RADARSAT-2 Applications Technical Note RN-TN-53-0076 Issue 1/3: April 08, 2015

TITLE: Geolocation of RADARSAT-2 Georeferenced Products			
PREPARED BY: Ron Caves Dan Williams	**Date:** 05-03-2015	**APPROVED BY:** Gordon Rigby	**Date:** 08-04-2015

SUMMARY DESCRIPTION:

This note gives information and status based on results available at the time of writing. This information is subject to change. The information described here is part of an ongoing improvement campaign in RADARAT-2 mission operations and users should expect updates to this information in the future.

SCOPE:

- Description of methods for geolocating RADARSAT-2 georeferenced SAR image products using the orbit data, tie-points and rational function coefficients provided in the information file (product.xml) provided with each product.

Changes since version 1/0

Section 1.3.1 clarified definition of the image coordinate system

Sections 2 and 3 are updated for the case of longer products having multiple time-stamped sets of ground range to slant range polynomial coefficients.

Section 2.2 step 7, the iteration method for calculating the ground range is changed to Newton-Raphson.

Section 2.3 step 9, tan(θ) corrected to tan(ψ)

Changes since version 1/1

Section 2.3 added algorithm clarifications

Changes since version 1/2

Section 4.2 clarified application of rational function across longitude ±180º

13800 Commerce Parkway
Richmond, B.C., Canada
V6V 2J3
Telephone (604) 278-3411
Fax (604) 278-2117

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

ACRONYMS AND ABBREVIATIONS

COTS	Commercial Off-The-Shelf
DEM	Digital Elevation Model
ECEF	Earth Centered, Earth Fixed
ECR	Earth Centered Rotating
EGM	Earth Gravitational Model
GeoTIFF	Geographic Tagged Image File Format
GPS	Global Positioning System
MDA	MacDonald, Dettwiler and Associates Ltd.
MSL	Mean Sea Level
NGA	National Geospatial-Intelligence Agency
PIF	Product Information File (product.xml)
RPM	Rigorous Projection Model
SAR	Synthetic Aperture Radar
ScanSAR	Scanning SAR (sensor)
SGF	SAR Georeferenced Fine
SGX	SAR Georeferenced eXtra-fine resolution
SLC	Single-Look Complex (product)
SRTM	Shuttle Radar Topography Mission
UTM	Universal Transverse Mercator
WGS84	World Geodetic System (1984)
XML	eXtended Markup Language

1 INTRODUCTION

1.1 Purpose

This technote explains how users of georeferenced Synthetic Aperture Radar (SAR) image products (Single-Look Complex (SLC), SAR Georeferenced eXtra-fine resolution (SGX), SAR Georeferenced Fine (SGF)) from the RADARSAT-2 mission can use product metadata to geolocate features in images and to geocode entire images. Specifically, it answers the following two interrelated questions:

- Forward geolocation problem: given the image coordinates (line and pixel) of a feature identified in an image what are the geographic coordinates (latitude, longitude and height) of the corresponding feature on the ground?
- Reverse geolocation problem: given the geographic coordinates of a feature within the imaged scene what are the image coordinates of the corresponding feature in the image? This represents the geocoding problem, populating pixels on a map projection with values from the image.

1.2 Scope

The Product Information File (PIF), product.xml, of each RADARSAT-2 georeferenced image product includes three forms of information that can be used in geolocation: orbit data, tie-points and rational functions. This technote explains how to use all three of these information sources for geolocation.

The following approaches are described:

- Forward and reverse geolocation using orbit data (Section 2)
- Forward geolocation using tie points (Section 3)
- Reverse geolocation using rational functions (Section 4)

Geolocation using orbit data is based on a Rigorous Projection Model (RPM) that maps how image coordinates project to ground coordinates. The RPM is based on the intersection of the range sphere centered on the satellite position, the zero Doppler plane passing through the satellite position and an Earth model.

The tie points are derived using the RPM and provide a regular grid of paired image and ground coordinates between which the geolocation of intermediate points can be interpolated.

The rational functions are a set of polynomials that provide an accurate approximation of the reverse RPM. They are used to calculate the image coordinates corresponding to a given geographic coordinate.

All three approaches have been successfully tested on RADARSAT-2 stripmap and ScanSAR products from a range of geographic locations. The approaches have not yet been tested on the rare case of images acquired over the poles where additional care in interpolating coordinates may be required.

The tie points referred to in this document are those included in the PIF. Tie points are also provided in the GeoTIFF tags of each GeoTIFF imagery file, but GeoTIFF tags are not needed for the purpose of geolocating or geocoding georeferenced products, and therefore their use is not covered in this document.

1.2.1 Audience

This technote is intended for developers of software tools for geolocating and geocoding RADARSAT-2 products.

1.2.2 Recommended Approach

The tie point approach is recommended for forward geolocation of RADARSAT-2 products as it does not require any iterative calculation and it is the method used by MDA for checking geolocation accuracy.

The rational function approach is recommended for reverse geolocation and geocoding of RADARSAT-2 products as this is the simplest method to implement and use.

Image geolocation and geocoding using rational functions is supported by various Commercial Off-The-Shelf (COTS) satellite image analysis software. Sometimes this requires reformatting of the rational function coefficients provided in the product.xml file into a format that the COTS can ingest.

1.3 RADARSAT-2 Product Format and Conventions

The RADARSAT-2 image format is documented in [1]. All of the product parameters required for geolocation are listed in the PIF. The conventions used to define image coordinates, geographic coordinates, elevation and satellite orbits are explained below.

1.3.1 Image Coordinate System

RADARSAT-2 geo-referenced products are produced in the "Zero-Doppler" orientation: i.e., with each row of pixels representing points along a line perpendicular to the sub-satellite track.

The origin of the image coordinate system (0,0) is the centre of the pixel at the top left corner of the image. Pixel (column) values (*p*) range from 0 to *m*-1 and increase from left to right. Line (row) values (*l*) range from 0 to *n*-1 and increase from top to bottom, where the image has dimension *m* pixels per line (columns) by *n* lines (rows). Image coordinate (p,l) refers to the centre of pixel *p* on line *l*, counting from zero in both dimensions. Image coordinates can take floating point values, e.g. when the peak of a point target is interpolated within a pixel.

RADARSAT-2 images are oriented such that north is nominally at the top of the image and east is nominally on the right of the image. Depending on the acquisition geometry (ascending/descending pass, left/right-looking) near range is either at the left or right of the image and the earlier azimuth time is at the top or bottom of the image. This is illustrated in Figure 1-1.

The pixelTimeOrdering tag in the PIF indicates whether range increases or decreases with pixel number. Likewise the lineTimeOrdering tag indicates whether azimuth time increases or decreases with line number.

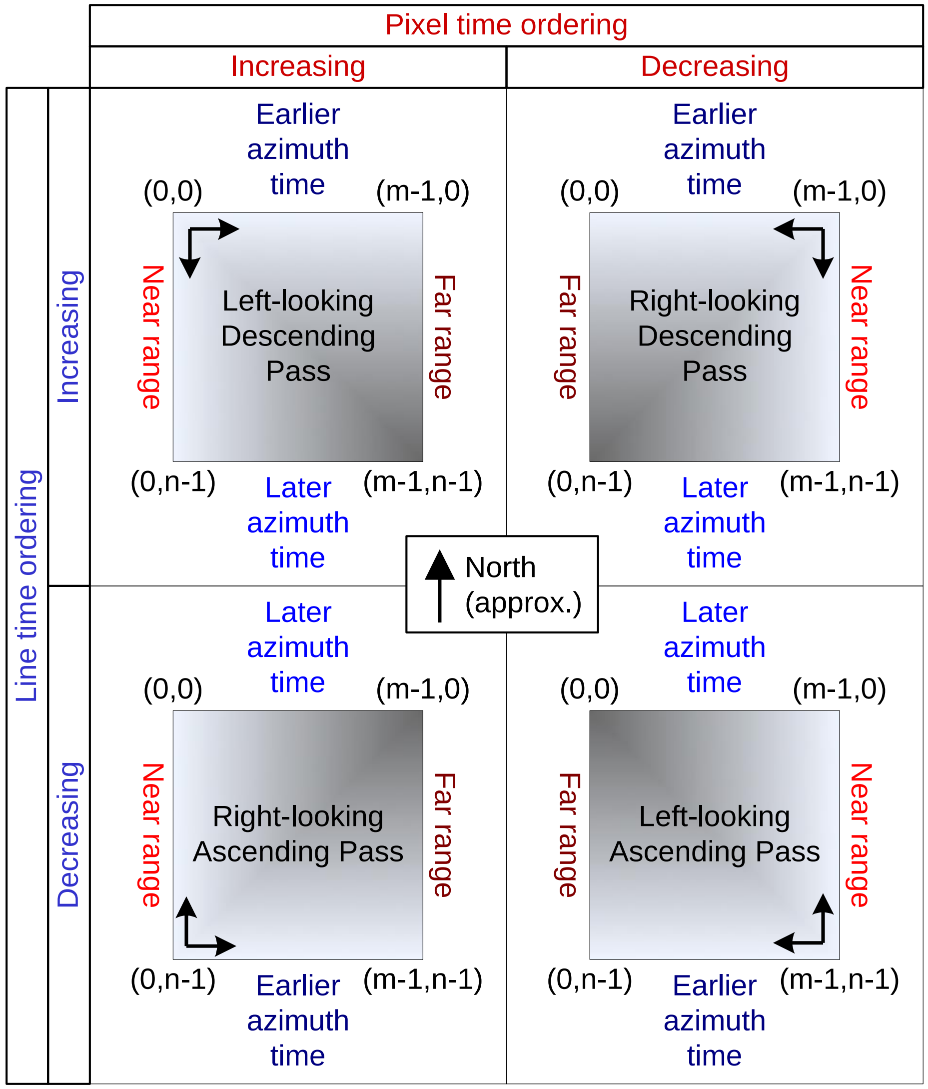


Figure 1-1 Image coordinate system for different pass and look directions.

1.3.2 Geographic Coordinate System

Geographic coordinates are defined in terms of geodetic latitude and longitude. The reference ellipsoid is WGS84.

Well defined algorithms are available for transforming geographic coordinates to and from other map projections such as Universal Transverse Mercator (UTM). These transformations are outside the scope of this technote.

1.3.3 Elevation Data

Elevation data for geocoding RADARSAT-2 data is defined perpendicular and relative to the WGS84 reference ellipsoid. The ellipsoid (or geodetic) height *h*, is not the same as the orthometric height *H*, measured relative to mean sea level (MSL) as provided by most maps, Digital Elevation Models (DEMs) and Global Positioning System (GPS) receivers. Corrections *N*, for transforming between orthometric height (based on the EGM96 geoid model) and WGS84 ellipsoid height are available online from National Geospatial-Intelligence Agency (NGA) (http://earth-info.nga.mil/GandG/wgs84/gravitymod/egm96/intpt.html).

$$h = H + N$$

The correction N represents the height of the EGM96 geoid above the WGS84 ellipsoid. It varies gradually over the Earth ranging between -106 m and +85 m. When starting from orthometric height, this correction to geodetic height must be applied to obtain accurate geolocation and geocoding results.

Digital Elevation Models (DEMs) for use in geocoding are available from multiple sources. Shuttle Radar Topography Mission (SRTM) DEM data with 90 m posting spacing are available between 60° S and 60° N from http://srtm.csi.cgiar.org/. Note, the SRTM data provide orthometric heights that need to be corrected to ellipsoid heights.

1.3.4 Satellite Position and Velocity

Satellite position and velocity are provided in Earth Centered Rotating (ECR) coordinates, which is also known as Earth Centered, Earth Fixed (ECEF) coordinates.

1.3.5 Processed Elevation

Each georeferenced product is processed assuming the entire imaged area is at a fixed elevation above the reference ellipsoid. This processed elevation is tagged as GeodeticTerrainHeight in the PIF and is calculated from a coarse DEM. The difference between the actual elevation of a feature and the processed elevation needs to be explicitly corrected by some of the geolocation approaches.

1.4 Slant Range Offset Correction

The RADARSAT-2 processor applies a fixed offset to slant range measurements to correct timing delays due to sensor electronics and atmospheric propagation. This slant range offset has been updated as geolocation knowledge has improved. The most recent update was implemented on Dec 18, 2009. The geolocation accuracy of products processed before Dec 18, 2009 can be improved by applying a correction to the geodetic heights used in geolocation. This height correction is given by

$$\Delta h = \frac{\Delta R}{\cos\theta}$$

where θ is the mean incidence angle for the image and ΔR is the slant range offset (Table 1-1). If H is the orthometric height derived from a map or DEM, the slant range corrected geodetic height h used in RADARSAT-2 geolocation is given by.

$$h = H + N + \Delta h$$

Table 1-1 Slant Range Offset Correction by Processing Date

Processing date	Slant Range Offset Correction
Up to September 11, 2008	22 m
September 12, 2008 to December 17, 2009	17 m
Dec 18, 2009 to present	0 m

Note, all data acquired before July 25, 2008 is subject to much larger geolocation uncertainties due to the initial setup of the orbit determination software on board the satellite.

2 GEOLOCATION USING ORBIT DATA

This section describes how to use the RADARSAT-2 orbit data to calculate:

- Image coordinates corresponding to given geographic coordinates (reverse geolocation algorithm).
- Geographic coordinates corresponding to given image coordinates (forward geolocation algorithm).

The reverse algorithm is simpler and is described first.

2.1 Input Parameters

The following parameters need to be extracted from the PIF.

Table 2-1 PIF parameters required for geolocation using orbit data.

Parameter	XML tag	Description	Symbol
Pixel spacing (m)	sampledPixelSpacing	Spacing between pixels in the slant/ground range direction	δ_{pixel}
Number of pixels per line	numberOfSamplesPerLine	Number of pixels (samples) per line	m
Number of lines	numberOfLines	Number of lines in the image	n
Pixel time ordering	pixelTimeOrdering	Range increases or decreases with pixel number	Increasing or decreasing
Semi Major Axis (m)	semiMajorAxis	Semi major axis length of the reference ellipsoid used in processing (e.g. WGS 84)	a
Semi Minor Axis (m)	semiMinorAxis	Semi minor axis length of the reference ellipsoid used in processing (e.g. WGS84)	b
Slant range at near edge (m)	slantRangeNearEdge	Slant range at the near edge of the image	R_{near}
Ground range origin (m)	groundRangeOrigin	Ground range reference position	D_{origin}
Ground to slant range coefficients	groundToSlantRangeCoefficients	5th order Polynomial coefficients for transforming ground range to slant range. This transformation changes along-track. Longer products (such as ScanSAR) have multiple sets of coefficients each with a different azimuth time-stamp.	$s_k(t_i): k = 0, \ldots, 5; i = 1, \ldots, \mu$

Parameter	XML tag	Description	Symbol
State Vectors	stateVector timestamp xPosition yPosition zPosition xVelocity yVelocity zVelocity	Time stamped satellite position (m) and velocity (m/s) in ECR coordinates (state vectors from ground processed definitive orbits are also available from MDA Geospatial Services)	$\mathbf{p}(t_i) = (p_{x_i} \quad p_{y_i} \quad p_{z_i})$ $\mathbf{v}(t_i) = (v_{x_i} \quad v_{y_i} \quad v_{z_i})$
Zero Doppler Time First Line	zeroDopplerTimeFirstLine	Zero Doppler date/time of first line of image (*l* = 0). If north-south flipping has occurred, this value refers to the first line after flipping.	t_{first}
Zero Doppler Time Last Line	zeroDopplerTimeLastLine	Zero Doppler date/time of last line of image (*l* = *n*-1). If north-south flipping has occurred, this value refers to the last line after flipping.	t_{last}
Antenna Pointing	antennaPointing	Antenna pointing direction (left or right)	left or right

2.2 Reverse Geolocation Algorithm Using Orbit Data

This section describes how to convert geographic coordinates (latitude φ, longitude λ, height h) into image coordinates (pixel p, line l).

Step 1, fit the orbit state vectors to an orbit propagation model that provides position $\mathbf{p}(t)$ and velocity $\mathbf{v}(t)$ vectors for any time within the azimuth time limits of the product. A simple and reasonably accurate orbit propagation model is a cubic spline.

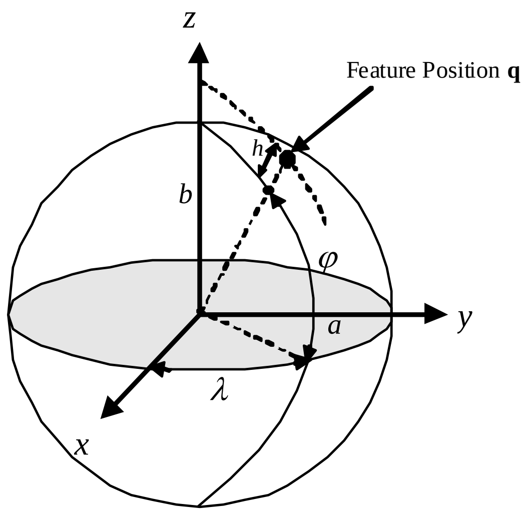
Step 2, convert the geographic coordinates (φ, λ, h) of the feature to Earth Centre Rotating coordinates

$$x = \left(\frac{a^2}{\sqrt{a^2 \cos^2 \varphi + b^2 \sin^2 \varphi}} + h \right) \cos\varphi \cos\lambda$$

$$y = \left(\frac{a^2}{\sqrt{a^2 \cos^2 \varphi + b^2 \sin^2 \varphi}} + h \right) \cos\varphi \sin\lambda$$

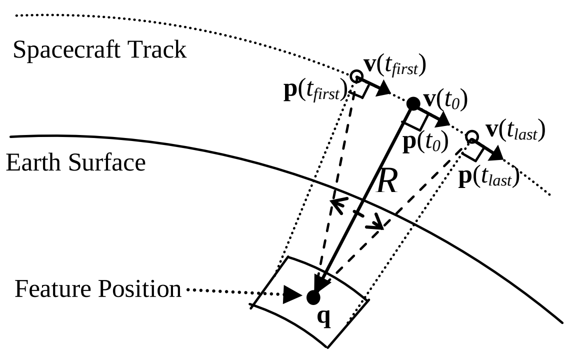
$$z = \left(\frac{b^2}{\sqrt{a^2 \cos^2 \varphi + b^2 \sin^2 \varphi}} + h \right) \sin\varphi$$

$$\mathbf{q} = (x \quad y \quad z)^t$$



Step 3, find the zero Doppler azimuth time of the feature t_0, which is the time when the slant range from sensor to feature is perpendicular to the sensor velocity. This is done by iterating the azimuth time until the dot product between the sensor velocity vector and the slant range vector from sensor to target is zero.

$$\mathbf{v}(t_0)\cdot(\mathbf{p}(t_0)-\mathbf{q})=0$$



Step 4, convert the zero Doppler time into image line number

$$l=\frac{t_0-t_{first}}{t_{last}-t_{first}}(n-1)$$

Step 5, calculate the slant range at zero Doppler

$$R=|\mathbf{p}(t_0)-\mathbf{q}|$$

Step 6, for products with more than one time-stamped set of ground to slant range coefficients, linearly interpolate between these to determine the coefficients for the zero Doppler time. Extrapolation may be required at the start and end of an image.

$$\text{For } k = 0, \ldots 5$$
$$s_k = \frac{t_{i+1} - t_0}{t_{i+1} - t_i} s_k(t_i) + \frac{t_0 - t_i}{t_{i+1} - t_i} s_k(t_{i+1})$$
where
$$i = \begin{cases} 1 & \text{if} \quad t_0 < t_1 \\ i & \text{if} \quad t_i \le t_0 \le t_{i+1} \\ \mu - 1 & \text{if} \quad t_\mu < t_0 \end{cases}$$

This step is skipped for SLC products or when the product contains only a single set of ground to slant range coefficients.

Step 7, convert the slant range into the distance D, from the near-range edge of the scene. This is done by applying Newton-Raphson iteration to the ground range to slant range polynomial (for SLC products, no iteration is required since $s_1 = 1, s_2 = s_3 = s_4 = s_5 = 0$ and $D = R - s_0$).

$$D = \begin{cases} 0 & \text{Initially} \\ \dfrac{R - s_0 + s_2 D^2 + 2 s_3 D^3 + 3 s_4 D^4 + 4 s_5 D^5}{s_1 + 2 s_2 D + 3 s_3 D^2 + 4 s_4 D^3 + 5 s_5 D^4} & \text{Iterate until convergence} \end{cases}$$

Convergence is attained when the change in D is significantly less than 1 metre.

Step 8, scale the distance from the near-range edge of the scene into pixel number

$$p = \begin{cases} \dfrac{D + D_{origin}}{\delta_{pixel}} & \text{if pixel time ordering is increasing} \\ m - 1 - \dfrac{D + D_{origin}}{\delta_{pixel}} & \text{if pixel time ordering is decreasing} \end{cases}$$

Note: Although the term ‘ground range’ is specific to ground range products, as opposed to slant range products, the formulas in steps 6 to 8 apply to both ground range and slant range products.

2.3 Forward Geolocation Algorithm Using Orbit Data

This section describes how to convert image coordinates (l, p) into geographic coordinates at a given height (φ, λ, h).

Step 1, fit the orbit state vectors to an orbit propagation model that provides position $\mathbf{p}(t)$ and velocity $\mathbf{v}(t)$ vectors for any time within the azimuth time limits of the product. A simple and reasonably accurate orbit propagation model is a cubic spline.

Step 2, convert the line number into zero Doppler time

$$t_0 = \frac{t_{first}(n - 1 - l) + t_{last} l}{n - 1}$$

Step 3, using the orbit propagation model, find the sensor position $\mathbf{p}(t_0)$ and sensor velocity $\mathbf{v}(t_0)$ at zero Doppler time

Step 4, convert the pixel number into a distance from the near-range edge of the image

$$D = \begin{cases} p\delta_{pixel} - D_{origin} & \text{if pixel time ordering is increasing} \\ (m-1-p)\delta_{pixel} - D_{origin} & \text{if pixel time ordering is decreasing} \end{cases}$$

Step 5, for products with more than one time-stamped set of ground to slant range coefficients, linearly interpolate between these to determine the coefficients for the zero Doppler time. Extrapolation may be required at the start and end of an image.

$$\text{For } k = 0, \ldots 5$$

$$s_k = \frac{t_{i+1} - t_0}{t_{i+1} - t_i} s_k(t_i) + \frac{t_0 - t_i}{t_{i+1} - t_i} s_k(t_{i+1})$$

where

$$i = \begin{cases} 1 & \text{if} \quad t_0 < t_1 \\ i & \text{if} \quad t_i \le t_0 \le t_{i+1} \\ \mu - 1 & \text{if} \quad t_\mu < t_0 \end{cases}$$

This step is skipped for SLC products or when the product contains only a single set of ground to slant range coefficients.

Step 6, convert the distance from the near-range edge of the image into zero Doppler slant range.

$$R = s_0 + s_1 D + s_2 D^2 + s_3 D^3 + s_4 D^4 + s_5 D^5$$

The following steps describe how to project the slant range onto an Earth ellipsoid that is “inflated” by a fixed known elevation *h*.

Step 7, find the approximate Earth radius at a point directly below the sensor position. Note: This is only approximate because the nearest surface point is not in exactly the same direction as the Earth centre.

$$r = \frac{|\mathbf{p}(t_0)|}{\sqrt{\dfrac{p_{x_0}^2 + p_{y_0}^2}{a^2} + \dfrac{p_{z_0}^2}{b^2}}} + h$$

Step 8, find the tangent of the angle ψ between the zero Doppler plane and the vertical direction, which is the same as the ratio between the radial and tangential components of the sensor velocity (see figure below):

$$\tan\psi = \frac{v_{\text{radial}}}{v_{\text{tangential}}}$$

where

$$v_{\text{radial}} = \frac{\mathbf{v}(t_0) \cdot \mathbf{p}(t_0)}{|\mathbf{p}(t_0)|}, \quad v_{\text{tangential}} = \sqrt{|\mathbf{v}(t_0)|^2 - v_{\text{radial}}^2}$$

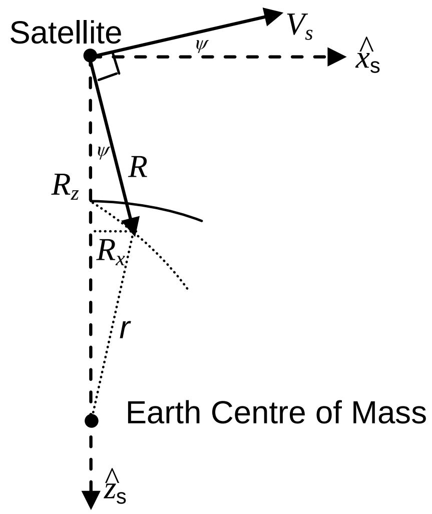
Step 9, define a satellite coordinate system centred on the sensor position, where the Z axis points towards the Earth centre, the X axis points along the tangential component of the sensor velocity, and the Y axis completes the right-handed coordinate system:

$$\hat{\mathbf{z}}_s = -\mathbf{p}(t_0) / |\mathbf{p}(t_0)|$$

$$\hat{\mathbf{y}}_s = \hat{\mathbf{z}}_s \otimes \mathbf{v}(t_0) / |\hat{\mathbf{z}}_s \otimes \mathbf{v}(t_0)|$$

$$\hat{\mathbf{x}}_s = \hat{\mathbf{y}}_s \otimes \hat{\mathbf{z}}_\mathbf{s}$$

The following figure illustrates the satellite coordinate system, with the satellite Y axis pointing out of the page.



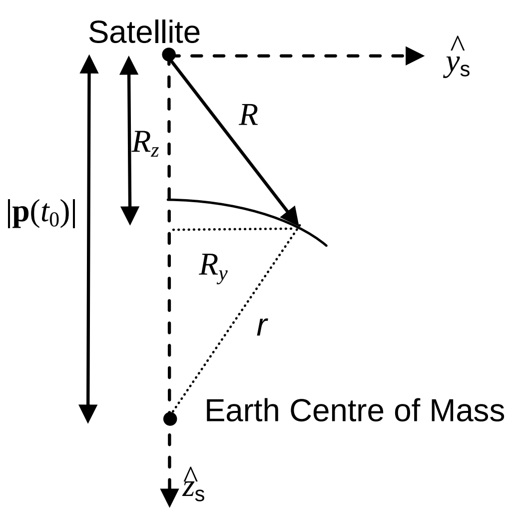
Step 10, form a triangle whose sides are the distance from the centre of the earth to the sensor, the slant range and the local earth radius and using the cosine law find the intersection point between the slant range vector and the Earth surface in the satellite coordinate system:

$$R_z = \frac{|\mathbf{p}(t_0)|^2 + R^2 - r^2}{2|\mathbf{p}(t_0)|}$$

$$R_x = R_z \tan\psi$$

$$R_y = \begin{cases} \sqrt{R^2 - R_z^2 - R_x^2} & \text{if } antennaPointing \text{ is 'Right'} \\ -\sqrt{R^2 - R_z^2 - R_x^2} & \text{if } antennaPointing \text{ is 'Left'} \end{cases}$$

The following figure illustrates this geometry, with the satellite X axis pointing into the page.



Step 11, transform the coordinates of the intersection point from satellite to target ECR coordinates:

$$\mathbf{q} = \mathbf{QR}$$

where

$$\mathbf{R} = \left(R_x \quad R_y \quad R_z - \left|\mathbf{p}(t_0)\right|\right)^t$$

$$\mathbf{q} = \left(x \quad y \quad z\right)^t$$

and the columns of the matrix **Q** are defined by the unit vectors defined in step 9

$$\mathbf{Q} = \left(\hat{\mathbf{x}}_s \quad \hat{\mathbf{y}}_s \quad \hat{\mathbf{z}}_s\right)$$

Step 12, find the Earth radius at a point directly below the intersection point (this approximation becomes more and more precise as the intersection point becomes closer to the surface):

$$r = \frac{|\mathbf{q}|}{\sqrt{\dfrac{x^2 + y^2}{a^2} + \dfrac{z^2}{b^2}}} + h$$

Repeat steps 10 to 12 until the change in *r* is less than a threshold (e.g. 0.1 m)

Step 13, convert the target ECR coordinates (*x*,*y*,*z*) into geographic coordinates (φ, λ, h) based on the WGS84 ellipsoid Earth model.

a) The following is a general method that assumes the height *h* is not yet known and solves for all 3 geographic coordinates iteratively:

$$\lambda = \text{atan2}(y, x)$$

$$\phi' = \text{atan2}\left(z, \sqrt{x^2 + y^2}\right)$$

$$CP = 1, \quad h = 0 \quad (\text{initially})$$

$$\phi = \arctan\left(\frac{\tan\phi'}{1 - \dfrac{CP \cdot \left(1 - b^2/a^2\right)}{CP + h}}\right), \quad CP = \frac{a}{\sqrt{1 - \left(1 - b^2/a^2\right)\sin^2\phi}}, \quad h = \frac{\sqrt{x^2 + y^2}}{\cos\phi} - CP$$

$$(\text{iterate the above 3 expressions until convergence})$$

b) Since the height *h* is known, method (a) above can be simplified by skipping the expressions to evaluate *h*.

c) Alternatively, the known value of *h* can be used in a simpler analytical expression that avoids the need to iterate:

$$\lambda = \text{atan2}(y, x)$$

$$\phi' = \text{asin}\left(z \middle/ \sqrt{x^2 + y^2 + z^2}\right)$$

$$\phi = \arctan\left(\tan\phi' \frac{(a+h)^2}{(b+h)^2}\right),$$

3 GEOLOCATION USING TIE POINTS

This section describes how to use the tie-point grid provided with RADARSAT-2 products to interpolate geographic coordinates corresponding to given image coordinates (forward geolocation algorithm). Two cases are presented:

- No correction for elevation, it is assumed that the actual elevation corresponds to the fixed elevation used in processing.
- Correcting the geographic coordinates for a known elevation that differs from the fixed elevation used in processing.

3.1 Input Parameters

The following parameters need to be extracted from the PIF.

Table 3-1 PIF parameters required for geolocation using tie points.

Parameter	XML tag	Description	Symbol
Pixel spacing (m)	sampledPixelSpacing	Spacing between pixels in the slant/ground range direction	δ_{pixel}
Line spacing (m)	sampledLineSpacing	Spacing between lines in the azimuth direction	δ_{line}
Number of pixels per line	numberOfSamplesPerLine	Number of pixels (samples) per line	m
Number of lines	numberOfLines	Number of lines in the image	n
Tie points	geolocationGrid	$M \times N$ grid of tie points: Regular grid of image coordinates (pixel, line) Corresponding geographic coordinates (geodetic latitude (deg), longitude (deg) and height (m) above WGS84 ellipsoid)	For $u = 0$ to $M-1$ and $v = 0$ to $N-1$ $p_{uv} = \frac{u(m-1)}{M-1}, l_{uv} = \frac{v(n-1)}{N-1}$ $\varphi_{uv}, \lambda_{uv}, h_{uv}$
Processed terrain height (m)	geodeticTerrainHeight	Terrain height used in processing	h_{proc}
Line time ordering	lineTimeOrdering	Time increases or decreases with line number	Increasing or decreasing
Pixel time ordering	pixelTimeOrdering	Range increases or decreases with pixel number	Increasing or decreasing
Semi Major Axis (m)	semiMajorAxis	Semi major axis length of the reference ellipsoid used in processing (WGS84 by default)	a
Semi Minor Axis (m)	semiMinorAxis	Semi minor axis length of the reference ellipsoid used in processing (WGS84 by default)	b
Slant range at near edge (m)	slantRangeNearEdge	Slant range at the near edge of the image	R_{near}

Parameter	XML tag	Description	Symbol
Ground range origin (m)	groundRangeOrigin	Ground range reference position	D_{origin}
Ground to slant range coefficients	groundToSlantRangeCoefficients	5th order polynomial coefficients for transforming ground range to slant range. This transformation changes along-track. Longer products (such as ScanSAR) have multiple sets of coefficients each with a different azimuth time-stamp.	$s_k(t_i): k = 0, \ldots, 5; i = 1, \ldots, \mu$
Zero Doppler Time First Line	zeroDopplerTimeFirstLine	Zero Doppler date/time of first line of image (l = 0). If north-south flipping has occurred, this value refers to the first line after flipping.	t_{first}
Zero Doppler Time Last Line	zeroDopplerTimeLastLine	Zero Doppler date/time of last line of image (l = n - 1). If north-south flipping has occurred, this value refers to the last line after flipping.	t_{last}

3.2 Forward Geolocation Algorithm Using Tie Points

Image coordinates are geolocated by interpolating between the tie-points provided in the PIF. The tie points match a regular grid of image coordinates (pixel, line) to corresponding geographic coordinates (latitude, longitude). Interpolation is performed using a bi-linear approach. Bi-quadratic interpolation can be applied for greater accuracy.

3.2.1 Without Elevation Correction

This initial approach assumes that the elevation used during processing corresponds to the actual elevation. Given an image coordinate (p, l) its corresponding geographic coordinates (φ,λ) are interpolated as follows:

Step 1, transform the tie-points from geographic to Cartesian coordinates to support interpolation in Cartesian space. The processed height is ignored; as a result the Cartesian coordinates lie on the reference ellipsoid.

$$x_{uv} = \frac{a^2 \cos \varphi_{uv} \cos \lambda_{uv}}{\sqrt{a^2 \cos^2 \varphi_{uv} + b^2 \sin^2 \varphi_{uv}}}$$

$$y_{uv} = \frac{a^2 \cos \varphi_{uv} \sin \lambda_{uv}}{\sqrt{a^2 \cos^2 \varphi_{uv} + b^2 \sin^2 \varphi_{uv}}}$$

$$z_{uv} = \frac{b^2 \sin \varphi_{uv}}{\sqrt{a^2 \cos^2 \varphi_{uv} + b^2 \sin^2 \varphi_{uv}}}$$

For $u = 0, M - 1; v = 0, N - 1$

Step 2, normalize the given image coordinates to the size of the tie-point grid

$$P = \frac{p}{m-1} M - 1$$

$$L = \frac{l}{n-1} N - 1$$

Step 3, identify the tie-point (u,v) to the top left of the normalized image coordinates (P, L),

$$u = \text{floor}(P)$$
$$v = \text{floor}(L)$$

Step 4, bi-linearly interpolate the Cartesian coordinates (x,y,z) using

$$x = (u+1-P)(v+1-L)x_{u,v} + (P-u)(v+1-L)x_{u+1,v} + (u+1-P)(L-v)x_{u,v+1} + (P-u)(L-v)x_{u+1,v+1}$$
$$y = (u+1-P)(v+1-L)y_{u,v} + (P-u)(v+1-L)y_{u+1,v} + (u+1-P)(L-v)y_{u,v+1} + (P-u)(L-v)y_{u+1,v+1}$$
$$z = (u+1-P)(v+1-L)z_{u,v} + (P-u)(v+1-L)z_{u+1,v} + (u+1-P)(L-v)z_{u,v+1} + (P-u)(L-v)z_{u+1,v+1}$$

Step 5, transform the interpolated Cartesian coordinates to latitude and longitude

$$\varphi = \text{atan}\left(\frac{az}{b\sqrt{b^2 - z^2}}\right)$$
$$\lambda = \text{atan2}(y, x)$$

Note, for greater accuracy bi-linear interpolation can be replaced with bi-quadratic interpolation. Steps 3 and 4 then become.

Step 3a, identify the tie-point (u,v) closest to the normalized image coordinates (P, L), subject to maintaining a one tie-point boundary around the edges of the image

$$u = \begin{cases} 1 : P < 0.5 \\ \text{round}(P) : 0.5 \le P < M - 1.5 \\ M - 2 : P \ge M - 1.5 \end{cases}$$

$$v = \begin{cases} 1 : L < 0.5 \\ \text{round}(L) : 0.5 \le L < N - 1.5 \\ N - 2 : L \ge N - 1.5 \end{cases}$$

Step 4a, bi-quadratically interpolate the Cartesian coordinates (x,y,z) using the matrix products

$$x = \mathbf{L}^t \mathbf{V}^{-1} \mathbf{X} \left(\mathbf{U}^{-1}\right)^t \mathbf{P}$$
$$y = \mathbf{L}^t \mathbf{V}^{-1} \mathbf{Y} \left(\mathbf{U}^{-1}\right)^t \mathbf{P}$$
$$z = \mathbf{L}^t \mathbf{V}^{-1} \mathbf{Z} \left(\mathbf{U}^{-1}\right)^t \mathbf{P}$$

where ^{t} indicates matrix transpose, **X**, **Y** and **Z** represent 3 × 3 matrices of tie-point Cartesian coordinates centred on the closest tie-point

$$\mathbf{X} = \begin{pmatrix} x_{u-1,v-1} & x_{u,v-1} & x_{u+1,v-1} \\ x_{u-1,v} & x_{u,v} & x_{u+1,v} \\ x_{u-1,v+1} & x_{u,v+1} & x_{u+1,v+1} \end{pmatrix} \quad \mathbf{Y} = \begin{pmatrix} y_{u-1,v-1} & y_{u,v-1} & y_{u+1,v-1} \\ y_{u-1,v} & y_{u,v} & y_{u+1,v} \\ y_{u-1,v+1} & y_{u,v+1} & y_{u+1,v+1} \end{pmatrix} \quad \mathbf{Z} = \begin{pmatrix} z_{u-1,v-1} & z_{u,v-1} & z_{u+1,v-1} \\ z_{u-1,v} & z_{u,v} & z_{u+1,v} \\ z_{u-1,v+1} & z_{u,v+1} & z_{u+1,v+1} \end{pmatrix}$$

the vectors **P** and **L** define quadratic exponents of the normalized image coordinates of the point being interpolated

$$\mathbf{P} = \begin{pmatrix} P^2 & P & 1 \end{pmatrix}^t \qquad \mathbf{L} = \begin{pmatrix} L^2 & L & 1 \end{pmatrix}^t$$

and the 3 × 3 matrices **U** and **V** define quadratic exponents of the tie-point grid

$$\mathbf{U} = \begin{pmatrix} (u-1)^2 & u-1 & 1 \\ u^2 & u & 1 \\ (u+1)^2 & u+1 & 1 \end{pmatrix} \qquad \mathbf{V} = \begin{pmatrix} (v-1)^2 & v-1 & 1 \\ v^2 & v & 1 \\ (v+1)^2 & v+1 & 1 \end{pmatrix}$$

For completeness, the matrix inverses of **U** and **V** are given by

$$\mathbf{U}^{-1} = \frac{1}{2}\begin{pmatrix} 1 & -2 & 1 \\ -2u-1 & 4u & -2u+1 \\ u^2+u & -2u^2+2 & u^2-u \end{pmatrix} \quad \mathbf{V}^{-1} = \frac{1}{2}\begin{pmatrix} 1 & -2 & 1 \\ -2v-1 & 4v & -2v+1 \\ v^2+v & -2v^2+2 & v^2-v \end{pmatrix}$$

3.2.2 Correction for a Known Elevation

A difference (δh) between the actual elevation (h) and the elevation used during processing (h_{proc}) generates a geolocation offset in the across-track plane. Where the actual elevation is known this offset can be calculated and corrected for. The across-track imaging geometry on which the correction is based is shown in Figure 3-1. δR and δD are the slant and ground range offsets resulting from the elevation difference δh.

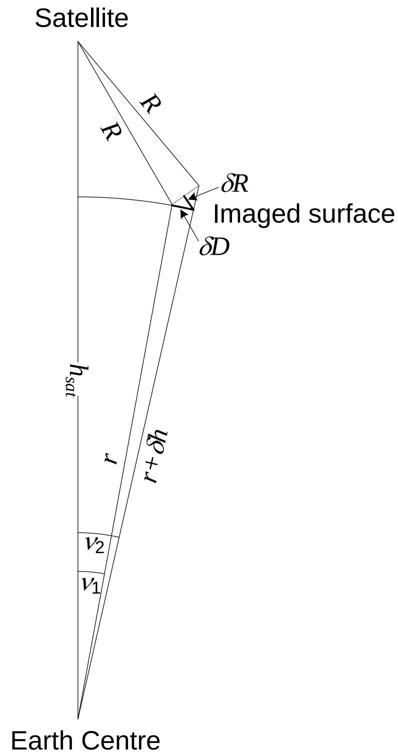


Figure 3-1 Across-track imaging geometry on which the range correction is based.

The steps for correcting the geolocation offset are outlined below.

Step 1, determine values of the near range (p_{near}) and far range (p_{far}) pixels based on pixel time ordering.

$$p_{near} = \begin{cases} 0 & : \text{ increasing pixel time ordering} \\ m-1 & : \text{ decreasing pixel time ordering} \end{cases}$$

$$p_{far} = m - 1 - p_{near}$$

Step 2, using the approach defined in Section 3.2.1 interpolate the Cartesian coordinates of the locations listed in Table 3-2 without any elevation correction

Table 3-2 Cartesian coordinates interpolated from tie-points

Location	Image coordinates	Interpolated Cartesian coordinates
Given target	p, l	x, y, z
Near range pixel on the same line as the target	p_{near}, l	$x_{near}, y_{near}, z_{near}$
Far range pixel on the same line as the target	p_{far}, l	$x_{far}, y_{far}, z_{far}$

Step 3, calculate the slant range at the target (R) and at far range (R_{far}). The slant range at near range (R_{near}) is listed in the PIF.

For SLC products, which are in slant range geometry, the slant range is simply given by

$$R = \begin{cases} R_{near} + p\delta_{pixel} & : \text{ for increasing pixel time ordering} \\ R_{near} + (m-1-p)\delta_{pixel} & : \text{ for decreasing pixel time ordering} \end{cases}$$

$$R_{far} = R_{near} + (m-1)\delta_{pixel}$$

For all other products (SGX, SGF etc) the slant range is derived from the relative ground range given by

$$D = \begin{cases} p\delta_{pixel} - D_{origin} & : \text{ for increasing pixel time ordering} \\ (m-1-p)\delta_{pixel} - D_{origin} & : \text{ for decreasing pixel time ordering} \end{cases}$$

$$D_{far} = (m-1)\delta_{pixel} - D_{origin}$$

The relative ground range is then transformed to slant range by the polynomial

$$R = \sum_{k=0}^{5} s_k D^k$$

Longer products (e.g. ScanSAR products) may contain more than one set of time-stamped ground range to slant range polynomial coefficients. In this case the polynomial coefficients for line *l* are given by linear interpolating between the time-stamped polynomial coefficients based on the zero Doppler time of line *l*. Extrapolation may be required at the start and end of an image.

$$t_0 = \frac{t_{first}(n-1-l) + t_{last} l}{n-1}$$

For $k = 0, \ldots 5$

$$s_k = \frac{t_{i+1} - t_0}{t_{i+1} - t_i} s_k(t_i) + \frac{t_0 - t_i}{t_{i+1} - t_i} s_k(t_{i+1})$$

where

$$i = \begin{cases} 1 & \text{if} \quad t_0 < t_1 \\ i & \text{if} \quad t_i \le t_0 \le t_{i+1} \\ \mu - 1 & \text{if} \quad t_\mu < t_0 \end{cases}$$

Step 4, calculate the distance (h_{sat}) of the satellite from the centre of the ellipsoid at the time of acquisition based on the across-track imaging geometry shown in Figure 3-2.

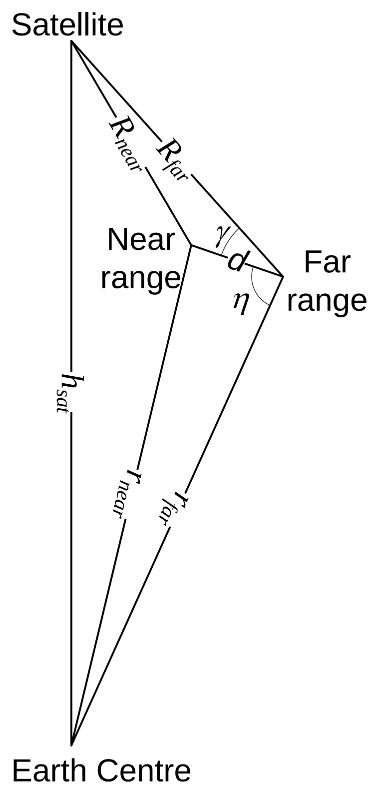


Figure 3-2 Across-track imaging geometry for calculating the satellite distance.

The satellite distance is given by

$$h_{sat} = \sqrt{r_{far}^2 + R_{far}^2 - 2r_{far}R_{far}\cos(\eta + \gamma)}$$

where η is the angle subtended at the far range location by the centre of the reference ellipsoid and the near range location

$$\eta = \arccos\left(\frac{d^2 + r_{far}^2 - r_{near}^2}{2dr_{far}}\right)$$

γ is the angle subtended at the far range location by the satellite and the near range location

$$\gamma = \arccos\left(\frac{d^2 + R_{far}^2 - R_{near}^2}{2dR_{far}}\right)$$

d is the Euclidean distance between the near range and far range locations

$$d = \sqrt{(x_{far} - x_{near})^2 + (y_{far} - y_{near})^2 + (z_{far} - z_{near})^2}$$

and r_{near} and r_{far} are the ellipsoid radii at the near range and far range locations

$$r_{near} = \sqrt{x_{near}^2 + y_{near}^2 + z_{near}^2} \; .$$
$$r_{far} = \sqrt{x_{far}^2 + y_{far}^2 + z_{far}^2}$$

Note: the above algorithm was originally developed for RADARSAT-1, with RADARSAT-2 the satellite distance can also be derived from orbit state vectors. This is simpler and potentially more accurate than the method above.

Step 5, calculate the radius of the ellipsoid at the target

$$r = \sqrt{x^2 + y^2 + z^2}$$

Step 6, calculate the angles subtended at the centre of the ellipsoid by the satellite and target without and with the height offset (Figure 3-1)

$$\cos \nu_1 = \frac{h_{sat}^2 + r^2 - R^2}{2 h_{sat} r}$$

$$\cos \nu_2 = \frac{h_{sat}^2 + (r + \delta h)^2 - R^2}{2 h_{sat} (r + \delta h)}$$

where

$$\delta h = h - h_{proc}$$

Step 7, calculate the range offset. SLC products require a slant range offset given by

$$\delta R = \sqrt{h_{sat}^2 + r^2 - 2 h_{sat} r \cos \nu_2} - R$$

All other products require a ground range offset given by

$$\delta D = (\nu_2 - \nu_1) r$$

Step 8, calculate the corrected pixel position. For SLC products the corrected pixel position is given by

$$p_{cor} = \begin{cases} \mathrm{p} + \dfrac{\delta \mathrm{R}}{\delta_{pixel}} & : \text{ increasing pixel time ordering} \\ \mathrm{p} - \dfrac{\delta \mathrm{R}}{\delta_{pixel}} & : \text{ decreasing pixel time ordering} \end{cases}$$

For all other products the corrected pixel position is given by

$$p_{cor} = \begin{cases} \mathrm{p} + \dfrac{\delta \mathrm{D}}{\delta_{pixel}} & : \text{ increasing pixel time ordering} \\ \mathrm{p} - \dfrac{\delta \mathrm{D}}{\delta_{pixel}} & : \text{ decreasing pixel time ordering} \end{cases}$$

Step 9, using the approach defined in Section 3.2.1 interpolate the Cartesian coordinates ($x_{cor}, y_{cor}, z_{cor}$) corresponding to the corrected image coordinates (p_{cor}, l).

Step 10, transform the corrected and interpolated Cartesian coordinates to geographic coordinates

$$\varphi_{cor} = \mathrm{atan}\left(\frac{a z_{cor}}{b \sqrt{b^2 - z_{cor}^2}} \right)$$

$$\lambda_{cor} = \mathrm{atan2}(y_{cor}, x_{cor})$$

4 GEOLOCATION USING RATIONAL FUNCTIONS

This section describes how to use the rational functions provided with RADARSAT-2 products to calculate the image coordinates corresponding to a given geographic coordinate (reverse geolocation algorithm).

4.1 Input Parameters

The following parameters need to be extracted from the PIF.

Table 4-1 PIF parameters required for geolocation using rational functions.

Parameter	XML tag	Description	Symbol
Line offset	lineOffset	Offset used to linearly transform line values to range of [-1, 1]	l_0
Pixel offset	pixelOffset	Offset used to linearly transform pixel values to range of [-1, 1]	p_0
Latitude offset (deg)	latitudeOffset	Offset used to linearly transform latitude values to range of [-1, 1]	φ_0
Longitude offset (deg)	longitudeOffset	Offset used to linearly transform longitude values to range of [-1, 1]	λ_0
Height offset (m)	heightOffset	Offset used to linearly transform height values to range of [-1, 1]	h_0
Line scale	lineScale	Scale used to linearly transform line values to range of [-1, 1]	σ
Pixel scale	pixelScale	Scale used to linearly transform pixel values to range of [-1, 1]	μ
Latitude scale	latitudeScale	Scale used to linearly transform latitude values to range of [-1, 1]	β
Longitude scale	longitudeScale	Scale used to linearly transform longitude values to range of [-1, 1]	α
Height scale	heightScale	Scale used to linearly transform height values to range of [-1, 1]	ν
Line numerator coefficients	lineNumeratorCoefficients	20 coefficients representing the polynomial in the numerator of the rational function mapping latitude, longitude and height to line	$\mathbf{d} = (d_1 \ldots d_{20})$
Line denominator coefficients	lineDenominatorCoefficients	20 coefficients representing the polynomial in the denominator of the rational function mapping latitude, longitude and height to line	$\mathbf{e} = (e_1 \ldots e_{20})$
Pixel numerator coefficients	pixelNumeratorCoefficients	20 coefficients representing the polynomial in the numerator of the rational function mapping latitude, longitude and height to pixel	$\mathbf{f} = (f_1 \ldots f_{20})$

Parameter	XML tag	Description	Symbol
Pixel denominator coefficients	pixelDenominatorCoefficients	20 coefficients representing the polynomial in the denominator of the rational function mapping latitude, longitude and height to pixel	$\mathbf{g} = (g_1 \ldots g_{20})$

4.2 Reverse Geolocation Algorithm Using Rational Functions

This algorithm is also described in [R-1] but with different notation. Note, geolocation accuracy may be slightly reduced when the actual height differs from the height offset (h_0) by more than 700 m.

Step 1, normalize the latitude, longitude and height values into the range [-1,1] using the offsets and scale factors defined in the PIF

$$A = \frac{\lambda - \lambda_0}{\alpha}, B = \frac{\varphi - \varphi_0}{\beta}, H = \frac{h - h_0}{\nu}$$

Note, in images straddling ±180º longitude, care is needed to avoid a step function in the normalized longitude. At points where the difference between the longitude and the longitude offset is greater than 180º then

$$A = \frac{\lambda - 360 - \lambda_0}{\alpha}$$

and if the difference between the longitude and the longitude offset is less than -180º then

$$A = \frac{\lambda + 360 - \lambda_0}{\alpha}$$

Step 2, generate the following vector of polynomial exponents from the normalized variables

$$\mathbf{q} = (1 \quad A \quad B \quad H \quad AB \quad AH \quad BH \quad A^2 \quad B^2 \quad H^2 \quad ABH \quad A^3 \quad AB^2 \quad AH^2 \quad A^2B \quad B^3 \quad BH^2 \quad A^2H \quad B^2H \quad H^3)$$

Step 3, calculate the normalized line and pixel values using the rational functions (range [-1,1])

$$L = \frac{\mathbf{d} \cdot \mathbf{q}}{\mathbf{e} \cdot \mathbf{q}}$$

$$P = \frac{\mathbf{f} \cdot \mathbf{q}}{\mathbf{g} \cdot \mathbf{q}}$$

where · indicates the inner product.

Step 4, Un-normalize the line and pixel values using the offsets and scale factors defined in the PIF

$$l = \sigma L + l_0$$

$$p = \mu P + p_0$$

5 REFERENCES

R-1 RN-RP-51-2713 RADARSAT-2 Product Format Definition, Issue 1/10. 2011. http://gs.mdacorporation.com/SatelliteData/Radarsat2/Products.aspx