5-2012

Synthetic Aperture Radar Tool and Libraries: A Framework for Geo-Referenced Data Processing and Algorithm Prototyping

Nathan R. Crookston
Utah State University

SYNTHETIC APERTURE RADAR TOOL AND LIBRARIES: A FRAMEWORK FOR GEO-REFERENCED DATA PROCESSING AND ALGORITHM PROTOTYPING

by

Nathan R. Crookston

A report submitted in partial fulfillment
of the requirements for the degree

of

MASTER OF SCIENCE

in

Computer Engineering

Approved:

Dr. Jacob Gunther Major Professor	Dr. Todd K. Moon Committee Member

Dr. Randy Jost
Committee Member

UTAH STATE UNIVERSITY
Logan, Utah

2011

Abstract

Synthetic Aperture Radar Tool and Libraries: A Framework for Geo-Referenced Data Processing and Algorithm Prototyping

by

Nathan R. Crookston, Master of Science

Utah State University, 2011

Major Professor: Dr. Jacob Gunther
Department: Electrical and Computer Engineering

Creating a system for Synthetic Aperture Radar (SAR) image formation can be a huge undertaking as it requires knowledge of several disparate domains. Researchers may be prevented from applying interesting techniques in a particular domain due to hurdles in working with those areas outside their area of interest. This paper presents the SyntheTic Aperture Radar Tool and Libraries (STARTAL) framework for SAR processing that simplifies adding new data formats and prototyping algorithms.

STARTAL provides a user interface for viewing the full data region on ground geometry, selecting sub-regions to process, and viewing processed results. Many common, difficult tasks are provided as libraries for general use. To validate the STARTAL framework, this paper also shows imagery which has been processed with algorithms developed at Utah State University (USU) which are derived from a model-based expression of the relationship between collected SAR data and ground geometry.

(48 pages)

Public Abstract

Synthetic Aperture Radar Tool and Libraries: A Framework for Geo-Referenced Data
Processing and Algorithm Prototyping

by

Nathan R. Crookston, Master of Science
Utah State University, 2011

Major Professor: Dr. Jacob Gunther
Department: Electrical and Computer Engineering

Creating a system for Synthetic Aperture Radar (SAR) data processing can be a huge undertaking as it requires knowledge of several loosely related areas. This can be a barrier to some SAR research as researchers with deep knowledge in only some areas may not be able or willing to gain the required knowledge in other areas. The SyntheTic Aperture Radar Tool and Libraries (STARTAL) framework provides a system which simplifies creating and testing algorithms. It is composed of several interchangeable concepts and allows researchers to deal only with those areas that align with their interests. The STARTAL project provides capabilities to generate images based on SAR research done at Utah State University (USU).

For Ashley, Andrew, Anne, and Kate

Acknowledgments

I'm very grateful for the many people throughout my educational and professional career who taught, inspired, and encouraged me. I'm grateful to my parents who had high expectations of me but gave a higher level of support. I'm indebted to Dr. Brandon Eames's mentoring and guidance for much of my professional success and for encouraging me to return to school to pursue this degree. Thanks to Dr. Jacob Gunther for taking me as his student and providing me with both an interesting project and countless explanations of the underlying theory. Hopefully I've got a handle on it now. Thanks also to Dr. Roger West for his work, help, and encouragement. I'm grateful to my employer, Space Dynamics Laboratory, for funding my academic pursuits and tolerating my irregular work hours.

Most importantly, I thank my wife for her love and support. Though the importance of this project pales in comparison with the importance of our family, she let me work on it anyway (for long hours and at strange times). I love you, Ashley. Finally, I'm grateful for my children's help and curiosity. Maybe my programs run better without their help, but the entertainment value is much lower.

Nathan Crookston

Contents

List of Figures

Figure Page

Acronyms

CBP	Convolution Backprojection
CCD	Coherent Change Detection
COM	Component Object Model
CSS	Cascading StyleSheets
DTED	Digital Terrain Elevation Data
ECEF	Earth-Centered Earth-Fixed
GPU	Graphics Processing Unit
	Datastream-based processing unit which achieves a high number of floating point operations per second when the same operations must be applied to a large number of data elements, as is the case for rasterization.
GUI	Graphical User Interface
KML	Keyhole Markup Language
	Files written in this XML-formatted language may be loaded in Google Earth. KML files may contain marker points, lines, areas, and volumes. In STARTAL, KML files often contain Icon tags which allow images to be displayed on the globe.
KMZ	Zipped KML file
	When a KML file references a local image or other resource, that file must be available when the KML file is loaded. To allow this (rather than forcing KML files to only reference files with absolute and publicly accessible URLs), the KML file may be zipped together with any media required to display it. STARTAL follows Google's recommendation to zip a subdirectory containing the media together with the KML file. Thus, if the user wishes to see the overlay image in some other application, the KMZ file must first be unzipped.
HTML	Hyper-Text Markup Language
LLA	Latitude-Longitude-Altitude
LFM	Linear Frequency Modulated
	A signal with good pulse compression properties often used to illuminate ground targets in SAR.
NASA	National Aeronautical and Space Administration

NITF	National Imagery Transmission Format
NGA	National Geospatial-Intelligence Agency
NRL	Naval Research Laboratory
NWU	North West Up
PRF	Pulse Repetition Frequency
RCS	Radar Cross Section
SAR	Synthetic Aperture Radar
SDL	Space Dynamics Laboratory
SNR	Signal-to-Noise Ratio
SRTM	Shuttle Radar Topography Mission
STARTAL	SyntheTic Aperture Radar Tool and Libraries
TRE	Tagged Record Extension
USGS	United States Geological Survey
USU	Utah State University
XML	eXtensible Markup Language

Chapter 1

Introduction

Utah State University (USU)'s development of a linear model relating Synthetic Aperture Radar (SAR) data samples to ground geometry suggests innovative algorithms which may produce significantly higher-quality SAR images than existing methods [1]. While the effectiveness of these algorithms at producing images from synthetic data has been demonstrated, the application to actual SAR data requires more than just signal processing technology. In addition to providing matched filtering capabilities, a SAR data processing system must provide methods and abstractions for dealing with elevation models and geographic transformations. Furthermore, since the datasets are often quite large, the processing needs to be as efficient as possible and, for the most expensive algorithms, must allow only portions of the data to be processed. The SyntheTic Aperture Radar Tool and Libraries (STARTAL) project provides the technology necessary to process SAR data and demonstrates that ability by implementing an algorithm suggested by the linear model.

1.1 Interoperable Abstractions

The STARTAL project's main goal is to facilitate algorithm prototyping and validation. It may be possible to write more efficient algorithms which take advantage of the raw data format or other data specific parameters. This is intentionally avoided in STARTAL because rewriting algorithms for each new data type leads to a large number of slightly varying algorithms. This can be a maintenence burden because, in addition to needing to write each new version, corrections or improvements to an instance of the algorithm must be made in all copies. Furthermore, data-specific efficiency improvements generally do not affect the worst-case complexity of the algorithms. STARTAL provides an abstraction of the actual data used which allows a single algorithm to be used with all data which fit that abstraction.

The SAR data abstraction assumes that the data is georeferenceable and that enough signal information is known to determine the expected signal phase for any data sample given a ground pixel and plane location. This is the case for the NuSAR data used, but, surprisingly, not for many of the freely available SAR datasets. Many such datasets have been fastidiously scrubbed of both geographic markers and relevant signal information. While it is conceivable to use STARTAL utilities to process such data (by re-purposing some abstraction methods to return different information), the fact that it does not fit the abstractions means that it will not interoperate with algorithms for data that do fit the abstraction. Supporting data which does not fit STARTAL's abstraction may be accomplished by either broadening the abstraction to cover both cases, or by creating a second abstraction with a separate set of algorithms. The former option would allow any registered data type to be used with any algorithm, but the formulation of those algorithms sometimes require information that simply is not present in those datasets. It would also require more setup work inside the algorithm abstraction to convert the loaded data to a supported format which is not necessary with the current abstraction. The latter option is preferable if there is a fundamental difference between what different SAR data types provide and what some algorithms require. However, it would increase the complexity of process task creation and processing dispatch. Because the main goal of STARTAL is algorithm prototyping (and the algorithm is specifically based on USU's model-based formulation), supporting data types lacking geographical or signal information is not a top priority and thus is not currently done.

Free elevation data are available online through many services, and others exist for United States government contractors or paying customers. Since different datasets may have competing strengths (e.g., some datasets may be spatially dense and very accurate for limited areas), STARTAL allows different datasets to be chosen.

1.2 Ease of Use

Many SAR processing algorithms are computationally expensive. That expense can sometimes be ameliorated by processing only certain sub-regions. Determining both the region which the data encompasses and the region to process should not involve tedious hand-manipulation and entry of geographic coordinates. The user should instead be able to view a graph showing the relative location of the data footprint, the sensor platform, and the regions to process. Ideally, that information would be overlaid on high-resolution imagery to provide better contextual awareness.

High-resolution imagery for some areas of the United States is available for download from United States Geological Survey (USGS), but the process is slow and laborious. Furthermore, creating a system to display merged map and elevation data with reasonable user and programming interfaces is a huge task likely to yield poorer results than using or modifying an existing system. There are many map services available on-line, but most provide either poor resolution or do not allow programmatic manipulation in a cross-platform manner. The Google Earth executable provides very good map data and has been ported to several platforms, but the Component Object Model (COM) interface for programmatic manipulation was always limited and finally deprecated in August 2010. In 2008, Google released a web browser plug-in which provides the same imagery and allows sophisticated, programmatic view manipulation using the Javascript programming language. Because this plug-in supports all desired operations in a manner likely to work on other platforms, STARTAL uses it to display all geographic information.

Because STARTAL's main focus is algorithm prototyping, the ability to record and re-use parameters supplied to previous processing invocations is also very desirable. Tying the map display into task creation and termination further improves usability. Because the geographic display mandates the use of a web page, and because task elements and outputs should be recorded, STARTAL uses a client-server architecture. A dynamic web page is served to clients connected via web browser which allows task elements to be created. Those elements are stored in a database on the server where they may be retrieved when creating and dispatching tasks. Once a task is dispatched, the server runs an executable with the task elements passed as arguments. The results of those tasks are also stored in the database and may be served to clients upon demand.

Once an algorithm has been developed, it should also be convenient to load into different instances of STARTAL. This may be accomplished by modifying the STARTAL code to be aware of the algorithm and providing either the necessary source code or the needed executables. If source code is provided, the recipient must be able to compile the code, which may require obtaining all libraries used in the new algorithm and in the STARTAL-based system. Additionally, there are cases where providing source code is impossible. Providing pre-built executables avoids issues with source disclosure and compilation. This, however, can make algorithm comparisons more difficult, since separately developed algorithms will exist in different STARTAL framework executables. STARTAL makes mixing algorithms and data types easier by loading shared libraries at runtime. This avoids the weaknesses of previously mentioned approaches without compromising other goals. Creating shared libraries places some constraints on the implementation language and often requires that the shared library code be compiled with the same compiler and flags as the core STARTAL system.

1.3 Cross-Platform Portability

The STARTAL project is most useful to researchers if it can be executed on a variety of hardware platforms and works independent of the operating system. This requirement has already been mentioned where it affected the choice of map display. It also affects the choice of server, database, programming languages, and libraries. There are many good frameworks available for serving web pages. The well-designed, portable Django project is used in STARTAL. Any Django-supported, cross-platform web server, and database may be used when deploying STARTAL; the easily set-up WEBrick server and SQLite database are used by default. The STARTAL project uses Hyper-Text Markup Language (HTML), Cascading StyleSheets (CSS), and Javascript in its front-end, and Python and C++ (1998 ISO Standard) in its back-end. All languages and all libraries used within those languages are portable to at least Windows and Linux systems.

1.4 Efficiency

A final consideration is how efficiently the dispatched task executes to generate the task output. Because of the general complexity of SAR data processing, and the specific complexity of processing based on USU's model, it is imperative that processing time be reduced as much as possible for the dispatched executable. A good way to increase efficiency is to choose algorithms with the best worst-case behavior. Where optimal algorithms are already being used, the implementation of those algorithms and the choice of language for implementation may be examined. Efficiency is still a lesser concern than proper abstraction, portability, and expressiveness; however, C++ fulfills all these requirements. Due to the unavailability of compilers that fully support the 2011 revision of the ISO C++ standard, STARTAL sacrifices some of the efficiency and expressiveness of the 2011 revision for the greater portability of the original 1998 standard.

Basic SAR processing concepts, including theoretical image resolution, are discussed in Chapter 2. Chapter 3 discusses elevation models and gives a precise definition for occasionally ambiguous geographic terms. It also defines coordinate systems commonly used by STARTAL and describes how to convert between them.

A description of USU's SAR data collection model follows in Chapter 4. As STARTAL was the first application to form non-synthetic data into the model's constituent matrix and vectors, a reasonably thorough discussion of each element is included. Model-based methods of forming images are also discussed in that chapter.

Finally, a description of the user interface, the programming interface, and extension points is given in Chapter 5. Specific elevation models, antenna types, and algorithms are discussed along with the abstractions. Screen shots of algorithm outputs, including imagery formed using USU's model-based approach [1] are shown in that chapter.

Chapter 2

SAR Processing Concepts

Synthetic Aperture Radar transmits energy from an antenna in a manner which allows the signal reflections from the ground to be recorded. Because the energy is radiated from an antenna, SAR systems do not require ambient illumination to operate. The wavelengths of the signals used for SAR also suffer little distortion when passing through clouds and precipitation. These properties make SAR a compelling choice for imaging in a variety of conditions. A full introduction to SAR may be found in Cumming and Wong [2].

2.1 SAR Data Generation and Collection

2.1.1 SAR Terminology

The direction of platform motion, or more precisely, the velocity vector is referred to as the azimuth direction [3]. The slant range is the length of the slant vector which originates at the antenna and continues down the antenna bore-sight. The ground range is the distance of the result of projecting the slant vector onto the ground. In this paper, unqualified references to range refer to the slant range. It is important to note that the slant range is not necessarily orthogonal to the azimuth vector; that only occurs when the antenna pointing direction is orthogonal to platform motion.

2.1.2 Signal Properties

The signal reflected by the ground must be distinguishable from any noise present in the environment. This project focuses on pulsed Linear Frequency Modulated (LFM) signals due to their simplicity and ubiquity [4]. The pulses for the signal must occur frequently enough that aliasing between samples of different pulses does not occur. The rate of chirp reoccurrence is the Pulse Repetition Frequency (PRF) and is based on the velocity of the platform, the desired resolution, and the expected range to both the target and the nearest ground point [3]. An equation for the transmitted signal is

$$s(t) = w(t)cos(2\pi f_c t + \pi\alpha t^2).$$

2.1.3 Data Collection

Data collection occurs after a pulse is emitted. The receiving antenna (which may be the same as the transmitting antenna) is polled every T_s seconds and the results are stored. The return for a single reflector is

$$r(t) = \sigma_0 w(t - \tau_0)cos(2\pi f_c(t - \tau_0) + \pi\alpha(t - \tau_0)^2),$$

where τ_0 is propagation time from the transmitter to the reflector and back to the receiver, and σ_0 is the reflector's complex-valued Radar Cross Section (RCS). The complex phase is extracted from the received signal via quadrature demodulation. The equation for the demodulated data is

$$r_C(t) = \sigma_0 w(t - \tau_0)e^{-j(2\pi f_c\tau_0 - \pi\alpha(t-\tau_0)^2)}.$$

Because of the distance of the platform from the region of interest, there is usually a delay, known as the range gate delay (τ_{rgd}), between pulse emission and collection commencement. If the delay were eliminated then many more samples would be collected, but none of them would possibly have returns from the region of interest. A collection of pulses and all samples for each pulse forms a data grid, represented in this document with all samples from a single pulse filling a row of the grid. Because the sampling period and range gate delay are also known, it is possible to determine an associated range for each sample using the formula

$$R = \frac{c(\tau_{rgd} + n * T_s)}{2}.$$

2.2 SAR Image Formation

Each sample in the unprocessed data grid contains the sum of the reflections of ground points at a matching range from the platform [1]. Unlike many passive sensors with optics that attempt to ensure that a single light-sensitive element is excited by a single area on the ground, raw SAR data looks like noise until the individual data elements are decorrelated to determine the reflectivity of individual ground areas. The maximum theoretical resolution of imagery, or the smallest distinguishable distance between different targets, is dependent on the signal properties, the platform velocity, the PRF, and distance from the target.

2.2.1 Range Resolution

The first type of SAR imagery resolution to consider is range resolution, or minimum distance between two targets in the cross-track direction which still allows those targets to be distinguished. If a simple rect function of width T_p is used as the pulse, then the range resolution would be $\frac{T_p c}{2}$. Making the rect function thin enough to provide good range resolution requires that the peak signal power be increased to maintain a good Signal-to-Noise Ratio (SNR). Platform capabilities may restrict such a design; fortunately, similar results may be achieved using pulse compression techniques. The interested reader may consult other sources for full detail on pulse compression and matched filtering [4]. Briefly, pulse compression and matched filtering allow the signal-to-noise ratio to be proportional to the bandwidth of a chirp signal. This naturally requires modification of the transmitted pulse and also requires some preprocessing of the received data prior to image formation. Some SAR platforms perform this preprocessing prior to transmitting or storing the data. Interestingly, in radar the range from the target does not directly affect the theoretical resolution (unlike optical sensors). As range increases, however, the SNR decreases.

2.2.2 Azimuth Resolution

The second type of resolution to consider is azimuth resolution. Azimuth resolution for a traditional radar is inversely proportional to the azimuth beam-width, which is itself inversely proportional to the antenna aperture size. Therefore, increasing the aperture size will improve the azimuth resolution. However, this is usually expensive and infeasible for smaller platforms. It is possible to use signal processing to synthesize a larger aperture using multiple correlated pulses. The azimuth beam-width θ_{bw} may be approximated as

$$\theta_{bw} \approx 0.886\frac{\lambda}{L}$$

[3], where λ is the radar wavelength and L is the physical antenna length. The azimuth resolution is approximately $\frac{2}{\theta_b w}$, and is therefore approximately equal to half the physical antenna length. Thus, unlike traditional radar, a smaller antenna will increase the image resolution. Qualitatively, a smaller antenna has a wider beam which illuminates each reflector longer. Thus, a smaller antenna gives a larger synthetic aperture. As with range resolution the SNR and the PRF place a lower limit on how wide the antenna beam can be.

2.2.3 Post Image Formation Algorithms

There are also algorithms which are applied to focused images. For example, processed SAR imagery may have several artifacts which reduce image quality. Excessive side lobes, due to inter-pixel interference and measurement noise, may make features of the target image blurry or otherwise indistinguishable. Many forms of post-image formation auto-focus are used in practice. Coherent Change Detection (CCD) is another algorithm that compares two focused images to detect changes between them. Post-image creation processing is not within the scope of this project.

Chapter 3

Geographic Concepts

3.1 Elevation Models

Elevations and altitudes are specified relative to a surface which approximates that of the earth. The approximating surfaces relevant to the current discussion are the ellipsoid [5] and the geoid [6].

3.1.1 Ellipsoid

The ellipsoid is like a sphere whose diameter measured at the poles differs from the diameter measured at opposite points along the equator. The radius from the earth's center point to the intersection of the prime meridian and the equator is the semi-major axis. The radius from the earth's center point to the north pole is the semi-minor axis. The ellipsoid is often described by many other derived parameters like the flattening and the first and second eccentricity. The ellipsoid is generally specified as any two of the flattening, semi-major and semi-minor axes.

3.1.2 Geoid

The geoid is the surface of gravitational equipotential. It is often described as the mean sea level. The surface is approximated by a mathematical model described by spherical harmonics. Determining a specific value in the model requires solving a polynomial equation with 360 terms, although fewer terms may be used depending on accuracy requirements. It is common to pre-compute specific values and interpolate those when the equation solution time is prohibitive for a given application. In STARTAL, this is done in those cases where conversion from geoid to ellipsoid is required.

3.2 Elevation Data Sources

There are several commonly used, machine-readable elevation data sources. The following sources were the most convenient to use when initial STARTAL development began.

3.2.1 Shuttle Radar Topography Mission (SRTM) Data

The National Geospatial-Intelligence Agency (NGA) and the National Aeronautical and Space Administration (NASA) have created a freely available dataset which supplies the elevation of the ground for selected latitude/longitude points. There is a point for each arc-second (approximately 30 meters) over the United States (referred to as SRTM1) and a point for every three arc-seconds (approximately 90 meters) elsewhere (referred to as SRTM3). Elevation readings are available in files which cover one degree latitude by one degree longitude. There are some areas without data, most notably the poles. SRTM data is distributed by the Jet Propulsion Laboratory at `http://www2.jpl.nasa.gov/srtm/`.

3.2.2 Digital Terrain Elevation Data (DTED)

NGA uses DTED data as its official standard format for elevation data. There are multiple data levels. DTED 0 has an elevation point every 900 meters, DTED 1 every 90 meters, and DTED 2 every 30 meters. While DTED 0 is freely available on-line, other levels are restricted to the Department of Defense, United States Government agencies, and United States Government contractors. Most images produced by STARTAL during initial development were made using DTED-based elevation data, but due to distribution restrictions only SRTM-based elevation loading is provided.

3.3 Coordinate Systems

STARTAL defines several coordinate systems which describe local and global platform positions. Some of the following coordinate systems are widely used while others are specific to STARTAL. Even some commonly used coordinate systems may have ambiguous specifications. The following precisely specifies their meaning in STARTAL.

3.3.1 Latitude-Longitude-Altitude (LLA)

Latitude and longitude [7] are specified in degrees. Zero degrees in latitude corresponds to the equator, 90 degrees to the north pole, and -90 degrees to the south pole. Zero degrees in longitude corresponds to the prime meridian and ±180 degrees correspond to the international date-line. Moving east from the prime meridian gives positive longitude values while moving west gives negative values. Altitude is given in meters and is relative to the reference ellipsoid. Negative altitudes move toward the center of the ellipsoid, positive values move away from the center.

3.3.2 Earth-Centered Earth-Fixed (ECEF) Coordinate System

ECEF coordinates, also referred to as geocentric coordinates [8], are described by a right-handed coordinate system with origin at the center of the earth, the positive x-axis formed by the vector from the origin to the point at the intersection of the equator and the prime meridian. The z-axis is described by the vector from the origin to the geographic north pole, and the y-axis is described as the result of computing the cross-product of the z-axis and the x-axis. All distances in this coordinate system are specified in meters.

3.3.3 North West Up (NWU) Coordinate System

The NWU coordinate system is relative to a local tangent plane. Typically its origin coincides with a ground elevation. A vector orthogonal to the plane and pointing away from the earth's center is the z-axis, the x-axis lies in the tangent plane and points north (it is the intersection of the tangent plane and the plane formed by the z-axis and a vector pointing at the geographic north pole). The y-axis is formed by crossing the z-axis into the x-axis.

3.3.4 Aircraft Coordinate System

The aircraft coordinate system's origin is coincident with the aircraft location. The vector from the center to the nose is the positive x-axis, the orthogonal vector from the center toward the left wing is the y-axis, and the z-axis is described as the result of crossing the x-axis into the y-axis, which results in a vector pointing out the top of the plane.

3.3.5 Antenna Coordinate System

The antenna coordinate system is a right-handed Cartesian coordinate system defined with the origin at the aircraft location, the positive x-axis along the bore-sight of the antenna, and the positive z and y axes in the plane orthogonal to the x-axis. The y-axis is parallel to the horizontal antenna pattern and the z-axis is the cross product of the x-axis into the y-axis. If the antenna is side-looking without squint (i.e., rotated 90 degrees from the front of the plane), then the positive y-axis is pointing toward the plane tail.

3.3.6 Coordinate System Conversions

A common requirement in geospatial image processing is converting between coordinate systems. Converting from LLA to ECEF requires the solution of these equations:

$$X = (\nu + h)\ cos\phi\ cos\lambda Y = (\nu + h)\ cos\phi\ sin\lambda Z = [\nu(1-e)^2 + h]\ sin\phi,$$

$$\nu = \frac{a}{\sqrt{1 - e^2 sin^2\phi}},$$

$$e = \sqrt{2f - f^2},$$

$$f = \frac{a-b}{a},$$

where a is the semi-major axis length, b is the semi-minor axis length, ϕ is the latitude, λ is longitude, and h is the altitude above the ellipsoid. Converting from ECEF to LLA is more computationally intensive, typically requiring an iterative solution to arrive at an accurate value. Converting between the Cartesian coordinate systems is actually much simpler. Because the transformations between coordinate systems are affine it is possible to construct an invertible matrix representing those transformations. Furthermore, it is possible to aggregate transformations via matrix multiplication.

3.3.7 Yaw, Pitch, and Roll Rotation Descriptions

Some of the traditional methods describing transformations between coordinate systems may also be ambiguous without an explicit convention. Yaw, pitch, and roll are used differently (and incompatibly) in different places. For example, in the SENSRA Tagged Record Extension (TRE) of the National Imagery Transmission Format (NITF) standard, the angles are order-independent projections into the coordinate planes (which actually underspecifies the transformation). In STARTAL, the angles (supplied in radians) are treated as an ordered set which fully specify the translation. A vector coincident with the positive x-axis is first rotated by yaw radians in the $z = 0$ plane (where clockwise is positive). That rotated vector is then rotated about the the normal vector in the $z = 0$ plane by pitch radians, where negative pitch values move into the $z < 0$ volume. Finally, the coordinate system is rotated about that vector by roll radians. Clearly the x-axis vector needs only two angles for full specification, but mapping other points into the coordinate system not on that axis requires that the roll be specified.

3.3.8 Azimuth and Elevation Descriptions

Vectors in a coordinate system may be specified using azimuth and elevation pairs, where azimuth and elevation are given in radians. Unlike yaw, pitch, and roll, azimuth and elevation descriptions are projections into two axial planes. Azimuth is a projection into the x, y plane while elevation is a projection into the x, z plane.

Chapter 4

SAR Algorithm Descriptions

4.1 Model-Based Image Formation

STARTAL currently supplies some algorithms which are based on a model developed at USU [1, 9]. West demonstrates that the relationship between the collected data and the estimated reflectivity of ground areas may be represented by a set of linear equations $d = Fg$ [1], where d is a vector of data samples relevant to a given target area, g is a vector of unknown ground reflectivities, and F is a matrix where each element specifies the modulation that a single ground pixel's reflection undergoes prior to contributing to a single sample of a single pulse.

4.1.1 Data of Interest

Once a specific target area is chosen, the d vector may be formed by identifying that data which includes contributions from the region of interest. That is, each pulse is checked with the region to determine if it illuminates the region. Pulses which do not illuminate the area (and thus do not contribute to an estimation of the region's pixels) are discarded. The ranges of the samples of the remaining pulses are then compared with the near and far ranges of the region of interest (for the given pulse) and only those samples which include contributions from the region of interest are kept. The samples which remain are then stacked into the d vector. The data may also be matched filtered to dramatically reduce the number of samples present in d. Despite the large amounts of data which are discarded, d is still usually very large due to the high PRF. The vector d is typically referred to as the data of interest. An example of the data of interest is given by figure 4.1. The full image (both black and white portions) represents all samples considered to be within the beamwidth. Each row represents the returns for a single sample. Within each row is a band of white pixels which represent those samples which contribute to the region of interest. The data of interest consists of those white pixels.

4.1.2 Region of Interest and Region of Interest Closure

The g vector consists of all pixels whose ranges must be known to decorrelate the data in d. This is created by finding the near and far range of the samples for each pulse in the data of interest and projecting the ground points which correspond to those ranges. The envelope formed by those ground points is subdivided into a grid whose points are spaced at predetermined resolution. The g vector is referred to as the region of interest closure. Intuitively, the region of interest closure contains all the pixels whose reflectivities should be estimated, plus all the pixels whose range and angles must also be known to correctly remove their contributions from the data used for the region of interest. Figure 4.2 shows the region of interest and the region of interest closure for a wide-beam antenna with points spaced one meter apart. The yellow points represent the region of interest and both the yellow points and the blue points represent the region of interest closure.

4.1.3 F Matrix

The F matrix describes how the ground reflectivities are modulated and summed to yield the data of interest. Each element gives the expected magnitude and the expected phase, modulated by the antenna pattern, of a single data of interest element for a single region of interest closure element. The formula for computing a single element of F is

$$F_{ij} = \beta \frac{(T_p - |\eta|) sin(\pi\alpha(T_p - |\eta|)\eta)}{\pi\alpha\eta} e^{-2j\pi f_c \tau} e^{-j\pi\alpha T_p \eta},$$

where $\eta = \tau_{rgd} + nT_s - \tau$ (where n is the data sample index), $\alpha =$ the chirp slope rate in $\frac{Hz}{s}$, $\beta =$ attenuation due to the camera pattern (dependent on the ground pixel), $\tau =$ time delay for signal to propagate to and from the reflector (also dependent on the ground pixel), $\tau_{rgd} =$ time delay between pulse transmission and sampling commencement, $T_p =$

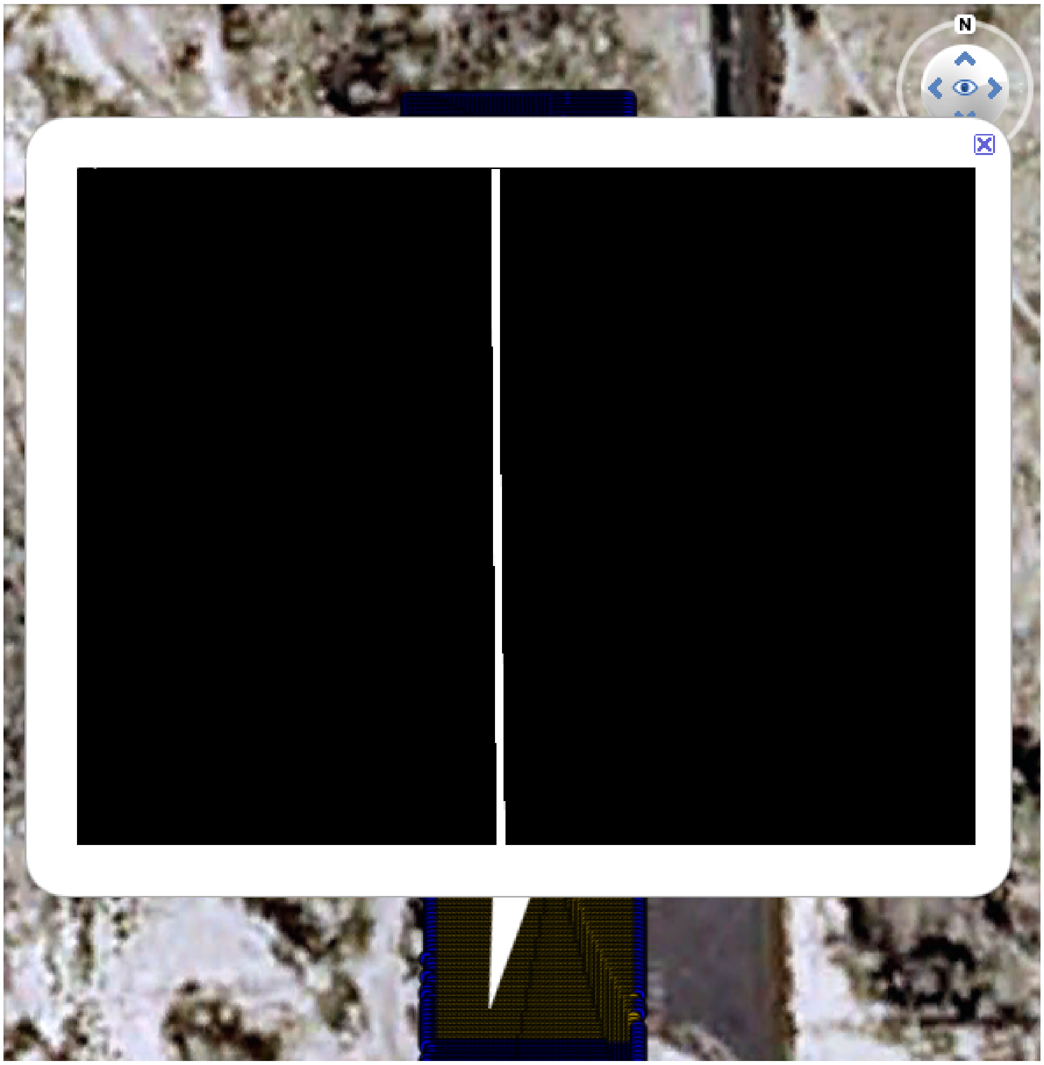


Fig. 4.1: Data summary for L-band SAR data.

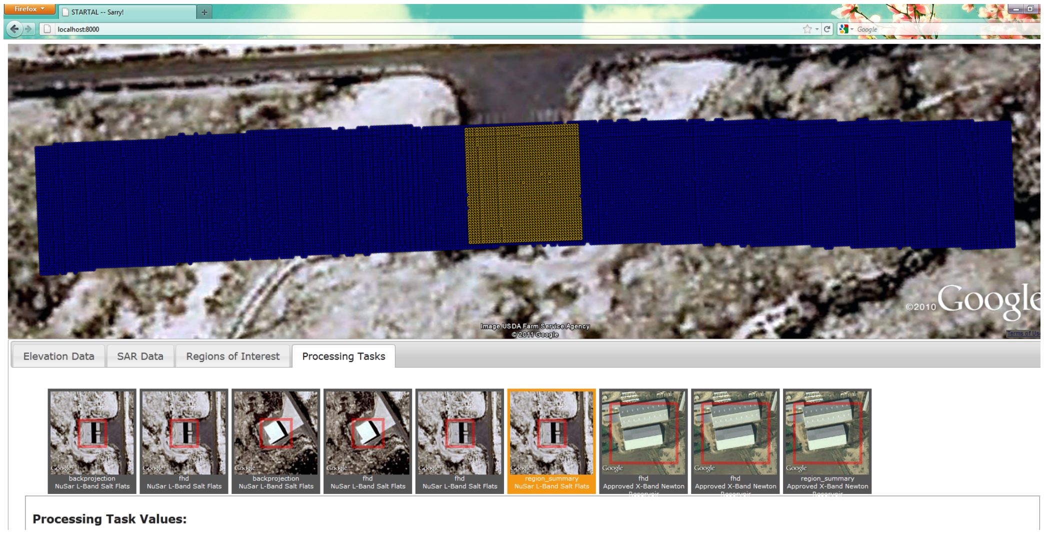


Fig. 4.2: Region summary for L-band SAR data.

time between pulses, T_s = time between samples, and f_c = the carrier frequency.

Because the data of interest is always larger than the region of interest closure, F is always a tall matrix – generally much taller than it is wide. The structure of F does lend itself favorably to caching and updating so that its computation time can be significantly reduced.

4.2 Adjoint Algorithm

The equation $d = Fg$ can be thought of as a projection from the region of interest closure space onto the data of interest space. To form SAR imagery from collected data, a projection from the data of interest space must be determined. Along with many other things, West's dissertation explores the idea of minimizing the error of such a projection via traditional linear algebraic techniques. Relevant to this discussion, the adjoint of F may be considered a projection from the data of interest space to the region of interest closure space, albeit without any guarantee of error minimization by some norm. West argues that the result of $F^H d \approx g$ is functionally equivalent to traditional Convolution Backprojection (CBP).

West was able to demonstrate the effectiveness of various projections using synthetic data, but the application to real data was first performed by STARTAL. The images produced using STARTAL's Adjoint algorithm are very comparable to CBP-produced images. This validates both the theory and the implementation of the construction of the equation elements.

4.3 Maximum Likelihood Algorithm

A projection which minimizes the error by the l_2 norm may be written as

$$(F^H F)^{-1} F^H d = g.$$

Though this projection may be performed using STARTAL's existing matrix and vector objects, the solution requires large amounts of memory and larger amounts of processing time. The matrix dimensions (the number of rows, in particular) make the $F^H F$ multiplication very time consuming. Additionally, the gram matrix $(F^H F)$, a square matrix whose dimensions equal the number of elements in the region of interest closure, is still very large. Thus, it is computationally important that the size of the region of interest closure be small, as the complexity of solving the described equations is $O(n^3)$. Some exploration of ways to reduce the complexity of solving this system has been performed by West et al. [10,11], but more investigation must occur before this approach can be extensively applied.

Chapter 5

Overview of SyntheTic Aperture Radar Tool and Libraries

5.1 STARTAL User Interface

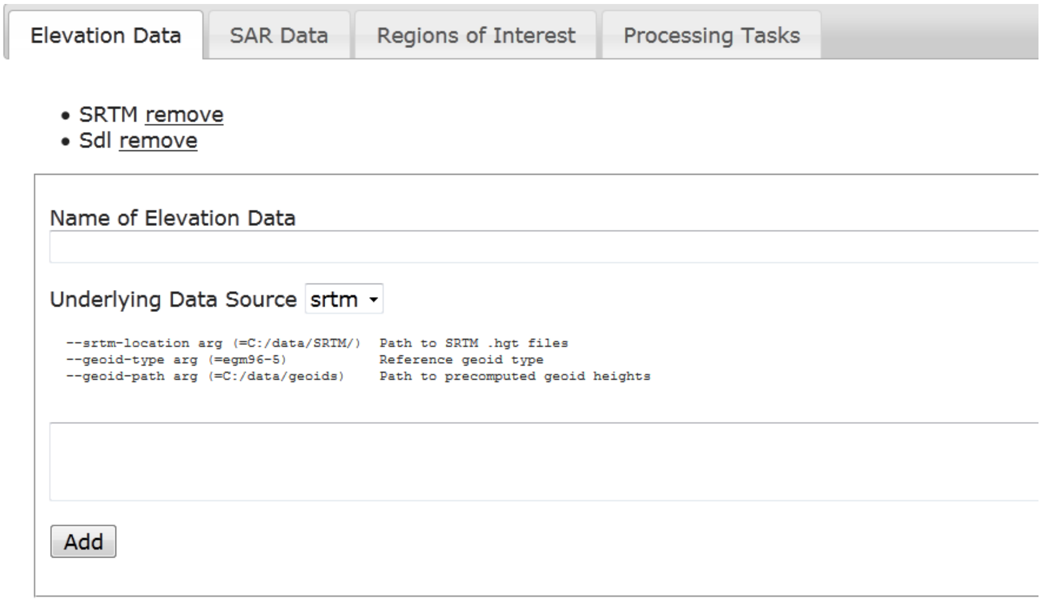
The upper portion of the STARTAL GUI is the Google Earth globe and provides some spatial context when choosing data and regions to process. A set of tabs is displayed below the globe. These allow the user to create instances of data types required to generate output products.

5.1.1 Elevation Data Tab

The first tab, shown in figure 5.1, allows instances of the elevation type to be created. A list of already-created elevation data instances is displayed, followed by a link to generate a new elevation data object. A new elevation data instance may be given a descriptive name and assigned one of the underlying data sources (e.g., SRTM data). Different data sources may require additional parameters. Once a source has been selected, the optional and required parameters for that source are displayed, with a text input allowing the user to specify the additional information. When data is added it is appended to the list of data sources. Selecting an existing data source on this page displays the "Add Elevation Data" form and fills those entries with the selected item's data.

5.1.2 SAR Data Tab

SAR data to import must already be locally available on the server. The typically large size of SAR data collections make uploading user-supplied data impractical. To import data, the user must select the SAR data tab from the STARTAL main GUI. A list of imported data is shown in the tab, followed by a link to add SAR data. The format of the data, the chirp parameters, and the aircraft mounting position of the antenna must be represented by



- SRTM remove
- Sdl remove

Name of Elevation Data

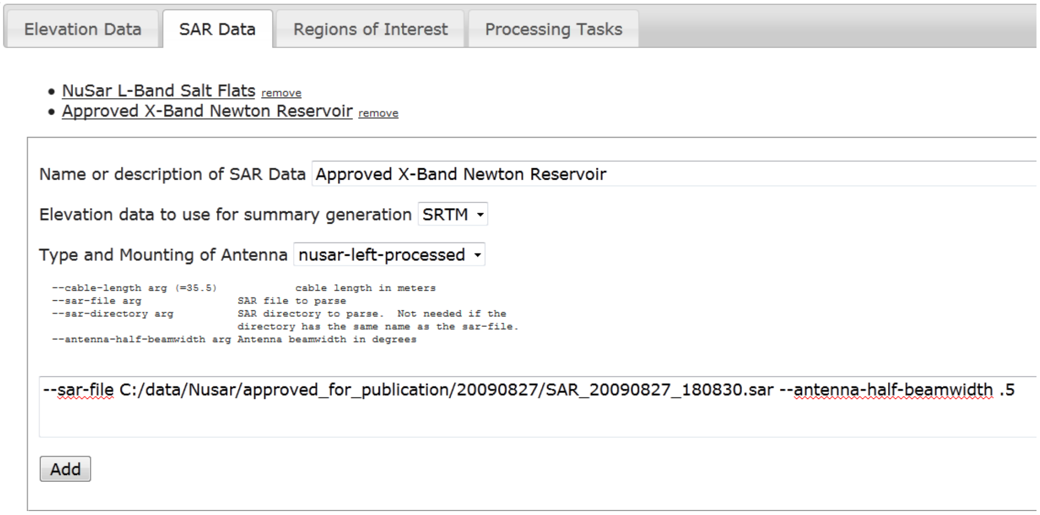
Underlying Data Source srtm

```
--srtm-location arg (=C:/data/SRTM/)  Path to SRTM .hgt files
--geoid-type arg (=egm96-5)           Reference geoid type
--geoid-path arg (=C:/data/geoids)    Path to precomputed geoid heights
```

Fig. 5.1: Elevation data tab.

some instance of an Antenna object. A screen-shot of the SAR data tab and the “Add SAR Data” form are displayed in figure 5.2. Registered antennas are displayed in a selection box. Different antenna models may require additional information. Required and optional command-line arguments are displayed upon selecting a valid antenna type and a text entry box accepting those parameters is displayed below.

After the data and required processing parameters have been specified, a summary Zipped KML file (KMZ) file is generated which shows the location of the aircraft for an evenly distributed subset of pulses. It also shows the near and far range points on the ground. Although the antenna instances do not require elevation data, it is necessary to perform the aforementioned point projection. When an item from the imported data list is selected, the overview KMZ file is displayed on the globe and the view is focused on the newly added data. The “Add SAR Data” form is also displayed and populated with the previous, data-specific processing parameters. The elevation data selection is not stored (as it is not a property of the data themselves), and is thus not modified when selecting an item. Showing the summary KMZ file provides some context for the user in choosing the target area for which an image should be created.



- NuSar L-Band Salt Flats remove
- Approved X-Band Newton Reservoir remove

Name or description of SAR Data: Approved X-Band Newton Reservoir

Elevation data to use for summary generation: SRTM

Type and Mounting of Antenna: nusar-left-processed

```
  --cable-length arg (=35.5)            cable length in meters
  --sar-file arg              SAR file to parse
  --sar-directory arg         SAR directory to parse.  Not needed if the
                              directory has the same name as the sar-file.
  --antenna-half-beamwidth arg Antenna beamwidth in degrees
```

```
--sar-file C:/data/Nusar/approved_for_publication/20090827/SAR_20090827_180830.sar --antenna-half-beamwidth .5
```

Fig. 5.2: SAR data tab.

5.1.3 Region of Interest Selection Tab

STARTAL was created to demonstrate the validity of a SAR data model created at USU [1]. The methods to form imagery based on the data model typically much more processing than even CBP. To be a practical demonstration, it is therefore necessary to reduce the amount of data on which the algorithm must operate. The primary means of reducing processing time is to select smaller regions of the globe for which images should be formed. Once the SAR data overview has been displayed, the user may select regions between the near and far range ground points. In fact, the user may select regions outside those points, but there will typically not be enough SAR data samples to permit image reconstruction. New regions may be created on the globe by holding the "ALT" key while pressing the left mouse button, dragging the mouse, and releasing. Previously created regions are displayed in a grid of images, shown in figure 5.3. Images displaying the regions of interest are used instead of a textual representation strictly due to human difficulty distinguishing points which differ only in the lower-order bits. Selecting different regions displays a Keyhole Markup Language (KML) overlay placing the region on the globe. STARTAL currently only supports four-sided regions for regions of interest due to the simplicity of selecting box-shaped regions and the additional bookkeeping and pixel masking which would be required to form imagery for arbitrary regions.

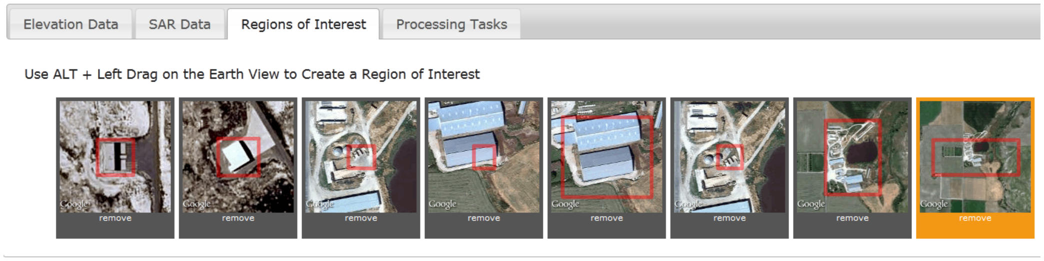


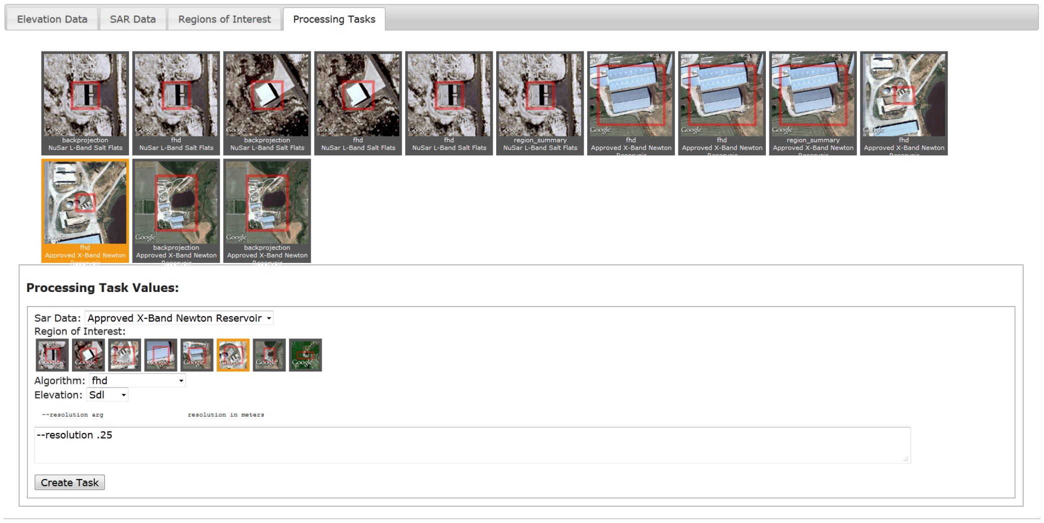
Fig. 5.3: Regions of interest tab.

5.1.4 Processing Task Tab

The Processing Tasks tab (figure 5.4) unites the information contained on the other three tabs. To create and execute an algorithm requires choosing the algorithm, the SAR data, and an elevation data type. (This is necessary here and in the "SAR Data" tab since the latter choice affects only summary information generation.) Previously created data objects will be listed by name in selection boxes, existing regions of interest will be shown by selectable image, and registered algorithms will be displayed in a drop-down. Optional and required parameters for the selected algorithm are displayed along with a text area where those parameters may be supplied.

The data items, regions, and tasks created and displayed in each tab are stored in a database for later retrieval. No item may be deleted which has been used by another item in STARTAL. For example, if the Adjoint algorithm makes use of a particular region of interest, it is programmatically enforced that the region may not be deleted until after the picture is.

Once the processing parameters have been chosen STARTAL creates an instance of the Sarry! executable. When execution terminates it stores the output KML/KMZ file and adds the task to the list of created tasks. Tasks may be selected to display their processing parameters in the "Add Task" form and their output on the globe. One shortcoming of STARTAL is that there is no intermediate status reported on the progress of the executable. Future work should address that to give some insight on how it is processing and make the application seem more responsive.



Processing Task Values:

Sar Data: Approved X-Band Newton Reservoir

Region of Interest:

Algorithm: fhd

Elevation: Sdl

```
--resolution arg                    resolution in meters
```

```
--resolution .25
```

Fig. 5.4: Processing tasks tab.

5.2 STARTAL Programming Interfaces

The primary means of configuring STARTAL's behavior is through customization points. Instances of the customization point must fulfill customization point requirements and be accessible to the executable. In STARTAL, this means the instance must be accessible as a shared library, and the shared library must be stored in a directory expected by STARTAL.

5.2.1 Shared Library Customization Interface

The public interface of a customization point is specified by an abstract C++ class, and instances are created using a factory pattern. Clearly, to create an instance of the customization point, all pure virtual methods must be overridden and all other virtual methods should be overridden or verified to work correctly.

Because the instance must be loaded without knowing the C++ type beforehand, it is also necessary that the class constructors all accept the same values. Because different instances of customization points may require additional parameters, all customization points accept command-line arguments in the traditional argc/argv format. Instances of a customization point should check the arguments for a "help-*abstraction*" parameter. For example, an instance of the Elevation customization point should check for the "–help-elevation" flag. If it is present the instance should throw an exception whose text specifies which flags are accepted by that instance. That is the mechanism whereby STARTAL displays the available flags when the instance is selected in the GUI.

Shared libraries containing one or more customization point instances must register those instances. This is done using the proposed Boost.Extension library. It exports a function which associates a specific instance of a customization point with a descriptive string and the abstract customization point. Attempting to compile the shared library will verify conformance to the customization point abstraction. Multiple instances of a single customization point may be registered in a single method. Likewise, instances of different customization points may be registered in a single method.

Both STARTAL and Sarry! load shared libraries and partition the instances by customization point. STARTAL uses the strings associated with each instance to form the selection boxes in the tabs, while Sarry! instantiates the specified instances to either perform data processing or to determine what additional parameters are required for a given instance.

STARTAL may be customized with respect to the antenna/data types, algorithms, and elevation models. The abstractions for each were carefully chosen to decouple data formats and antenna mounting variations from algorithm application. While this decoupling may result in a certain amount of inflexibility and inefficiency (e.g., processing using 32-bit floating-point numbers may be acceptable for some collected data), it greatly simplifies algorithm prototyping and new format addition.

5.2.2 The Elevation Abstraction

Because many SAR processing algorithms require some notion of the terrain reflecting pulses, a mechanism for determining the ground height at a specific latitude/longitude point is required. Latitude and longitude values should be supplied in degrees. Though many digital elevation models supply elevations relative to the geoid, instances of the elevation customization point should provide elevations in meters relative to the ellipsoid. This is because of the comparative computational simplicity of transforming from one coordinate system to another with the more mathematically simple ellipsoid specification.

5.2.3 SRTM Elevation Instance

The SrtmElevation class models the elevation abstraction by reading the raw ".hgt" files and performing endianness conversions as needed. The class bilinearly interpolates between valid data points (discarding any invalid points) and converts the height above the geoid to the height above the ellipsoid. While SRTM data is freely available on the Internet, STARTAL expects that any needed .hgt SRTM files will have already been downloaded. The path to the directory containing those files may be passed as a parameter to SrtmElevation construction. Converting from geoid-relative to ellipsoid-relative uses the open-source GeographicLib. It allows the use of different geoid models (Egm96, Egm2008, etc.) and requires a file containing precomputed conversions. Because the data is relative to the 1996 gravitational model, egm96 is the default. This, as well as the path to the geoid conversions file, can be modified by supplying additional flags on instance construction.

5.2.4 The Antenna Abstraction

STARTAL uses a single class to represent the signal parameters and the data format. This is intended to reduce the number of shared objects which must be registered and avoid needing to specify the constructor arguments of the signal and data loading classes. The data loading class (PerPulseInfoLoader) and the signal class (Chirp) are also intended to be used polymorphically.

The PerPulseInfoLoader must be able to specify the aircraft location in LLA for each pulse, where latitude and longitude are in degrees and altitude is in meters above the ellipsoid. It must also be able to transform arbitrary points in ECEF to points in the antenna coordinate system. The loader must also be able to retrieve the raw and/or matched-filtered data collected by a given pulse. If the data is already matched-filtered then only matched-filtered data requests should be successful. This introduces a slight dependency between the algorithms and the data used, but most algorithms require that the data be matched filtered anyway. Finally, the data loader must define a monotonically increasing ordering on the pulses (usually the order of collection) and allow random access of the pulse information. Models of the data loader may provide employ caching to improve speed for specific access patterns.

The Chirp object represents the signal for an LFM pulse. STARTAL is designed to support non-continuous-wave LFM signals and must furthermore be for a side-looking antenna. The Chirp must specify signal parameters like bandwidth, pulse duration, and the pulse repetition interval. It also may supply physical antenna parameters such as beamwidth, dimensions, etc.

5.2.5 NuSar Data Instance

The NSLeftProcessedAntenna class is an instance of the Antenna customization point which uses processed data from from the Naval Research Laboratory (NRL)/Space Dynamics Laboratory (SDL) NuSar sensor. It has a NuSar-specific Chirp object and a pulse information loader which accepts NuSar data which has been processed into raw files of little-endian double-precision values. Although this antenna instance assumes a fixed antenna mounting on the left-hand side of the aircraft, it would be possible to specify mount parameters as command-line arguments, or to have an antenna type in which the antenna is gimbaled. Processed NuSar data is stored in a numerically indexed set of files. To speed reading information for a single pulse, when pulse information for a file is requested for the first time, the remaining file information is cached into memory. This prevents a single instance of NSPerPulseInfoLoader from being shared between threads, but greatly reduces the amount of time parsing pulse information files.

5.2.6 The Algorithm Abstraction

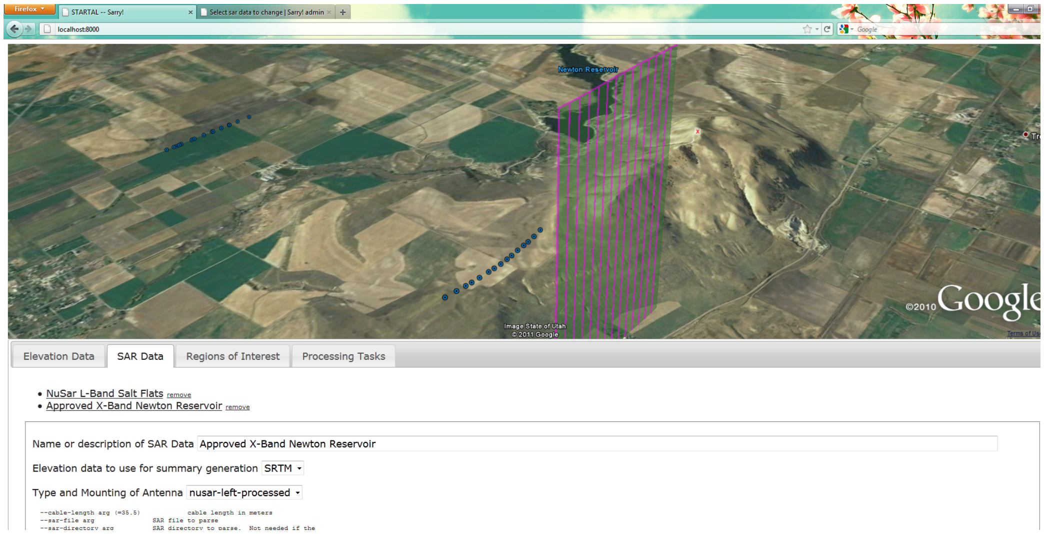
Models of the algorithm abstraction must be constructible with the antenna object and an elevation object, in addition to the argc and argv parameters that all customization points in STARTAL accept. Its output must be directly loadable into Google Earth. For non-imagery data like GMTI, the KML format is usually most appropriate; for imagery data a KMZ file is usually most appropriate. Many algorithms require a target point or region of interest to form imagery, but others do not. In STARTAL, region of interest points do not form part of the class interface, but enough algorithm instances require that those points be passed as command-line arguments that regions of interest have a tabbed display, are stored in the database, and are required when dispatching algorithms. Algorithm instances which do not require a region of interest may simply ignore those command line parameters. In the future, parsing of the optional and required arguments may allow users of STARTAL to only specify a region of interest when needed.

5.2.7 Data Summary Algorithm

The DataSummary algorithm is the most basic instance of the Algorithm abstraction. It loads one of every thousand pulses, finds the aircraft location, and determines the near and far ground range for each loaded pulse. A KML file showing the path of the aircraft and the region for which the sensor has collected data is output. Unlike the following algorithms, DataSummary is not compiled into a shared library. It is explicitly loaded into the Sarry! core because it is a required algorithm for STARTAL's operation and so that it does not appear in the list of potential algorithms in STARTAL's algorithm tab. The output of DataSummary is shown in figure 5.5.

5.2.8 Region Summary Algorithm

The linear algebra-based image formation formulation described in section 4 requires knowledge of the region of interest, region of interest closure, and the data of interest. Those may be examined using the region summary algorithm. Selecting the region summary will generate, for some supplied resolution, the mesh of points consisting the region of interest (shown in yellow), the region of interest closure (both the yellow and blue points), and the data of interest. An example of output from the region summary algorithm is shown in figure 5.6. The examples of the region of interest and the data of interest described in sections 4.1.1 and 4.1.2 were also formed with the region summary algorithm.



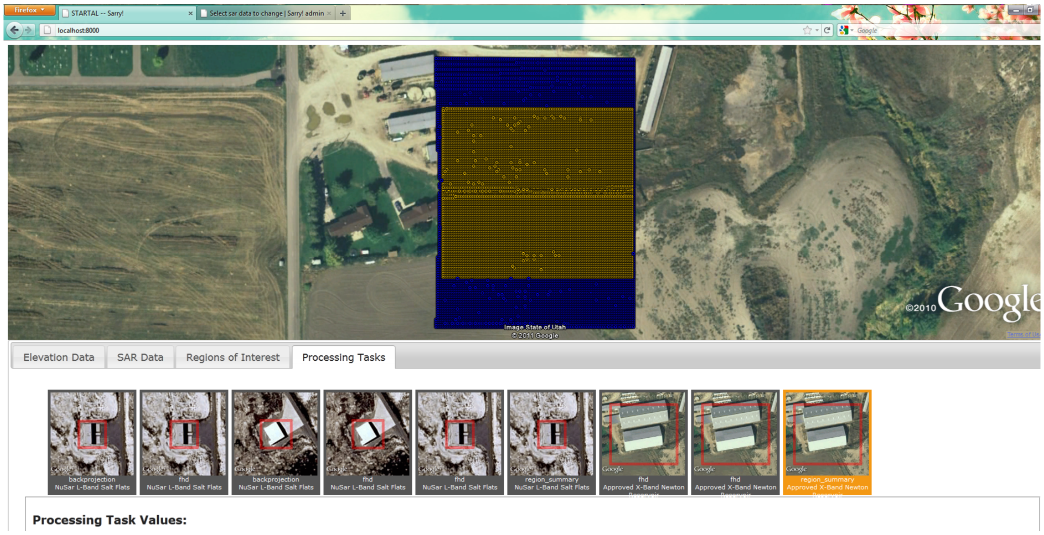
Name or description of SAR Data Approved X-Band Newton Reservoir

Elevation data to use for summary generation SRTM

Type and Mounting of Antenna nusar-left-processed

```
  --cable-length arg (=35.5)           cable length in meters
  --sar-file arg              SAR file to parse
  --sar-directory arg         SAR directory to parse.  Not needed if the
```

Fig. 5.5: Data summary for X-band SAR data.


STARTAL -- Sarry!
Select sar data to change | Sarry! admin
localhost:8000
©2010 Google
Image State of Utah
© 2011 Google
Elevation Data
SAR Data
Regions of Interest
Processing Tasks
backprojection NuSar L-Band Salt Flats
fhd NuSar L-Band Salt Flats
backprojection NuSar L-Band Salt Flats
fhd NuSar L-Band Salt Flats
fhd NuSar L-Band Salt Flats
region_summary NuSar L-Band Salt Flats
fhd Approved X-Band Newton Reservoir
fhd Approved X-Band Newton Reservoir
region_summary Approved X-Band Newton Reservoir

Processing Task Values:

Fig. 5.6: Region summary for X-band SAR data.

5.2.9 Adjoint and Maximum Likelihood Algorithms

STARTAL supplies two linear-algebra algorithms: Adjoint and Maximum Likelihood. These correspond to the adjoint and minimum error projections mentioned in section 4.2 and section 4.3. These algorithms both require a four-sided region of interest so that samples which do not include returns from the region may be discarded. They also require an output image pixel resolution. Because of the computational complexity of image formation via these methods (both of which exceed back-projection) using the highest possible resolution for the produced images may extend processing times excessively. Both algorithms determine which pulses illuminate the region of interest and determine the corresponding data of interest. They also both determine the same region of interest closure.

A MatchedMatrix class is responsible for lazily computing the F matrix. The equation for F is given in section 4.1.3. It uses internal caching to speed column-major traversal, which is most beneficial to the Adjoint algorithm. The caching optimization takes advantage of the fact that a significant number of computations in the determination of a single element in a column are dependent only on the pulse location. Thus, while the sample index changes with each column element, the pulse location remains constant for many rows. The actual number of rows depends on the region width and the range resolution. Other computations may be stored which depend only on the pulse location and the ground pixel location. In fact, the only elements from section 4.1.3 which must be computed for each element access are η and those equations which depend on it. Care must be taken since unsynchronized caching is dangerous when the object is shared between multiple threads. One option is to ensure synchronous modification of the cache; however, this would be very expensive and could lead to cache-thrashing if multiple threads are looking at different portions of the matrix. Since the class is inexpensive to copy, each thread using the matrix should instead make a copy of it prior to accessing its elements. This avoids needing explicit synchronization at the expense of memory. While column-major traversal is the best-suited for the Adjoint algorithm, row-major could be convenient for Maximum Likelihood. This is not currently supported; the user desiring the matrix in row-major order is encouraged to form a matrix in column-major order and transpose it.

5.2.10 Adjoint Algorithm

For the Adjoint algorithm, the elements of g which correspond to pixels in the region of interest are partitioned and processed by separate threads. Because computing the adjoint is embarrassingly parallel, near-linear speedup and efficient processor utilization are common. When an element of g is completed, a message is displayed to the screen indicating how many elements of g are complete and how many remain. As noted earlier, when run from STARTAL the output is not currently relayed to the user. It is only visible if the stand-alone Sarry! executable is run. Examples of Adjoint-generated imagery are shown in figures 5.7, 5.8, 5.9, and 5.10.

5.2.11 Maximum Likelihood Algorithm

The Maximum Likelihood algorithm has also been implemented. Unlike the Adjoint algorithm, Maximum Likelihood requires that F (or at least $F^H F$) be present in memory. It is not uncommon for F to have over 100,000 rows. Assuming that each element of the matrix is 128 bytes, it is easy to see that even a modestly sized region of interest will consume many gigabytes of memory. Considering that more than twice that memory must also be present to create $F^H F$, it is possible to consume all available memory on a system and still lack. A combination of choosing smaller regions and forming $F^H F$ differently may alleviate the memory problem; however, the complexity of the g vector estimation requires several $O(n^3)$ operations, which, for large n, could consume all available time. Due to these issues, the Maximum Likelihood implementation has not yet generated an image. Creating a Maximum Likelihood image is a top priority for future work with this framework.

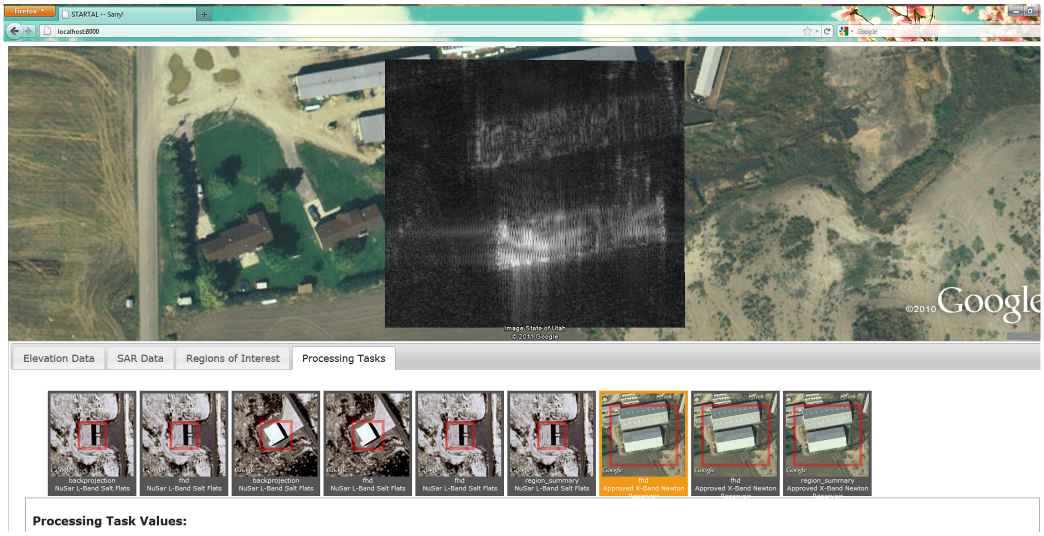

Firefox
STARTAL -- Sarry!
localhost:8000
Google
Image State of Utah
© 2011 Google
©2010 Google
Elevation Data
SAR Data
Regions of Interest
Processing Tasks
backprojection NuSar L-Band Salt Flats
fhd NuSar L-Band Salt Flats
backprojection NuSar L-Band Salt Flats
fhd NuSar L-Band Salt Flats
fhd NuSar L-Band Salt Flats
region_summary NuSar L-Band Salt Flats
fhd Approved X-Band Newton Reservoir
fhd Approved X-Band Newton Reservoir
region_summary Approved X-Band Newton Reservoir
Processing Task Values:

Fig. 5.7: One-quarter meter resolution Adjoint-created image of barns using X-band SAR data.

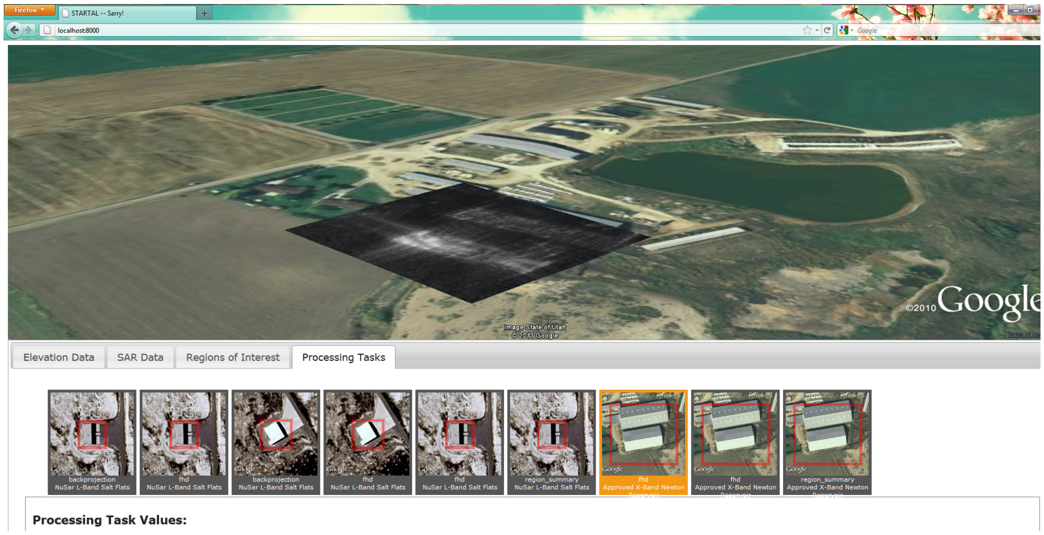

Firefox
STARTAL -- Sarry!
localhost:8000
Google
Image State of Utah
© 2011 Google
©2010 Google
Elevation Data
SAR Data
Regions of Interest
Processing Tasks
backprojection NuSar L-Band Salt Flats
fhd NuSar L-Band Salt Flats
backprojection NuSar L-Band Salt Flats
fhd NuSar L-Band Salt Flats
fhd NuSar L-Band Salt Flats
region_summary NuSar L-Band Salt Flats
fhd Approved X-Band Newton Reservoir
fhd Approved X-Band Newton Reservoir
region_summary Approved X-Band Newton Reservoir
Processing Task Values:

Fig. 5.8: Another one-quarter meter resolution Adjoint-created image of barns using X-band SAR data.

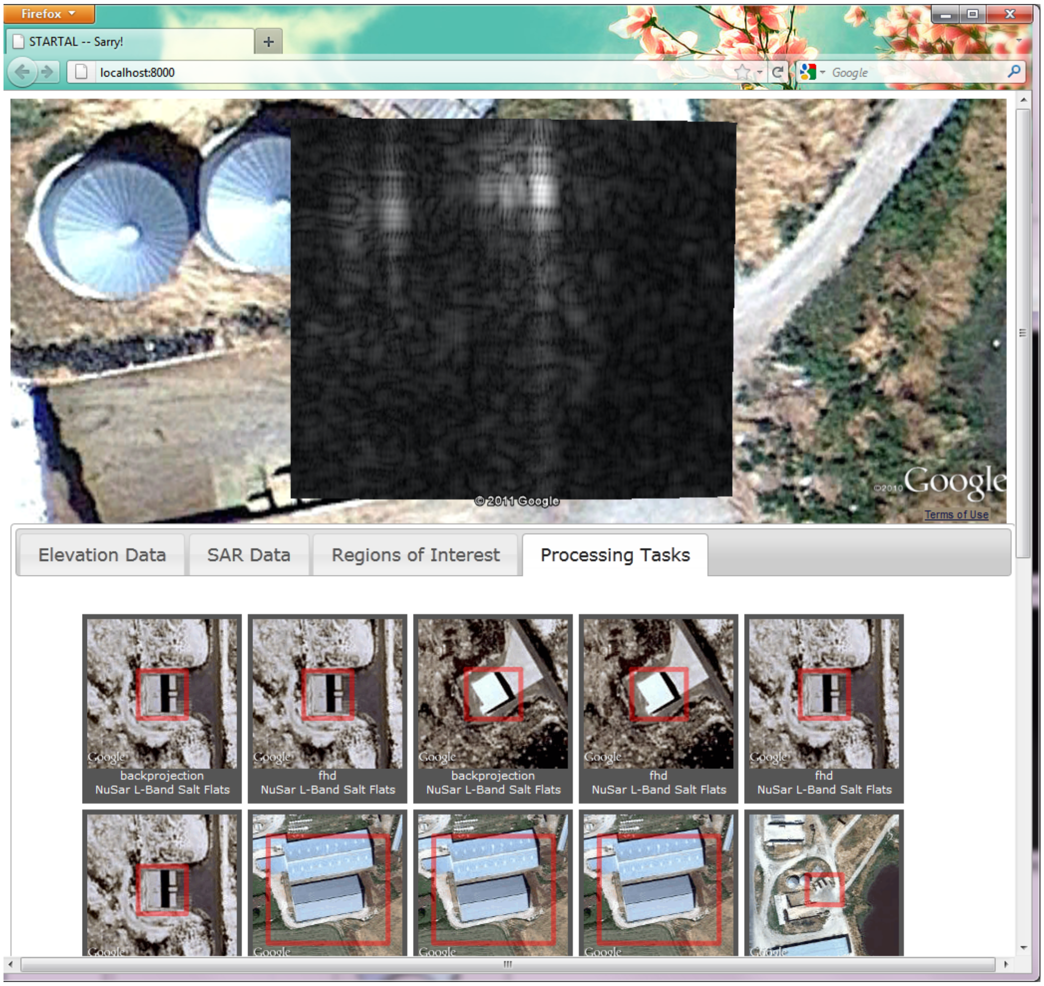


Fig. 5.9: One-half meter resolution Adjoint-created image of silos using X-band SAR data.

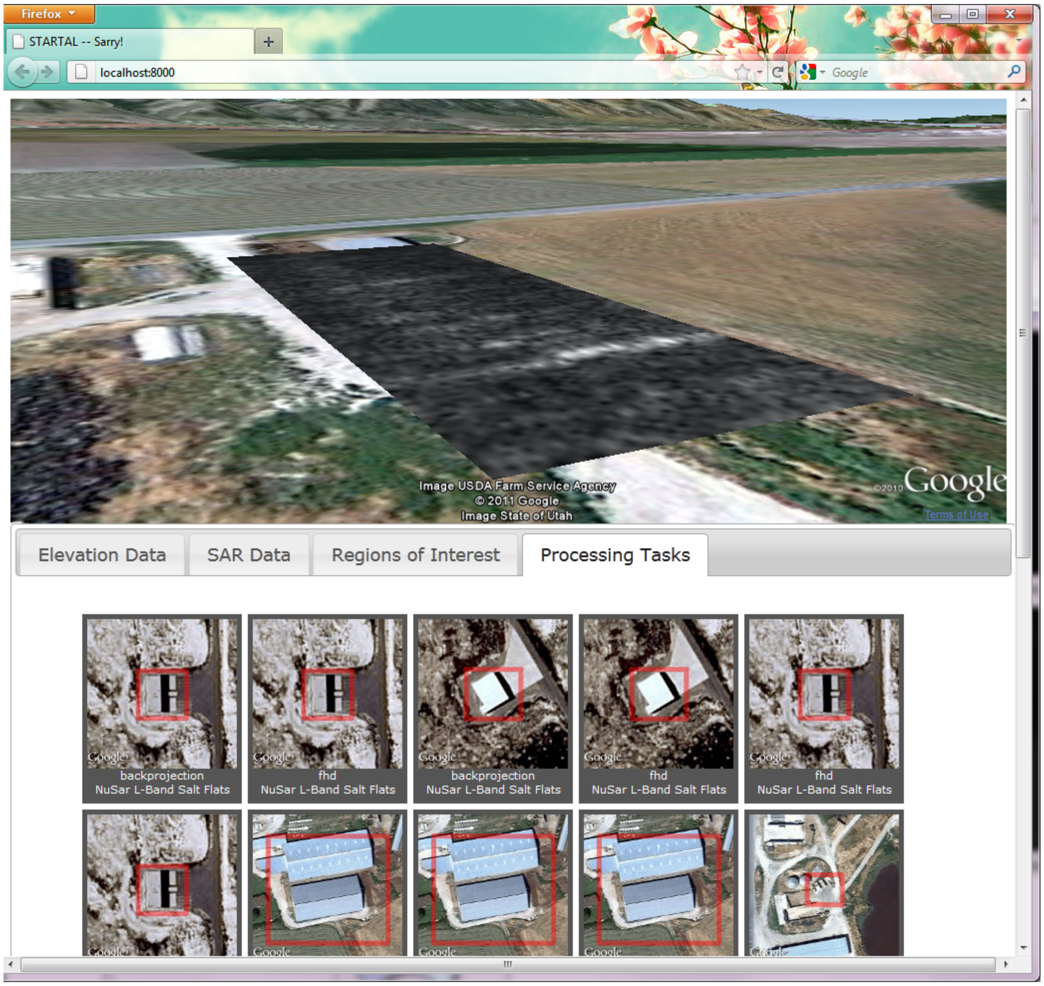


Fig. 5.10: One-half meter resolution Adjoint-created image of shed using X-band SAR data.

Chapter 6

Conclusion

STARTAL is an algorithm prototyping platform which validates the SAR model developed at USU [1]. It has processed SAR data collected from an airborne platform and produced images comparable to those produced by CBP. STARTAL provides the technology necessary to implement data processing algorithms, including elevation data loading, signal processing libraries, and usable abstractions for the data and algorithms.

6.1 STARTAL Goals

STARTAL's primary goal is to facilitate algorithm prototyping and validation. This is achieved by requiring algorithms to conform to a very generic abstraction, and by abstracting how data used in algorithms is stored and loaded. This allows conforming data to be immediately usable with all conforming algorithms, and allows new algorithms to be applicable to all conforming data.

Another STARTAL goal is to simplify the creation, display, and comparison of processing tasks. The user interface features a globe on which dataset information, regions of interest, and processed images are overlaid. Algorithm comparison is facilitated since processing task information is easily displayed and recombined to form new tasks. New instances of algorithms, SAR data, or elevation data may be added to the system as shared without needing to modify the core STARTAL system. This allows algorithms to be easily shared among different STARTAL systems, further easing comparison.

The final goals guiding STARTAL's design and implementation are that it be portable across platforms and that it be efficient. All programming languages and libraries in STARTAL are natively supported on Microsoft Windows and Linux. Programming languages used are chosen based the quality of libraries and frameworks available for STARTAL's tasks. In those places where efficiency is most important, algorithms with good complexity are chosen, and a language with good performance characteristics is used.

6.2 Future Directions

While STARTAL achieves the goals defined at project outset, there are still many interesting directions which future work could pursue. An obvious area for growth is in algorithm addition. The ability to compare algorithm output is more useful with a wealth of algorithms to compare. It may also be interesting to add support for heterogeneous computing platforms (such as Graphics Processing Unit (GPU)s) and computing clusters to reduce processing time.

More fundamentally, because STARTAL was designed to work with geographically referenced data, and because typical publicly available SAR data excludes geographical information, it would be beneficial to further examine the SAR data abstraction. It may be possible to replace the current abstraction with a more general abstraction, or more likely, include both abstractions. This would allow more of the publicly available data to be processed and displayed in STARTAL without removing the ability to process data using current algorithms that require more information.

A usable implementation of the maximum likelihood algorithm is another area where more study is warranted. The major obstacles to creating a working algorithm are related. Forming the F matrix requires too much memory, and multiplying $F^H F$ takes too much time due to the matrix's large size. It is possible to form $F^H F$ directly, but the processing time to naively form the matrix is prohibitive. The structure of $F^H F$ for non-range compressed data can yield significant improvements in the speed of element determination; however, it is unknown if similar improvements will result from forming $F^H F$ for range-compressed data.

References

[1] R. D. West, *Model-Based Stripmap Synthetic Aperture Radar Processing.* Ph.D. dissertation, Utah State University, 2011.

[2] I. G. Cumming and F. H. Wong, *Digital Processing of Synthetic Aperture Radar Data.* Norwood, MA: Artech House, 2005.

[3] I. G. Cumming and F. H. Wong, *Digital Processing of Synthetic Aperture Radar Data*, ch. 4. Norwood, MA: Artech House, 2005.

[4] I. G. Cumming and F. H. Wong, *Digital Processing of Synthetic Aperture Radar Data*, ch. 3. Norwood, MA: Artech House, 2005.

[5] J. C. McGlone, Ed., *Manual of Photogrammetry*, ch. 3, pp. 182–184. Bethesda, Maryland: The American Society for Photogrammetry and Remote Sensing, 2004.

[6] J. C. McGlone, Ed., *Manual of Photogrammetry*, ch. 3, pp. 192–194. Bethesda, Maryland: The American Society for Photogrammetry and Remote Sensing, 2004.

[7] J. C. McGlone, Ed., *Manual of Photogrammetry*, ch. 3, pp. 183–188. Bethesda, Maryland: The American Society for Photogrammetry and Remote Sensing, 2004.

[8] J. C. McGlone, Ed., *Manual of Photogrammetry*, ch. 3, p. 188. Bethesda, Maryland: The American Society for Photogrammetry and Remote Sensing, 2004.

[9] J. Gunther, R. West, N. Crookston, and T. Moon, "Maximum likelihood synthetic aperture radar image formation for highly nonlinear flight tracks," in *Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE), 2011 IEEE*, pp. 449 –454, Jan. 2011.

[10] R. West, T. Moon, and J. Gunther, "A novel block fast array rls algorithm applied to linear flight strip-map sar imaging," in *Signals, Systems and Computers (ASILOMAR), 2010 Conference Record of the Forty Fourth Asilomar Conference on*, pp. 979 –983, Nov. 2010.

[11] R. West, J. Gunther, and T. Moon, "Forming regularized maximum likelihood strip-map synthetic aperture radar images using the block rls algorithm," in *Digital Signal Processing Workshop and IEEE Signal Processing Education Workshop (DSP/SPE), 2011 IEEE*, pp. 455 –460, Jan. 2011.