SANDIA REPORT

SAND2014-18347
Unlimited Release
Printed October 2014

Bistatic SAR: Signal Processing and Image Formation

Daniel E. Wahl and David A. Yocky

Prepared by
Sandia National Laboratories
Albuquerque, New Mexico 87185 and Livermore, California 94550

Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energy's National Nuclear Security Administration under contract DE-AC04-94AL85000.

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

Available to DOE and DOE contractors from
U.S. Department of Energy
Office of Scientific and Technical Information
P.O. Box 62
Oak Ridge, TN 37831

Telephone: (865) 576-8401
Facsimile: (865) 576-5728
E-Mail: reports@adonis.osti.gov
Online ordering: http://www.osti.gov/bridge

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

Telephone: (800) 553-6847
Facsimile: (703) 605-6900
E-Mail: orders@ntis.fedworld.gov
Online order: http://www.ntis.gov/help/ordermethods.asp?loc=7-4-0#online

SAND2014-18347
Unlimited Release
Printed October 2014

Bistatic SAR: Signal Processing and Image Formation

Daniel E. Wahl and David A. Yocky
Signal & Image Processing Technologies

Sandia National Laboratories
P.O. Box 5800
Albuquerque, New Mexico 87185-MS1207

Abstract

This report describes the significant processing steps that were used to take the raw recorded digitized signals from the bistatic synthetic aperture RADAR (SAR) hardware built for the NCNS Bistatic SAR project to a final bistatic SAR image. In general, the process steps herein are applicable to bistatic SAR signals that include the direct-path signal and the reflected signal. The steps include preprocessing steps, data extraction to for a phase history, and finally, image format. Various plots and values will be shown at most steps to illustrate the processing for a bistatic COSMO SkyMed collection gathered on June 10, 2013 on Kirtland Air Force Base, New Mexico.

ACKNOWLEDGMENTS

This work would not have been possible without the support of many people from several organizations. The authors wish to express their gratitude to the National Nuclear Security Administration, Defense Nuclear Nonproliferation Research and Development (DNN R&D). Our team would like to thank our program leader, Veraun Chipman from NSTec for his help and support of our efforts. This work was done by Sandia National Laboratories under award number DE-AC52-06NA25

We would like to thank Paul Eichel (Sandian, retired) for writing and maintaining the image formation computer code used in this research.

CONTENTS

FIGURES

NOMENCLATURE

BW	bandwidth
IFP4	Image Formation Processor (Version 4)
Mocomp	motion compensation
PRF	pulse repetition frequency
PRI	pulse repetition interval (PRI = 1/PRF)
RADAR	radio detection and ranging
SAR	synthetic aperture RADAR
TLE	Two Line Element

1. INTRODUCTION

This report describes the significant processing steps that were used to take the raw recorded digitized signals from the bistatic synthetic aperture RADAR (SAR) hardware described in [1] to a final bistatic SAR image. Various plots and values will be shown at most steps to illustrate the processing for a bistatic COSMO-SkyMed collection gathered on June 10, 2013 on Kirtland Air Force Base, New Mexico. The processing presented here was used to form imagery collected over Kirtland Air Force Base, New Mexico and Nevada National Security Site, Nevada. There were nine bistatic SAR collections processed using these steps.

[This page intentionally left blank]

2. HIGH-LEVEL PROCESSING DESCRIPTION

The signal processing steps are broken into two distinct parts. The preprocessing software (Block 1) in Figure 1 takes the continuously recorded direct-path channel data and reflected-path channel data along with the two line element (TLE) file and a configuration ASCII file called 'MainInfo.txt' and produces standard wide-band and narrow band files. This report primarily documents this step. The output narrow-band and wide-band files in this case have the extension *phs* for the wide-band file and *au4* for the narrow-band file. These files are then used as inputs to IFP4 which forms the bistatic SAR image using standard polar-format image processing [2].

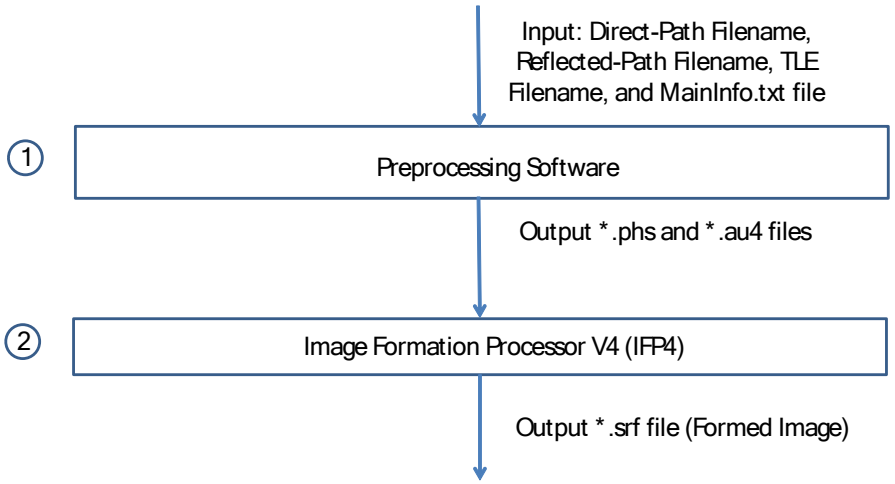

Figure 1. High-level processing block diagram.

IFP4 is a general spot-light polar-format image formation algorithm that can be used to form both monostatic and bistatic SAR imagery. The IFP4 algorithm is well documented and therefore the details will not be discussed here.

[This page intentionally left blank]

3. DETAILED PROCESSING DESCRIPTION

The details of the pre-processing step represented by Block 1of Figure 1 will be discussed here. Figure 2 shows in more detail the primary steps involved in this block. Each step of the preprocessing algorithm will be described in detail. Various plots and values will be shown at most steps to illustrate the processing for a specific collect experiment. The experiment chosen to show these values is the COSMO-SkyMed collect that occurred June 10, 2013, of the Tijeras Arroyo Golf Course near Albuquerque, New Mexico.

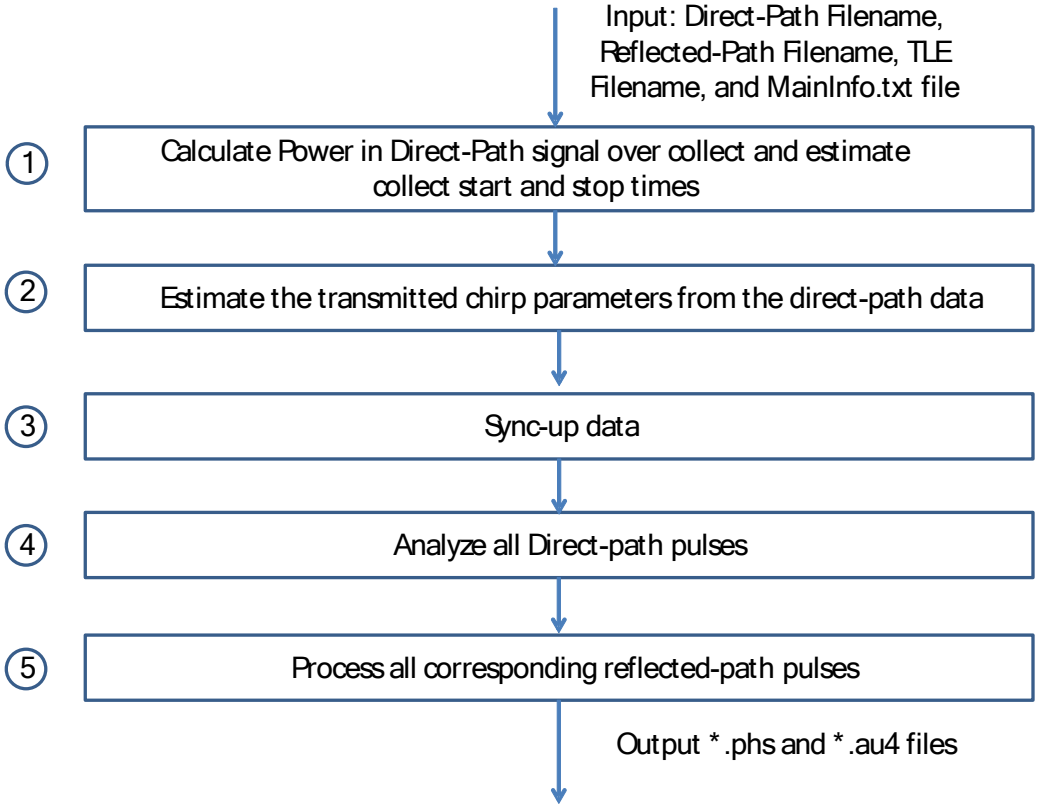

Figure 2. Detailed block diagram of preprocessing software.

3.1 Required Inputs

The inputs to the procedure are the two files collected from the bistatic hardware recorder (direct-path channel and the reflected path channel), the TLE file, and a configuration file with the name 'MainInfo.txt'.

Both the direct-path and the reflected path file have headers with important information about the collect. The specific entries in the headers used in this algorithm are the time-stamp associated with the first digitized sample, the system sample rate (1.6 GHz for the collects described here), and the GPS location of the direct-path and reflected-path antenna's.

The third input is the configuration file with the name 'MainInfo.txt'. The format of the file is shown in Figure 3. It is assumed this file resides in the same directory as the other files. The parameters of this file are used in various parts of the algorithm.

The last input is the TLE file. The TLE file is a standardized file associated with a particular satellite giving the required variables to determine its location as a function of time. The TLE files are ASCII coded file with two lines. The TLE used in this example is shown in Figure 4. It is for the Italian commercial X-Band SAR called COSMO-SkyMed. The position location of the COSMO-SkyMed vs. time was calculated using the TLE, the time of interest, and the MATLAB software described in [3].

```
Phase History Oversample:  1.2
Number Of Pulses to Sinc:  64
Approximate PRF:           3068
Data Threshold:            0.2
Patch Size (m):            10000
Collect GRP  (Lat (Deg); Lon (Deg); Ht (m)):       35.015985; -106.528301; 1645.378
Direct Ant Loc (Lat (Deg); Lon (Deg); Ht (m)):     35.023051; -106.487485; 2085.762
Reflect Ant Loc (Lat (Deg); Lon (Deg); Ht (m)):    35.023126; -106.487415; 2087.808
f0_if (Hz):   400000000
fdemod (Hz): 9600000000
```

Figure 3. Data in 'MainInfo.txt' file for Cosmo-SkyMed collect.

```
1  32376U  07059A     13156.80652889   .00000354     00000-0   50950-4  0   620
2  32376   097.8732  342.2019   0001530  083.5716  276.5681  14.82159748297090
```

Figure 4. TLE file used in the formation of the Cosmo-SkyMed collect that occurred June 10, 2013.

3.2 Block 1: Calculate power in direct-path signal over collect and estimate collect start and stop times

The recorder is typically started prior to the illumination time and stopped after the collect is over. The first step, therefore, is to determine a suitable interval of time to process the collected data. This step scans the entire direct-path signal recording interval and logs the power as a function of time. The start and stop times are determined by locating the time where the maximum power occurs. The start time is determined by moving backward from this point and stopping at the location where the power reaches a threshold (specified in the MainInfo.txt file). The end time is calculated by moving forward from the maximum power location until the same threshold is reached.

The power curve for the example collect is shown in Figure 5. The threshold in this case is 0.2 (specified in the MainInfo.txt file). The red line shows the threshold and the corresponding location of the start and stop times. In this case, approximately 3 seconds of the collect were processed.

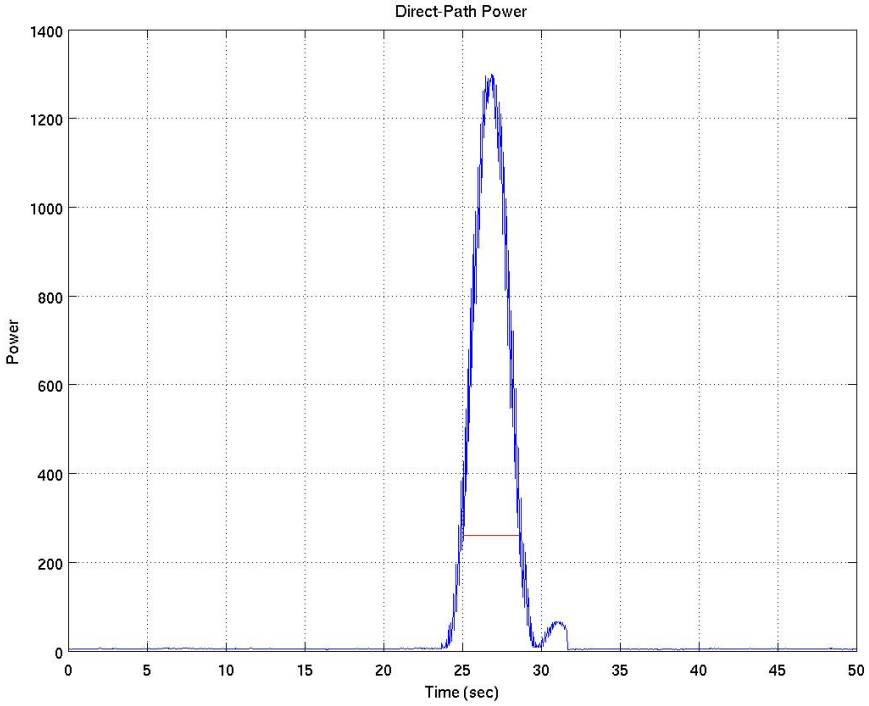

Figure 5. Power of the direct-path signal over the 50 second collect. The red line also shows how the collect start and end are determined using the threshold.

3.3 Block 2: Estimate the transmitted chirp parameters from the direct-path data

Successful image formation requires accurate knowledge of the transmitted chirp. It is assumed this information is not provided and therefore an algorithm was developed to estimate the transmitted chirp from the data recorded in the direct-path channel. The algorithm assumes the transmitted chirp characteristics are constant and do not change over the coherent dwell period.

The algorithm takes a small section of data (approximately 15 pulses) from the direct-path data channel where the peak power is detected. From this data, the routine estimates the following parameters which fully characterize the chirp: Chirp start frequency, chirp-rate, chirp bandwidth, chirp duration, and an accurate estimate of the collect pulse repetition frequency (PRF). The steps in the chirp parameter estimation algorithm are described below:

- The hardware converts the received data with center frequency of 9.6 GHz down to a signal with a center frequency of 400 MHz before it digitizes the data. No information is lost in this process but does allow the signal to be digitized at a much lower rate. This data however, is still not yet in the format required to process it for image formation. An additional frequency conversion is necessary which converts the 400 MHz signal down to DC or 0 Hz. This process is called "basebanding" and is done using a series of Fourier transform operations. Once this is done, the base-banded chirp can now be modeled as a complex exponential.

$$S(t) = , \; ae^{(j(\omega_0 t+\alpha t^2+\phi_0))} \; 0 < t < \tau_c \qquad \textbf{Equation 1}$$

- Estimate τ_c . This is done by detecting the chirp start and end times for each pulse in the extracted data and measuring the average time between these points. The algorithm assumes the magnitude of the data is in one of two states: 1) A high-level when a chirp is present and 2) a low level when only noise is present. A threshold for distinguishing the two levels is calculated using a k-means clustering algorithm.
- Determine the PRF from the location of the measured transitions from noise to chirp from the previous step.
- Determine the chirp bandwidth (BW) by Fourier transforming each detected pulse individually. Sum the magnitude of each Fourier transformed pulse to produce and average frequency content measurement of the signal. Estimate the frequency content (or the bandwidth) by calculating a threshold by again using the k-means clustering algorithm. The threshold is then used to measure the frequency locations where the signal

frequency starts and ends. The distance between these locations produces the bandwidth measurement.

- There are two possibilities of transmitted chirps. An ‘up-chirp’ or a ‘down-chirp’. An up-chirp starts at a low frequency and proceeds to a high frequency. A ‘down-chirp’ will do the opposite. It starts at a high frequency and moves to a lower frequency. This step determines if the chirp is an up-chirp or a down-chirp. It does this by comparing the frequency content of the first-half of the chirp to the last-half of the chirp.
- Get a rough estimate of the chirp rate by dividing the estimated bandwidth by the estimated chirp duration.
- Determine a more accurate value for the chirp-rate by doing a brute-force search about the calculated chirp-rate above. The procedure compresses the extracted data with a chirp signal that has the chirp-rate of interest. The compressed signal that has the largest compressed peak is selected and retained.

3.4 Block 3: Sync-up data

The purpose of this block is to produce an exact time (or sample number) when the first pulse after the received power level exceeds the threshold arrives. This pulse serves as a benchmark for the rest of the collect and is assumed to be the first pulse of the coherent-data period.

The algorithm determines this time by reading a section of data starting at the time the direct-path power exceeds the threshold (see section 3.2). This data is down-converted and compressed with a chirp with the estimated transmitted chirp parameters. The location of the first-peak in the compressed data gives an accurate arrival time of the first pulse.

3.5 Block 4: Analyze all direct-path pulses

Before processing the reflected-path data, two additional critical timing items need to be estimated for each pulse. Precise knowledge of when the direct-path pulse arrives, and the phase associated with each direct-path pulse. This block estimates these values by compressing each direct-path pulse in the direct-path channel.

Figure 6 shows a typical received pulse in the direct-path channel. This data is then down-converted to base band producing a complex signal of the form shown in Equation 1. The data is then compressed or convolved with a copy of the conjugate of the transmitted chirp (The chirp values are estimated in Block 2). Theoretically the result of this convolution will be a shifted and scaled complex ‘sinc’ function. A sinc function is mathematically defined as:

$$\text{sinc}(x) = \frac{\sin\square(x)}{x} \qquad \textbf{Equation 2}$$

which produces a peak at x = 0. The location and the phase at the peak of the sinc function are the two measured values required for each received direct-path pulse.

Figure 7 shows the resulting data after compression. In this case, the resulting spike or sinc function is easily seen and the location of the peak of the spike relative to the start of the read location is the first critical measurement needed. The second is the phase at the peak location. This value is retrieved by simply measuring the phase at the peak of the sinc function as shown in the blowup in Figure 8.

.

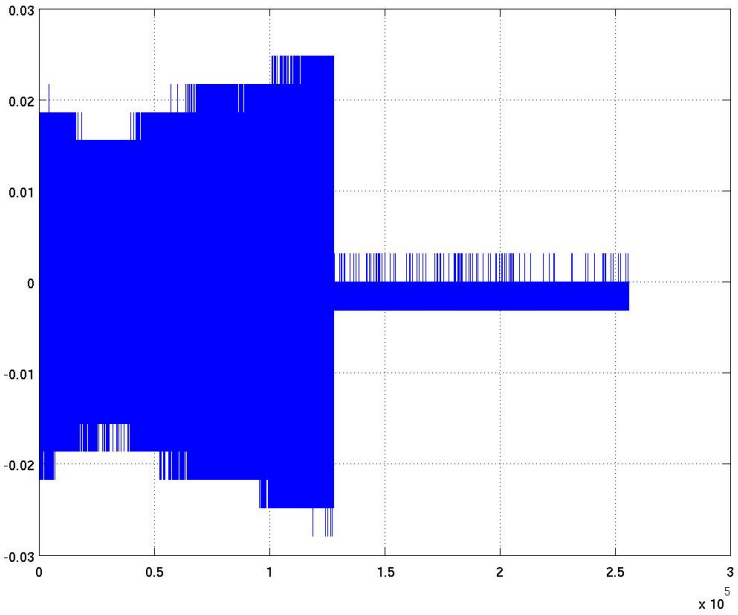

Figure 6. First pulse from direct-path signal.

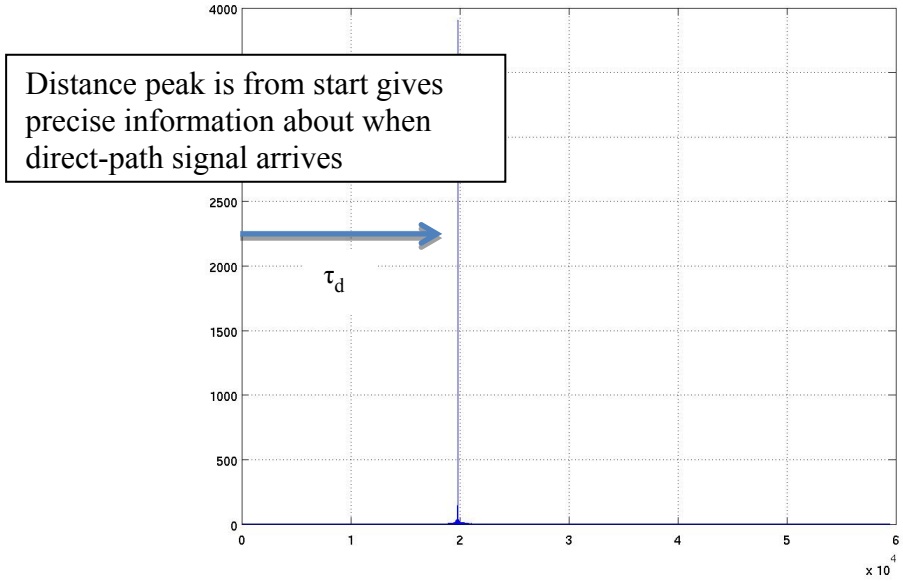

Figure 7. Pulse from the direct-path signal after compression. The symbol τ_d represents the delay in seconds the center of the transmitted chirp arrives relative to the start of the data read.

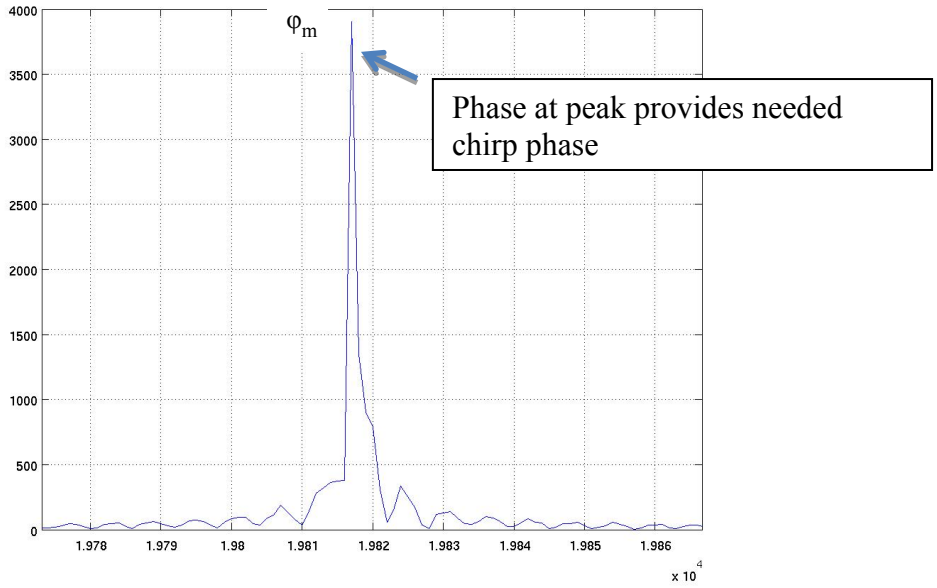

Figure 8. Expanded view of sinc function shown in Figure 7. The phase φ_m is the measured phase at the peak of the function.

3.6 Block 5: Process all corresponding reflected-path pulses

This block process all the reflected path pulses and calculates the necessary phase stabilization coefficients required to phase compensate each pulse. The basic steps involved for each pulse are:

3.6.1 Read the raw data

The first step simply reads the entire 'pulse' return or PRI. An example of the raw data for one of the pulses is shown in Figure 9.

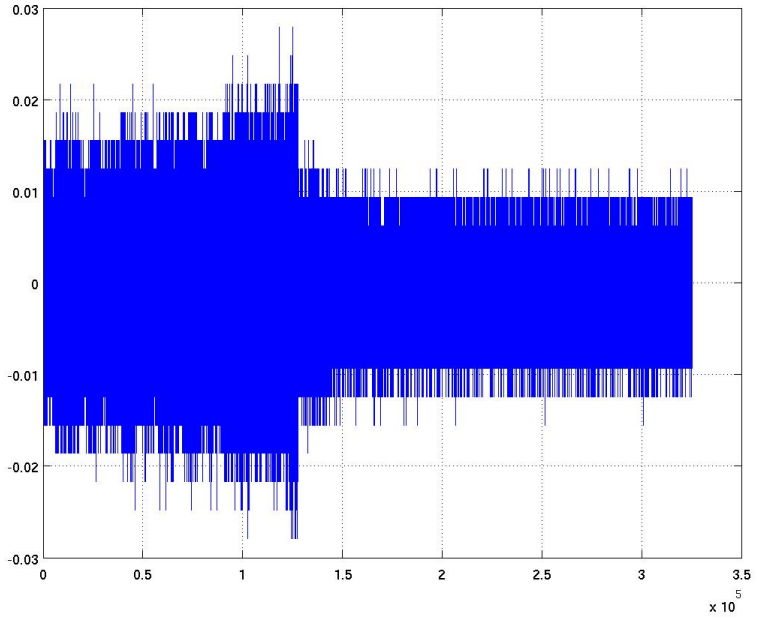

Figure 9. Raw data from single pulse in the reflected-path channel.

3.6.2 Compress the pulse

The raw data is now compressed. This step is the same process that is done on the direct-path data and is described in Section 3.5. In this case, the raw data contains the returns from a patch of ground containing many scatterers. The result of the compression will not be a single spike as was seen from the compression of the direct-path signal but will contain many overlapping sinc functions.

Figure 10 shows the result of compressing the data of Figure 9. The large spike seen in the figure corresponds to either direct-path energy leaking into the signal via the antenna back-lobe or returns coming from directly around the reflected-path antenna.

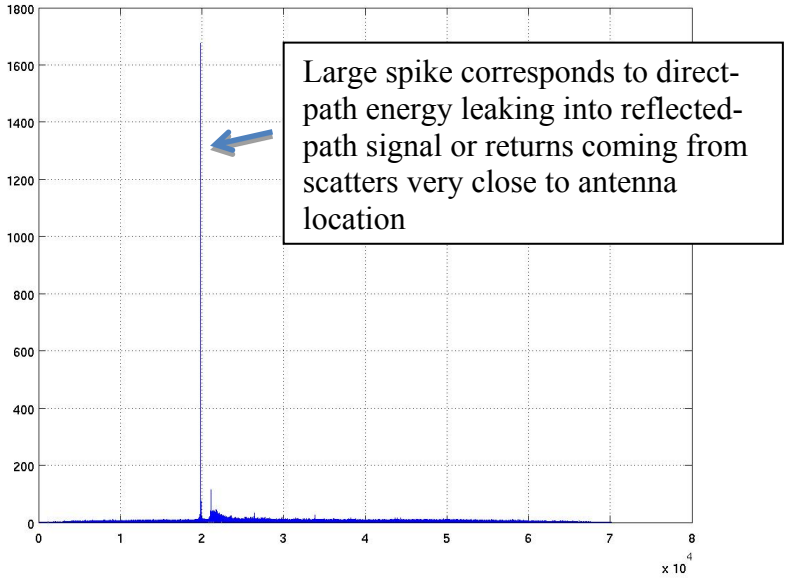

Figure 10. Result of compressing the reflected-path data shown in Figure 9. The large spike corresponds to either direct-path energy leaking into the signal via the antenna back-lobe or returns coming from directly around the reflected-path antenna.

3.6.3 Extract the desired data from the compressed pulse

The correct samples to extract from the compressed pulse are always calculated relative to where the direct-path sinc was measured as described in Section 3.5. The relative distance from this location is a function of the patch size, location of the transmitter, location of the receiver, and the scene-center-point.

Figure 11 shows a graphical view of how the size and location of the data to extract is calculated. The extracted data is used as the phase-history data for this particular pulse. The sample corresponding to the return from the center of the patch is calculated using the geometry of Figure 12 and is given as a time delay relative to the arrival of the direct path. This time delay is shown in Equation 3.

$$\tau_p = B + C - A \qquad \textbf{Equation 3}$$

The amount of data to extract about the scene-center point is proportional to the desired reconstructed scene size and the geometry of the collect.

The extracted data is then Fourier transformed and placed in a raw data phase history file *.phs fie). The Fourier transform is necessary because the IFP4 algorithm assumes the raw data is in the Fourier domain – requiring the transform to go from range-compressed to frequency domain.

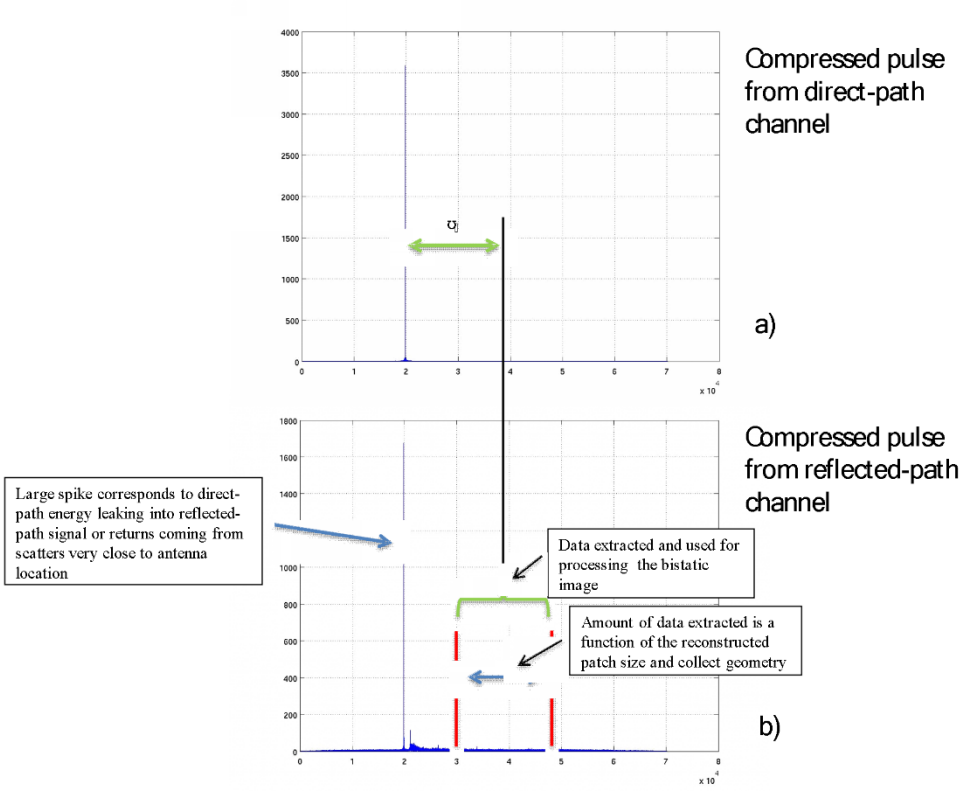

Figure 11 a) Compressed direct-path pulse. b) Compressed reflected-path pulse. These plots show how the location and size of the data extraction in the reflected path pulse is calculated.

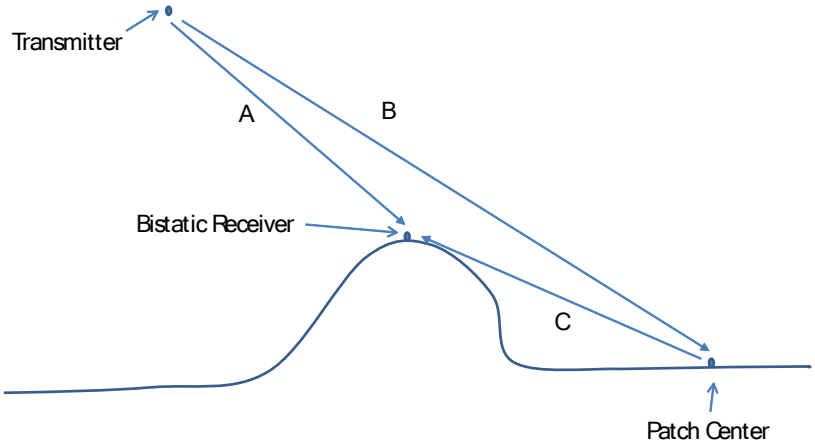

Figure 12. Basic bistatic geometry used to determine samples to extract from range-compressed data.

3.6.4 Calculate the phase compensation values

This step calculates all the phase compensation values to motion compensate or 'mocomp' each pulse properly so the IFP4 algorithm forms the image correctly. As detailed in the documentation of IFP4, there are three constant phase values used to do this task. These are referred to the C0, C1, and C2 coefficients. IFP4 uses these coefficients to apply a phase function to a particular pulse using the following relationship:

$$\beta(n) = C0 + C1 * n + C2 * n^2 \qquad \textbf{Equation 4}$$

where β represents the phase function and *n* is the sample number of the extracted pulse.

The value for C0 is determined using the following equation:

$$C0 = -\varphi_m + 2 * \pi * f_0 * \tau_p - \pi * S_1 \qquad \textbf{Equation 5}$$

where φ_m represents the measured direct-path phase associated with this particular pulse (described in 3.5), τ_p is the delay between the arrival of the direct-path and the return from the mocomp point or the patch center as described in 3.6.3, and f_0 is the radar center frequency. S_1

represents a sub-pixel shift amount. This number is needed because the calculated start and end time of the compressed signal rarely falls on an integer pixel value. Instead, the closest pixel value corresponding to these times are used. This introduces an error in the process and is corrected by additions to the *C0* and *C1* coefficients. More will be said about this in the description of *C1* below.

C1 is needed solely to correct for the fact we usually extract data from the compressed pulse on an integer pixel value rather than the calculated time or corresponding pixel value. The form of *C1* is:

$$C1 = \frac{2 * \pi * S_1}{N} \qquad \textbf{Equation 6}$$

where N is the number of extracted samples in the pulse and

$$S_1 = (\tau_d + \tau_p) * f_s - \text{round}(\tau_d + \tau_p) * f_s. \qquad \textbf{Equation 7}$$

S_1 is a number ranging from -0.5 <= S_1<= 0.5, and f_s is the sample rate in units of samples/sec.

Extracting the data on the closest pixel location rather than on the correct sub-pixel location essentially introduces a shift into the compressed signal. A shift can be removed by applying a phase ramp in the frequency domain which is what Equation 6 implements.

Unfortunately, this phase ramp will change the constant phase at the center of the pulse (at sample N/2). In order to preserve the desired constant phase (in *C0*), we must remove the effect of *C1* which is why we see the term $- \pi * S_1$ in Equation 5.

C2 was set to zero for these collects. This term would apply a quadratic phase to the pulse and may be needed where the transmitter has a combination of high velocity and a significant squint during the collect. All collects with this system as the date of this report have not needed this correction. The correct C2 compensation will be included in later versions.

The critical coefficients C0, C1, C2 (described above) along with some other collect information such as the transmitter and receiver locations, collect center frequency, bandwidth, and timing information are written out in the *.au4 file.

4. RESULTING IMAGE AND ADDITIONAL COMMENTS

The output image for the example described in this report is shown in Figure 13. The desired patch size for the processing was set to 10000 meters. This is a variable in the 'MainInfo.txt' file as shown in Figure 3. Setting the patch size to be this large puts the bistatic receiver at the top of the reconstructed scene. Another way to think of this is that the patch size represented by the two red lines of Figure 11 is so large that the bright return on the left of the figure is included in the reconstruction.

The final image was constructed using 10902 pulses and had a size of 8600 meters in azimuth and 10000 meters in range. The azimuth and range resolutions were 1.48 meters and 0.84 meters respectively. In bistatic reconstructions where the receiver is stationary, the azimuth resolutions is typically about half of the range resolution.

Note that this image appears 'distorted' in that the roads are curved. This is because the image was formed using the IFP4 algorithm which is based on the polar-format algorithm. The distortions in the image can be corrected using a post-processing warping algorithm. Another option is to use a back-projection algorithm that reconstructs the image on a predetermined grid on the earth leaving no image distortions and is automatically orthorectified assuming the grid incorporates the correct height of the earth.

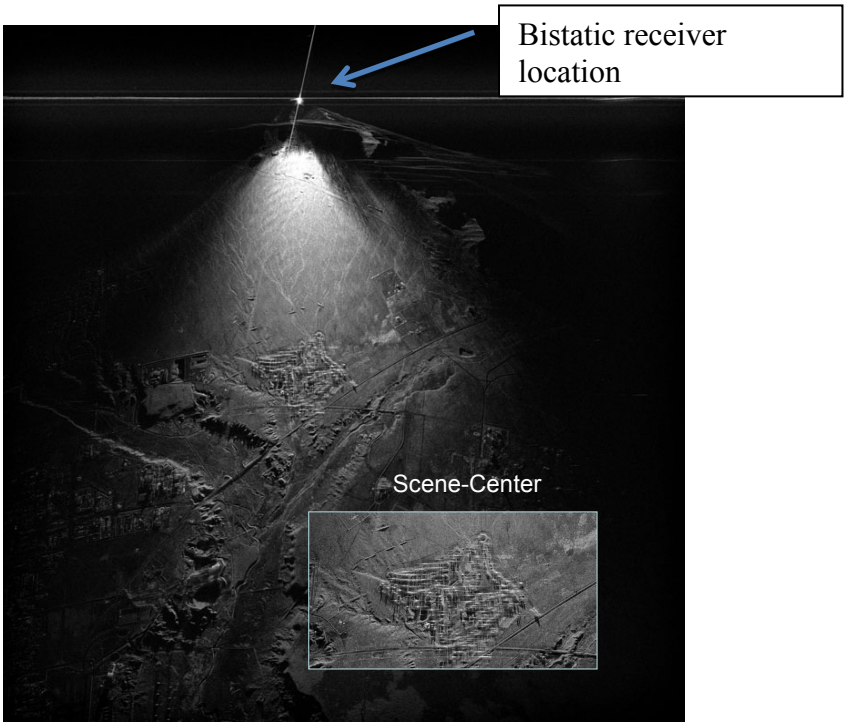

Figure 13. Resulting image for the COSMO-SkyMed bistatic collect that occurred June 10, 2013, of the Tijeras Arroyo Golf Course.

5. CONCLUSION

This report outlines the steps the processing software takes to convert the raw recorded signals digitized by the hardware of [1] to a final SAR bistatic image. Each of the significant steps was described in detail. Various plots and values were shown to illustrate the processing for the COSMO-SkyMed collect that occurred June 10, 2013, of the Tijeras Arroyo Golf Course.

6. REFERENCES

[1] D. A. Yocky, N. E. Doren, T. A. Bacon, D. E. Wahl, P. H. Eichel, C. V. Jakowatz, G. G. Delaplain, D. F. Dubbert, B. L. Tise and K. R. White, "Bistatic-SAR: Proof of concept," 2013.

[2] P. H. Eichel, "IFP V4.0: a polar-reformatting image formation processor for synthetic aperture radar.," Sandia National Laboratories Sandia Report: SAND2005-5232, 2005.

[3] D. A. Vallado, "Revisiting Spacetrack Report #3," in *AIAA/AAS Astrodynamics Specialist Converence*, Keystone, CO, 2006.

[4] C. Jakowatz, D. Wahl, P. Eichel, D. Ghiglia and P. Thompson, Spotlight-mode Synthetic Aperture Radar: A Signal Processing Approach, Kluwer, 1996.

DISTRIBUTION

1 Veraun Chipman
NCNS/CIT Program Manager
National Security Technologies (NSTec)
PO Box 98521
M/S NLV101
Las Vegas, NV 89193-8521

1	MS0971	Robert M. Huelskamp	5750
1	MS1207	Charles V. Jakowatz, Jr.	5962
4	MS1207	Daniel E. Wahl	5962
2	MS1207	David A. Yocky	5962
1	MS1209	John E. Gronager	5960
1	MS0899	Technical Library	9536 (electronic copy)


Sandia National Laboratories