** Applies a simplified Automatic Target Recognition scheme that compares the original SAR image with the target-populated version and produces a disparity map showing only the detected target templates.

Since we are only interested in IF methods for this study, we restrict our analysis to SSCA3's second phase of execution, which implements a frequency-domain ω-K Image Formation algorithm with compute complexity $O(N^2 \log_2 N)$.

3.2 Time-domain SAR Workload – Unoptimized Filtered Backprojection

Although there are several SAR backprojection implementations in the literature, e.g., [Neophytou, et al. 2007; Cordes 2009; Fasih & Hartley 2010; Park & Shires 2010], we chose the implementation described in [Gorham & Moore 2010] for this preliminary study. Written by researchers at Air Force Research Laboratory (AFRL), this code implements a simple, unoptimized filtered backprojection algorithm for SAR Image Formation with time complexity $O(N^3)$. In this study, we refer to this algorithm as the Unoptimized Filtered Backprojection algorithm or UFB for short.

We chose the UFB implementation for two main reasons: 1) It was readily available at the beginning of this study, thus, allowing us to get some early preliminary results (Section 7 describes our plans to analyze more sophisticated and realistic implementations). 2) UFB is a sequential implementation written in MATLAB. We found the latter preferable as it enabled us to study the backprojection implementation in the same execution environment as we are studying SSCA3, thus, giving us some potential space for comparison between the methods, notwithstanding their different datasets.

UFB can process two different SAR input datasets, both of which are publicly available through AFRL's Sensor Data Management System website [SDMS]:

1) "SAR-Based GMTI in Urban Environment Challenge Problem" [Scarborough, et al. 2009]: AFRL created this dataset to facilitate the development of Ground Moving Target Indicator (GMTI) systems. It consists of a 2-pass, 71-second portion of X-band SAR sensor measurements taken over a scene consisting of numerous buildings and civilian vehicles. The radar signals were transmitted and received horizontally, i.e., the radar performed in an HH polarization configuration.
2) "GOTCHA 2D/3D Volumetric Imaging Challenge Problem" [Casteel, et al. 2007]: The dataset consists of an 8-pass set of X-band SAR sensor measurements taken in circular mode over a parking lot and adjacent grass area. The data were captured in four radar polarization modes (HH, HV, VH and VV), each partitioned into 360 azimuth angle observations. This study ran the UFB workload for each polarization mode in this dataset. However, we restricted the data in each polarization input set to the 260-360 azimuth angle measurements so that the execution times were long enough for power measurements to stabilize, yet short enough to perform the experiments in a timely manner.

Table 1. Summary of target SAR workloads and datasets used in this study

Target SAR Workload	Image Formation Method	Computational Complexity	Input Dataset	Dataset Size
DARPA-SSCA3	Frequency-domain (ω-K)	$O(N^2 \log_2 N)$	Uniform point reflectors (synthetically generated)	488MB
AFRL-UFB	Time-domain (filtered-backprojection)	$O(N^3)$	AFRL GMTI Challenge (HH polarization)	2.95GB
			AFRL GOTCHA Challenge (230-360 azimuth range) (HH, HV, VH, and VV polarization input sets)	146MB (for each input set)

4. Power Efficiency Techniques

As discussed in Section 1, as weight and size gradually play a smaller role in limiting the computational power of embedded HEC systems, e.g., MR SAR processing systems, power consumption is a significant and growing concern. Since compute capacity is expected to outpace power capacity for the near future, intelligent means of increasing power efficiency will be the main approach for developing energy-feasible HEC systems [Brown and Reams 2010].

In general, there are two main ways of regulating power consumption and, thus, increasing the power efficiency of a computing system: 1) management of *system resource availability* and 2) management of *application resource requirements*. We discuss these two approaches in detail below.

4.1 Management of System Resource Availability

Restricting resource availability places the responsibility of power management on the shoulders of the computing system. In this scenario, the system enforces resource capacity limits to meet power consumption constraints. Current computing hardware and system software already provide many methods of restricting resource availability. These system-initiated power-saving functionalities can be placed into two categories [Liu and Zhu 2010]:

1) **Dynamic Resource Scaling**: Reduces *active power*, i.e., usage-based power consumption, by slowing/reducing the speed/intensity of a resource. Examples include reducing the frequency of a processor (known as Dynamic Voltage Scaling), network bandwidth, and/or antenna signal strength.
2) **Dynamic Resource Sleeping**: Reduces both active power and *passive power*, i.e., idle consumption due to power leakage, by hibernating or turning off a resource. Examples include turning off processors, processing cores, circuit components (known as Power Gating), memory banks, and/or disks.

These system-initiated techniques often are implemented as black-box solutions, where the system is largely unaware of the running application's resource characteristics and, thus, manages resource availability based largely on runtime monitoring and adaptation heuristics.

4.2 Management of Application Resource Requirements

Unlike the previous system-initiated approach, managing an application's resource requirements places the responsibility of power management on the shoulders of the running process. This approach to power management has received less attention, compared to the system-initiated approach; most adaptive application-initiated resource management techniques largely target execution time and output quality rather than power efficiency. Additionally, often these application-initiated approaches are case-by-case solutions that require significant effort on the part of the application developer and are not very generalizable to other applications. However, in the case of floating-point intensive workloads, there are two general ways of reducing power consumption that require relatively less in-depth knowledge of the target workload:

1) **Reduction of Floating-point Precision:** Currently, most general-purpose processor Floating-Point Units (FPUs) support both single-precision (32-bit) and double-precision (64-bit) floating-point operations in hardware. This gives us the option of selecting which precision to use in our workloads. Choosing to run in single-precision mode over double-precision is potentially beneficial, from a power-consumption perspective, in three ways:
    i. Since single-precision instructions do not use all the FPU hardware required for double-precision instructions [Preiss, et al. 2009], this means that, theoretically, they require less active power to execute and, thus, reduce overall power consumption as compared to double-precision operations.
    ii. If the hardware supports Power Gating technology, as mentioned in 4.1, single-precision execution allows the system to turn off the unused portions of the FPU, thus, reducing not only active power but passive power as well.
    iii. Single-precision instructions run faster than double-precision instructions, thus, even if for some reason active or passive power is not reduced, the overall energy consumption, which is the product of power and execution time, can decrease. Low energy consumption is ultimately what we strive for in a power-constrained environment.
2) **Removal of Floating-point Operations:** Most mainstream processors have different functional units that compute integer/logical operations (ALUs) and floating-point operations (FPUs) separately. In general, FPUs have a larger and denser transistor footprint than ALUs, making floating-point operations more power expensive than integer operations. Given this fact, special-purpose low-power processors often have been designed without FPUs, thus, requiring floating-point operations to be simulated in software using equivalent, but slower, integer operations, i.e., fixed-point arithmetic. We can take advantage of the large body of work on systematic floating-point-to-fixed-point code translation [Han 2006] to create fixed-point, low-power equivalents of floating-point algorithms enabling us to decide statically, or at runtime, which version to run given power and time-to-solution constraints.

In summary, power-efficiency techniques can be 1) system-initiated, where resource availability is restricted, or 2) application-initiated, where resource requirements are constrained. This study focuses on the latter approach by analyzing functionally equivalent single-precision,

double-precision, and, in the near future, fixed-point implementations of IF algorithms to explore their power vs. performance (including image quality) tradeoffs. The ultimate goal of this work is to implement an adaptive power management system that can switch between functionally equivalent IF implementations at runtime to maintain power consumption constraints in an MR environment. Although we concentrate on application-initiated power management in this study, we will also explore system-initiated techniques in the near future.

5. Experimental Setup

As mentioned earlier, this study is preliminary, focusing on CPU-, rather than GPU-, based measurements. Also, the experimental environment is very rudimentary, especially in comparison to the more powerful, massively parallel, computational environments associated with embedded HEC systems. Nonetheless, it has permitted us to gain experience with target (SAR IF) workloads and datasets, and as described below, it has allowed us to study functionally equivalent single- and double-precision, and, in the near future, fixed-point, implementations of frequency- and time-domain IF algorithms in terms of power vs. performance (including image quality). Shortly after the publication of this report, a more sophisticated experimental environment will be available to us: it will include sophisticated power measurement and capture equipment, as well as a GPGPU with floating-point capability (an NVIDIA Fermi-based GPU). Please refer to Sections 7.2-7.4 for more details.

5.1 Execution Environment

For this study, we ran the target SAR IF workloads and datasets in a mainstream Macbook Pro laptop environment. This system consists of a dual-core Intel Core Duo 2.53GHz processor with 30 watts of maximum power consumption, based on its Thermal Design Power (TDP) specification, and has 4GB of main memory.

5.2 Power and Energy Measurement

To measure execution power consumption, as shown in Figure 1, we plugged the laptop power adapter into a simple Kill-a-Watt power meter. This power meter displays watt usage at one-second intervals and has a measurement error margin of 0.2%. Since this simple power meter does not log power usage measurements across time, we recorded the power and workload output on video and later analyzed the recording to estimate average power consumption during the execution of the Image Formation IF) phase. Since this is a rather coarse-grain method of computing power consumption, we selected IF problem sizes that ran sufficiently long, e.g., on the order of hours, so that potential observation inaccuracies, within a margin of seconds or minutes, were amortized. As mentioned above, we are in the process of acquiring more sophisticated measuring equipment for the next phase of this study.

To reduce system perturbation of power measurements, we ran all experiments in Mac Safe Mode, which runs only essential kernel modules and disables all startup programs. Additionally, we manually disabled all non-essential third-party processes that were not automatically turned off by the Safe Mode boot routine.

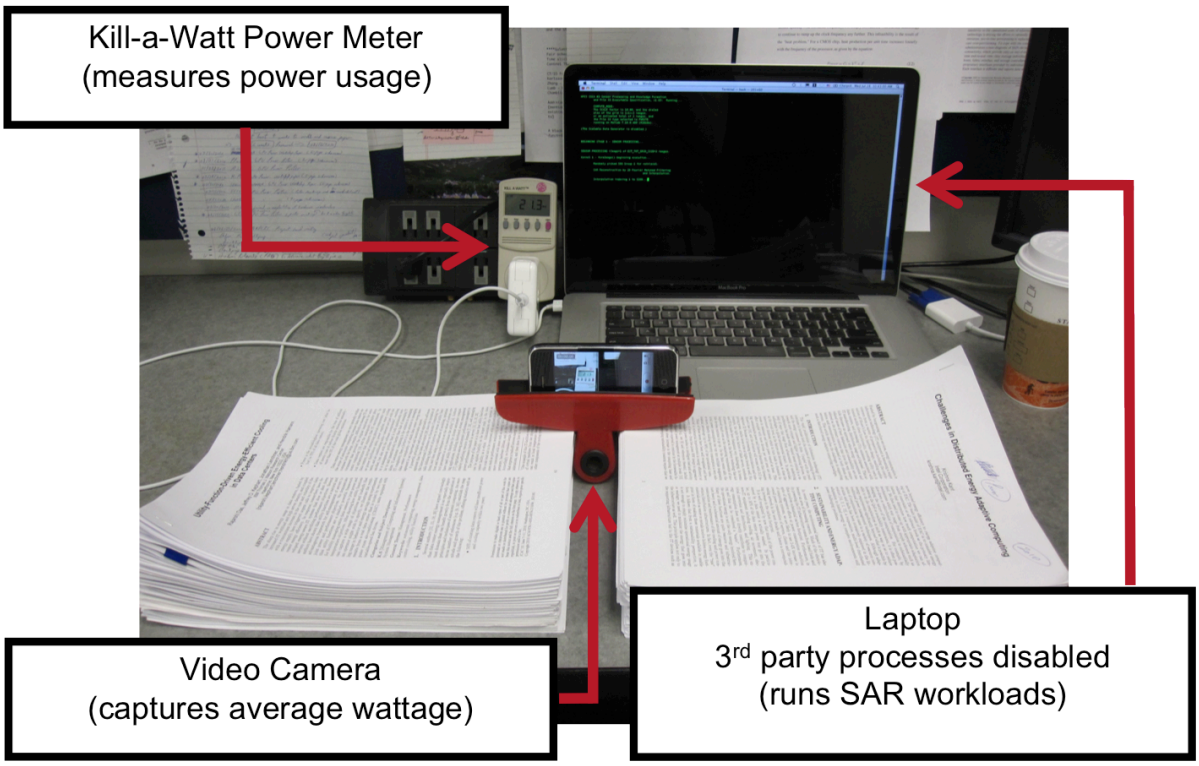

Figure 1. Preliminary experimental environment

Based on our observations, this simplified runtime configuration consumed around nine watts of power when idle. Thus, the average idle power, i.e., nine watts, was subtracted from all power usage measurements to compute power consumption attributed to the workload itself.

5.2 SAR Image Quality Evaluation

Since the goal of this study is to determine whether power reduction techniques can be applied to SAR IF phases without significant degradation of image quality, we investigated several different methods to evaluate SAR image output. In practice, SAR output images are handled by human observers or by image processing algorithms. In the former case, the perceptual quality of the image is important since it is processed by the Human Vision System (HVS). In the latter case, objective image quality measures may take precedence as the processing algorithms need not follow an HVS-based framework. Fortunately, the image processing community has developed a wide range of perceptual and objective image comparison metrics to choose from and we use both types of metrics in this study to determine SAR image quality.

5.2.1 Perceptual Image Comparison Metrics

There is a wide array of metrics in the image processing literature that compute image fidelity based on how the human brain perceives the quality of an image. For this study, we chose the Visual Signal to Noise Ratio (VNSR) metric [Chandler & Hermami 2007] because it was one the best performing in a large and comprehensive analysis of HVS-based metrics [Eerola 2010]. VSNR computes the perceptual similarity of two images and maps this value to a Signal-to-Noise Ratio

(SNR) scale. Since this metric outputs an SNR-type value, a high VSNR number signifies a high perceptual similarity between two images.

5.2.2 Objective Image Comparison Metrics

Compared to perceptual quality measures, there is a wider, and much more diverse, array of objective metrics for image comparison. Because of the diversity of these metrics, we decided to use several of them in our evaluation of SAR images. Specifically, we focused on the metrics produced by a publicly available MATLAB program called "Image/Picture Quality Measures" [Athinarayanan 2009] that uses MATLAB's Image Processing Toolbox to compute several common objective comparison metrics:

- **Mean Square Error (MSE):** The cumulative squared pixel error between a distorted image and the original image (low value is good)
- **Peak Signal to Noise Ratio (PSNR):** The peak fidelity between a distorted image and the original image (high value is good)
- **Normalized Cross-Correlation (NCC):** Normalized intensity, i.e., brightness, similarity between the distorted and source image (high value is good)
- **Average Difference (AD):** Average same-coordinate pixel-per-pixel difference (value closer to zero is good)
- **Structural Content (SC):** Object boundaries and/or regions of high entropy that are similar between the distorted and source image (high value is good)
- **Maximum Difference (MD):** Maximum same-coordinate pixel-per-pixel difference between the distorted and source image (low value is good)
- **Normalized Absolute Error (NAE):** Normalized sum of same-coordinate pixel-per-pixel absolute differences between the distorted and source image (low value is good)

6. Results

Given the target SAR workloads and datasets, which are summarized in Table 1, we developed fully single-precision and double-precision versions and ran them in the experimental environment described in Section 5. The following subsections discuss the results from these experiments, i.e., the observed power vs. performance tradeoffs of the IF phase of each workload, where performance refers to both IF image quality and execution time, and power refers to average power usage and total energy consumption.

6.1 SAR IF Image Quality

SAR IF image quality was measured in two ways: in terms of visual comparisons (Section 6.1.1) and in terms of automatic metric-based comparisons (Section 6.1.2).

6.1.1 Visual Comparisons

Among the three experimental runs (see Table 1, presented again below for convenience), the SAR image output from the single- and double-precision versions of SSCA3 were the most similar from a visual perspective. Figure 2 displays a side-by-side view of the resulting image formations

given the synthetic input data for a uniform lattice of point reflectors. As can be seen by the zoomed-in portions of the SAR images, they are virtually the same.

Among the UFB experiments, the output from the AFRL GMTI dataset performed the best. As can be seen in Figure 3, the single-precision output is slightly blurrier than the double-precision output, but image objects and patterns are still clearly discernable.

Figure 4 shows the worst performing single-precision run, i.e., UFB with the AFRL GOTCHA VV polarization dataset. The zoomed-in portions of the single- and double-precision outputs show that some content is completely blurred when single-precision is used.

In summary, Figures 2, 3, and 4 show that, from a visual standpoint, single-precision is good enough in some cases, i.e., SSCA3, and may not be sufficient for others, i.e., UFB GOTCHA VV. However, visually comparing images to decide the acceptability of single-precision computation requires human participation and, thus, cannot be part of any automated decision system geared towards satisfying power consumption and output quality constraints. Accordingly, the following subsection evaluates these images based on the comparison metrics that were discussed in Section 5.2, which would enable a resource management system to evaluate image quality constraint satisfaction automatically and, thus, facilitate autonomous power versus image-fidelity management in a Mobile Radar environment.

Table 1. Summary of target SAR workloads and datasets used in this study

Target SAR Workload	Image Formation Method	Computational Complexity	Input Dataset	Dataset Size
DARPA-SSCA3	Frequency-domain (ω-K)	$O(N^2 \log_2 N)$	Uniform point reflectors (synthetically generated)	488MB
AFRL-UFB	Time-domain (filtered-backprojection)	$O(N^3)$	AFRL GMTI Challenge (HH polarization)	2.95GB
			AFRL GOTCHA Challenge (230-360 azimuth range) (HH, HV, VH, and VV polarization input sets)	146MB (for each input set)

6.1.2 Automatic Metrics-based Comparisons

Table 2 shows the image comparison values for each workload/dataset precision pair in this study. Each column contains the computed image similarity between the single-precision and double-precision outputs of a workload/dataset for a particular metric. Since each metric computes image-similarity differently, i.e., sometimes a high value means high similarity, sometimes the opposite is true, please refer to section 5.2.2 for the significance of each value. However, for convenience, each column contains a bolded black value and a bolded red value each representing the most similar and dissimilar comparisons, respectively, for that metric. As Table 2 demonstrates, the single- and double-precision images for SSCA3 were judged the most similar for most of the metrics. While, UFB GOTCHA VV single- and double-precision outputs were judged the least similar for most of the metrics. The consensus of these automatic evaluations coincides with the visual observations presented in Section 6.1.1 and, thus, lends some credence to the feasibility of a

decision system that can judge whether a lower-power single-precision execution can satisfy image quality constraints.

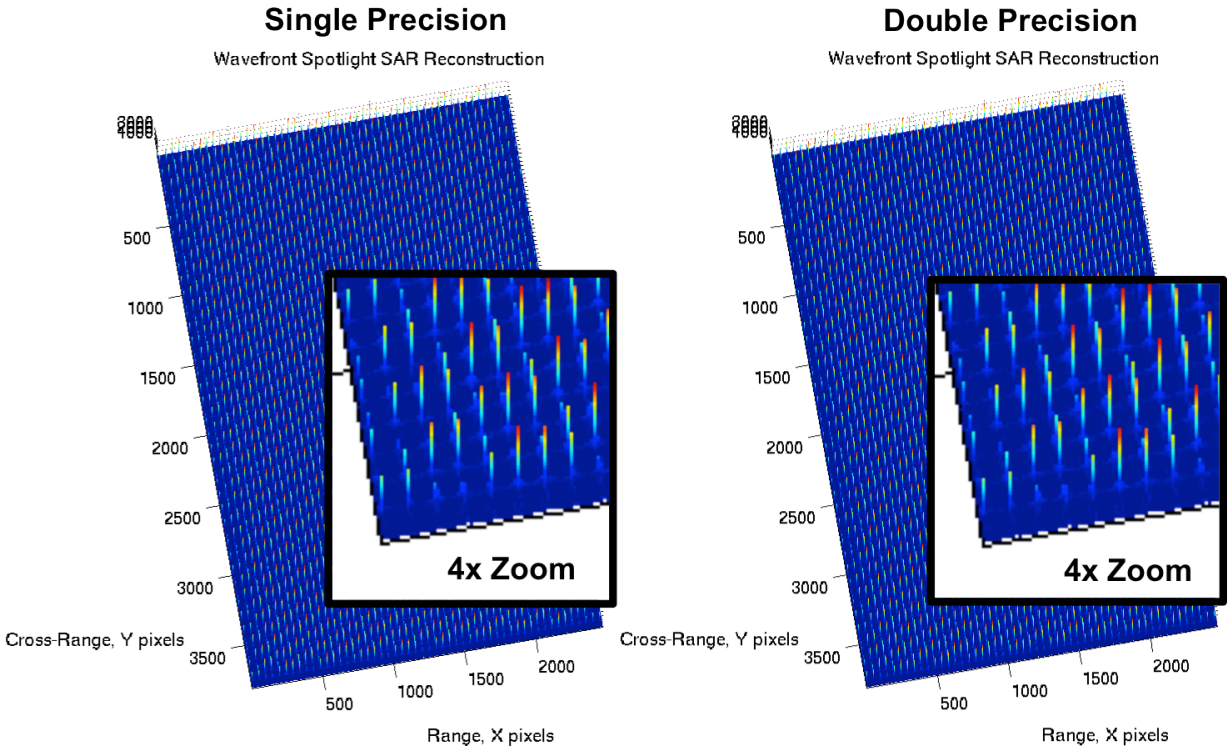

Figure 2. SAR image output of single- (left) and double-precision (right) versions of SSCA3 with zoomed-in portions for detail – images are visually the same

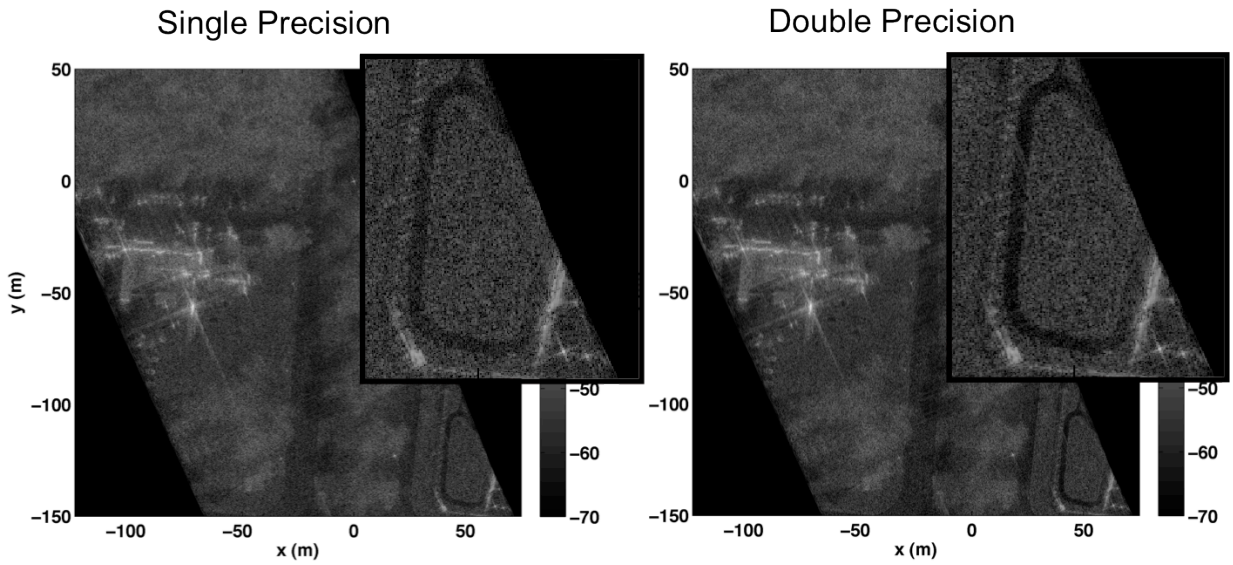

Figure 3. SAR image output for single- (left) and double-precision (right) versions of UFB with the AFRL GMTI input dataset with zoomed-in portions for detail – double-precision output is slightly clearer

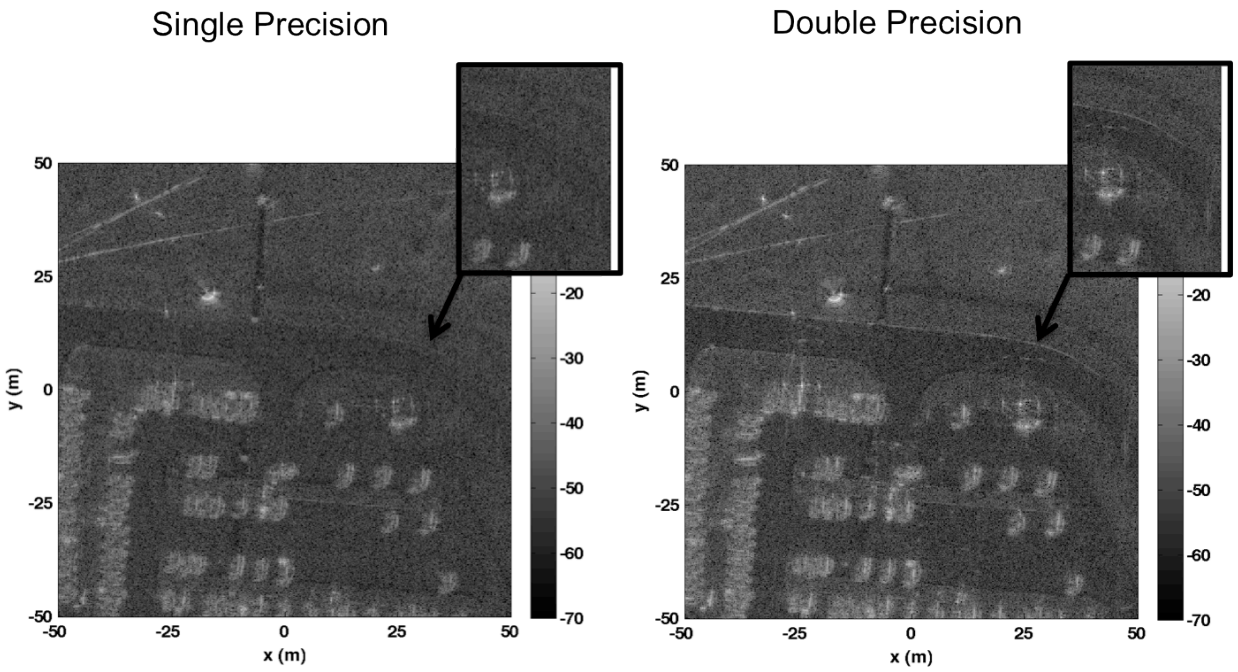

Figure 4. SAR image output for single- (left) and double-precision (right) versions of UFB with the AFRL GOTCHA VV polarization input dataset with zoomed-in portions for detail – double-precision is much clearer

6.2 Image-Formation Power, Energy, and Execution-time Results

As described in Section 5.2, we computed average power usage for the Image Formation (IF) phase of each workload/dataset/precision run using a simple power meter. We also recorded IF execution time by instrumenting the workload MATLAB codes with timing functions. Given the average power and execution-time values of each run, we acquired the total energy consumption of the respective IF phase by computing the product of the two values, i.e., *average power * execution time*.

Interestingly, as Table 3 shows, the average power consumption of the single- and double-precision runs were the same, disagreeing with the assumption that single-precision would be less power demanding. In fact, for the single case of SSCA3, single-precision actually used *more* power on average compared to the double-precision run. It may be that there were other runtime factors that masked the power benefits of single-precision during the execution of these workloads.

Table 3 also shows that SSCA3 experienced the largest execution-time speedup, almost 2x, when single-precision was used. Since frequency-domain solutions such as SSCA3 have relatively large memory requirements [Ulander, et al. 2003], this suggests that SSCA3 was memory-bound for the observed input/machine pair making it more susceptible to precision changes since single-precision not only affects what processor logic is used but also cuts the floating-point memory footprint in half compared to double-precision. This phenomenon, along with the power similarity between single- and double-precision, will be analyzed further in the next steps of our study.

Even though there were no observed power reductions for single-precision, all the single-precision runs completed quicker than their double-precision counterparts, thus, ultimately leading to less energy consumption. As can be seen in Figure 5, all single-precision executions produced significant energy reductions, i.e., between 14 and 51 percent. When we consider the image quality results in Table 2 and the energy reduction results in Figure 5, we see that, as was the case for UFB/GMTI and SSCA3, using single-precision in lieu of double-precision computations can result in significant energy reductions with little or no image degradation.

Table 2. Summary of single- vs. double-precision image similarity results: bolded values signify most similar for that metric, while red values signify least similar for that metric

SAR Workload and Input Set	Visual Signal to Noise Ratio	Mean Square Error	Peak Signal to Noise Ratio	Normalized Cross-Correlation	Average Difference	Structural Content	Maximum Difference	Normalized Absolute Error
SSCA #3	**22**	**2.5341**	**44.0926**	**1**	**-0.0004**	**1**	**239**	**0.0002**
Backprojection (GMTI Dataset)	21.6367	51.5091	31.012	1.0004	-0.6709	0.9977	85	0.0199
Backprojection - GOTCHA Dataset - HH polarization	18.7583	62.6727	30.16	1.0004	-0.6354	0.9973	**89**	0.0225
Backprojection - GOTCHA Dataset - HV polarization	16.1703	55.3237	30.7017	1.0006	-0.4797	0.9974	101	0.0179
Backprojection - GOTCHA Dataset - VH polarization	**16.1544**	54.9834	30.7285	1.0003	-0.3903	0.9979	101	0.0178
Backprojection - GOTCHA Dataset - VV polarization	16.8281	**379.7102**	**22.3363**	**0.9879**	**2.6844**	**1.0133**	161	**0.0654**

Table 3. Summary of power, execution-time, and energy results: bolded values signify largest energy difference, percentage-wise, between single- and double-precision runs

SAR Workload and Input Set	Average Watts (above idle)		Execution Time (Seconds)		Energy Consumption (Joules)	
	Single Precision	Double Precision	Single Precision	Double Precision	Single Precision	Double Precision
SSCA #3	13	11.5	246	566	**3198**	**6509**
Backprojection (GMTI Dataset)	11.5	11.5	4262	6036	49013	69414
Backprojection - GOTCHA Dataset - HH polarization	11.5	11.5	2702	3151	31073	36236.5
Backprojection - GOTCHA Dataset - HV polarization	11.5	11.5	2706	3174	31119	36501
Backprojection - GOTCHA Dataset - VH polarization	11.5	11.5	2711	3199	31176.5	36788.5
Backprojection - GOTCHA Dataset - VV polarization	11.5	11.5	2708	3227	31142	37110.5

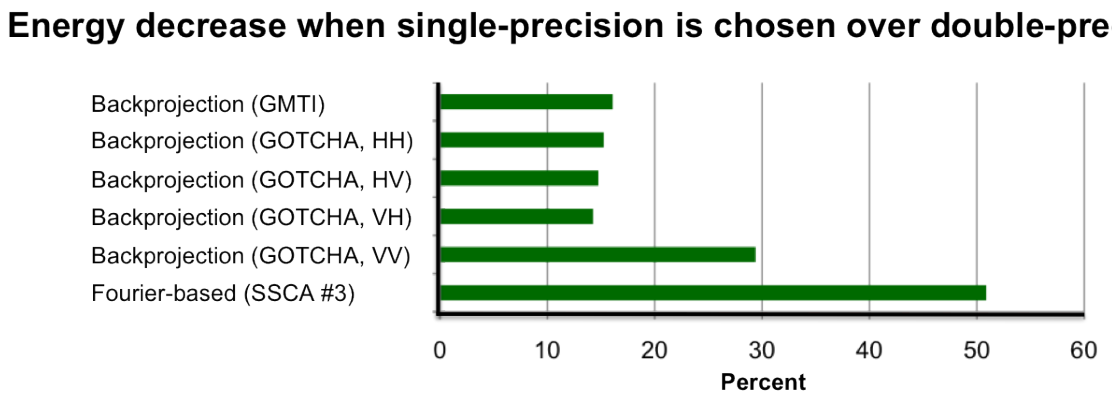

Figure 5. Reduction in total IF energy consumption when single-precision is used instead of double-precision – SSCA3 benefits the most from using single-precision (energy-wise).

7. Next Steps

As this was only the first phase of a more comprehensive study, there are many pending research tasks that we plan to accomplish in the near future – these are described below. Note that these tasks do not include further investigation of frequency-domain based approaches to image formation.

7.1 Evaluate more sophisticated SAR algorithms

A relatively simple backprojection SAR Image Formation (IF) code was used in this preliminary study. The backprojection-based Ultra-wideband (UWB) Synchronous Impulse Reconstruction (SIRE) radar code from the Army Research Lab (ARL) uses a more sophisticated backprojection algorithm that implements Recursive Sidelobe Minimization (RSM) [Nguyen 2009], an IF optimization that improves image quality and facilitates target detection during SAR post-processing. We are working with our ARL collaborators to evaluate their implementation of SIRE/RSM – we plan to use this as our target workload for the next phase of our power vs. performance study. Since this study only explored 2D SAR, we also will evaluate SAR image formation in 3D space with this SIRE/RSM workload.

7.2 Employ finer grain power measurement methods

This study used a very simple and coarse-grain method to capture the power consumption of SAR image formation phases of execution. We will extend this study with data from experiments that will use finer-grain power-measurement equipment, i.e., voltage and current probes hooked up to a Data Acquisition (DAQ) device, that can capture and store time-stamped power consumption of the whole system as well as of individual hardware components. Additionally, these tools will allow us to capture power consumption at finer timescales, thus, enabling us to potentially map power consumption to individual computation phases of SIRE/RSM.

7.3 Evaluate GPU implementations of SAR

One of the advantages of the backprojection algorithm is that it is embarrassingly parallel [Cordes 2009] and, thus, it is ideal for many-core devices such as GPGPUs. We're in the process of acquiring an NVIDIA Fermi-based GPU that implements fast double-precision floating-point arithmetic. We plan to use this testbed, along with the CPU/GPU hybrid version of SIRE/RSM from ARL [Park and Shires 2010], to evaluate fixed-point vs. single-precision vs. double-precision performance/power tradeoffs in a many-core environment.

7.4 Evaluate realistic SAR executables

This preliminary study focused on MATLAB implementations of SAR. The next phase of our study will analyze power vs. performance tradeoffs for non-interpreted, i.e., compiled instead of MATLAB, executions of SAR. The evaluation of SAR binaries may expose single-precision power benefits that were not visible under MATLAB runs, i.e., where MATLAB run-time instruction translation may have been the main energy consumer. We will start this study with the parallel C version of SIRE/RSM written by our collaborators at ARL [Park and Shires 2010].

7.5 Analyze mixed-mode precision implementations of SAR

This preliminary study focused only on workloads that were purely single- or double-precision implementations. In the next phase of this study, we will analyze smart mixed-mode precision implementations, i.e., that selectively use single- and double-precision computation within SIRE/RSM – this may reduce power consumption and execution time, with little or no loss in image quality.

7.6 Develop and evaluate fixed-point implementations of SAR

As discussed in Section 4.2, a processor's Floating-Point Unit (FPU) is larger and more complex than its Arithmetic-Logic Unit (ALU), thus, making floating-point arithmetic potentially more power intensive than fixed-point, i.e. integer-based, equivalents. Thus, we will translate the existing SIRE/RSM floating-point code to an equivalent fixed-point version. A tool that may help in this endeavor is the MATLAB Floating-Point to Fixed-Point Transformation Toolbox [Han, et al. 2005], which can facilitate the translation of floating-point DSP code to fixed-point equivalents. Then, as mentioned above, we will quantify the respective power benefits.

7.7 Employ more radar-centric output quality metrics

Although we employed a wide array of image quality metrics in this study, there are additional quality metrics that are geared more towards radar systems. These include Impulse Response (IPR), Peak to Sidelobe Ratio (PSLR), Integrated Sidelobe Ratio (ISLR), Multiplicative Noise Ratio (MNR), and Additive Noise Levels (ANL). We will be collaborating with radar specialists from the Army's Communications-Electronics Research, Development, and Engineering Center (CERDEC) who have expertise in these metrics, which will allow us to evaluate radar output quality within this more restricted problem domain.

7.8 Explore and implement dynamic power management of SAR image formation

Once we quantify SIRE/RSM's power vs. performance tradeoffs between fixed-point vs. single-precision vs. double-precision computation on CPUs and GPGPUs, as part of a doctoral dissertation, we will investigate dynamic adaptation mechanisms that will switch between these computational modes based on runtime measurements of power consumption and performance constraints, e.g., execution time and image quality, as well as adaptation overhead.

8. Conclusions

The first phase of this study focused on single- and double-precision implementations of two popular SAR Image Formation (IF) methods, i.e., frequency- and time-domain based approaches executed on a CPU. Using simple experimental and power measurement environments, and a set of image quality metrics, we made the following observations:

- For the observed workload/dataset pairs and quality metrics, single-precision IF image quality was often comparable to double-precision IF image quality.
- SAR image quality can be judged automatically using established image comparison metrics, thus, facilitating the implementation of a power manager that takes image quality constraints into account.
- For this experimental environment, single-precision did not offer any power benefits over double-precision. We are investigating this phenomena further.
- Single-precision consistently reduced IF execution time, thus, leading to significant reductions in total IF energy consumption, i.e., between 14 and 51 percent compared to double precision.
- The SSCA3 and UFB/GMTI experimental runs were good examples of cases where single-precision can offer significant energy benefits at little or no cost to IF image output quality.

Acknowledgments

We thank Ahmed Fasih from Ohio State University for his help in providing us with the source code for the UFB workload. In addition, we thank our ARL collaborators at Aberdeen Proving Ground - Dale Shires, Song Park, and James Ross - and at Adelphi Laboratory Center - Lam Nguyen and Anders Sullivan - for their input and expertise regarding SAR and embedded HEC systems. We would also like to thank Joseph C. Deroba from CERDEC for his additional feedback on SAR output quality evaluation and future research areas.

This work is supported by AHPCRC research grant no. W11NF-07-2-0027.

References

AMARASINGHE, S., CAMPBELL, D., CARLSON, W., CHIEN, A., DALLY, W., ELNOHAZY, E., HALL, M., HARRISON, R., HARROD, W., HILL, K., HILLER, J., KARP, S., KOELBEL, C., KOESTER, D., KO, P., LEVESQUE, J., REED, D. A., SARKAR, V., SCHREIBER, R., RICHARDS, M., SCARPELLI, A., SHALF, J., SNAVELY, A., AND STERLING. T. 2009. *ExaScale software study: Software challenges in extreme scale systems*. Study Report. AFRL contract number FA8650-07-C-7724, DARPA IPTO.

ATHINARAYANAN, S. 2009. *Image/Picture quality measures*. MATLAB Central File Exchange. http://www.mathworks.com/matlabcentral/fileexchange/25005-image-picture-quality-measures

BADER, D. A., MADDURI, K., GILBERT, J. R., SHAH, V., KEPNER, J., MEUSE, T., AND KRISHNAMURTHY, A. 2006. Designing scalable synthetic compact applications for benchmarking high productivity computing systems. *Cyberinfrastructure Technology Watch Quarterly*, vol. 2, no. 4B, pp 41–51.

BASU, S., AND BRESLER, Y. 2000. $O(N^2 \log_2 N)$ filtered backprojection reconstruction algorithm for tomography. *IEEE Transactions on Image Processing*, vol. 9, no. 10, pp. 1760–1773.

BROWN, D. J. AND REAMS, C. 2010. Toward Energy-Efficient Computing. *ACM Queue*, vol. 8, no. 2, pp. 30-43.

CASTEEL, C. H., GORHAM, L. A., MINARDI, M. J., SCARBOROUGH, S., AND NAIDU., K. D. 2007. A challenge problem for 2D/3D imaging of targets from a volumetric data set in an urban environment. In *Proceedings of the SPIE Defense and Security Symposium on Algorithms for Synthetic Aperture Radar Imagery XIV*. Orlando, FL. April 9–13 2007.

CHANDLER, D. M., AND HEMAMI, S. S. 2007. VSNR: A wavelet-based visual signal- to-noise ratio for natural images. *IEEE Transactions on Image Processing*, vol. 16, no. 9, pp. 2284–2298.

CORDES, B., AND LEESER, M. 2009. Parallel backprojection: a case study in high-performance reconfigurable computing. *EURASIP Journal on Embedded Systems*, vol. 2009, article 727965, pp 1–14.

EEROLA T. 2010. *Computational visual quality of digitally printed images*. Ph.D. Dissertation, Lappeenranta University of Technology, Finland.

FASIH, A. R., AND HARTLEY, T. D. R. 2010. GPU-accelerated synthetic aperture radar backprojection in CUDA. In *Proceedings of the IEEE Radar Conference*. Arlington, VA. May 10–13 2010. pp. 1408–1413.

GORHAM, L. A., AND MOORE, L. J. 2010. SAR image formation toolbox for MATLAB. In *Proceedings of the SPIE Defense and Security Symposium on Algorithms for Synthetic Aperture Radar Imagery XVII*. Orlando, FL. April 5–9, 2010.

HAN, K., EVANS, B. L., AND SWARTZLANDER JR., E. E. 2005. Low-power multipliers with data word length reduction. In *Proceedings of the Asilomar Conference on Signals, Systems, and Computers*. Pacific Grove, CA. Oct. 30–Nov. 2 2005. pp. 1615-1619.

HAN, K. 2006. *Automating transformations from floating-point to fixed-point for implementing digital signal processing algorithms*. Ph.D. dissertation, The University of Texas at Austin, Austin TX USA.

HUNTER, A. J., HAYES, M. P., AND GOUGH, P. T. 2003. A comparison of fast factorized back-projection and wave number algorithms for SAS image reconstruction. In *Proceedings of the World Congress on Ultrasonics*. Paris, France. Sep 5–9 2003.

LIU, Y, AND ZHU, H. 2010. A survey of the research on power management techniques for high-performance systems. *Software: Practice and Experience*, DOI 10.1002/spe.952.

MILMAN, A. S. 1993. SAR imaging by ω-k migration. *International Journal of Remote Sensing*, vol. 14, no. 10, pp. 1965–1979.

MUNSON, D. C., O'BRIEN, J. D., AND JENKINS, W. K. 1983. A tomographic formulation of spotlight-mode synthetic aperture radar. *Proceedings of the IEEE*, vol. 71, no. 8, pp. 917–925.

NA, Y., LU, Y., AND SUN, H. 2006. A comparison of back-projection and range migration algorithms for ultra-wideband SAR imaging. In *Proceedings of the 4th IEEE Workshop on Sensor Array and Multi-Channel Processing*. Waltham, MA. July 12–14 2006. pp. 320-324.

NEOPHYTOU, N., XU, F. AND MUELLER, K. 2007. Hardware acceleration vs. algorithmic acceleration: can GPU-based processing beat complexity optimization for CT?. In *Proceedings of the SPIE Symposium on Medical Imaging*. San Diego, CA. February 17–22 2007.

NGUYEN, L. 2009. *Signal and image processing algorithms for the U.S. army research laboratory ultra-wideband (UWB) synchronous impulse reconstruction (SIRE) radar*. Technical Report. ARL-TR-4784, U.S. Army Research Laboratory, Adelphi, MD.

PARK, S. J., AND SHIRES, D. 2010. *Central processing unit/graphics processing unit (CPU/GPU) hybrid computing of synthetic aperture radar algorithm*. Technical Report. ARL-TR-5074, U.S. Army Research Laboratory, Aberdeen, MD.

PREISS, J., BOERSMA, M., AND MUELLER, S. M. 2009. Advanced clock gating schemes for fused-multiply-add-type floating-point units. In *Proceedings of the 2009 19th IEEE Symposium on Computer Arithmetic*. Portland, OR. June 9–10 2009.

POINTER, D. 2008. *Projecting Images in Radar and Medical Applications*. DSP-FPGA. http://www.dsp-fpga.com/articles/id/?3133

RAHMAN S. 2010. *Focusing moving targets using range migration algorithm in ultra wideband low frequency synthetic aperture radar*. M.S. thesis, Blekinge Institute of Technology, Sweden.

SCARBOROUGH, S. M., CURTIS, J., CASTEEL, H., GORHAM, L., MINARDI, M. J., MAJUMDER, U. K., JUDGE, M. G., ZELNIO, E., BRYANT, M., NICHOLS, H., AND PAGE, D. 2009. A challenge problem for SAR-based GMTI in urban environments. In *Proceedings of the SPIE Defense and Security Symposium on Algorithms for Synthetic Aperture Radar Imagery XVI*. Orlando, FL. April 13–17, 2009.

SDMS. *Gotcha and GMTI Synthetic Aperture Radar Public Datasets*. AFRL Sensor Data Management System (SDMS) Public Web Site Home Page. https://www.sdms.afrl.af.mil

SRC. *Airborne, Portable and Embedded Processing Solutions*. SRC Computers LLC, http://srccomputers.com/products/embeddedportable.asp

ULANDER, L. M. H., HELLSTEN, H., AND STENSTRÖM, G. 2003. Synthetic-aperture radar processing using fast factorized back-projection. *IEEE Transactions on Aerospace and Electronic Systems*, vol. 39, no. 3, pp. 760–776.

WILLIS, N. J., AND GRIFFITHS, H. D. 2007. *Advances in Bistatic Radar*. SciTech Publishing, Raleigh, NC.

VU, V. T., SJÖGREN, T. K, AND PETTERSSON, M. I. 2008. A comparison between fast factorized backprojection and frequency-domain algorithms in UWB low frequency SAR. In *Proceedings of the IEEE International Geoscience & Remote Sensing Symposium*. Boston, MA. July 6–11 2008.