

# Area Efficient Backprojection Computation with Reduced Floating-Point Word Width for SAR Image Formation

Jon J. Pimentel, Aaron Stillmaker, Brent Bohnenstiehl, and Bevan M. Baas  
 Department of Electrical and Computer Engineering  
 University of California, Davis  
 {jppimentel, astillmaker, bvbohen, bbaas}@ucdavis.edu

**Abstract**—The widths of data words in digital processors have a direct impact on area in application-specific ICs (ASICs) and FPGAs. Circuit area impacts energy dissipation per workload and chip cost. Floating-point exponent and mantissa widths are independently varied for the seven major computational blocks of an airborne synthetic aperture radar (SAR) engine. The circuit area in 65 nm CMOS and the PSNR and SSIM metrics are found for 572 design points. With word-width reductions of 46.9–79.7%, images with a 0.99 SSIM are created with imperceptible image quality degradation and a 1.9–11.4× area reduction.

## I. INTRODUCTION

Synthetic aperture radar (SAR) imaging uses pulses of microwave energy transmitted from a series of locations towards a target and reflected back towards an antenna to provide a means for day, night, and all weather imaging while producing resolution that otherwise requires a large antenna aperture [1]. SAR imaging is applied in many fields including environmental monitoring, navigation, and reconnaissance.

The focus of this work is to reduce the widths of data words in a SAR backprojection image formation data path to much smaller widths than the floating-point (FP) double-precision (DP) and single-precision (SP) commonly used in programmable processors as well as in custom hardware. Reducing data word widths directly reduces circuit area which is easy to unequivocally measure and thus it is the metric we use in this work. Energy dissipation and computational latency are also directly reduced by word-width reduction but unfortunately also depend on factors such as radar data and architectural details and so are more difficult to compare.

The SAR backprojection algorithm is a widely used, compute intensive method for forming images, and is known for its inherent parallel nature [2], [3]. With the shift towards many-core processing-arrays and interest in energy-efficiency, there has been specific algorithmic study to parallelize SAR backprojection algorithms for a graphics processing unit (GPU) [4], an Intel Xeon Phi many-core accelerator [5], and a TI digital signal processor (DSP) [6]. Additionally, FPGAs have been considered as image processors for real-time SAR processing [7]. Previously, SAR compression algorithms have been necessary to allow for remote computation through a bandwidth limited link [8], [9]. However, performing computation on an embedded system can remove the need for a raw-data link if performed in an energy-efficient fashion.

The circuit area of an application-specific IC (ASIC) is a useful metric to predict the energy dissipation per calculation (among similar designs) and fabrication cost per chip. In addition, comparing ASIC areas among similar designs can also provide a basis for predicting the resource requirements, energy/calculation, and performance of implementations on FPGAs.

FP arithmetic is typically chosen for performing backprojection computations due to the dynamic range and precision of the raw data. Previous work has been done on optimal design trade-offs of

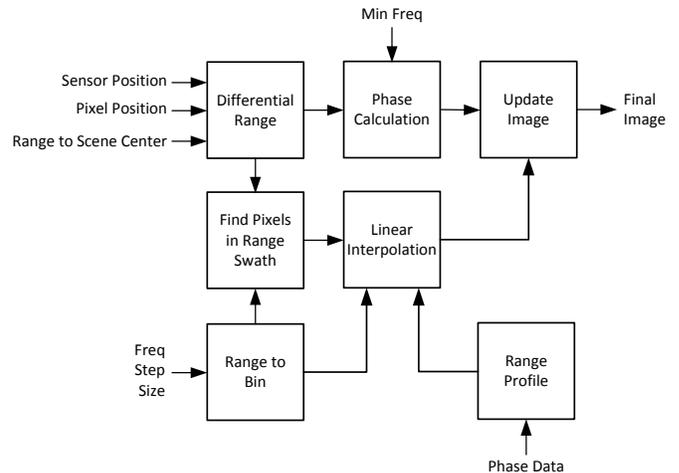


Fig. 1. Block diagram of the seven major computation blocks of the SAR backprojection engine.

general FP hardware [10], [11]. However, little exploration has been done into reducing FP mantissa and exponent width to observe the effect this has on SAR image quality and chip area.

We determine the FP hardware requirements of a backprojection algorithm for an airborne spotlight-mode SAR system and the effect that reducing the FP word width has on final image quality. Identifying the necessary precision for each of these blocks can improve future ASIC design and algorithm development. The image formation algorithm is broken into seven functional blocks. The FP word width for each of these blocks is reduced, SAR images are formed and then assessed against images created using DP-FP arithmetic. The potential width reduction and area savings that maintain acceptable image quality are identified. X-band data collected via spotlight-mode SAR, in which the radar beam is continually steered to illuminate the same target scene [1], is used to form images.

The remainder of this paper is organized as follows: Section II presents the functional blocks for the backprojection algorithm. Section III discusses the data sets considered. Section IV explains the methods used for reducing FP word width and determining chip area. Section V evaluates the effect of FP word width reductions on image quality and area. Section VI concludes the paper.

## II. BACKPROJECTION ALGORITHM FUNCTIONAL BLOCKS

A previously developed SAR signal model is used for the backprojection imaging algorithm [12]. This model has been used to generate a backprojection imaging algorithm in MATLAB [13]. The functional blocks of the backprojection implementation are shown in Fig. 1. For the data sets considered, the range to the motion compensation

point was always around 10,000 m, and each coordinate of the antenna phase center ranged between 0–7000 m, conditions which are common with many airborne SAR systems [14]. Each functional block of the backprojection algorithm is described below.

#### A. Range to Bin

The range to bin data is used for finding pixels that fall within the range swath and for linear interpolation. The number of bins is set by the number of points of the inverse fast Fourier transform (IFFT). The values of these bins are evenly spaced along the scene range. The maximum alias-free range extent of the image is calculated by dividing the speed of light by twice the frequency step size. This module uses a multiplication, addition/subtraction, and division unit; which is implemented using the non-restoring algorithm [15].

#### B. Range Profile

The range profile is computed by zero padding the phase history data and computing the IFFT. The size of the IFFT was chosen as 4096 points to maintain computational efficiency of the IFFT, reduce the occurrence of artifacts, and produce a smoother image. The range profile data is then used during linear interpolation, along with the differential range and the range to each bin, to determine an interpolated value for the range profile.

The areas for storing the phase history data, the computed transform, and twiddle factors in memory are considered separately from the computation area. The computation area is calculated for a single radix-2 butterfly. As with the other functional blocks, the number of computation units can be replicated depending on the desired throughput, however the area relationship between using different FP word widths remains the same.

The computational area is provided in Table I. The area for data storage for the DP-FP implementation was  $3\,060\,000\ \mu\text{m}^2$ . The minimum area for the data storage to obtain an  $\text{SSIM} \geq 0.5$  was  $680\,000\ \mu\text{m}^2$ . The minimum area to obtain an  $\text{SSIM} \geq 0.9\text{--}0.99$  was identical, and required  $1\,020\,000\ \mu\text{m}^2$ . The memory area is estimated from a large on-chip shared memory fabricated in 65 nm CMOS [16], for a many-core processor array [17].

#### C. Differential Range

The differential range is computed using the three-dimensional position of each pixel, the sensor at each pulse, and the range to the scene center. The differential range is the difference between the distance from the antenna phase center to the scatterer, and the distance from antenna phase center to the origin. This block uses five addition/subtraction, three multiplication, and one square root unit; which is implemented using the non-restoring algorithm [18]. The differential range data is used to compute the phase corrections and perform linear interpolation.

#### D. Phase Calculation

Determining the phase correction for the receiver output relies on the minimum frequency of the received samples and the differential range [13]. This calculation requires a complex exponential function which can be transformed into equivalent trigonometric functions. First, inputs to the phase calculation block are range reduced into a smaller interval [19], then the sine and cosine are computed using polynomial approximations for values within this interval [20]. In order to pipeline the computation of the polynomial approximations for sine and cosine, this module uses 29 multiplication units and 14 addition/subtraction units. The phase correction is then applied to the data following linear interpolation.

#### E. Find Pixels in Range Swath

Using the minimum and maximum values of the range to every bin in the range profile and the differential range values, this module uses a comparison unit to determine which pixels are within the range swath. Linear interpolation is then performed only on these pixels.

#### F. Linear Interpolation

Following calculation of the range profile, the range to each bin in the range profile, and the pixels in the range swath, a linear interpolation operation is performed since the values of the differential range do not exactly line up with the discrete range to bin values [13]. This module uses one division, five addition/subtraction, and four multiplication modules. This number of functional blocks allows simultaneous processing of the real and imaginary data. Phase correction is then applied to the data in the image update block.

#### G. Image Update

Following the linear interpolation step, the phase correction is applied to the data through a complex multiplication and then the image responses are summed for each pulse to create the final image data. This module utilizes four multiplication modules and four addition/subtraction modules.

### III. SYNTHETIC APERTURE RADAR DATA SETS

Three publicly available data sets released by the Air Force Research Laboratory (AFRL) are utilized for image formation [21], [22]. The phase history data for all three data sets is in the X-band region with a circular flight path and collected via spotlight passes.

#### A. Volumetric SAR Data Set

This data set is formed from imaging stationary civilian vehicles and calibration targets [21]. For each azimuth angle, there are 117 pulses on average, each with 424 frequency samples. Images formed from this data set have a scene extent of  $100\ \text{m} \times 100\ \text{m}$  and form a  $501 \times 501$  pixel image.

#### B. Point Target Data Set

This data set consists of synthetically generated data for three point targets [13]. The targets were simulated with 128 pulses and 512 frequency samples per pulse. Images formed from this data set have a scene extent of  $10\ \text{m} \times 10\ \text{m}$  and a form a  $501 \times 501$  pixel image.

#### C. Ground Moving Target Indicator Data Set

The SAR-based Ground Moving Target Indicator (GMTI) motion compensated radar data set includes data from imaging a moving vehicle in an urban environment [22]. The data includes 8000 pulses and 384 frequency samples per pulse. Images formed from this data set have a scene extent of  $200\ \text{m} \times 200\ \text{m}$  and form a  $1001 \times 1001$  pixel image.

### IV. METHODS FOR REDUCING FLOATING-POINT WORD WIDTH AND DETERMINING AREA FOR BACKPROJECTION

Using a combination of C++ and MATLAB, each functional block of the backprojection algorithm is written to support computations at any exponent and mantissa width up to DP-FP. For each FP word width, the FP standard is appropriately considered to account for changing exponent bias, and rounding positions. Modifying the exponent and mantissa width allows control over the available dynamic range and precision for each FP format. The IEEE-754 default round-to-nearest mode, round-to-nearest is used for all computations [23]. The dependency on dynamic range is evaluated by modifying the exponent width to between 1 and 11 bits. The effects of precision on image

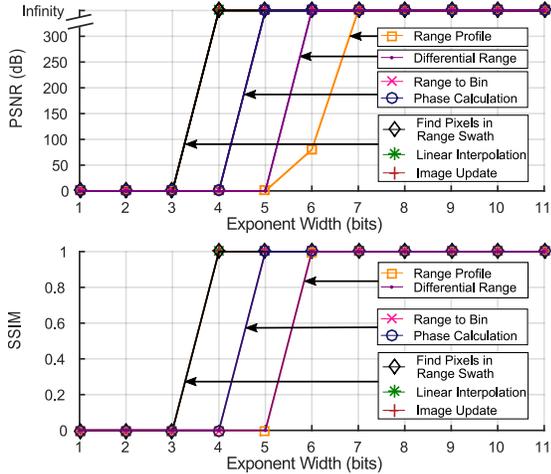


Fig. 2. PSNR (upper plot) and SSIM (lower plot) of resulting images versus the exponents widths of the seven functional blocks used to compute those images. The values of SSIM and PSNR are used for determining the minimum exponent widths for each functional block. Data are determined by the worst-case PSNR and SSIM across the three data sets. The mantissa width is kept at 52 bits.

quality are explored by modifying the mantissa width to between 1 and 52 bits after the decimal point. Before modifying mantissa width, the minimum exponent width needed to accommodate the dynamic range is determined. Fig. 2 plots PSNR and SSIM versus exponent width for each functional block.

To determine the area requirements at each exponent and mantissa width, each functional block is written in Verilog RTL, and synthesized in 65 nm CMOS at 1.3 V and a clock frequency of 1.2 GHz. Each functional block is pipelined. Since each functional block has a different pipeline depth, this results in different numbers of pixels that can be operated on simultaneously. Therefore, area percentages are made relative to the same functional block. To achieve a throughput requirement, the computational units can simply be replicated and the area relationship between using different FP word widths will remain the same.

## V. COMPARISON OF IMAGE QUALITY WITH REDUCED FLOATING-POINT WORD WIDTHS

To obtain gold-standard images to measure against, each functional block is first configured to perform computations using DP-FP arithmetic. The backprojection algorithm is then performed on all data sets to form each gold-standard image.

For each image quality comparison, the mantissa and exponent width for one of the seven functional blocks is reduced from DP-FP. DP arithmetic is then utilized for the remaining six functional blocks. Each data set is processed to form an image. The images formed when modifying the FP word width of each functional block are then evaluated against the gold-standard image. Images are saved as gray level (8-bit) jpegs using lossless compression before comparison.

Peak Signal-to-Noise Ratio (PSNR) and the Structural Similarity (SSIM) index are used to quantify the quality of images formed using each FP word width configuration [24]. Each image is measured against the gold-standard image. PSNR measures the ratio between the maximum signal power and the noise corrupting the image, however this metric does not map well with the human visual system. However, SSIM is based on the notion that human visual perception is adapted for extracting structural information about an image.

Fig. 3 plots the worst-case PSNR and SSIM for images constructed using the three data sets while varying the mantissa width. PSNR and

SSIM versus mantissa width are plotted in the upper and lower figure, respectively. All mantissa widths between 1 and 52 bits are traversed. The exponent width of each functional block is chosen to satisfy the dynamic range requirements of all data sets as shown in Fig. 2. The results when using SP-FP and DP-FP arithmetic are also denoted.

Although PSNR is a commonly used image quality metric, SSIM proves to be more useful for determining human perceived image quality. Among the data sets utilized, images appearing identical can differ in PSNR value by as much as 80 dB. Alternatively, no visibly detectable differences are found between the gold-standard images and images formed using a reduced FP word width when the SSIM is  $\sim 0.95$  or higher. Therefore, it is considered that images produced with an  $SSIM \geq 0.99$  are indiscernible from the gold-standard images. After achieving a value  $\geq 0.99$ , the SSIM asymptotically approaches a value of 1 without visibly improving image quality, therefore the additional FP word width and hardware are unnecessary.

The results shown in Table I demonstrate that the mantissa width requirements for each functional block range between 6–27 bits to form an image with an  $SSIM \geq 0.99$ . These reductions in widths amount to average area savings of 75.5%. The largest area savings are obtained by reducing mantissa width, rather than exponent width.

The range profile functional block has the largest potential area savings. It is possible to reduce the FP word width for this block from DP-FP to a format using a 6-bit exponent and 6-bit mantissa and obtain a resulting SSIM value of 0.99. This reduction amounts to an area savings of 91.2%. Conversely, the differential range block required the largest mantissa width. To achieve an SSIM of 0.99, a mantissa width of 27 bits is required; reducing area 48.4%.

The image quality produced when reducing the exponent and mantissa width of all functional blocks simultaneously is also considered. Fig. 4 shows four images formed using the volumetric data set. Each image is examined against Fig. 4d, which is the gold-standard image created by using DP-FP arithmetic for each functional block. For Fig. 4a–4c each functional block is configured to use the exponent and mantissa widths shown in Table I to achieve an SSIM of 0.5, 0.9, and 0.99, respectively. For Fig. 4a, although each functional block is configured to achieve an SSIM of 0.5, the image produced when using these reduced widths together forms a visibly degraded image with an SSIM of 0.42. However, for Fig. 4b and Fig. 4c each functional block is configured to achieve an SSIM of 0.9 and 0.99, respectively, and the image quality is not visibly different from Fig. 4d. Similar results are observed for the other data sets in which using the settings for achieving SSIM values  $\geq 0.9$  maintained image quality.

## VI. CONCLUSION

In this paper, the effect that reducing the FP word width has on image quality and chip area when performing the backprojection algorithm to form airborne spotlight-mode SAR images from the X-band is presented. The backprojection image formation algorithm is split into seven functional blocks and the effect of reducing precision and dynamic range is quantified through image quality and area comparisons. These reductions in width and area are considered as a first step towards future SAR backprojection ASIC design and algorithm development. The image quality metrics of PSNR and SSIM are utilized to determine potential area savings while maintaining high image quality. The effect on final image quality when the FP word width was reduced for all blocks simultaneously is also demonstrated and shows no visible image quality degradation when using settings to obtain an SSIM of 0.9 or higher. Each functional block uses 48.4–91.2% less area than that required by DP-FP hardware.

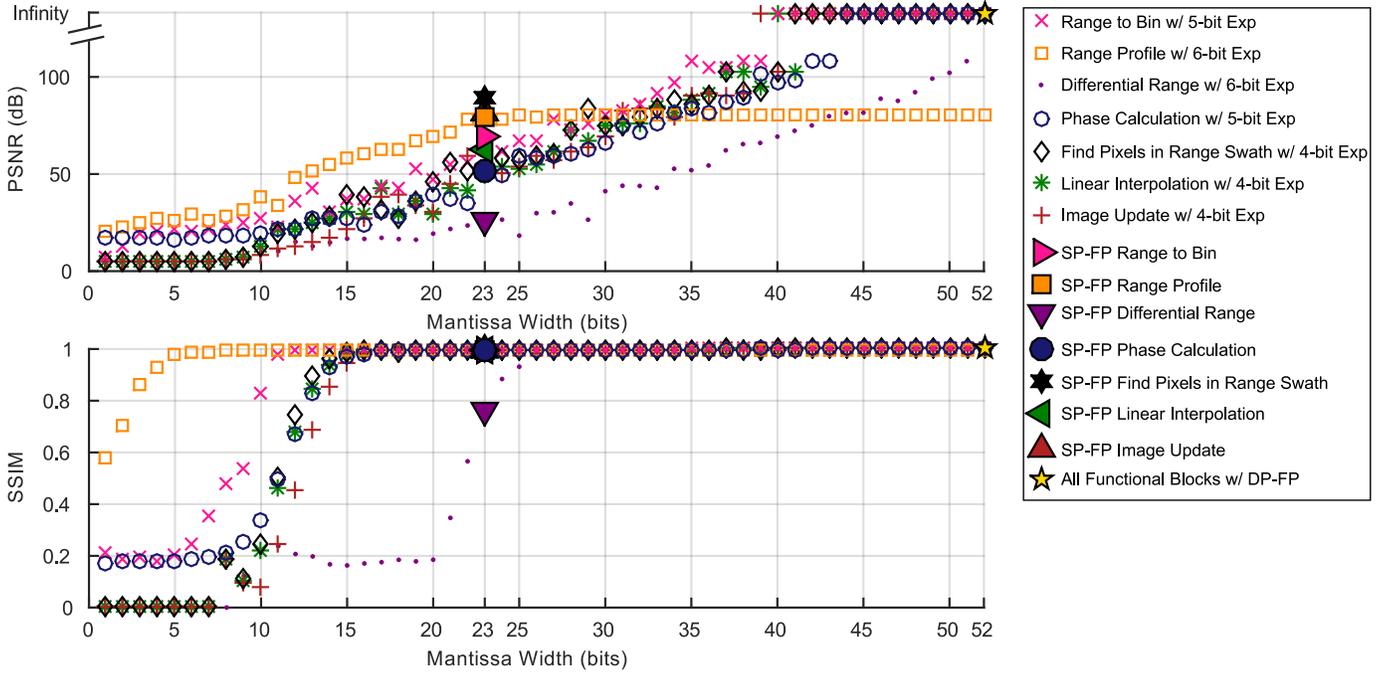


Fig. 3. PSNR (upper plot) and SSIM (lower plot) of resulting images versus the mantissa widths of the seven functional blocks used to compute those images. Images are measured against images formed using double-precision floating-point (DP-FP) and single-precision floating-point (SP-FP) arithmetic. Data are determined by the worst-case PSNR and SSIM across the three data sets. The exponent width is chosen to satisfy the dynamic range requirement of all data sets as shown in Fig. 2. All functional blocks using DP arithmetic are denoted by the gold star symbol. Functional blocks using SP-FP arithmetic have a mantissa width of 23 bits.

Table I  
SSIM AND PSNR OF BACKPROJECTION IMAGES FORMED WITH REDUCED FLOATING-POINT WORD WIDTHS MEASURED AGAINST SINGLE-PRECISION AND DOUBLE-PRECISION ARITHMETIC

	Functional Block						
	Range to Bin	Range Profile	Differential Range	Phase Calculation	Find Pixels in Range Swath	Linear Interpolation	Image Update
Area w/ Single Precision Arithmetic ( $\mu\text{m}^2$ )	23365	102409	92708	552409	2469	108609	95212
Area w/ Double Precision Arithmetic ( $\mu\text{m}^2$ )	66040	250694	214185	1754742	5171	273014	257678
Exponent Width used for All Optimized Designs*	5	6	6	5	4	4	4
Minimum Mantissa Width for SSIM $\geq 0.5$	9	1	22	12	11	12	13
Minimum Mantissa Width for SSIM $\geq 0.9$	11	4	25	14	14	14	15
Minimum Mantissa Width for SSIM $\geq 0.99$	12	6	27	17	16	17	17
Minimum Area ( $\mu\text{m}^2$ ) for SSIM $\geq 0.5$	8185	4415	82255	221803	1126	47390	43128
(% Area of DP-FP)	(12.4%)	(1.8%)	(38.4%)	(12.6%)	(21.8%)	(17.4%)	(16.7%)
Minimum Area ( $\mu\text{m}^2$ ) for SSIM $\geq 0.9$	10270	16422	93361	249950	1299	51998	52058
(% Area of DP-FP)	<b>(15.6%)</b>	<b>(6.6%)</b>	<b>(43.6%)</b>	<b>(14.2%)</b>	<b>(25.1%)</b>	<b>(19.0%)</b>	<b>(20.2%)</b>
Minimum Area ( $\mu\text{m}^2$ ) for SSIM $\geq 0.99$	10716	22080	110425	354374	1438	66287	57973
(% Area of DP-FP)	<b>(16.2%)</b>	<b>(8.8%)</b>	<b>(51.6%)</b>	<b>(20.2%)</b>	<b>(27.8%)</b>	<b>(24.3%)</b>	<b>(22.5%)</b>

Results based on synthesis in 65 nm CMOS with a supply voltage of 1.3 V at 1.2 GHz and evaluated against the images formed when using DP-FP arithmetic. SSIM values are shown for reducing mantissa width of only the given block while all other blocks have 52-bits (DP-FP).

\* These values are the smallest exponent word widths which satisfy the dynamic range requirement as shown in Fig. 2.

#### ACKNOWLEDGMENT

The authors gratefully acknowledge support from C2S2 Grant 2047.002.014, NSF Grant 1018972 and 0903549 and CAREER

Award 0546907, SRC GRC Grant 1598, 1971, and 2321 and CSR Grant 1659, SEM, LeRoy Gorham, and Linda Moore. The authors would also like to thank the Air Force Research Laboratory for providing the data sets.

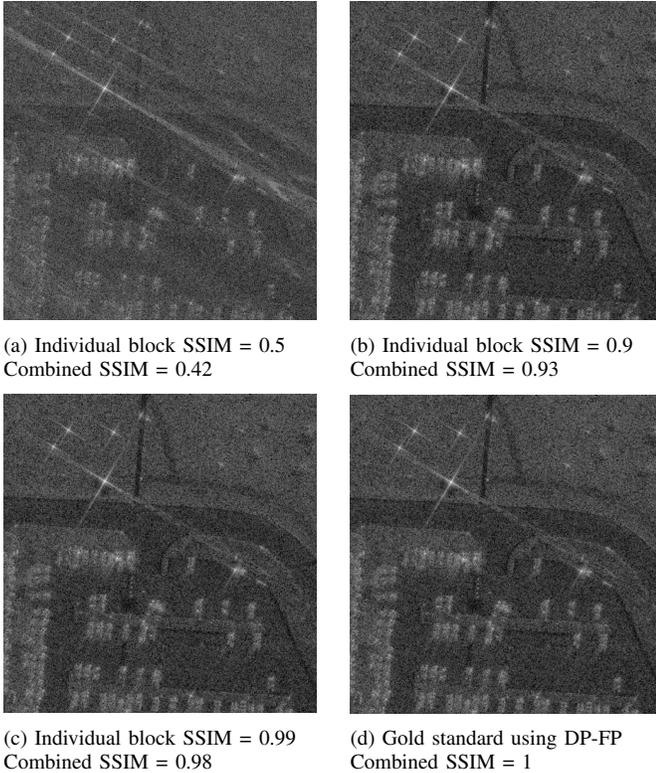


Fig. 4. Images formed using the backprojection algorithm and the volumetric data set. An integration angle of  $4^\circ$  centered at  $60^\circ$  azimuth is used. Each functional block is connected together and configured to achieve a specific SSIM value. (a) Image formed using widths for each functional block to provide at least SSIM = 0.5. The SSIM of the image degraded to 0.42. For (b) and (c), configuring each functional block to provide a SSIM = 0.9 and SSIM = 0.99 does not visibly degrade the final image quality. (d) Gold-standard image created using DP-FP arithmetic.

#### REFERENCES

[1] C. Jakowatz Jr. *et al.*, *Spotlight-mode Synthetic Aperture Radar: A signal processing approach*. Boston, MA: Kluwer Academic Publishers, 1996.

[2] A. Yegulalp, "Fast backprojection algorithm for synthetic aperture radar," in *Proc. IEEE RadarCon*, 1999, pp. 60–65.

[3] M. Desai and W. Jenkins, "Convolution backprojection image recon-

struction for spotlight mode synthetic aperture radar," *IEEE TIP*, vol. 1, no. 4, pp. 505–517, Oct 1992.

[4] R. Portillo *et al.*, "Power versus performance tradeoffs of GPU-accelerated backprojection-based synthetic aperture radar image formation," in *Proc. SPIE*, vol. 8060, Jun. 2011, pp. 806 008–806 008–21.

[5] J. Park *et al.*, "Efficient backprojection-based synthetic aperture radar computation with many-core processors," in *Proc. of Intl. Conf. for High Perf. Comp., Net., Storage and Analysis (SC)*, Nov. 2012, pp. 1–11.

[6] D. Wang and M. Ali, "Synthetic aperture radar on low power multi-core digital signal processor," in *Proc. IEEE HPEC*, Sept 2012, pp. 1–6.

[7] K.-N. Chia *et al.*, "High-performance automatic target recognition through data-specific vlsi," *IEEE TVLSI*, vol. 6, no. 3, Sept 1998.

[8] N. Agrawal and K. Venugopalan, "Analysis of complex SAR raw data compression," in *Proc. PIERS*, Cambridge, MA, July 2008, pp. 155–160.

[9] S. Bhattacharya *et al.*, "Synthetic aperture radar raw data encoding using compressed sensing," in *Radar Conf., 2008. RADAR '08. IEEE*, May 2008, pp. 1–5.

[10] J. J. Pimentel and B. M. Baas, "Hybrid floating-point modules with low area overhead on a fine-grained processing core," in *Proc. IEEE Asilomar Conf. on Sig., Systems and Computers (ACSSC)*, Nov. 2014.

[11] J. Y. F. Tong, D. Nagle, and R. A. Rutenbar, "Reducing power by optimizing the necessary precision/range of floating-point arithmetic," *IEEE TVLSI*, vol. 8, no. 3, pp. 273–285, Jun. 2000.

[12] B. Rigling and R. Moses, "Taylor expansion of the differential range for monostatic SAR," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 41, no. 1, pp. 60–64, Jan 2005.

[13] L. A. Gorham and L. J. Moore, "SAR image formation toolbox for MATLAB," in *Proc. of SPIE*, vol. 7699, no. 1. SPIE, 2010, p. 769906.

[14] M. Richards, "A beginner's guide to interferometric SAR concepts and signal processing," *IEEE Aerosp. Electron. Syst. Mag.*, vol. 22, no. 9, pp. 5–29, Sept 2007.

[15] H. Nikmehr, "Architectures for floating-point division," Ph.D. dissertation, The University of Adelaide, 2005.

[16] D. Truong *et al.*, "A 167-processor computational platform in 65 nm CMOS," *IEEE JSSC*, 2009.

[17] —, "A 167-processor 65 nm computational platform with per-processor dynamic supply voltage and dynamic clock frequency scaling," in *VLSI Circuits, IEEE Symp. on*, Jun. 2008.

[18] Y. Li and W. Chu, "Implementation of single precision floating point square root on FPGAs," in *Proc. IEEE FCCM*, Apr 1997, pp. 226–232.

[19] K. C. Ng, "Argument reduction for huge arguments: Good to the last bit," Tech. Rep., 1992.

[20] J. F. Hart, *Computer Approximations*. Krieger Publishing Co., 1978.

[21] C. H. Casteel, Jr. *et al.*, "A challenge problem for 2D/3D imaging of targets from a volumetric data set in an urban environment," in *Proc. SPIE*, vol. 6568, 2007, pp. 65 680D–65 680D–7.

[22] S. M. Scarborough *et al.*, "A challenge problem for SAR-based GMTI in urban environments," in *Proc. SPIE*, vol. 7337, 2009.

[23] *IEEE Standard for Floating-Point Arithmetic*, 2008, IEEE Std 754-2008.

[24] Z. Wang *et al.*, "Image quality assessment: from error visibility to structural similarity," *IEEE Trans. Image Process.*, vol. 13, no. 4, pp. 600–612, April 2004.