

SANDIA REPORT

SAND96-2460 • UC-900

Unlimited Release

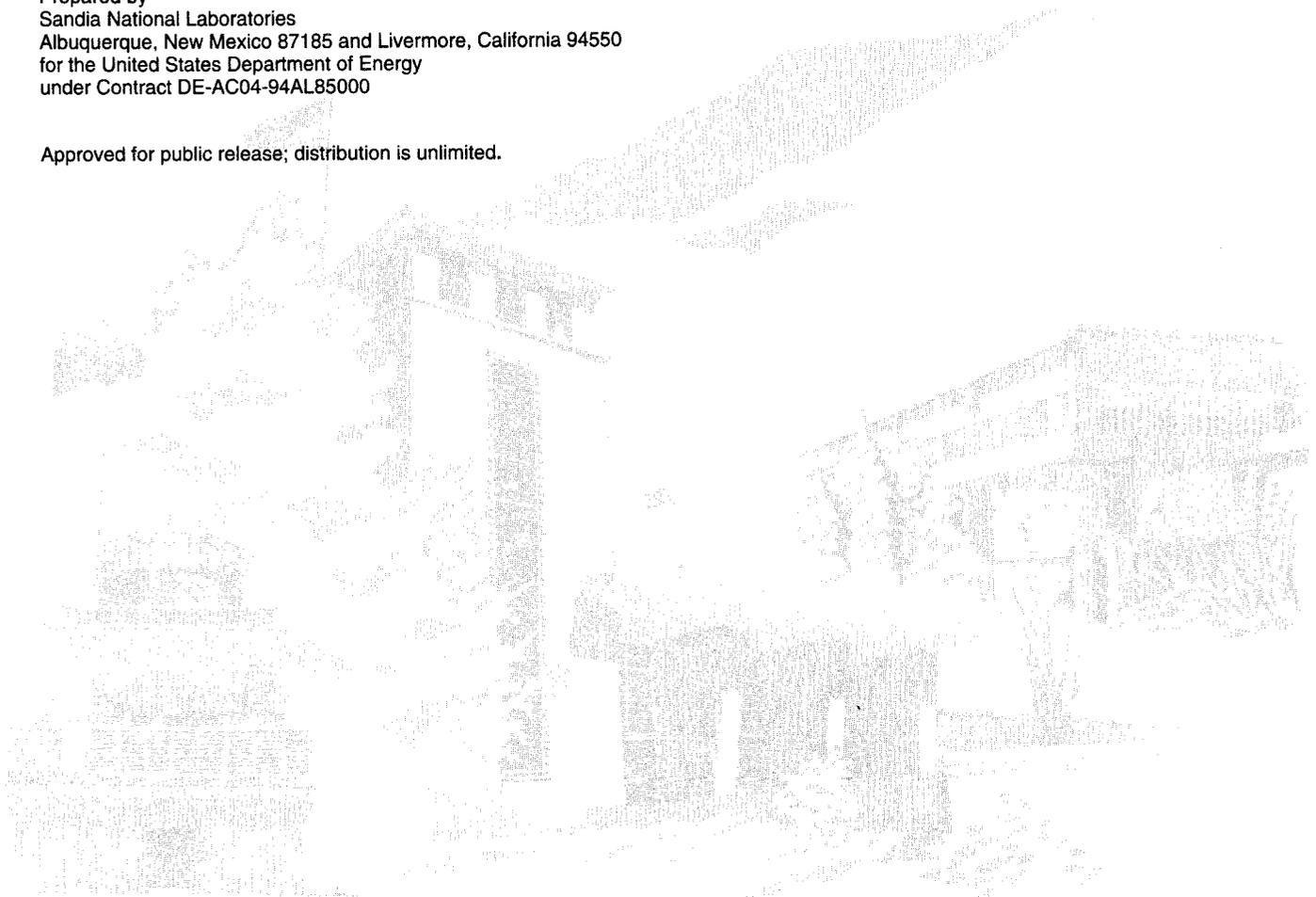
Printed October 1996

A Comparison of Spotlight Synthetic Aperture Radar Image Formation Techniques

Curtis D. Knittle, Neall E. Doren, Charles V. Jakowatz

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550
for the United States Department of Energy
under Contract DE-AC04-94AL85000

Approved for public release; distribution is unlimited.



Issued by Sandia National Laboratories, operated for the United States Department of Energy by Sandia Corporation.

NOTICE: This report was prepared as an account of work sponsored by an agency of the United States Government. Neither the United States Government nor any agency thereof, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness, or usefulness of any information, apparatus, product, or process disclosed, or represents that its use would not infringe privately owned rights. Reference herein to any specific commercial product, process, or service by trade name, trademark, manufacturer, or otherwise, does not necessarily constitute or imply its endorsement, recommendation, or favoring by the United States Government, any agency thereof or any of their contractors or subcontractors. The views and opinions expressed herein do not necessarily state or reflect those of the United States Government, any agency thereof or any of their contractors.

Printed in the United States of America. This report has been reproduced directly from the best available copy.

Available to DOE and DOE contractors from
Office of Scientific and Technical Information
PO Box 62
Oak Ridge, TN 37831

Prices available from (615) 576-8401, FTS 626-8401

Available to the public from
National Technical Information Service
US Department of Commerce
5285 Port Royal Rd
Springfield, VA 22161

NTIS price codes
Printed copy: A03
Microfiche copy: A01

A Comparison of Spotlight Synthetic Aperture Radar Image Formation Techniques

Curtis D. Knittle, Neall E. Doren and Charles V. Jakowatz
Advanced Analysis Department
Sandia National Laboratories
Albuquerque, NM 87185

Abstract

Spotlight synthetic aperture radar images can be formed from the complex phase history data using two main techniques: 1) polar-to-cartesian interpolation followed by two-dimensional inverse Fourier transform (2DFFT), and 2) convolution backprojection (CBP). CBP has been widely used to reconstruct medical images in computer aided tomography, and only recently has been applied to form synthetic aperture radar imagery. It is alleged that CBP yields higher quality images because 1) all the Fourier data are used and 2) the polar formatted data is used directly to form a 2D cartesian image and therefore 2D interpolation is not required.

This report compares the quality of images formed by CBP and several modified versions of the 2DFFT method. We show from an image quality point of view that CBP is equivalent to first windowing the phase history data and then interpolating to an exscribed rectangle. From a mathematical perspective, we should expect this conclusion since the same Fourier data are used to form the SAR image.

We next address the issue of parallel implementation of each algorithm. We dispute previous claims that CBP is more readily parallelizable than the 2DFFT method. Our conclusions are supported by comparing execution times between massively parallel implementations of both algorithms, showing that both experience similar decreases in computation time, but that CBP takes significantly longer to form an image.

This page intentionally left blank

Contents

1	Introduction	1
2	Two-Dimensional FFT Method	2
3	Convolution Backprojection Method	4
4	Modified CBP for SAR Image Formation	6
4.1	Discrete Implementation Issues	7
5	SAR Simulations	8
5.1	Comparisons	10
5.2	One Dimensional Interpolation of Filtered Projections	18
6	Massively Parallel Implementations	24
7	Conclusions	27

Figures

1	Fourier-domain annulus region where SAR collects data.	1
2	Two steps in interpolation process first yields a keystone grid, then a cartesian grid.	2
3	Grid point locations for inscribed and exscribed rectangles.	4
4	Image and filtered projection arrangement for backprojection step.	6
5	Frequency domain specifications of filter for SAR data.	7
6	Target arrangement on ground patch for simulations.	9
7	Taylor window IPR showing quality measures.	10
8	Shows spatial domain sidelobe structure corresponding to Fourier domain shape.	11
9	Images formed from “A” parameters. (a) IW. (b) XW. (c) CBP. (d) WX.	12
10	Images formed from “D” parameters. (a) IW. (b) XW. (c) CBP. (d) WX.	13
11	“A” image (a) Center target IPR’s. (b) Upper left target IPR’s. Solid=IW, dash=XW, dot=WX, dot/dash=CBP.	15
12	“D” image (a) Center target IPR’s. (b) Upper left target IPR’s. Solid=IW, dash=XW, dot=WX, dot/dash=CBP.	16
13	Phase history domain images from “D” parameters. (a) WX. (b) CBP.	19
14	Center row from CBP “D” image, $N_s = 332$, $M = 512$	20
15	Center row from CBP “D” image, $N_s = 332$, $M = 1024$	20
16	Center row from CBP “D” image, $N_s = 332$, $M = 2048$	21
17	Center row from CBP “D” image, $N_s = 332$, $M = 4096$	21
18	CBP-formed images (a) “D” image with $M = 512$. (b) “D” image with $M = 4096$	23
19	Projection function for image with single impulse.	24
20	Center row from CBP “D” image, $N_s = 332$, $M = 4096$. Solid line=nearest neighbor interpolation; dashed line=linear interpolation.	25

Tables

1	IPR Measurements for “A” Image Data	17
2	IPR Measurements for “B” Image Data	17
3	IPR Measurements for “C” Image Data	17
4	IPR Measurements for “D” Image Data	18
5	Running Time (sec) for CBP and 2DFFT on a 1024 Node nCUBE2	27

1 Introduction

The tomographic viewpoint of spotlight mode synthetic aperture radar has established that a demodulated, reflected pulse represents a bandpass-filtered radial slice of the two-dimensional Fourier transform of the ground-patch reflectivity [1]. The angle of the radial slice is equal to the slant plane squint angle of the radar at the time the pulse was transmitted. The nonzero region of the radial slice (i.e. the passband) is centered at the radar carrier frequency, f_c , and the width is governed by the bandwidth of the transmitted pulse, Δf . (We assume throughout that frequencies are in spatial units, i.e. $f_c = \frac{2}{c}(\text{carrier in Hz})$). The collection of demodulated returns gathered over the entire synthetic aperture yields Fourier data in the annulus segment shown in Fig. 1. Pulses are transmitted at discrete points along the aperture, so consequently the total angle subtended by the synthetic aperture is sampled according to the pulse repetition frequency of the radar and the velocity of the radar platform. Similarly, the demodulated return pulses are sampled for storage and processing. Hence, the collection of return pulses actually represents a sampled version of the two-dimensional Fourier transform of the ground-patch reflectivity, but only data in the annulus segment shown in Fig. 1 is nonzero. The radial dimension is alternatively called range, while cross-range or azimuth refers to the dimension orthogonal to range.

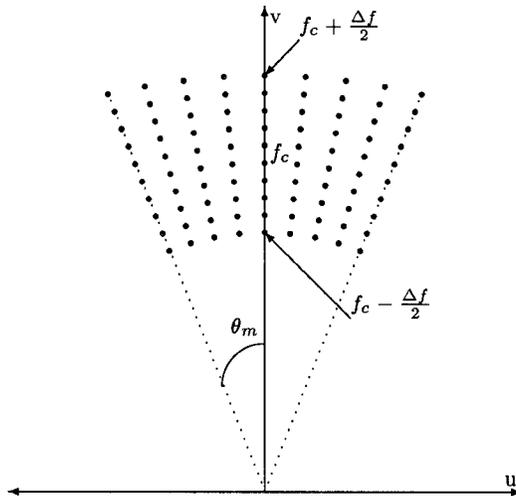


Figure 1: Fourier-domain annulus region where SAR collects data.

Whether the viewpoint is tomographic or Doppler, the image formation problem becomes one of inverting offset Fourier data that is recorded on a polar grid. One inversion method, called the 2DFFT method, strives to exploit the speed of the Fast Fourier Transform (FFT) algorithm by reformatting the data on the polar grid to lie on a cartesian grid. This “polar reformatting” is then followed by a two-dimensional inverse FFT (IFFT) to obtain the image.

More recently, a method known as convolution backprojection (CBP) has been proposed for spotlight mode SAR image reconstruction [1], [2]. CBP uses the data on the polar grid directly to form a cartesian image, and allegedly yields a higher quality image because all the Fourier data are used and two-dimensional interpolation is not required.

This report describes in detail the implementation issues of the CBP algorithm, and more importantly addresses some of the claims made in [2]. In particular, we examine the conclusion that CBP provides better image quality than 2DFFT because “polar-to-cartesian interpolation is computationally intensive and error prone due to interpolation inaccuracies” [2] and “polar-to-rectangular interpolation, which limits the achievable resolution, and therefore, the quality of the final image” [2]. We also dispute the assertion that CBP is better suited to parallel implementation, and show by way of examples that linear interpolation (and certainly nearest neighbor) is not acceptable under certain conditions. This report accomplishes these goals in the following manner. In the next section, a brief overview of the 2DFFT method is given, followed by the theoretical basis and implementation issues of CBP. Following the CBP presentation, we direct attention to SAR image quality obtained using each method. A comparison using simulated SAR data is performed using more meaningful quality measures than that used in [2], and we show experimentally and theoretically that CBP does not yield higher quality images than the 2DFFT method. Then, we examine the parallel implementation of both image formation methods and show that the 2DFFT method is also inherently parallel. We use actual image formation times to reinforce our conclusion that both algorithms experience equivalent increases in speed. We summarize our results and offer additional conclusions at the end of this report.

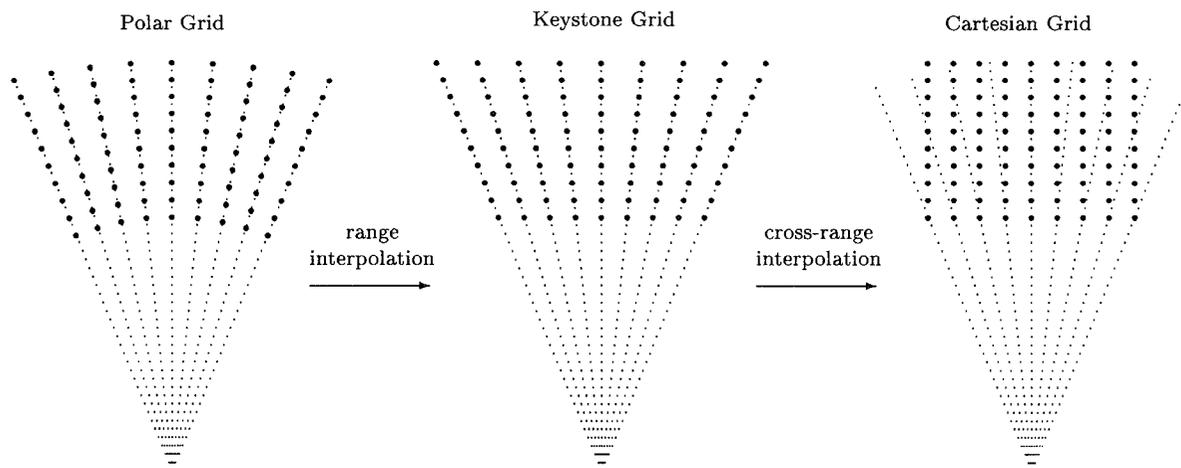


Figure 2: Two steps in interpolation process first yields a keystone grid, then a cartesian grid.

2 Two-Dimensional FFT Method

The 2DFFT method is very straight forward in concept. Given Fourier data lying on a polar grid, two-dimensional interpolation techniques are used to derive values on a cartesian grid. Once data are available on a cartesian grid, the data are windowed and a 2D IFFT

is performed to obtain spatial domain data, also on a cartesian grid. The magnitude of the spatial domain data is then displayed on any suitable monitor.

The interpolation step is usually computationally expensive and must be very accurate. Several techniques exist to accomplish polar-to-cartesian reformatting [3]. The algorithm of choice for this report uses separable one-dimensional, windowed, sinc functions. The first step, shown in Fig. 2, involves interpolating in the radial, or range, dimension to obtain a keystone grid, which yields Fourier data on equidistant horizontal lines. The second step interpolates data on the keystone grid to derive data on equidistant vertical lines, thus yielding data on a cartesian grid. This step is also shown in Fig. 2. More details of polar-to-cartesian interpolation are given in [4, 5, 6].

We must address the contention in [2] that interpolation is “inaccurate”. If we are to believe the sampling theorem, which states that any continuous analog signal can be exactly reconstructed by samples of the signal taken at the Nyquist rate, then it should follow that correctly reconstructing the analog signal from its samples, and then resampling at the desired locations should be perfectly acceptable. “Correctly” reconstructing the original signal implies using an infinite-length *sinc()* interpolator. Infinite-length *sinc()* interpolation in the Fourier domain corresponds to multiplying the terrain of interest in the spatial domain by an ideal 2D spatial rectangular window. Truncating an infinite-length *sinc()* interpolator to a k -length interpolator is equivalent to multiplying the infinite-length interpolator by a k -length Fourier domain rectangular window. In the spatial domain, this corresponds to convolving the ideal spatial domain rectangular window with the inverse Fourier transform of the k -length Fourier domain rectangular window. The effect is to introduce the Gibbs phenomenon at the edges and to alter the amplitude of the 2D spatial domain rectangular passband. The edge effects cause aliased energy to be folded back into the terrain of interest, while the amplitude effects cause the magnitude of the image to be weighted differently. The severity of both these effects depends on the length of the Fourier domain window, or, on the length of the interpolator. In addition, windows tapered at the edges, like Taylor, Hamming, etc., are applied to the *sinc()* interpolator to reduce the effects just described. Based on this qualitative description of interpolation, one may conclude that interpolation causes errors near the patch edges due to aliased energy folding into the patch, but to state that interpolation is inaccurate and yields poorer results across the entire spatial domain image would be “inaccurate”.

After the data has been reformatted to a rectangular cartesian grid, a window such as a Taylor window is applied to each dimension before performing a 2D IFFT. Note that there are many rectangular grid sizes that the polar data could be interpolated to. Two of these are shown in Fig. 3, where in one case a rectangular region is *inscribed*, and in the other a rectangular region is *exscribed*. Significant differences exist in the spatial domain image depending on whether inscribing or exscribing is used, and furthermore, where the windowing step is accomplished. One can see that when a rectangle is exscribed, some of the grid points in the lower corners will be zero because the data in this region is zero. However, only in the exscribed case is *all* the polar data used in the image formation; the inscribed case does not use all the polar formatted data. More will be said about this later in this report.

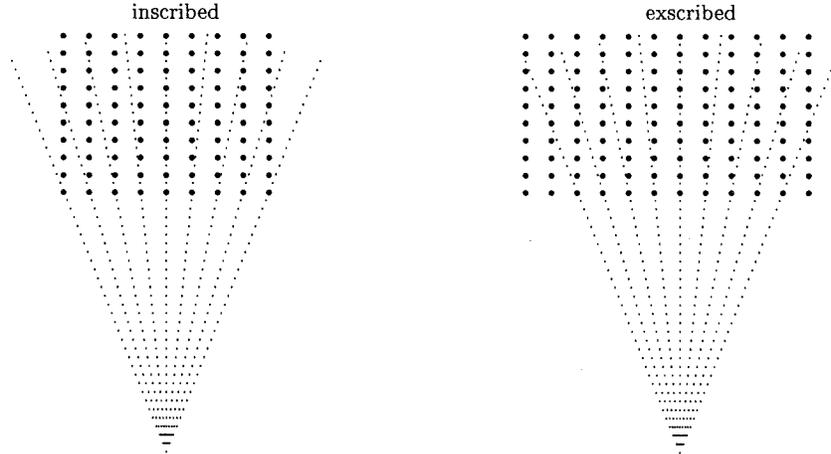


Figure 3: Grid point locations for inscribed and exscribed rectangles.

3 Convolution Backprojection Method

Convolution backprojection has historically been used to form images in computer aided tomography (CAT) [7, 8, 9]. We begin presenting the theoretical basis for CBP by writing the 2D Fourier transform pair:

$$G(u, v) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} g(x, y) e^{-j2\pi(ux+vy)} du dv \quad (1)$$

$$g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G(u, v) e^{j2\pi(ux+vy)} dx dy \quad (2)$$

where $g(x, y)$ represents the 2D spatial complex reflectivity function of the ground patch being imaged, and $G(u, v)$ is the cartesian Fourier transform of $g(x, y)$. It can be shown that if $G(\rho, \theta)$ is the polar representation of the 2D Fourier transform of $g(x, y)$, then the inverse Fourier transform Eq. 2 can be written as

$$g(x, y) = \int_0^{\pi} \int_{-\infty}^{\infty} G(\rho, \theta) |\rho| e^{j2\pi(\rho x \cos(\theta) + \rho y \sin(\theta))} d\rho d\theta, \quad (3)$$

Where θ is measured with respect to the positive abscissa axis. If we define a variable t such that

$$t = x \cos(\theta) + y \sin(\theta) \quad (4)$$

then Eq. 3 becomes

$$g(x, y) = \int_0^{\pi} \int_{-\infty}^{\infty} G(\rho, \theta) |\rho| e^{j2\pi\rho t} d\rho d\theta. \quad (5)$$

If we further define the “filtered projection” function $q_{\theta}(t)$ as

$$q_{\theta}(t) = \int_{-\infty}^{\infty} G(\rho, \theta) |\rho| e^{j2\pi\rho t} d\rho, \quad (6)$$

then Eq. 5 becomes

$$g(x, y) = \int_0^\pi q_\theta(t) d\theta. \quad (7)$$

The two steps represented by Eqs. (6) and (7) define the CBP algorithm and allow a cartesian spatial domain image to be formed directly from polar Fourier domain data. Note that Eqs. (2) and (5) are essentially mathematical identities. That is,

$$\int_{-\infty}^{\infty} \int_{-\infty}^{\infty} G(u, v) e^{j2\pi(ux+vy)} dx dy = \int_0^\pi \int_{-\infty}^{\infty} G(\rho, \theta) |\rho| e^{j2\pi\rho t} d\rho d\theta. \quad (8)$$

The 2D function $f(x, y)$ can be formed using either Eq. (2) or Eq. (5), but theoretically the results should be identical!

Equation (6) represents the “convolution” portion of the CBP algorithm because $q_\theta(t)$ is nothing more than a linear filtering operation, i.e. a convolution. To see this, note that for a particular value of θ , the radial slice of the Fourier transform, $G(\rho, \theta)$, can be considered a one-dimensional function of ρ ; call it $P_\theta(\rho)$. Equation (6) directs us to multiply $P_\theta(\rho)$ by $|\rho|$ and then perform an inverse Fourier transform on the product. This is frequency domain filtering. The equivalent time domain representation of Eq. (6) is

$$q_\theta(t) = p_\theta(t) * h(t) \quad (9)$$

where $*$ denotes convolution,

$$p_\theta(t) = \int_{-\infty}^{\infty} P_\theta(\rho) e^{j2\pi\rho t} d\rho. \quad (10)$$

and

$$h(t) = \int_{-\infty}^{\infty} |\rho| e^{j2\pi\rho t} d\rho. \quad (11)$$

That is, $h(t)$ is simply the impulse response of a filter whose frequency response is

$$H(\rho) = |\rho|. \quad (12)$$

After computing $q_\theta(t)$, this function must be “backprojected”, as specified by Eq. (7). The backprojection operation can be described by considering Eq. (7) together with Fig. 4, which shows a filtered projection for the angle θ . For every point on the cartesian grid, the value of t is computed, i.e.

$$t_1 = x_1 \cos(\theta) + y_1 \sin(\theta). \quad (13)$$

The contribution to $g(x_1, y_1)$ made by this projection function is $q_\theta(t_1)$ where t_1 is given by Eq. (13). Note that the point (x_2, y_2) will also give the same value of t , and thus the contribution to $g(x_2, y_2)$ is also $q_\theta(t_1)$. In fact, for this value of θ , every point on the line defined by the two points (x_1, y_1) and (x_2, y_2) will receive the same contribution from $q_\theta(t)$. In other words, all grid points lying on a line oriented at an angle θ with respect to the horizontal axis, and whose orthogonal distance to the origin is t_1 , will receive the same contribution from $q_\theta(t)$, namely $q_\theta(t_1)$. In this sense, the filtered projection is “backprojected” or “smeared” back onto the cartesian image. The integration of all filtered projections backprojected onto the image yields the final result. Note that for only one angle do the points (x_1, y_1) and (x_2, y_2) receive the same contribution.

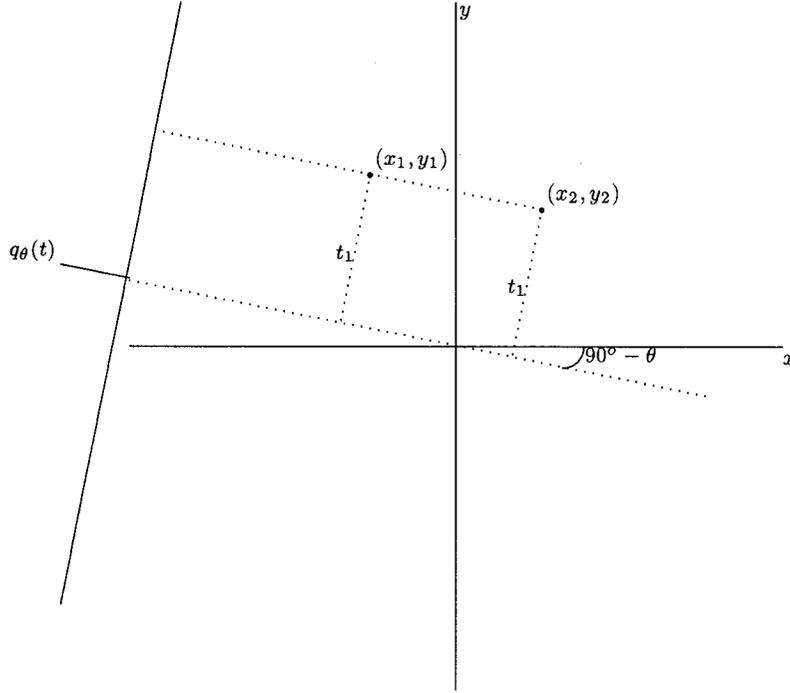


Figure 4: Image and filtered projection arrangement for backprojection step.

4 Modified CBP for SAR Image Formation

When considering data collected by a SAR, several modifications are made to the fundamental equations. Since data is nonzero only in the region shown in Fig. 1, the limits of integration in Eq. (5) must be changed. Specifically, Eq. (5) becomes

$$g(x, y) = \int_{\frac{\pi}{2}-\theta_m}^{\frac{\pi}{2}+\theta_m} \int_{f_c-\frac{\Delta f}{2}}^{f_c+\frac{\Delta f}{2}} G(\rho, \theta) |\rho| e^{j2\pi\rho t} d\rho d\theta. \quad (14)$$

One more simplification can be made by a simple change of variables:

$$g(x, y) = \int_{\frac{\pi}{2}-\theta_m}^{\frac{\pi}{2}+\theta_m} \int_{-\frac{\Delta f}{2}}^{\frac{\Delta f}{2}} G(\rho + f_c, \theta) |\rho + f_c| e^{j2\pi\rho t} d\rho e^{j2\pi f_c t} d\theta. \quad (15)$$

With these steps, the filtered projection function is

$$q_\theta(t) = \int_{-\frac{\Delta f}{2}}^{\frac{\Delta f}{2}} G(\rho + f_c, \theta) |\rho + f_c| e^{j2\pi\rho t} d\rho, \quad (16)$$

and the backprojection equation becomes

$$g(x, y) = \int_{\frac{\pi}{2}-\theta_m}^{\frac{\pi}{2}+\theta_m} q_\theta(t) e^{j2\pi f_c t} d\theta. \quad (17)$$

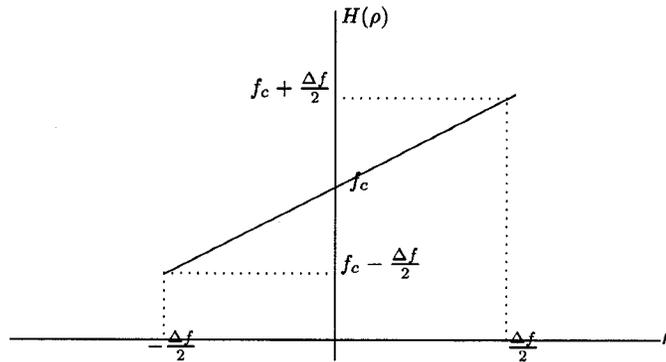


Figure 5: Frequency domain specifications of filter for SAR data.

4.1 Discrete Implementation Issues

We begin discussing implementation issues by first considering the filtered projection Eq. (16). The change of variables performed in the previous paragraph shifted the Fourier data from a center frequency of f_c to baseband. Thus, a one-dimensional FFT can be used to obtain $q_\theta(t)$ after the filtering operation. Note that due to the shift in frequency, the Fourier data must be multiplied by a filter whose frequency characteristics match those shown in Fig. 5 (i.e. a shifted segment of $|\rho|$).

Assuming that there are N_s samples per pulse, the interval between samples in the radial dimension is

$$\delta_\rho = \frac{\Delta f}{N_s}. \quad (18)$$

To use the FFT algorithm, the N_s samples must be zero-padded to a length which is a power of two. That is, we choose the FFT length to be M , where $2^M \geq N_s$. As a result, the spacing between samples of $q_\theta(t)$ will be

$$\delta_t = \frac{1}{M\delta_\rho} = \frac{N_s}{M\Delta f}, \quad (19)$$

and therefore the values $q_\theta(n\delta_t)$ will be available for use in the backprojection step. Note that both positive and negative values of n are available.

Conversion of the backprojection step is straight forward. Since the angle subtended by the radar is actually sampled, the backprojection formula Eq. (17) becomes

$$g(x, y) = \sum_{i=1}^{N_p} q_{\theta_i}(t) e^{j2\pi f_c t} \quad (20)$$

where θ_i represents the angle of the i^{th} pulse, and N_p is the number of pulses transmitted during data collection. Normally there is a scale factor in front of the summation, but removing it is inconsequential since all pixels are scaled by the same value.

During the backprojection step, the spacing between pixels in the spatial domain can be arbitrarily assigned. For example, if δ_x and δ_y represent scale factors for the x and y

dimensions, respectively, then the value of t for some grid location (x, y) is computed using

$$t = x\delta_x \cos(\theta) + y\delta_y \sin(\theta). \quad (21)$$

One can see that if x and y are in units of “pixels”, and δ_x and δ_y are in units of “distance/pixel”, then the units of t will be distance. One may choose δ_x and δ_y to “zoom in” or “zoom out” to any level desired.

Once the value of t is computed via Eq. (21), a decision must be made about what value is backprojected to the point (x, y) . One can easily see that it is likely $t \neq n\delta_t$ for all values of n . Therefore, one-dimensional interpolation must be performed to obtain the value $q_\theta(t)$, which is between $q_\theta(\hat{n}\delta_t)$ and $q_\theta((\hat{n} \pm 1)\delta_t)$, where

$$\hat{n} = \text{integer value of } \left(\frac{t}{\delta_t}\right), \quad (22)$$

and $+$ is used if $t \geq 0$ and $-$ is used if $t < 0$.

The one-dimensional interpolation step mentioned in the previous paragraph is a critical issue. It is believed in [2] that linear interpolation or nearest neighbor assignment can be used to obtain $q_\theta(t)$. We believe that nearest neighbor assignment is not conducive to SAR image formation because this nonlinear operation would inject phase errors into the data. In addition, linear interpolation may be used but only if $\delta_t \ll 1$. Examples of the effects of linear interpolation will be shown in a later section.

Multiplication by $e^{j2\pi f_c t}$ in Eq. (20) is also a critical step. Unlike the 2DFFT method, where a shift of the data to the origin does not affect the magnitude of the spatial domain image, the complex exponential in Eq. (20) is a function of θ and therefore must be included in the integration over θ . Interpolation is not required to obtain the proper value of $e^{j2\pi f_c t}$; use the exact value for t from Eq. (21).

5 SAR Simulations

We have the benefit of using a spotlight mode synthetic target generator to simulate actual SAR data collections. All required parameters for SAR data collection can be specified; some of these include bandwidth, center frequency, chirp rate, pulse repetition frequency, collection geometry, and the number and location of as many targets desired. The simulator provides phase history data (i.e. Fourier domain data) as though it were a real SAR.

If there are differences in image quality depending on the formation technique used, we are more likely to see the variations if we have high resolution. For this reason, we maintain the SAR parameters to provide one foot (0.3m) resolution in both range and azimuth (cross-range). We accomplish 0.3m resolution in range by using a pulse bandwidth of 500 MHz, which translates to $\Delta f = \frac{2}{c}5 \times 10^8 = 3.3333\text{m}^{-1}$. If γ_r represents the achievable range resolution, then

$$\gamma_r \approx \frac{1}{\Delta f} = 0.3\text{m}. \quad (23)$$

The achievable azimuth resolution, γ_a , is approximately given by [1]

$$\gamma_a \approx \frac{c}{4f_c\theta_m}, \quad (24)$$

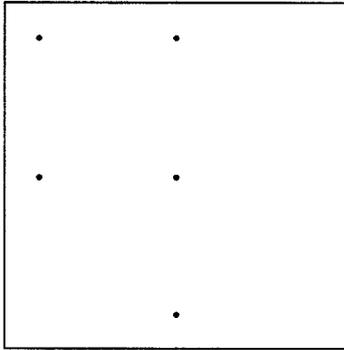


Figure 6: Target arrangement on ground patch for simulations.

where Δf , f_c , and θ_m are shown in Fig. 1. We choose four sets of pairs (f_c, θ_m) to maintain $\gamma_a \approx 0.3\text{m}$. These are: A- $(223.3\text{m}^{-1}, 0.4276^\circ)$, B- $(100\text{m}^{-1}, 0.9549^\circ)$, C- $(50\text{m}^{-1}, 1.9102^\circ)$, and D- $(25\text{m}^{-1}, 4.000^\circ)$.

A standard arrangement of targets on the ground is used for each simulation. The arrangement, shown in Fig. 6, encompasses all the extreme locations where a target may be located, i.e. near range, patch center, far range, azimuth edge, and a combination of far range - azimuth edge. It has been our experience that targets near the center of the patch behave nicely under most circumstances, so we include targets at the patch edges.

We form the simulated images exactly like we form non-simulated images. This is an important distinction from the work done in [2], where the grid size was chosen so that the nulls of the sidelobes fall on a grid point, thereby entirely removing sidelobes from the image. Sidelobes are an important and unavoidable consequence of using finite length data and therefore should be examined along with the mainlobe.

When we measure the quality of an image, we avoid using the multiplicative noise ratio [2] because it ignores sidelobe issues. Instead, we use three quality measures that we judge to represent how closely a target resembles an impulse: -3 dB width, -18 dB width, and peak-to-sidelobe (PSL) ratio. To illustrate our use of these measures, refer to Fig. 7, which shows the impulse response (IPR) of a Taylor window. The plots are always normalized so that the peak corresponds to 0 dB. Therefore, the PSL ratio can be measured by finding the level of the highest sidelobe. In the example of Fig. 7, the IPR has the following characteristics: PSL ratio is -40 dB, -3 dB width is 1.25, -18 dB width is 2.8125.

A target's IPR is measured by first extracting a slice of the image through the target, normally an azimuth (horizontal) slice or a range (vertical) slice. Then an FFT of the slice is computed, the Fourier sequence is zero padded to a longer length, and an inverse FFT is computed, the magnitude of which is displayed. Zero padding to a longer length interpolates between samples in the spatial domain and yields a smoother plot. The reason an azimuth or range IPR is studied is because the sidelobes are usually in the azimuth and range dimension. This is because sidelobes extend in a direction orthogonal to the discontinuities in the Fourier domain. For example, in the case of an inscribed rectangle, the Fourier domain edges are both horizontal and vertical, which gives rise to vertical and horizontal sidelobes, respectively, in the spatial domain. On the other hand, if the nonzero region in the Fourier domain were

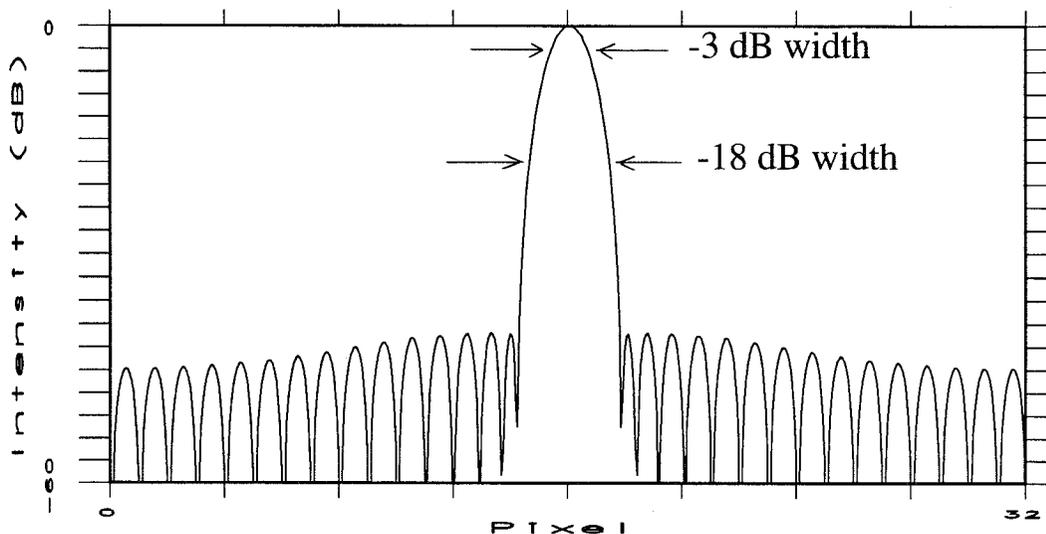


Figure 7: Taylor window IPR showing quality measures.

shaped similar to a trapezoid (as it is in the case of exscribing a rectangle), then multiple sets of sidelobes appear in the spatial domain image. This situation is depicted in Fig. 8.

It would not be fair to compare an azimuth IPR (horizontal slice) from the two situations shown in Fig. 8 because in the upper case, the IPR continues to slice through the sidelobes, while this is not the case for the lower sidelobe structure. A slight modification is required to our previous definition of an IPR plot to equalize the contribution from the sidelobes in either case. In essence, sidelobes represent a deviation from a perfect point target (an impulse), and they also interfere with other targets close by. For example, if a target is located on a sidelobe of a nearby target, the magnitude of the target could be artificially boosted or reduced depending on whether the sidelobe interfered constructively or destructively. Consequently, the most important sidelobe to measure is the maximum sidelobe in a certain region about the target. For this reason, when computing an azimuth IPR, for example, for a particular azimuth position (column), several range lines (rows) above and below the target range line will be scanned for the maximum magnitude. The maximum magnitude is used in the IPR plot. Using the new definition of the IPR will not alter the appearance of an IPR for a target generated by an inscribed rectangle because the largest magnitudes already lie on the horizontal line intersecting the target. However, the new IPR definition will drastically alter an IPR from an exscribed rectangle because it will use the maximum magnitude from either of the diagonal sidelobe traces.

5.1 Comparisons

For each set of SAR parameters, we form four different images. The steps taken to form the first image include polar-to-cartesian interpolation in an inscribed rectangle, followed by windowing, then computing the 2D inverse FFT. We refer to the formed image as an “IW” image (inscribe then window). An “XW” image (exscribe then window) is formed using interpolation to an exscribed rectangle, windowing, and 2D inverse FFT. A “WX” image

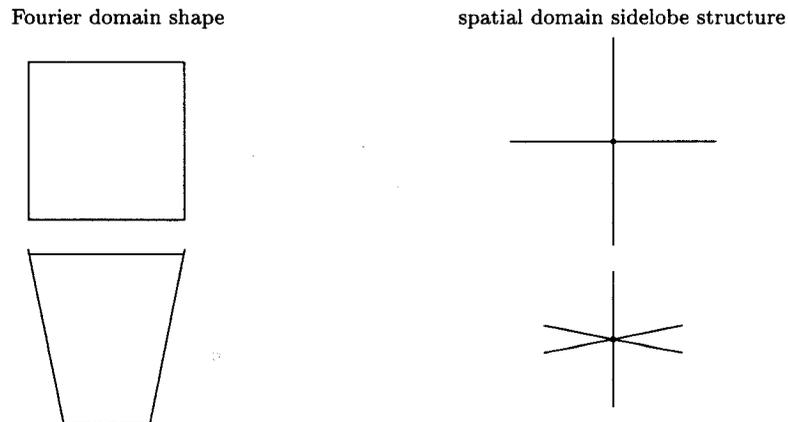


Figure 8: Shows spatial domain sidelobe structure corresponding to Fourier domain shape.

(window then exscribe) is formed by first applying what amounts to a polar windowing scheme, then interpolate to an exscribed rectangle, followed by a 2D inverse FFT. The fourth image is the “CBP” image formed using convolution backprojection. The windows applied were Taylor windows with -40 dB sidelobes.

We routinely use a 17-point sinc() interpolator for polar-to-cartesian interpolation, while in [2], the largest interpolator used was length 14. Judging from the results given in [2], a 17-point interpolator would improve the MNR for the direct Fourier method in each case.

We focus most of our attention on the images formed using the “A”-image parameters, $(f_c, \theta_m) = (223.3\text{m}^{-1}, 0.4276^\circ)$, and the “D”-image parameters, $(f_c, \theta_m) = (25\text{m}^{-1}, 4.000^\circ)$ since these two represent the most extreme angle diversities. The “A” images are shown in Fig. 9, while the “D” images are shown in Fig. 10. Note the sidelobes for each case fan out in a direction orthogonal to the discontinuities in the Fourier domain, as described earlier. Recall that the angle subtended in the “A” images is less than 1° ; hence, the sidelobes appear similar to the inscribed case only because the angle between them is very small.

It is asserted in [2] that one reason CBP renders higher quality images is because *all* the Fourier data are used. They refer to the fact that an inscribed rectangle does not use all the Fourier data in the annulus shown in Fig. 1. This is what the authors of [2] must have meant when they stated that “polar-to-cartesian interpolation limits the achievable resolution, and therefore, the final image quality.” This correlates well with our intuition and also with the fundamental equations given in [1] showing that higher resolution is attainable if larger bandwidths are used. However, as pointed out previously, we are not restricted to interpolate to an inscribed rectangle. Furthermore, from a theoretical point of view, we can not form an image with better resolution than what the size of the collected Fourier data dictates. We can not magically create more Fourier data resulting in an improved image. We can, however, massage the data in different ways to better utilize the structure or format of the data, as CBP appears to do by using the polar formatted data directly. Recall, however, that if the Fourier data is reformatted to an exscribed rectangular grid, or, if CBP is used to form the image, *all the Fourier data are used*. Given the mathematics in Eq. (8), we hypothesized that similar quality images should result *if* the same steps are taken, in the

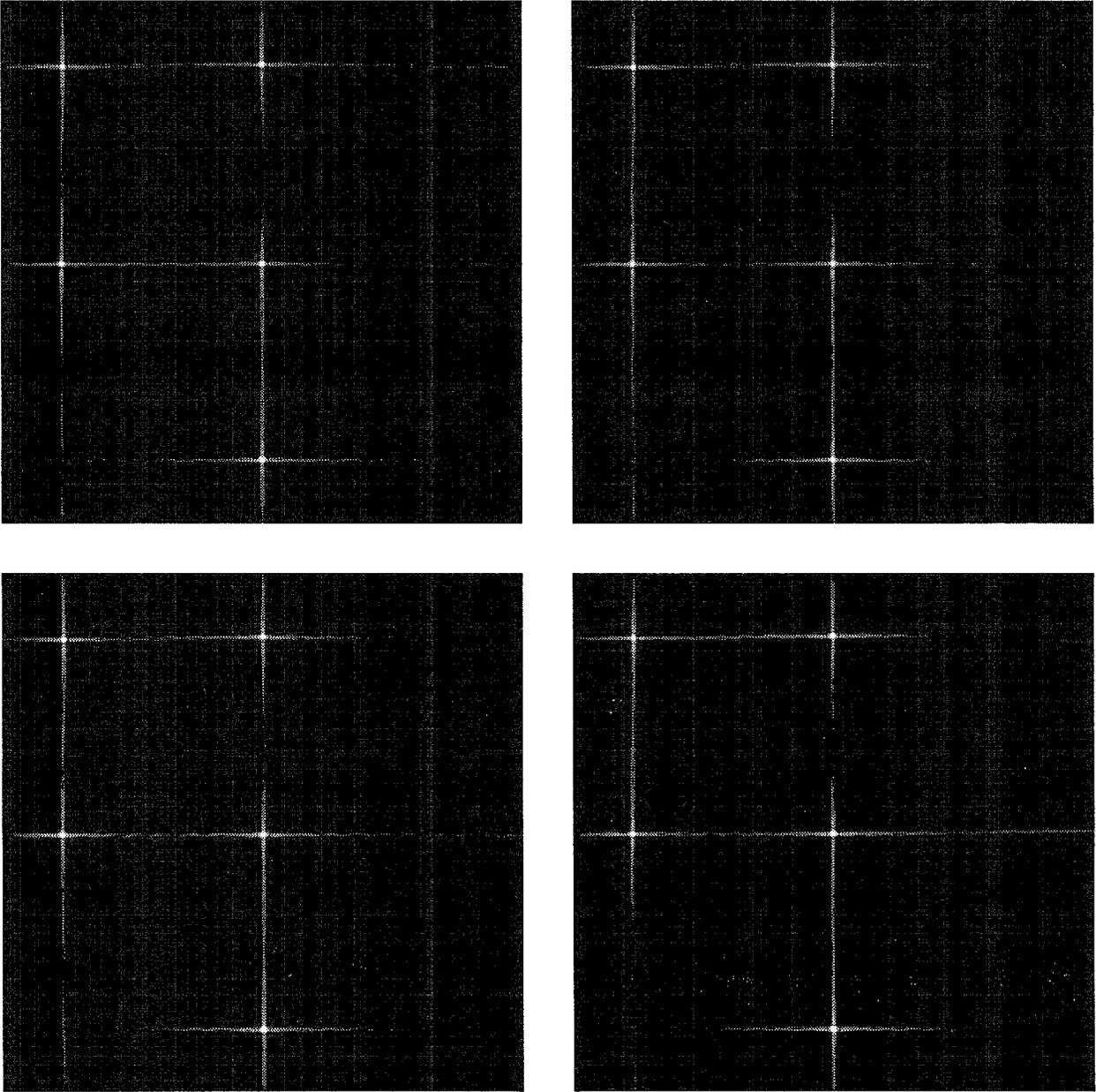


Figure 9: Images formed from "A" parameters. (a) IW. (b) XW. (c) CBP. (d) WX.
(labeled clockwise from upper left)

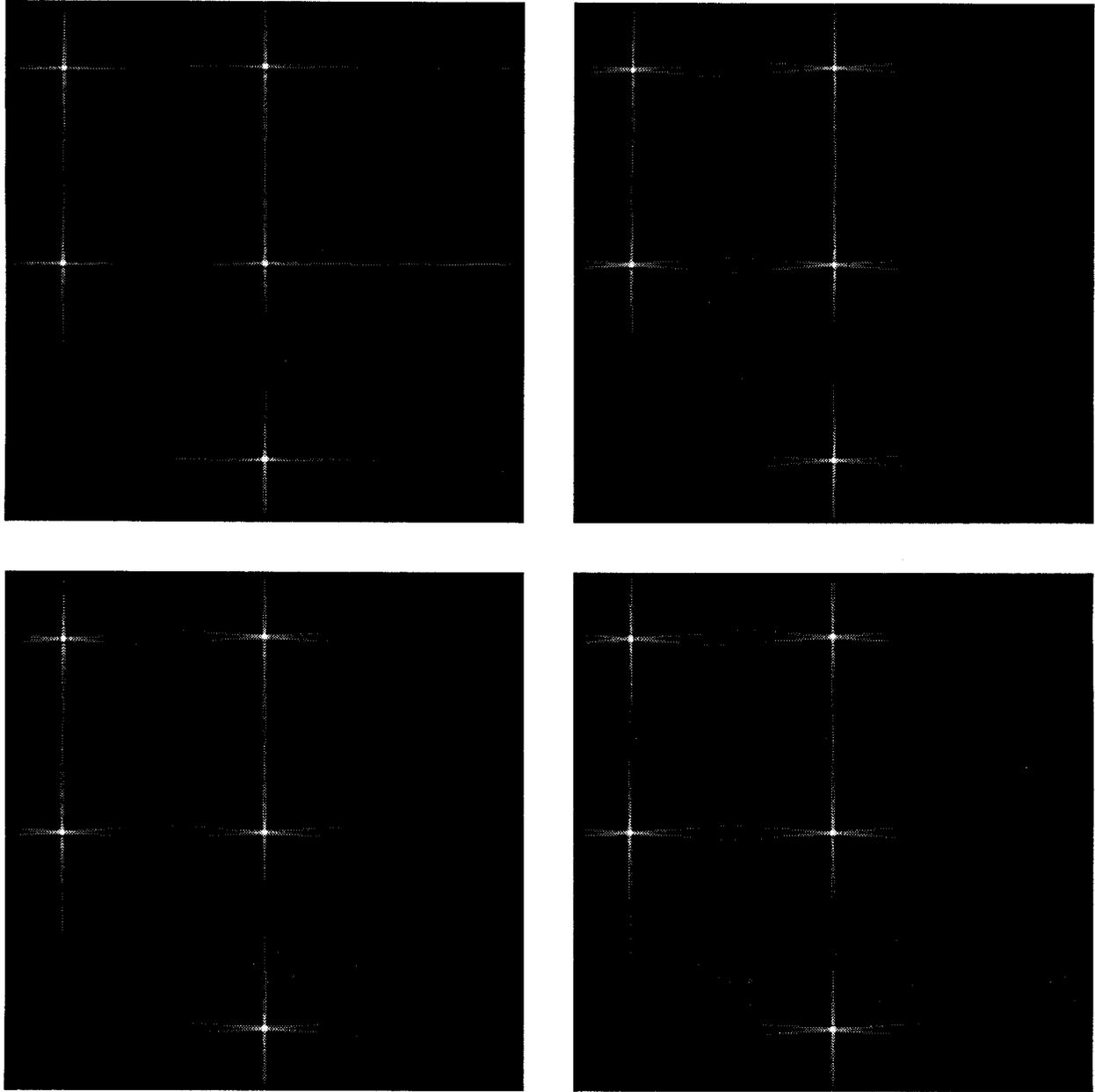


Figure 10: Images formed from “D” parameters. (a) IW. (b) XW. (c) CBP. (d) WX.
(labeled clockwise from upper left)

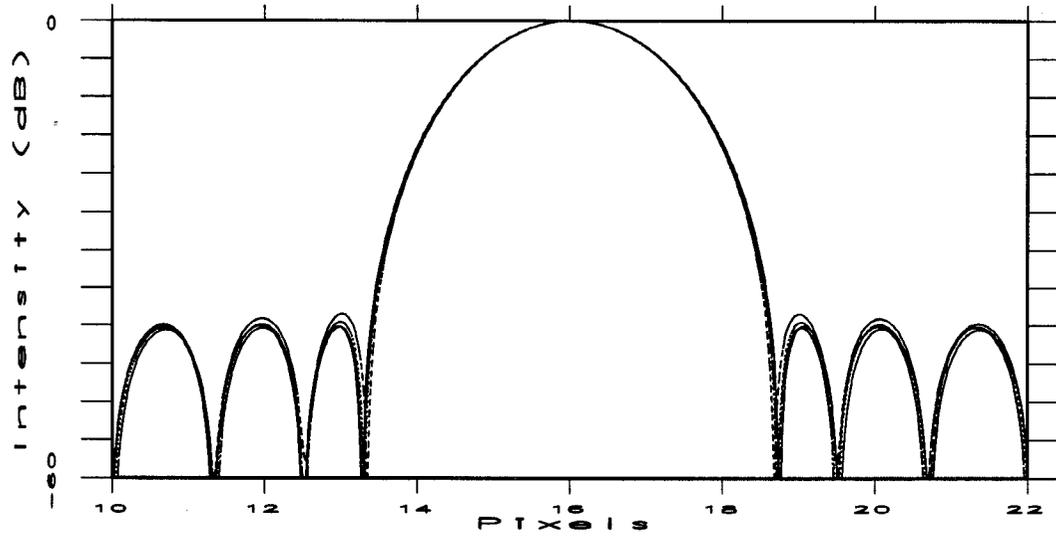
same order, and the same Fourier data are used, regardless of the inverse Fourier transform technique, CBP or interpolation/2DIFFT. This was the motivation for windowing the data *before* reformatting to an exscribed rectangle. In fact, the windowing scheme employed in the WX and CBP images are identical; the first step in either process is to window in the range (radial) dimension and then the cross-range (angular) dimension. If our hypothesis is correct, the difference between the WX and CBP images should be negligible.

Target IPR's from each image formation technique using the "A" parameters are shown in Fig. 11, where the plots in Fig. 11(a) are from the center target and Fig. 11(b) are from the upper left target (azimuth edge-far range). Because the total angle subtended is only $2\theta_m = 0.8552^\circ$, we do not expect much difference between the four image formation techniques. The reason is because the Fourier data patch is nearly rectangular, and inscribing a rectangle does not discard large quantities of Fourier data. The results in Fig. 11 show the following. First, the XW IPR exhibits the narrowest main lobe because the length of the Taylor window is longer in the cross-range dimension, and all the Fourier data are used. It is well known that a longer window provides a narrower main lobe than a shorter window, and, Eq. (24) shows that if more bandwidth is used, better resolution results. While it is difficult to see, the WX and CBP IPR's are overlapping in the main lobe region and provide the second narrowest main lobe. It is true that both these techniques use all the Fourier data, but the windows applied are only as wide as the actual data and therefore are shorter, resulting in a slightly wider main lobe than the XW case. The IW case yields the widest main lobe because less Fourier data are used and the windows are shorter. The second significant result is that while the PSL in the IW, WX, and CBP cases are nearly equal, the peak sidelobe in the XW case is considerably larger. This is due to the discontinuities in each of the lower corners of the Fourier domain patch. The quantitative values for these IPR's are given in Table 1. All these significant results should become more prominent as θ_m is increased.

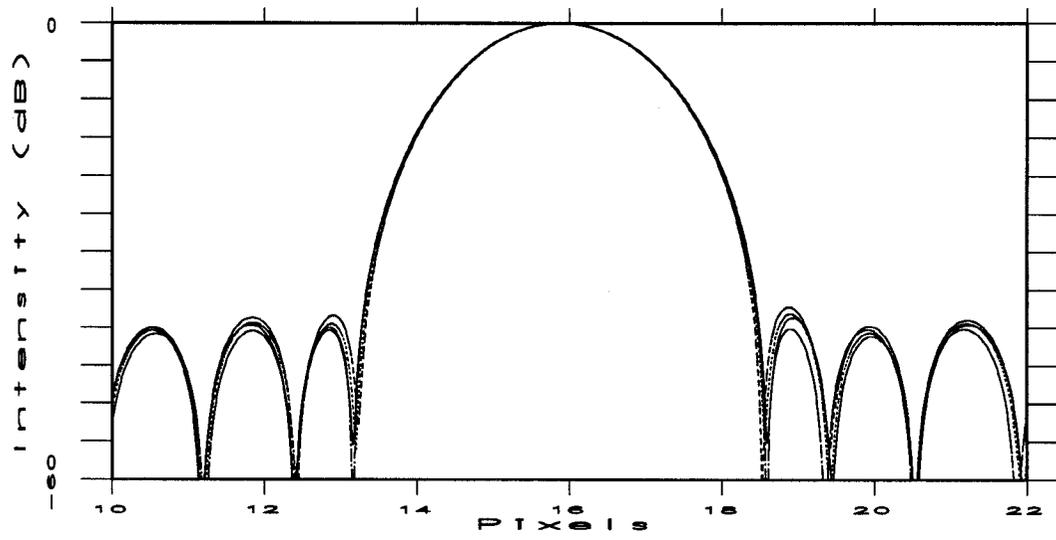
Figures 12(a) and 12(b) show IPR's from the center and upper left targets, respectively, generated using the "D" image parameters. The differences described in the preceding paragraph remain consistent for this case also, although they are more significant, as expected. To summarize, XW shows the narrowest main lobe but the largest PSL, IW exhibits the widest main lobe, and WX and CBP are again overlapping in the main lobe and only slightly different in PSL's. The differences are more pronounced in this case because the angle subtended is now $2\theta_m = 8^\circ$. Inscribing a rectangle discards a large amount of data, thus reducing the bandwidth considerably. Exscribing followed by windowing results in a large discontinuity in the lower corners oriented at angles $\pm\theta_m$ with respect to a vertical axis. It should be obvious that as the angle subtended is increased, this effect is accentuated even more. Quantitative values for the "D" image targets are shown in Table 4.

Although we have examined IPR's from only two targets from the "A" and "D" image data sets, these are representative of the entire data set. We have neglected to show range IPR's because there was no significant difference between any of the images, regardless of the formation technique or SAR parameters.

Based on the preceding results, we conclude that the mathematical identity given in Eq. (8) is maintained in the case of SAR image formation. In retrospect, we should not expect the IW and XW images to produce the same images as WX and CBP for the simple reason that different steps are taken to obtain the final image. Conversely, if the same steps are followed,



(a)



(b)

Figure 11: "A" image (a) Center target IPR's. (b) Upper left target IPR's. Solid=IW, dash=XW, dot=WX, dot/dash=CBP.

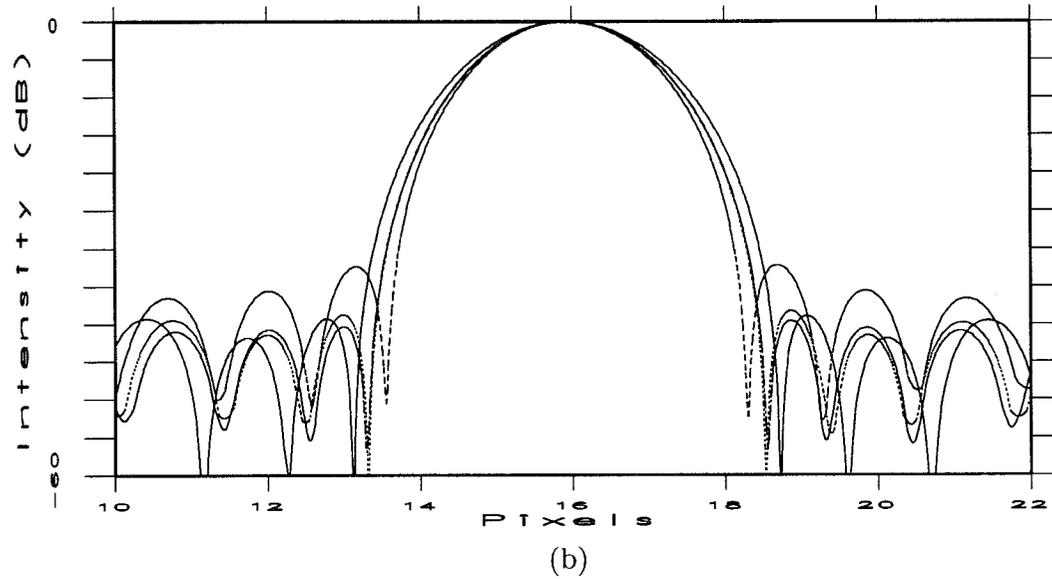
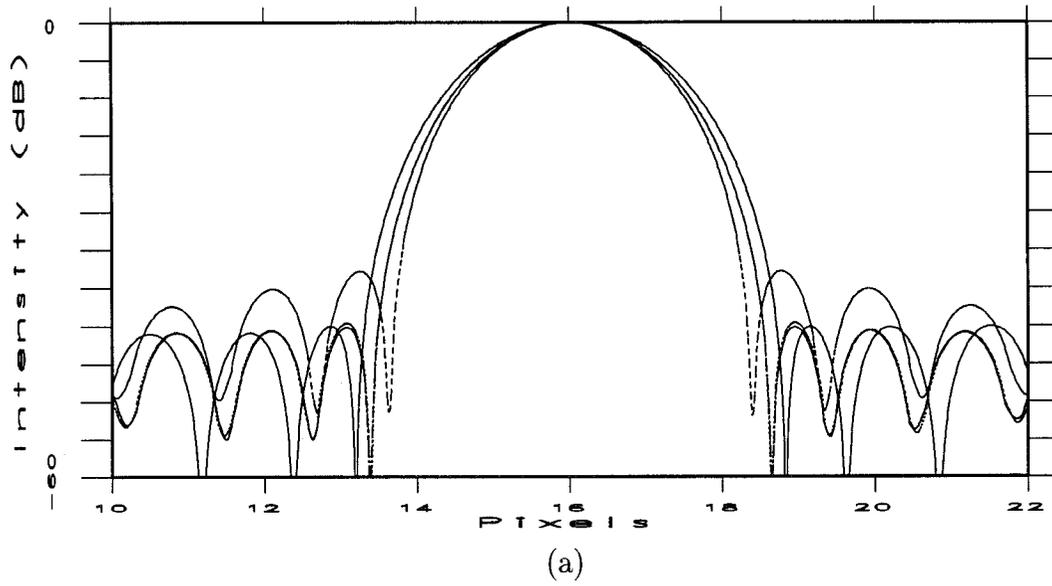


Figure 12: "D" image (a) Center target IPR's. (b) Upper left target IPR's. Solid=IW, dash=XW, dot=WX, dot/dash=CBP.

Table 1: IPR Measurements for "A" Image Data

	IW	XW	WX	CBP
Center Target				
-3dB	1.8612	1.8362	1.8443	1.8417
-18dB	4.1581	4.0943	4.1204	4.1217
PSL	-40.0896	-38.4982	-39.5972	-40.1557
Upper Left Target				
-3dB	1.8625	1.8377	1.8421	1.8407
-18dB	4.1546	4.0897	4.1118	4.1213
PSL	-38.6378	-37.2457	-38.1568	-40.1561

Table 2: IPR Measurements for "B" Image Data

	IW	XW	WX	CBP
Center Target				
-3dB	1.8500	1.8011	1.8189	1.8081
-18dB	4.1350	4.0067	4.0640	4.0450
PSL	-40.0988	-37.3462	-39.6283	-40.2694
Upper Left Target				
-3dB	1.8365	1.7893	1.8047	1.8081
-18dB	4.1088	3.9816	4.0356	4.0435
PSL	-38.2566	-35.8288	-37.7706	-40.2873

Table 3: IPR Measurements for "C" Image Data

	IW	XW	WX	CBP
Center Target				
-3dB	1.9452	1.8434	1.8799	1.8778
-18dB	4.3509	4.0842	4.2002	4.2026
PSL	-39.6552	-35.4441	-39.4776	-40.2289
Upper Left Target				
-3dB	1.9342	1.8348	1.8652	1.8780
-18dB	4.3305	4.0682	4.1768	4.2004
PSL	-38.3770	-34.3389	-37.7368	-40.0441

Table 4: IPR Measurements for “D” Image Data

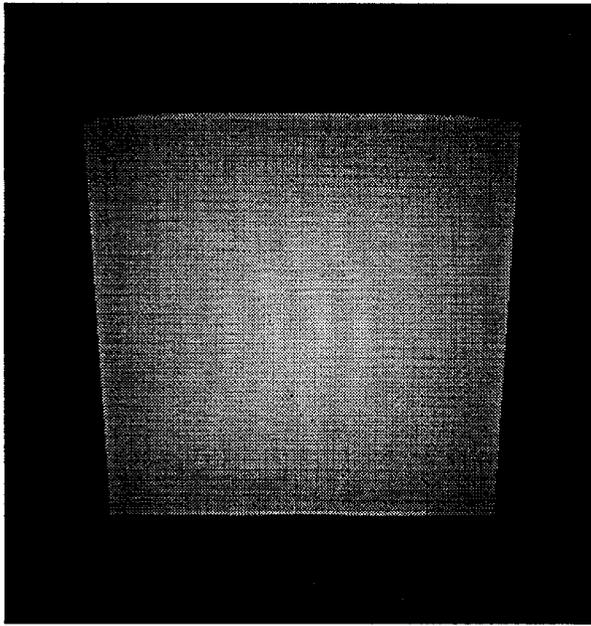
	IW	XW	WX	CBP
Center Target				
-3dB	1.9145	1.7205	1.7886	1.7848
-18dB	4.2874	3.7863	3.9989	3.9954
PSL	-40.0168	-32.7780	-39.6069	-40.1483
Upper Left Target				
-3dB	1.9111	1.7157	1.7790	1.8000
-18dB	4.2750	3.7799	3.9860	4.0127
PSL	-38.8986	-32.1347	-38.2308	-39.5324

i.e. window followed by inverse Fourier transform (either by CBP or interpolation/IFFT), then we should expect, based on Eq. (8), that the resultant images would be nearly identical. The only artifactual differences between the WX and CBP methods are due to numerical implementation error, which are likely manifested in varying sidelobe levels because these differences are extremely small.

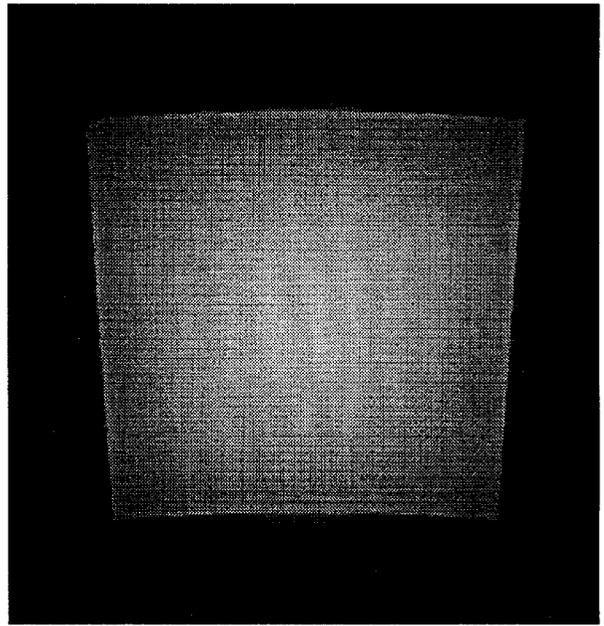
Another visual illustration of the above conclusion is shown in Fig. 13, where two phase history domain images from the “D” parameter set are offered. The phase history domain image is obtained from the spatial domain image via a 2D Fourier transform. Phase history is a term used to specify the actual Fourier domain data collected by the SAR. Figure 13(a) shows the 2D Fourier transform (via FFT) of the WX image (Fig. 10(c)), while Fig. 13(b) displays the 2D Fourier transform (via FFT) of the CBP image (Fig. 10(d)). One can see that the Fourier data used to form the CBP image (Fig. 13(b)) does in fact resemble that which was obtained by first windowing and then interpolating to an exscribed rectangle (Fig. 13(a)). Note the angle of the edges in each phase history domain image corresponds to $\theta_m = 4.0^\circ$ with respect to a vertical axis.

5.2 One Dimensional Interpolation of Filtered Projections

In this section we revisit the issue of how to compute $q_\theta(t)$ if $t \neq n\delta_t$ for any value of n . This topic was briefly addressed in [10] for nearest neighbor interpolation. We find that linear interpolation can be used but only if $\delta_t \ll 1$. To show the importance of this restriction, we present several examples, each using linear interpolation and having a different value for δ_t . For the “D” parameter set, the SAR collects 381 pulses and samples each pulse 332 times per second ($N_s = 332$). To obtain the filtered projection function $q_\theta(t)$ given in Eq. (16), we perform the multiplication specified, zero pad the sequence to a length M , and finish with a length M inverse FFT to obtain samples of $q_\theta(t)$ at points separated by δ_t given in Eq. (19). Normally, one would chose $M = 512$ if $N_s = 332$, which gives $\delta_t = 0.194531$. If we form a CBP image using $M = 512$, and then extract from the image the center row, which contains the center and left targets, and plot the magnitude (normalized with respect to the maximum value), we obtain the plot shown in Fig. 14. If we let $M = 1024$ then $\delta_t = 0.097266$ and the same row extracted from the new image is shown in Fig. 15. Plots in Figs. 16 and 17 result from $M = 2048, \delta_t = 0.048633$ and $M = 4096, \delta_t = 0.024316$, respectively.



(a)



(b)

Figure 13: Phase history domain images from “D” parameters. (a) WX. (b) CBP.

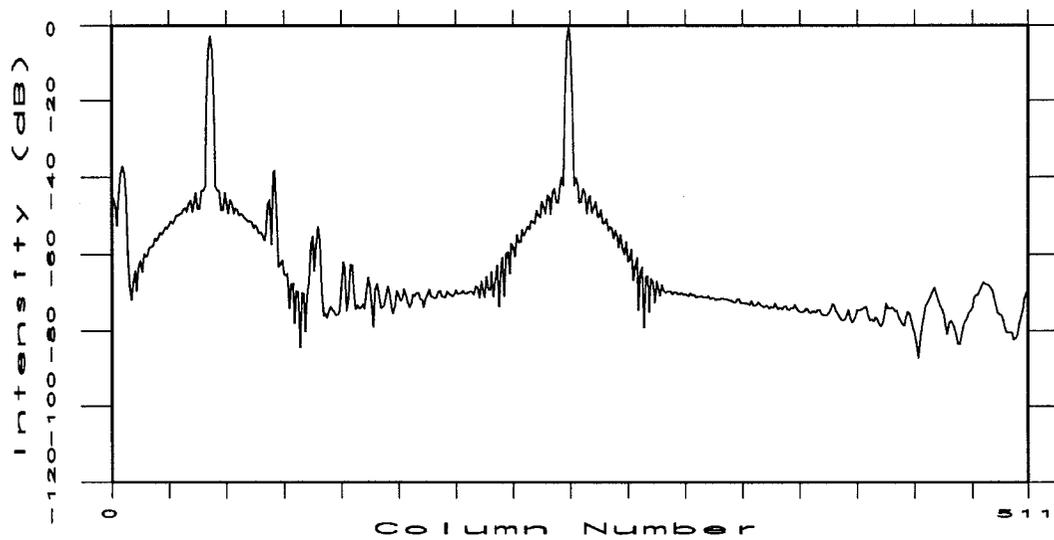


Figure 14: Center row from CBP "D" image, $N_s = 332$, $M = 512$.

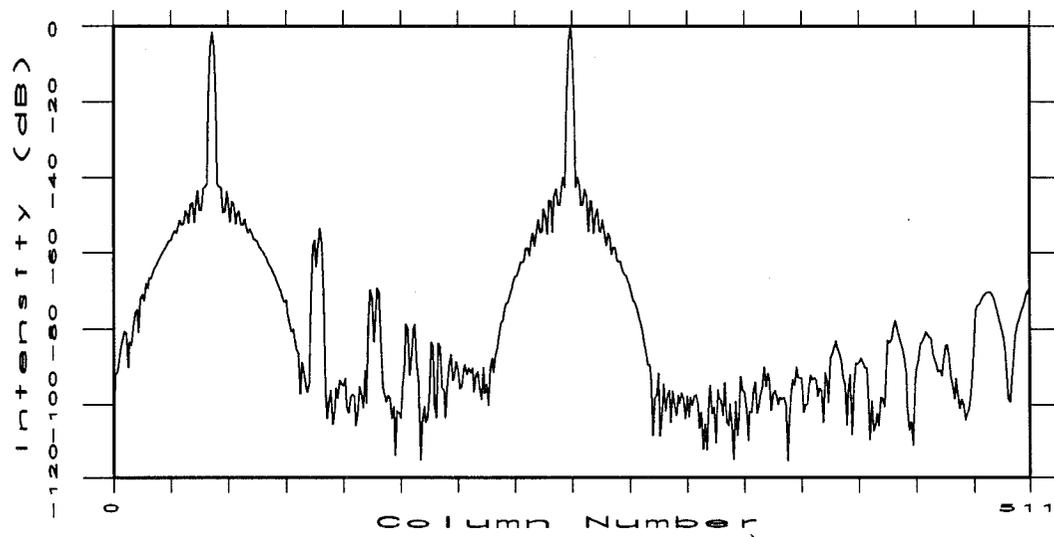


Figure 15: Center row from CBP "D" image, $N_s = 332$, $M = 1024$.

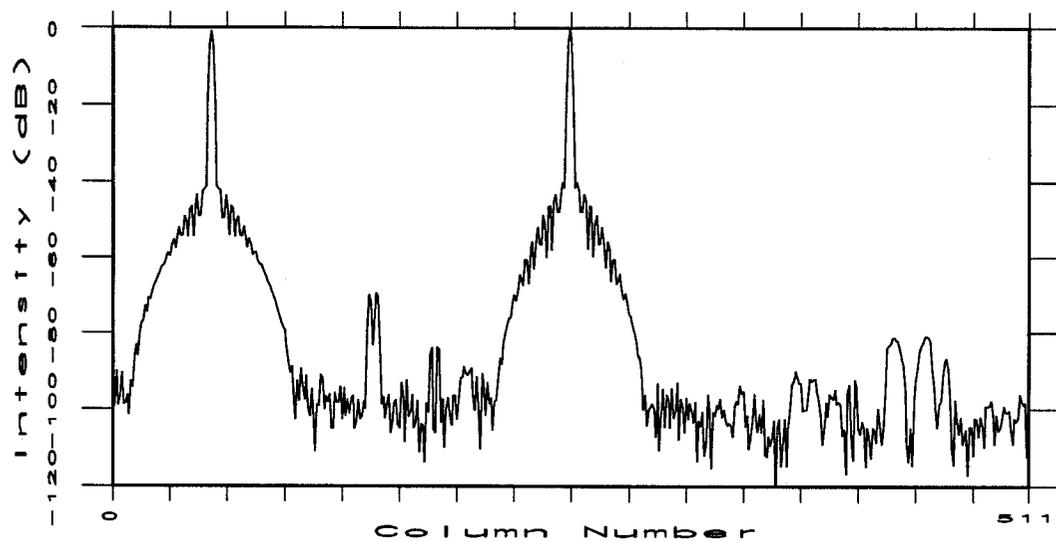


Figure 16: Center row from CBP "D" image, $N_s = 332$, $M = 2048$.

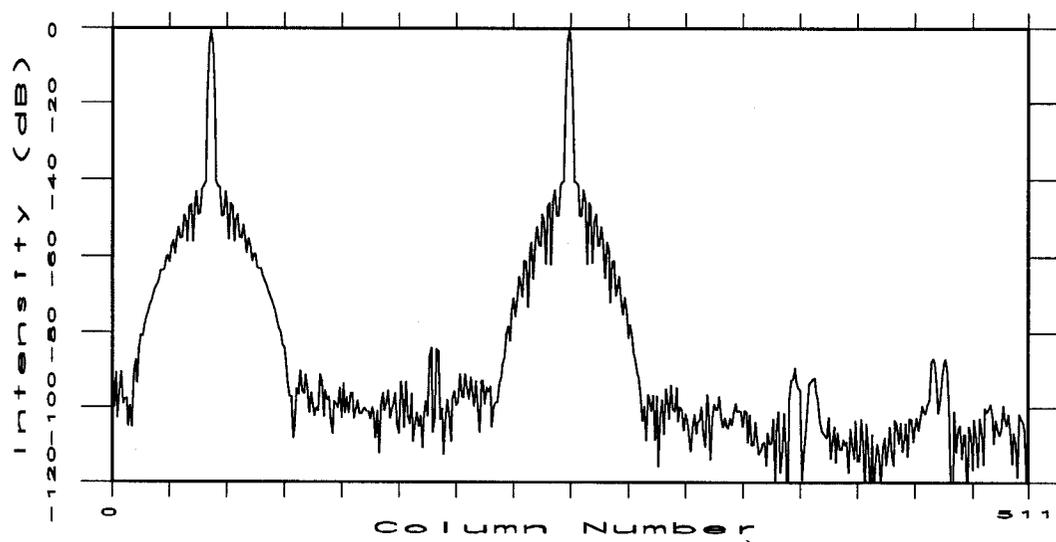


Figure 17: Center row from CBP "D" image, $N_s = 332$, $M = 4096$.

All plots in Figs. 14 through 17 use the same scale to simplify comparison. Two very important observations can be made from these plots. First, for the left target, one can see that linear interpolation is inadequate for the larger values of δ_t because additional targets have begun to appear. These additional targets were first termed “false targets” in [11], but to our knowledge, the false targets were not mentioned again in any literature concerning CBP and SAR, nor was an explanation given for their existence. The second important observation from these plots is that the center target has not generated any false targets. *Our results have shown that for a specific angle θ_m , as targets move further from the cross-range center, larger false targets are generated and the false target distance from the true target increases. Furthermore, as the angle θ_m becomes smaller, either by increasing f_c and decreasing θ_m to maintain resolution, or by decreasing θ_m to reduce resolution, then the effects shown in Figs. 14 through 17 become less severe.*

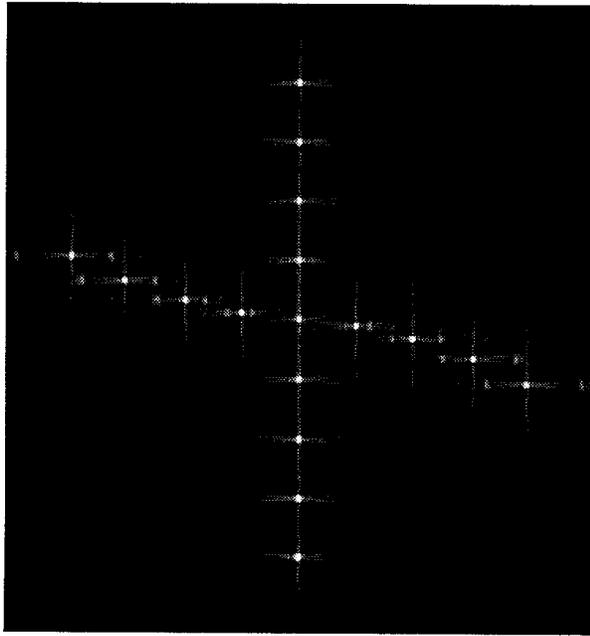
To show this effect more dramatically, refer to the images in Fig. 18. We have strategically placed targets in the ground patch so that the predominant position variation is either range or azimuth. The images in (a) and (b) were formed using the “D” image parameters, so the angle subtended by the radar is $2\theta_m = 8^\circ$. The only difference between these two images is that Fig. 18(a) used $M = 512$ and Fig. 18(b) used $M = 4096$. Note in Fig. 18(a) that as a target’s azimuth position is increased from zero, the false targets become larger and more spread out. Note also that multiple false targets actually occur, but false targets further from the true target reduce in amplitude. Multiple false targets can also be seen in the one dimensional plots in Figs. 14 through 16. Finally, one can see in Fig. 18(b) that zero padding to a longer length removes these effects. It should be pointed out that the act of zero-padding to longer lengths essentially implements 1D sinc() interpolation!

While we can not offer theoretical guidelines for using linear interpolation, the conditions outlined in the previous paragraph, under which linear interpolation is not acceptable, do have a common thread. Referring to Fig. 19, we see that a point target at location $(x_1, 0)$ gives a projection function

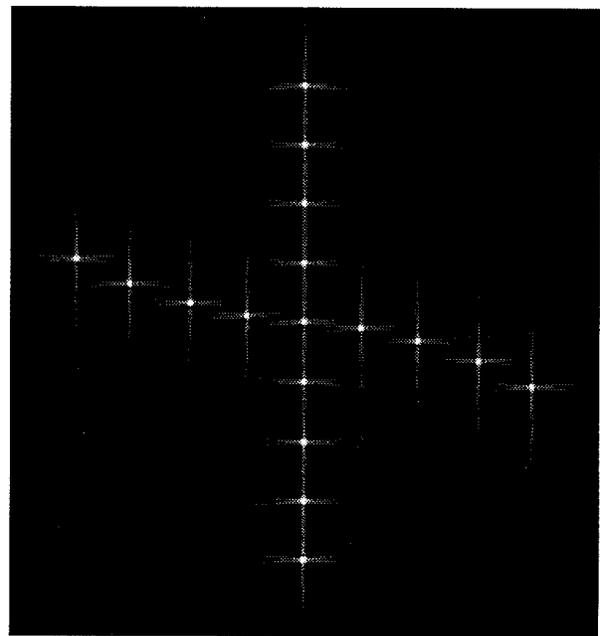
$$q_\theta(t) = \delta(t - t_1) \quad (25)$$

where $t_1 = x_1 \cos(\theta)$ and $\delta()$ is the unit sample function. If $\frac{\pi}{2} - \theta_m < \theta < \frac{\pi}{2} + \theta_m$, then the range of t_1 is $-x_1 \sin(\theta_m) < t_1 < x_1 \sin(\theta_m)$. Thus, as x_1 becomes larger, the range of values t_1 takes on also becomes larger as θ progresses through its range. Similarly, as θ_m is increased, the range of t_1 also increases for x_1 constant. It is apparently the large range of values of t_1 , coupled with backprojecting a complex exponential (see Eq. (20)) onto a finite grid that produces the false targets. The reason this effect was neither noted nor shown in [2] is most likely because their simulations used $2\theta_m = 0.1875^\circ$.

It is instructive to compare linear interpolation with nearest neighbor interpolation to determine if the simpler interpolation scheme is adequate with the smaller value of δ_t . This comparison is shown in Fig. 20, where the center row from an image formed using nearest neighbor interpolation (solid graph) is compared with the same row using linear interpolation (dashed graph), with $M = 4096$, $\delta_t = 0.024316$. One can see that linear interpolation reduces the noise floor by approximately 30dB or more. It is recommended that the additional computations for linear interpolation be used to achieve this advantage.



(a)



(b)

Figure 18: CBP-formed images (a) “D” image with $M = 512$. (b) “D” image with $M = 4096$.

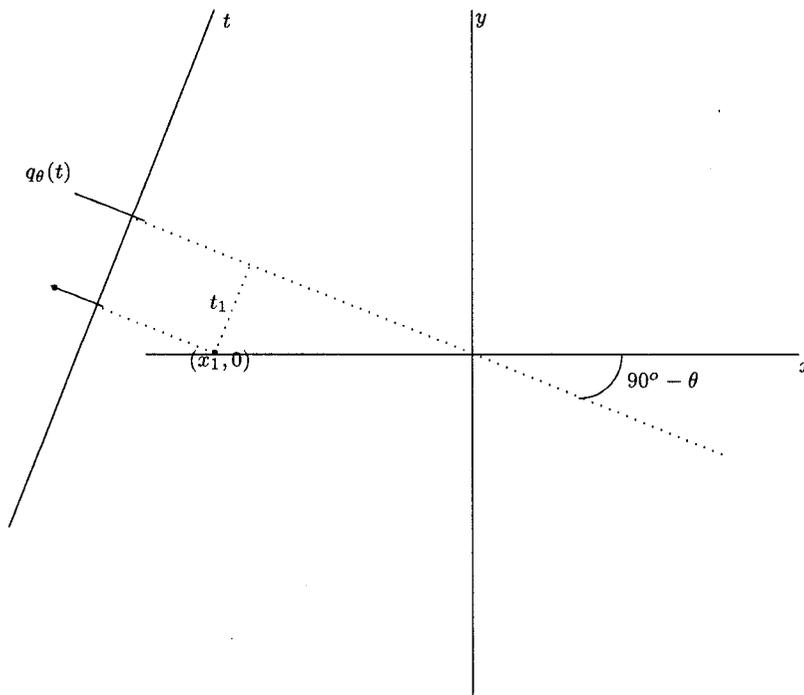


Figure 19: Projection function for image with single impulse.

6 Massively Parallel Implementations

From a conceptual standpoint, the CBP algorithm is straightforward to parallelize. This was mentioned in [2] as an advantage of the CBP algorithm over the 2DFFT method. However, from an implementation point of view, there are practical constraints that limit the performance of CBP when running on a parallel machine. Thus, the computation time required to form an image using CBP is significantly longer than with the 2DFFT method, while offering no real simplification in the ease of parallelization. The parallel program described herein was designed for execution on a 1024-node nCUBE 2. Its execution times will be compared to that of the 2DFFT algorithm on the same machine. While tests of CBP were not run on the Connection Machine CM-2 and Cray Y-MP, references will be made to implementation issues on those architectures as well.

The convolution and backprojection portions of CBP are inherently independent and are treated as two discrete phases in the parallel algorithm. The convolution portion of CBP is done in three steps. First, the complex phase history data are distributed as evenly as possible among the computer's p processors. The phase history pulses are distributed to the processors along with their respective angle θ , as measured with respect to the positive abscissa axis. When the total number of pulses, N_p , is divided evenly by the number of processors, p , then N_p/p pulses are given to each node. If the division results in a remainder of r pulses, then the first r processors will receive an additional pulse to process. The data are read from either a single disk or a disk array, with the data having been previously striped across the disks. As with the 2DFFT method, disk transfer times are

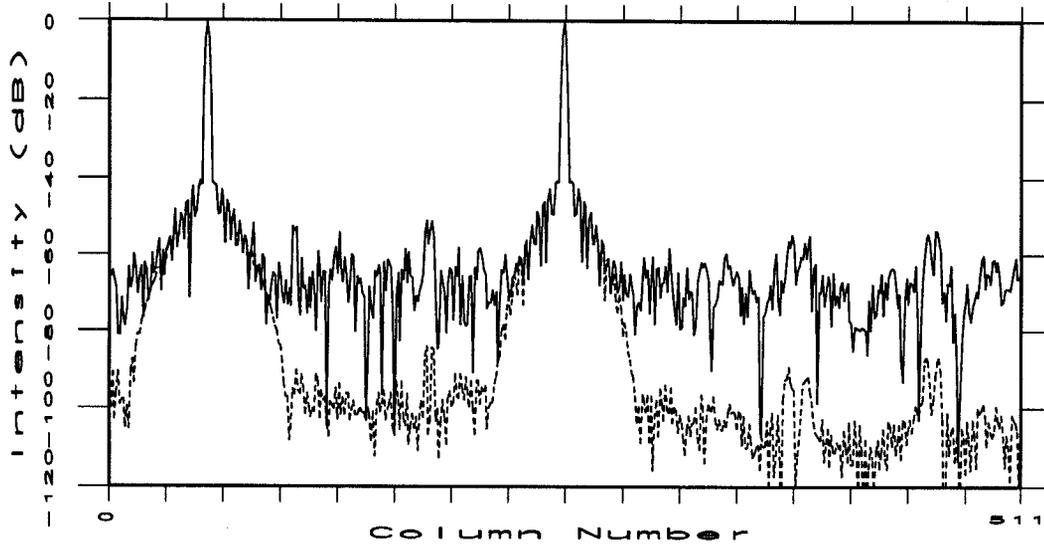


Figure 20: Center row from CBP “D” image, $N_s = 332$, $M = 4096$. Solid line=nearest neighbor interpolation; dashed line=linear interpolation.

not included in the performance measurements. In the second step of convolution, the filtered projection for each pulse is obtained by performing the multiplication method of Eq. (16), and then zero padding the projection to a length of M , where $2^M \geq N_s$. Finally, an inverse FFT of length M is performed on each projection to obtain samples of $q_\theta(t)$ at points separated by δ_t , as defined by Eq. (19). The entire convolution process is performed with an asymptotic computational complexity of $O(M \log_2 M)$. With the exception of the initial data distribution, which is ignored in this analysis, there is no interprocessor communication required for the convolution phase of CBP.

Having calculated the filtered projections $q_\theta(t)$ for each of the pulses, the backprojection phase begins. At this point, each processor node has the necessary data for generating its projections’ contributions to the overall formed image. In practice, the size of the formed image is variable, resulting in different pixel spacing in the spatial domain, as described in Section 4.1. For this discussion, assume the formed image is to have I image pixels in both the range and azimuth directions. At first glance, it seems reasonable to have each node to generate its contribution to the overall formed image, based on the filtered projections it contains. After completion, each node would contain a spacial domain contribution to the image of size I^2 , and the final image would be formed by integrating the contributions of all the nodes. There are several practical limitations that make this approach undesirable. First of all, very few parallel computers have enough memory in their individual nodes to hold a backprojected image of significant size. An image size of 512×512 , where $I = 512^2 = 262,144$, is already too large for an nCUBE 2’s node memory when dealing with eight byte complex numbers. In practice, images can exceed $14,000 \times 14,000$ pixels. Second, even if there is sufficient memory available, it is a formidable task in terms of interprocessor communication to integrate the contributions of 1024 nodes, when each contribution is I^2 . In a vector machine with a large shared memory, such as the Cray YMP or Connexion Machine CM-2, the integration can be performed without interprocessor communication, but

memory limitations would require repetitive swapping to and from disk of image portions during calculation.

To work more efficiently within the memory constraints of the parallel computer, consider the following method of parallel backprojection. Instead of having each processor compute its contribution to the overall space domain image, restrict the processor nodes to computing only a portion of the formed image. For an image size of I^2 , where I is a power of 2, let the first processor compute the first I/p rows of the image, the second processor the next I/p rows, etc. After each processor, in parallel, has integrated the contributions of its filtered projections, they pass their filtered projections to another processor, and receive those of another. This process continues until all nodes have received pulses from every other processor, and have backprojected and integrated the contributions into their portion of the final image. This method minimizes the amount of memory required and results in a portion of the final image residing in each node, allowing for easy storage onto the disk or disk array. However, this method has the disadvantage of requiring $O(p^2)$ communication steps, unless parallelism in communication is exploited.

To minimize communication steps within the nCUBE 2, the nodes are configured as an end-connected ring. After a node backprojects the filtered projections it contains, it sends them to the next higher processor, as numbered using a Gray coded ordering scheme. It then receives another set of filtered projections from its next lower node, as based on the Gray coding scheme. The backprojection is then performed on the new filtered projections within the node. In this way, all processors communicate with a hard-wired nearest neighbor, and the entire ring-shift of data occurs within one communication step, since all nearest neighbor communication takes place simultaneously. By the end of only $O(p)$ communication steps, all filtered projections have been seen by all processor nodes. Each processor requires $O(I)$ calculations to backproject each filtered projection, resulting in an overall complexity of $O(N_p^2)$, when the input phase history is similar in size to the formed space domain image. The communication time required to shift the filtered projections one position in the ring is a linear function of the length of the filtered projection, namely M . When significant zero padding of the input pulse is required, M increases and communication time suffers. This is a potential disadvantage of CPB over the 2DFFT method, since 2DFFT doesn't require padding beyond rounding to the next higher power of 2.

Regardless of the parallel method used to implement the CBP algorithm, it's likely to be less efficient than the 2DFFT method. This is due to CBP's 'expansion' of data, whereby a smaller filtered projection is backprojected onto a larger space domain image. This results in an increased amount of communication, since the space domain contributions within the nodes must be integrated into a single formed image through a data gathering operation. When a node does not backproject over the entire space domain, but instead, is responsible for the formation of a portion of the image, memory is conserved. However, the node must have access to all filtered projections within all other nodes. Either way, a considerable amount of interprocessor communication is required. While the CBP method requires $O(p)$ communication steps, the bulk of communication in the 2DFFT method is due to the matrix transposition, which takes only $O(\log_2 p)$ steps.

Table 5 contains actual running times for the CBP and 2DFFT algorithms on a 1024 node nCUBE 2. The running time of both algorithms increases proportionally with the square of the input data size. However, due to the data expansion in the CBP algorithm and the

Table 5: Running Time (sec) for CBP and 2DFFT on a 1024 Node nCUBE2

Ph. Hist. Size	CBP	2DFFT
1024 x 1024	76	6
2048 x 2048	543	19
4096 x 4096	4091	67
8192 x 8192	31,863	239

associated increase in communication cost, the constant multiplier for the CBP algorithm is higher. Data transfers to and from disk were not considered in these timings.

7 Conclusions

It should be evident that resolution and, therefore image quality, is dependent on the size of the Fourier data patch obtained during aperture synthesis, and not so dependent on the image formation technique used. In the case of inscribing a rectangle in the polar annulus, Fourier data is discarded, which reduces the achievable resolution. When the polar data is interpolated to an exscribed rectangle followed by windowing, the tapered window extends to the edges of the rectangular region, leaving a sizable discontinuity beginning in the upper corners, oriented at an angle corresponding to the collection geometry. While the main lobes of point targets are narrower, the sidelobes are much larger because of the discontinuity. This results in a poorer quality image.

If the entire Fourier data are windowed, followed by 2D Fourier inversion without discarding any data, then mathematics and experimental results show that regardless of the Fourier inversion technique, the image quality will be identical, save for insignificant numerical artifacts. This was the case for convolution backprojection and interpolating to an exscribed rectangle followed by 2D IFFT. The two methods appear to offer the best results in terms of the tradeoff between narrower mainlobe and very low sidelobes. However, some may argue that the more complex sidelobe structure detracts from the appearance of the image. For broadside mode collection, inscribing a rectangle generates sidelobes in only 4 directions, but the other methods force the user to contend with sidelobes in 6 directions.

Note that if the polar data is first interpolated to an exscribed rectangle, and then a progressively tapered window is applied so that the ends of the window correspond to the discontinuity, then results similar to CBP and WX are obtained. In this case, however, applying the window is much more complicated than in any of the other cases because the window length for each row must be computed and a new window generated.

The CBP algorithm is straightforward to parallelize. However, several practical constraints limit the efficiency of the computations. The zero padding of pulses during the convolution phase results in longer messages, thereby increasing communication time. Furthermore, an expansion of data results from the backprojection of individual filtered projections onto a larger space domain. This drives up memory requirements and results in increased communication overhead, since each node's contribution to the space domain image must be globally integrated. Alternatively, each node can be made responsible for the formation of a portion of the final image. This reduces memory requirements and eliminates

the global gathering operation. Unfortunately, this approach requires each node to have access to all other nodes' projections, thereby increasing communication time. The CBP algorithm does allow for the variable scaling of the formed image, which isn't a feature of the 2DFFT. However, the computation times of the CBP method are significantly longer than those of the CBP method, and as the size of the input data set increases, the CBP method falls farther behind the 2DFFT in efficiency.

References

- [1] D. C. Munson, J. D. O'Brien, and W. K. Jenkins, "A tomographic formulation of spotlight-mode synthetic aperture radar," *Proceedings of the IEEE*, pp. 917–925, August 1983.
- [2] M. D. Desai and W. K. Jenkins, "Convolution backprojection image reconstruction for spotlight mode synthetic aperture radar," *IEEE Trans. on Image Processing*, pp. 505–517, October 1992.
- [3] D. C. Munson, J. L. C. Sanz, W. K. Jenkins, G. Kakazu, and B. C. Mather, "A comparison of algorithms for polar-to-cartesian interpolation in spotlight mode sar," *Proc. Int. Conf. on Acoustics, Speech, and Signal Processing*, pp. 1364–1367, March 1985.
- [4] G. A. Mastin and D. C. Ghilia, "A research-oriented spotlight synthetic aperture radar polar reformatter," Tech. Rep. SAND90-1793, Sandia National Laboratories, Reprinted October 1993.
- [5] G. A. Mastin and D. C. Ghilia, "An enhanced spotlight synthetic aperture radar polar reformatter," Tech. Rep. SAND91-0718, Sandia National Laboratories, Reprinted October 1993.
- [6] G. A. Mastin, S. J. Plimpton, and D. C. Ghilia, "A massively parallel digital processor for spotlight synthetic aperture radar," *The International Journal of Supercomputer Applications*, vol. 7, pp. 97–112, Summer 1993.
- [7] G. T. Herman, *Image Reconstruction from Projections*. Academic Press, 1980.
- [8] A. Rosenfeld and A. C. Kak, *Digital Picture Processing*. Academic Press, 1982.
- [9] E. S. Haykin, *Array Signal Processing*. Prentice-Hall, 1985.
- [10] M. D. Desai, *A New Method of Synthetic Aperture Radar Image Reconstruction Using Modified Convolution Backprojection algorithm*. PhD thesis, Univ. of Illinois, 1985.
- [11] M. D. Desai and W. K. Jenkins, "Convolution backprojection image reconstruction for synthetic aperture radar," *Proceedings of ISCAS*, pp. 261–263, 1984.

This work was supported by the U.S. Department of Energy under contract DE-AC04-76DP00789.

Distribution

- 1 Dr. Gregory Donohoe
EECE Department
University of New Mexico
Albuquerque, NM 87131

- 1 Dr. Don Hush
EECE Department
University of New Mexico
Albuquerque, NM 87131

- 1 Dr. Neeraj Magotra
EECE Department
University of New Mexico
Albuquerque, NM 87131

- 2 MS 0572 C.V. Jakowatz, 5912
- 1 0572 D.E. Wahl, 5912
- 1 0572 P.A. Thompson, 5912
- 1 0572 T.J. Flynn, 5912
- 3 0572 N.E. Doren, 5912
- 1 0978 L.G. Stotts, 5934
- 1 1110 D.C. Ghiglia, 9223
- 2 1436 C. Meyers, 4523

- 1 9018 Central Technical Files, 8523-2
- 5 0899 Technical Library, 4414
- 2 0619 Review and Approval Desk, 12630
For DOE/OSTI
- 3 0161 Patent and Licensing Office, 11500