Implementing a SAR Processor

J. Thatcher Chamberlin*
MIT Department of Physics
(Dated: December 7, 2018)

This report describes the development and operation of a synthetic aperture radar (SAR) processor written for the 12.421 final project. In class we have discussed Maxwell's equations and radar and SAR so this project will synthesize all we have learned into one functional product that produces useful imagery. The program implements range compression, range cell migration correction, and azimuth compression. Hopefully it will serve as a helpful reference for others seeking to understand the full radar processing pipeline. Additionally, the work presented here provides a base for possible future work in interferometric SAR and multilook processing to generate even more useful images.

I. BACKGROUND

I.1. Synthetic Aperture Radar

Since 1978, scientists have used space-based synthetic aperture radar to study and monitor the surface of the Earth.[1] These radars produce high resolution imagery than contains valuable information about sea surface state, terrain elevation, and surface composition that aid in a wide range of military, commercial, and scientific applications.[2] This imagery comes as a result of sophisticated digital signal processing of reflected radio waves measured by these radar instruments. I proposed to process some of this data in an effort to learn how it is done and how different sources of error can affect the final result. There are and have been many synthetic aperture radar space missions, and since many are operated by non-military government institutions, their data are freely available.

I.2. ALOS PALSAR

PALSAR is a polarimetric L-band synthetic aperture radar instrument flying aboard JAXA's Advanced Land-Observing Satellite (ALOS) pictured in Fig. 1. It supports several imaging modes, but the one used in this report is the fine-beam dual-polarization mode (FBD).[3] This imaging mode in principle allows for the production of polarimetric images, a possibility explored with some mild success in the development of the program.

TABLE I. Some key parameters of the ALOS-PALSAR mission. Numbers from [4] and processed data files.

Carrier Wavelength	23.6 cm
Carrier Frequency	1.27 GHz
Antenna Length	8.9 m
Nominal Altitude	692 km

* jthatch@mit.edu

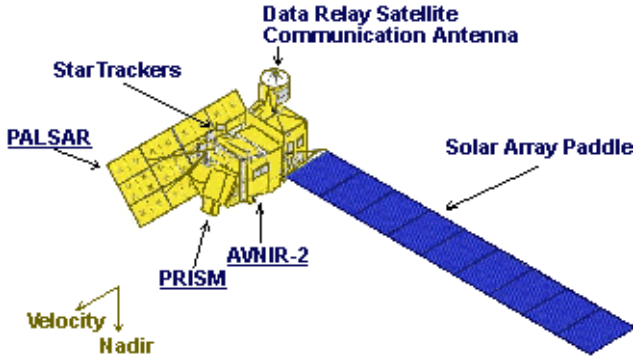

FIG. 1. A drawing of the ALOS satellite, with the PALSAR antenna array visible on the left.[4]

II. METHODS

I wrote a program using the language Julia[5], a modern language designed for high speed numerical processing. Julia's ecosystem provides excellent libraries for image processing, Fast Fourier Transformations, object serialization, and plotting, greatly reducing the amount of code required to write such a program.

II.1. File Processing/Data Formats

Level 1.0 PALSAR files were obtained from the Alaska Satellite Facility's Vertex website.[6] These files are archives containing several files. The files used in this report were the leader file and the two image files (HH and HV polarizations), both of which are described in the ALOS/PALSAR Product Format Description.[7]

The leader file contains a `DataSetSummaryRecord` which contains information necessary for processing the images in the archive. In particular, I extract the carrier wavelength, chirp modulation, chirp length, IQ sampling rate, and pulse repetition frequency (PRF).

The image file is composed of an `ImageFileDescriptorRecord` followed by many `SignalDataRecord`s. The `ImageFileDescriptorRecord` contains the number of radar echos in the image, the number of bytes per echo, the bits per sample, and the

data format (either real or complex).[7]

The information I took from the `SignalDataRecord`s contain was just the IQ encoded radar echos received by the instrument and the range distance to the first echo sample. 'IQ' signals are complex-valued samples of a signal, commonly used in signal processing. In RF systems like radars, IQ values are obtained by mixing the received signal with an in-phase oscillator and an oscillator shifted in phase by 90° (the quadrature oscillator) and measuring the results, resolving an ambiguity encountered with purely real signals. The file format document also specified locations of other values of interest like platform altitude and speed and electronic and mechanical antenna squint angles, but many of these fields were left blank or set to exactly zero. [7] These fields are used in higher-level data products when better estimates are available, but are included as blanks in the 1.0 files for consistency.

The processor developed in this report parses these files according to schema records (lists of tuples of form `("paramName", (startByte,endByte), parseFunction)`) that define which bytes to take from each file, how to interpret them, and what to name the result for easier access. The processor extracts the requisite bytes from the leader and image files creates a list of signal records which it then combines into an uncompressed complex radar image.

II.2. Range Compression

Range compression is the first step in creating a useful image from the blurry, unfocused raw image. This technique is not unique to synthetic aperature radar, but can be used in any radar system to give increased range resolution without requiring prohibitively high transmitter powers. In brief, range compressed radar mixes its carrier frequency with a signal that only cross-correlates with itself when offset by a small amount. In practice, this is accomplished by linearly sweeping the transmitter frequency over a small range to produced a "chirp" pulse. [1]

The radar echo received by the instrument is proportional to the true reflectivity of the illuminated scene convolved with the chirp. To recover the true signal, we need to perform a correlation of the received signal with the chirp. This is equivalent to passing the received signal through a matched filter sensitive to the radar's specific chirp pattern. One efficient way to perform the correlation is to work in Fourier space.[1] It can be shown that multiplying the complex Fourier coefficients of the the received signal with the conjugate of the coefficients of the reference signal and performing the inverse Fourier transform on the result yields the same signal as simple time-domain correlation operation, albeit with many fewer computer instructions. This is a very widely used technique to decrease processing time in signal processing applications.[1]

From the leader file, we find that the image in question uses a down chirp with a chirp rate of $-5.185\times10^{11} Hz/s$. Next, we generate a matched filter. The approach used in this report is to recreate the chirp signal, sample it at the sampling frequency, and correlate it with the radar echos. The documentation is not entirely clear, but it appears the SignalRecords are downconverted by their carrier frequency and then sampled. Thus, we expect the downcoverted chirp signal received by the instrument from a stationary scatterer to proportional to the Equation 1, where $\dot{f}$ is the chirp rate in Hz/s.

$$C(t) = exp(i\pi \dot{f} t^2) \tag{1}$$

Using Equation 1 as the reference signal, I apply range compression using the Fourier technique. Once range compression is applied to each echo, the magntitude of the complex images shows much sharper features in the cross-track direction, indicating that range compression was successful.

II.3. Range Cell Migration

Range cell migration is an effect where the the signal from a single bright scatterer moves along a track in both range and azimuth space in an uncompressed image. In extreme cases, it gives rise to images like the one shown in Fig. 2. This effect happened to be small in the ALOS images I used, but this step was still necessary to achieve good results.

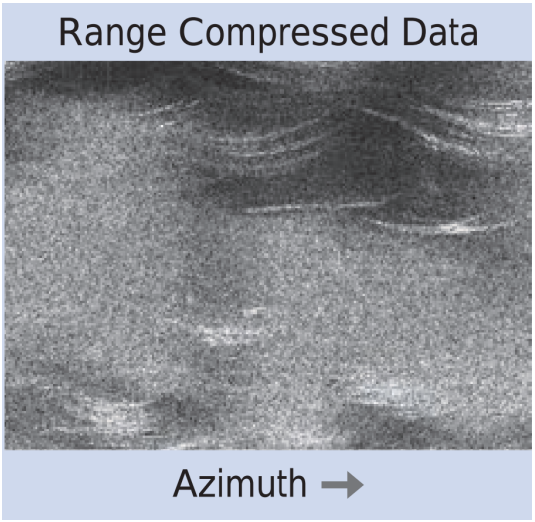

FIG. 2. An extreme example of the effects of range cell migration from [2]. The curved lines visible are from signal bright scatterers whose echos appear farther in the range direction when they are not direction abeam of the radar instrument.

The coupling between range and azimuth position caused by range cell migration must be corrected for to enable efficient azimuth compression. The Range-Doppler Algorithm provides an excellent way to do this.

The Fourier transform of the range compressed image is taken in the along-track direction. In this transformed space, range and azimuth frequency are not coupled, so each set of azimuth frequencies can be shifted in the range direction by an amount R_{rd}, which is derived from the Fourier transform of the approximate range equation (3).

The derivation as seen in [1],[2] begins with the distance R to a scatterer at time η, where η is the time that parameterizes the satellite's motion along its orbit:

$$R(\eta) = \sqrt{R_0 + V_r^2\eta^2} \tag{2}$$

Taylor expanding Equation 2 yields the approximate range in Equation 3.

$$R(\eta) \approx R_0 + \frac{V_r^2\eta^2}{2R_0} \tag{3}$$

The Range Doppler Algorithm corrects for range cell migration by shifting each echo by R_{rd}, removing the frequency dependence of azimuth phase.

$$R_{rd} = \frac{\lambda^2 R_0 f_\eta^2}{8V_r^2} \tag{4}$$

Once this shift is made, the range cell migration is corrected for as seen in Fig. 3.

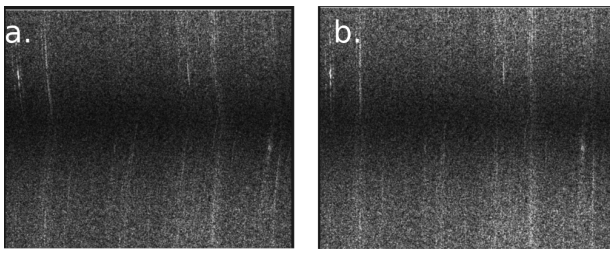

FIG. 3. Two images in the range-Doppler domain, where a Fourier transform has been performed on each column on the image. Visible in a., the uncorrection image, are parabolic curves. After a shift to the left of each row by a azimuth frequency (row index) amount R_{rd}, we obtain the image b. where curves are now straight. In this transformed space, all azimuth phase information for a specific target lies in the same column.

II.4. Azimuth Compression

Next, azimuth compression (sometimes also called focusing[2]), is performed. Azimuth compression is the essential operation of SAR and is what make usable, high-resolution images possible. The process relies on the fact that the phase of the signal returned from a scatterer changes as a function of radar along-track position.

The phase of a signal returned from a scatterer at range R is $4\pi R/\lambda$ (4π because the signal travels a distance $2R$ - there and back), so using the range equation shown in Equation 2 we have the phase

$$\phi(\eta) = -\frac{4\pi}{\lambda}R(\eta) \tag{5}$$

and using the approximate range equation in Equation 3 the target phase reduces to

$$\phi(f_\eta) \approx -\frac{4\pi R_0}{\lambda} - \frac{2\pi V_r^2}{\lambda R_0}\eta^2 \tag{6}$$

.

As with the range compression, the program written for this report generates a recreation of the phase signal and uses it as a matched filter along the along-track direction of the range-compressed complex image. Matched filtering is again performed in the frequency domain for efficiency.

After azimuth compression is performed, we have what is known as a single-look complex image (SLC), with a complex value at each pixel. SLCs can be used directly for interferometric SAR, or converted to from slant range to ground range coordinates and projected onto a map.

III. RESULTS AND DISCUSSION

The raw image shown in Fig. 4 is barely recognizable as the true scene. It is blurry because it has yet to be range compressed and focused and it has bright white stripes in the range axis. These stripes are presumably due to interference from another source of L-band radiation emitting perioding bursts. However, since these stripes are not associated with the PALSAR instrument, they do not contain the imprint of the frequency chirp, so the chirp matched filter greatly suppresses them in the range compressed image.

The range compressed image also shows greatly increased sharpness in the range direction: the long chirp pulses which smear the target scene in the range direction have been compressed down to one range cell each, eliminating the blurring effect. Some bright stripes are visible in the azimuth direction: these lines correspond to bright targets seen at approximately the same range distance in many successive pulses.

Azimuth compression compresses these long stripes into narrow points in azimuth space, using a matched filter for the changing phase of the return signal due to the changing distance of each image cell. Some exceptionally reflective targets (most likely buildings or urban areas) mask the visibility of less bright scatterers, so for viewability the image was thresholded to improve contrast. Also shown is a log-scale image, representing the scattered power in decibels. This image also suffers from poor viewability, but this is mostly due to the logarithm increasing the visibility of noise in low-brightness cells. Thresholding can also make this image more viewable.

There is a good deal of speckle noise visible in the final focused iamge, but this is common in SLC images because of small, angle-dependent scatterers found in each range cell.[1] A number of additional processing steps and image processing techniques can be used to reduce the visual noise in the final image, but they are beyond the scope of this report.

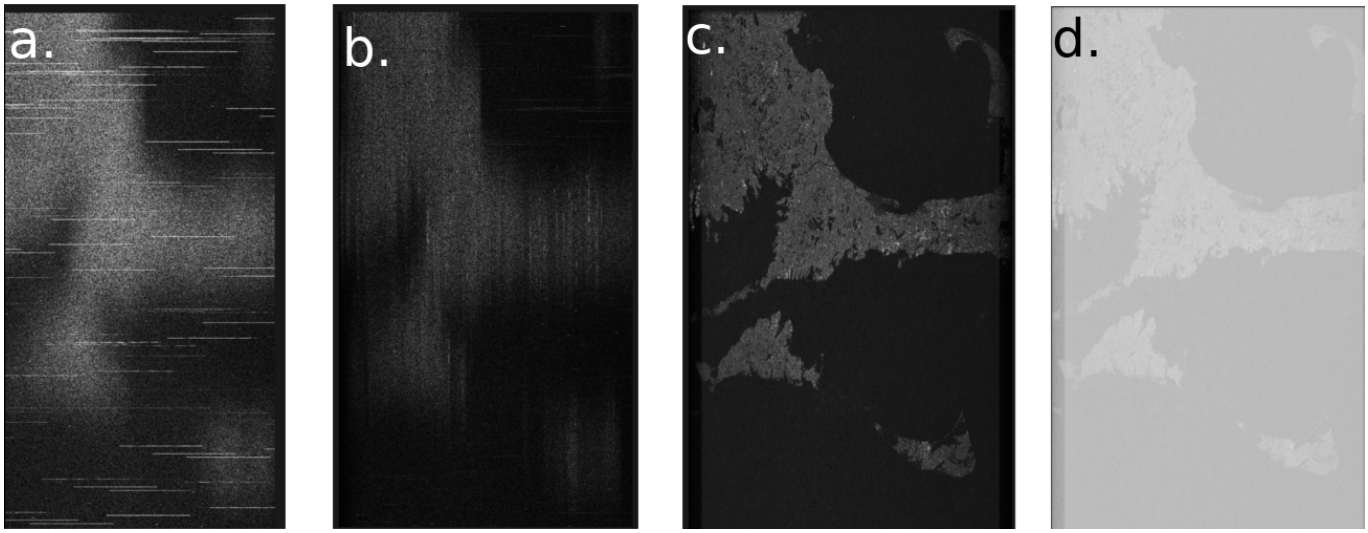

FIG. 4. Radar images at different stages of processing. The range axis is in the horizontal direction and the azimuth axis is in the vertical. a. The raw, unprocessed radar reflection image produced by taking the magnitude of the complex record of each echo. b. The magntitude of the complex range compressed image. c. The magntitude of the complex azimuth compressed image, with contrast adjusted for visiblity. d. The same image shown in log magnitude.

IV. CONCLUSION

This project provided me with the opportunity to apply much of what I learned this semester in a single project. The details of SAR processing build on E&M knowledge as well as radar-specific domain knowledge, and in building this program I solified my knowledge of both topics. I also now have a very good understanding of what goes into radar images and what more we can hope to learn from them. I also hope that my program could be useful to other students who want to know exactly what goes into producing these images. I intend to share the program online once I find time to make it more presentable.

Future work on this processing program could include generating multilook images with greatly reduced speckle noise. Multilook processing involves splitting the azimuth bandwidth into several "looks", which use the Doppler effect to make several overlapping images of the regions ahead of and behind the satellite. When these images are combined, the noise is suppressed and the true image comes out more clearly.[1]

I would also like to generate and interferogram, as that was the original goal of this project. All that would be required to generate an interferogram would be to produce SLCs of the same scene at different times, coregister the images, and then compare the phase of the complex value at each pixel. From such an image one can learn about terrain elevation or deformation to a very high precision.

[1] I. G. Cumming and F. H. Wong, *Digital Processing of Synthetic Aperture Radar Data* (Artech House, 2005).
[2] A. Moreira, P. Prats-Iraola, M. Younis, G. Krieger, I. Hajnsek, and K. P. Papathanassiou, IEEE Geoscience and Remote Sensing Magazine **1**, 6 (2013).
[3] A. Rosenqvist, M. Shimada, and M. Watanabe, 4th International Symposium on Retrieval of Bio- and Geophysical Parameters from SAR Data for Land Applications , 1 (2004).
[4] "About alos - overview and objectives," https://www.eorc.jaxa.jp/ALOS/en/about/about_index.htm, accessed: 2018-11-20.
[5] "The julia language," https://docs.julialang.org/en/v1/.
[6] "Vertex: Asf's data portal," https://vertex.daac.asf.alaska.edu/, accessed: 2018-09-25.
[7] J. E. O. R. Center, "Alos/palsar level 1.1/1.5 product format description ¡english version¿," https://earth.esa.int/web/guest/-/alos-palsar-level-1-product-format-description-vol1-level-1

(2006).