

Department of the Interior
U.S. Geological Survey

LS-IAS-06
Version 1.0

MULTISPECTRAL SCANNER (MSS) GEOMETRIC ALGORITHM DESCRIPTION DOCUMENT (ADD)

Version 1.0
June 2012



MULTISPECTRAL SCANNER (MSS)
GEOMETRIC ALGORITHM
DESCRIPTION DOCUMENT (ADD)
June 2012

Prepared By:

Reviewed By:

M. Choate Date
Calibration Analyst
SGT

L. Johnson Date
IAS Software Project Lead
SGT

Prepared By:

Reviewed By:

D. Steinwand Date
Computer Scientist
U.S. Geological Survey

R. Junker Date
IAS Systems Engineer
SGT

Prepared By:

Approved By:

R. Rengarajan Date
Cal/Val Lead
SGT

E. Fosnight Date
Data Acquisition Manager
U.S. Geological Survey

Reviewed By:

Reviewed By:

T. McVay Date
Level 1 Software Developer
SGT

K. Alberts Date
Landsat Ground System Lead
U.S. Geological Survey

EROS
Sioux Falls, South Dakota

Document History

Document Number	Document Version	Publication Date	Change Number	Status/Issue
LS-IAS-06	Version 1.0	June 2012	DCR 379	

Contents

Document History	iii
Contents	v
List of Figures	ix
1 Introduction	1
1.1 Document Overview	1
1.2 Instrument and Spacecraft Overview	2
1.3 Calibration Parameter File (CPF) Overview	4
2 Coordinate Systems	5
2.1 MSS Focal Plane Coordinate System	5
2.2 Navigational Reference Coordinate System	5
2.3 Orbital Coordinate System	6
2.4 Earth-Centered Rotating (ECR) Coordinate System	6
2.5 Geodetic Coordinate System	7
2.6 Map Projection Coordinate System	8
3 Coordinate Transformations	9
3.1 Sensor to Navigational System	9
3.2 Navigational Reference to Orbital	9
3.3 ECR to Geodetic	10
3.4 Geodetic to Map Projection	11
3.5 Orbital Propagation	11
4 Time Systems	15
4.0.1 Spacecraft Time	15
4.1 UTC	15

5	Algorithm Description	17
5.1	Common Methods	17
5.1.1	GCPs	17
5.1.2	Mensuration	17
5.1.3	Coordinate Axis Rotation	19
5.2	Read MSS-X Header	20
5.3	Read MSS-X Cal Data	21
5.4	Create MSS-X Geometric Model	21
5.5	Map Output Coordinates (UTM Projection) to Input L1R-H Pixel Locations	30
5.5.1	Defining the MSS-X Processing Parameters	31
5.5.2	Initialize Target Parameters	31
5.5.3	Initialize Orbital Propagation	32
5.5.4	Initialize Times	34
5.5.5	Iterate Solution	35
5.6	Create Hybrid Imagery – Along Scan Line Adjustments	42
5.6.1	Define MSS-X Detector Offsets	43
5.6.2	Read Scan Line Length from L0R MSCD	43
5.6.3	Build a Set of Cubic Convolution Resampling Weights	43
5.6.4	Resample L1R Imagery	43
6	Overview of MSS-A and MSS-P Processing	45
6.1	MSS-P Geometric Processing	46
6.2	HOM Projection	46
6.3	Determine Image Frame	48
6.4	MSS-A Geometric Processing	51
6.5	Assumptions Made in the Algorithm Development	53
6.6	MSS-A Geometric Processing	54
7	The GPYRAMID Algorithm	59
8	Create Resampling Grid	67
8.1	Read MSS-X Geometric Model File	67
8.2	Read MSS-X CPF	67
8.3	Determine Output Map Projected Image Frame	67
8.4	Map Output Grid Cell Locations to Hybrid Level 1 Radiometric (L1R-H) Pixels	69

9 Resample L1R-H Imagery	75
10 MSS REFINE	81
11 GVERIFY Algorithm Description	93
11.0.1 Known Limitations to GVERIFY Algorithm	101
Appendices	103
Acronyms	105
References	109

List of Figures

1.1	MSS Scanner Ground Scanning Pattern	3
2.1	MSS Focal Plane Coordinate System	6
2.2	Geodetic Coordinate System	7
3.1	Defining a Position and Velocity Vector	12
5.1	MSS Scanning Velocity	24
5.2	Position and Velocity Vector with Respect to Attitude xyz Axis . .	31
5.3	Apply Pitch-Yaw-Roll to Attitude xyz Axis	32
5.4	Line-of-Sight (LOS) Vector Within xyz Coordinate Axis	39
5.5	Satellite State at Iteration n and Final Iteration (Solution)	40
5.6	Non-Ideal LOS Vector with Respect to Perturbed xyz Axis	40
6.1	Arrangement of Tic-Mark locations on the Image Pixel Grid	49
6.2	Tic-Mark Location Information on the Image Grid	50
6.3	Image Space Grid with Tic-Mark Intersections Displayed	51
7.1	Gaussian Image Pyramid Levels	61
7.2	UTM Projected Scene with Evenly Spaced GCP Points	62
8.1	Geometric Resampling Grid	68
9.1	Terrain Table Calculations	76
10.1	Grid Cell Mapping	82
10.2	Curvature Test for Precision Correction Process	89
11.1	Neighborhood Analysis	97
11.2	Scene Divided into 3×3 Regions	100

1 Introduction

1.1 Document Overview

This document explains the method for geometrically processing Multispectral Scanner (MSS)-X, MSS – Archive Format (MSS-A), and MSS – Processed Format (MSS-P) datasets associated with the MSS instrument. This processing includes systematically corrected images (corrected using only payload and calibration data), precision corrected images (images that have ground control applied), and terrain corrected images (images with relief displacement removed and ground control applied). To calculate relief displacement, elevation information corresponding to individual image samples is required. A Digital Elevation Model (DEM) co-registered to the imagery is used for this purpose.

The first step in correcting MSS imagery is to populate the geometric satellite model data structure and corresponding file. This data structure is a convenient way to prepare and store the payload and instrument characterization data needed for geometrically correcting a Level 1 Radiometrically Corrected (L1R) image data set. This method facilitates the conversion between output latitude and longitude locations projected to a specified Earth model and the corresponding L1R image pixels.

After the model data structure is populated, a resampling grid is created. The resampling grid converts geographic latitude and longitude locations created into a map projected x/y coordinate. The resampling grid efficiently converts an output space line and sample location, or x/y projection coordinate, to an input line and sample location during the image resampling process. The MSS-X resampling grid does not take into account terrain effects; the process of compensating for elevation displacement is performed as a separate step within the resampler, using a Lookup Table (LUT) approach.

Image resampling follows the creation of the resampling grid data structure and corresponding file. A radiometrically corrected image is resampled

depending on the geometric characteristics of the input data. A geometrically corrected data set using only the telemetry data and characterization parameters is referred to as a systematically corrected image. A data set that has systematic correction removed using Ground Control Points (GCPs) and has relief adjustments due to terrain is referred to as a terrain corrected image.

For a terrain corrected image, ground control measures offset within the systematic image, which then updates the geometric resampling grid. The ground control removes any positional or attitude errors in the grid data structure due to payload data errors and changes in characterization parameters.

1.2 Instrument and Spacecraft Overview

The first five Landsat missions carried the MSS as a payload instrument. The MSS instruments were line scanners that used an oscillating mirror to produce an image scan of the Earth as a target in a direction perpendicular to the spacecraft's velocity. The instrument carried four bands, with each band containing six lines or detectors placed in the along track direction of the spacecraft's velocity. These six lines were scanned during the forward sweeps of the instrument's mirror operations. Reverse scans imagery was not obtained or acquired. See Figure 1.1.

The MSS optical system consisted primarily of a scan mirror, telescope, rotating shutter, and an array of 24 optical fibers. The telescope optics consist of a 9 inch Ritchey-Chretien with a 0.0889 meter (3.5 inch) secondary mirror with a focal length of 0.8255 meters (32.5 inches). The optical fibers transmit the image intensities to the appropriate detectors. The 24-square fibers arranged within the telescope's focal plane carry the radiant energy to the 24 detectors. The MSS mirror samples imagery only in one direction (termed the forward direction or forward scans); the six detectors placed in the along track direction produce an along track viewing angle that compensates for the spacecraft along track motion during reverse scans where imagery is not acquired. The effective across track viewing angle of the MSS mirror motion is 11.6° . The scanning frequency is 13.62 Hertz (Hz), producing a total scan time of 0.07342 seconds with a duty cycle of 0.0315 to 0.034 seconds.

The Landsat 1–5 satellites were placed in a nearly polar sun-synchronous orbit; however, Landsat 1–3 were in a slightly different orbit than Landsat 4–5. Landsat 1–3 were placed in an orbit referred to as the Worldwide Reference System (WRS)-1, which had a nominal inclination angle of 99° and an altitude

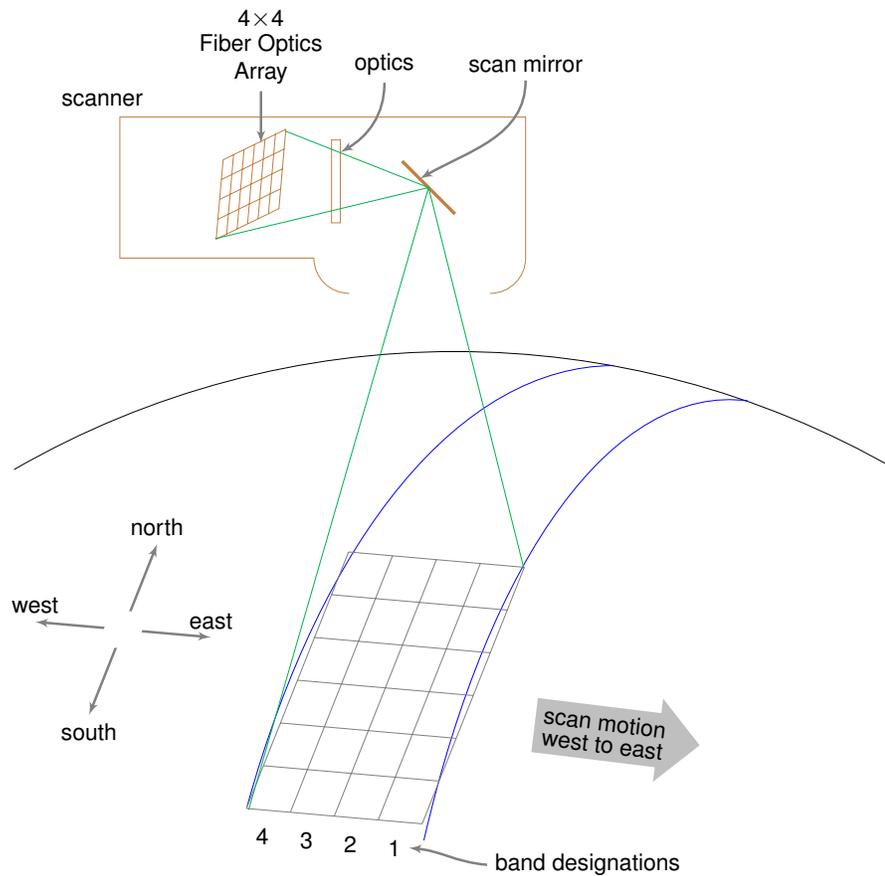


Figure 1.1. MSS Scanner Ground Scanning Pattern

of 920 kilometers (km) producing a repeat cycle of 18 days. WRS-1 scenes were split by day cycle and latitude such that there were 251 orbits and 248 scenes per orbit. This organization created a grid system where a single scene is identified by its corresponding path and row. Landsat 4–5 had a nominal inclination angle of 98.2° and an altitude of 705 km producing a repeat cycle of 16 days, 233 orbits or paths, and 248 rows. Landsat 4–5 are in the orbit referred to as the [WRS-2](#).

1.3 Calibration Parameter File (CPF) Overview

All MSS instrument-specific geometric and processing parameters are stored in an American Standard Code for Information Interchange ([ASCII](#)) file accessed during processing. The file containing these parameters is referred to as the Calibration Parameter File ([CPF](#)).

2 Coordinate Systems

2.1 MSS Focal Plane Coordinate System

The focal plane coordinate system defines the band and detector offsets from the instrument optical axis used to generate the image space viewing vectors for individual detector samples. The coordinate system is defined where the z -axis is along the optical axis and is positive toward the sensor chip assemblies. The origin is where the optical axis intersects the focal plane. The x -axis is in the along track direction of the satellite. The y -axis lies in the across track direction of the satellite. Figure 2.1 shows the layout of the MSS focal plane.

The detectors are sampled serially at a sampling rate of $0.398 \mu\text{sec}$. This sampling rate creates a ground pattern where each subsequent detector within a band is slightly displaced between two adjacent detectors (see Figure 1.1). The detector offsets represent the offsets within the CPF and are accounted for during the along scan resampling setup (see Section 5.5).

2.2 Navigational Reference Coordinate System

The navigation reference frame is the body-fixed coordinate system used for spacecraft attitude determination and control. The spacecraft Attitude Control System (ACS) defines the coordinate axes. Therefore, in nominal “non-pointing” mode, the ACS attempts to keep the navigation reference frame aligned with the orbital coordinate system. This method allows the Landsat optical axis to always point normal to the surface of the Earth.

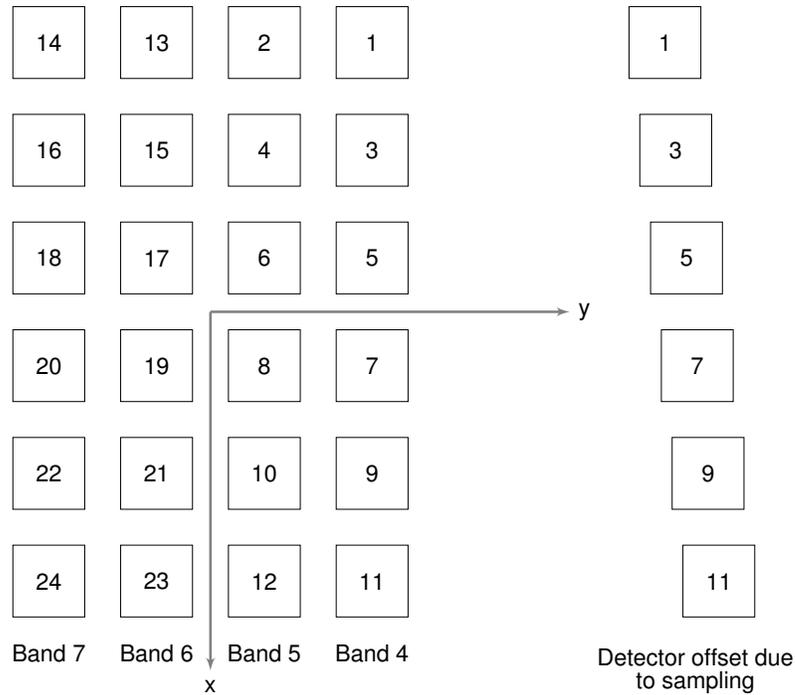


Figure 2.1. MSS Focal Plane Coordinate System

2.3 Orbital Coordinate System

The orbital coordinate system is centered on the satellite, and its orientation is based on the spacecraft position in inertial space. The origin is the spacecraft center of mass, with the z -axis of the spacecraft pointing at the Earth's center of mass when in its "nominal" pointing mode. The y -axis is the normalized cross product of the z -axis and the instantaneous (inertial) velocity vector. The y -axis corresponds to the negative of the instantaneous angular momentum vector direction. The x -axis is the cross product of the y - and z -axes.

2.4 Earth-Centered Rotating (ECR) Coordinate System

The Earth-Centered Rotating (ECR) coordinate system is Earth-fixed with its origin at the center of the Earth's mass. The ECR coordinate system corre-

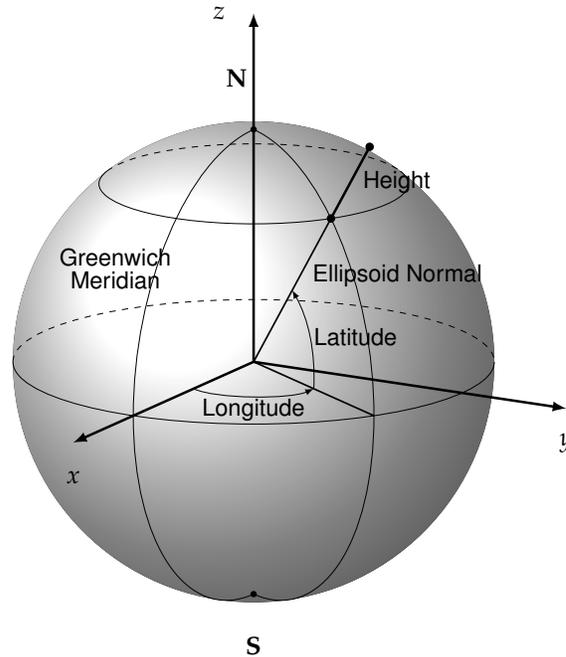


Figure 2.2. Geodetic Coordinate System

sponds to the Bureau International de l'Heure (BIH) defined Conventional Terrestrial System, which is the same as the U.S. Department of Defense (DoD) World Geodetic System 1984 (WGS84) geocentric reference system.

2.5 Geodetic Coordinate System

The geodetic coordinate system is based on the WGS84 reference frame with coordinates expressed in latitude, longitude, and height above the reference Earth ellipsoid (see Figure 2.2). The definition of the ECR coordinate system does not require an ellipsoid, but the geodetic coordinate system depends on the selection of an Earth ellipsoid. Latitude and longitude are defined as the angle between the ellipsoid normal and its projection onto the equator, and the angle between the local meridian and the Greenwich meridian, respectively. The scene center and scene corner coordinates in the Level 0 Reformatted (LOR) product metadata are expressed in the geodetic coordinate system.

2.6 Map Projection Coordinate System

Geometrically corrected image data sets are generated with respect to a map projection coordinate system, such as Universal Transverse Mercator ([UTM](#)). This generation provides a mapping from latitude and longitude to a plane coordinate system that is an approximation to a Cartesian coordinate system for a portion of the Earth's surface. It is used for convenience as a method of providing the digital image data in an Earth-referenced grid compatible with other ground referenced data sets. Although the map projection coordinate system is only an approximation to a true local Cartesian coordinate system at the Earth's surface, the mathematical relationship between the map projection and geodetic coordinate systems is precisely and unambiguously defined, see [Map Projections - A Working Manual \[Snyder, 1987\]](#) for more details.

3 Coordinate Transformations

3.1 Sensor to Navigational System

The MSS instrument alignment matrix describes the relationship between the sensor and navigation reference coordinate systems. This transformation is a three-dimensional rotation, implemented as a matrix multiplication, between the [ACS](#) reference and the instrument scan mirror. The nominal rotation matrix is the identity matrix. Currently, this matrix is set to its nominal value.

3.2 Navigational Reference to Orbital

The spacecraft attitude defines the relationship between the navigation reference and orbital coordinate systems. This transformation is three-dimensional with the components of the rotation matrix being functions of the spacecraft roll, pitch, and yaw attitude angles. The nature of the roll, pitch, and yaw functions depend on the exact definition of these angles (i.e., how the [ACS](#) generates these angles). In the initial model, it is assumed that the rotations are performed in the order pitch-yaw-roll. Because the spacecraft attitude is constantly changing, this transformation is time varying. The nominal rotation matrix is the identity matrix because the ACS maintains the MSS instrument pointing normal to Earth's surface.

3.3 ECR to Geodetic

The relationship between **ECR** and geodetic coordinates can be in its direct form:

$$\begin{aligned}e^2 &= 1 - b^2/a^2 \\n &= \frac{a}{\sqrt{(1 - e^2 \sin^2 \phi)}} \\x &= (n + h) \cos \phi \cos \lambda \\y &= (n + h) \cos \phi \sin \lambda \\z &= [n(1 - e^2) + h] \sin \phi\end{aligned}$$

Where:

x, y, z = ECR coordinates

ϕ, λ, h = Geodetic coordinates

n = Ellipsoid radius of curvature in the prime vertical

e^2 = Ellipsoid eccentricity squared

a, b = Ellipsoid semi-major and semi-minor axes

The closed-form solution for the general inverse problem (which is the problem of interest in this example) involves the solution of a quadratic equation and is not typically used in practice. Instead, an iterative solution is used for latitude and height for points that do not lie on the ellipsoid surface (i.e., for $h \neq 0$).

To convert ECR Cartesian coordinates to spherical coordinates:

Define:

$$\begin{aligned}r &= \sqrt{x^2 + y^2 + z^2} \\ \phi' &= \sin^{-1} \frac{z}{r} \\ \lambda &= \tan^{-1} \frac{y}{x}\end{aligned}$$

Initialize:

$$\begin{aligned}\theta &= \phi' \\ h_0 &= 0\end{aligned}$$

Iterate until $|h_i - h_{i+1}| < \text{TOL}$:

$$r_e = \frac{a\sqrt{1-e}}{\sqrt{1-e\cos^2\theta}}$$

$$\phi = \tan^{-1} \frac{\tan\theta}{1-e}$$

$$\Delta\phi = \phi - \theta$$

$$r_s = r^2 - r_e^2 \sin^2(\Delta\phi)$$

$$h_{i+1} = \sqrt{r_s} - r_e \cos(\Delta\phi)$$

$$\theta = \phi' - \sin^{-1} \left(\frac{h_{i+1}}{r \sin(\Delta\phi)} \right)$$

3.4 Geodetic to Map Projection

The transformation from geodetic coordinates to the output map projection depends on the type of projection selected. The mathematics for the forward and inverse transformations for [UTM](#), Lambert Conformal Conic, Transverse Mercator, Oblique Mercator, Polyconic, and Space Oblique Mercator ([SOM](#)) are given in Map Projections – A Working Manual [[Snyder, 1987](#)]. Further details of the SOM mathematical development are presented in Space Oblique Mercator Mathematical Development [[Snyder, 1981](#)]. The United States Geological Survey ([USGS](#)) General Cartographic Transformation Package ([GCTP](#)) handles map projections and is located at <ftp://edcftp.cr.usgs.gov/software/gctpc/>.

3.5 Orbital Propagation

The MSS-X header file contains a set of ephemeris positional values, typically eleven points, sampled at a rate of five seconds per sample. Satellite position is found using only the first, mid, and last point within the data stream. Satellite positions or locations are then found by treating locations as planar, having constant angular velocity and a variable radius. Angular velocity and the change in radius is then based on changes between either the first-to-mid or mid-to-last ephemeris points listed within the MSS-X header file.

Define the position and velocity vector (the example given is simplified to a two coordinate system (Figure [3.1](#)):

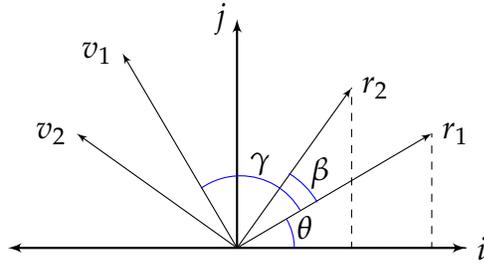


Figure 3.1. Defining a Position and Velocity Vector

Define two spacecraft positional and velocity vectors in both radial and Cartesian coordinates:

$$r_1 = x_1 \vec{i} + y_1 \vec{j}$$

$$x_1 = r_1 \cos \theta \rightarrow \cos \theta = \frac{x_1}{r_1}$$

$$y_1 = r_1 \sin \theta \rightarrow \sin \theta = \frac{y_1}{r_1}$$

$$r_2 = x_2 \vec{i} + y_2 \vec{j}$$

$$x_2 = r_2 \cos(\theta + \beta) = r_2(\cos \theta \cos \beta - \sin \theta \sin \beta)$$

$$y_2 = r_2 \sin(\theta + \beta) = r_2(\cos \theta \sin \beta + \sin \theta \cos \beta)$$

$$v_1 = v_{x1} \vec{i} + v_{y1} \vec{j}$$

$$v_{x1} = v_1 \cos(\theta + \gamma)$$

$$v_{y1} = v_1 \sin(\theta + \gamma)$$

$$v_2 = v_{x2} \vec{i} + v_{y2} \vec{j}$$

$$v_{x2} = v_2 \cos(\theta + \beta + \gamma)$$

$$v_{y2} = v_2 \sin(\theta + \beta + \gamma)$$

For a circular orbit, the velocity vectors are orthogonal to the positional vec-

tor:

$$v_{x1} = v_1 \cos(\theta + \pi/2) = -v_1 \sin \theta \Rightarrow \frac{v_{x1}}{v_1}$$

$$v_{y1} = v_1 \sin(\theta + \pi/2) = v_1 \sin \theta \Rightarrow \frac{v_{y1}}{v_1}$$

Therefore, the relationship between the two vector states is:

$$x_2 = r_2 \frac{x_1}{r_1} \cos \beta \vec{i} + r_2 \frac{v_{x1}}{v_1} \sin \beta \vec{j}$$

$$y_2 = r_2 \frac{y_1}{r_1} \cos \beta \vec{i} + r_2 \frac{v_{y1}}{v_1} \sin \beta \vec{j}$$

$$v_{x2} = \frac{dx_2}{d\beta} = -r_2 \frac{y_1}{r_1} \sin \beta + r_2 \frac{v_{x1}}{v_1} \cos \beta$$

$$v_{y2} = \frac{dy_2}{d\beta} = -r_2 \frac{y_1}{r_1} \sin \beta + r_2 \frac{v_{y1}}{v_1} \cos \beta$$

4 Time Systems

Two time systems are of primary interest for the MSS geometric algorithms: Spacecraft Time and Universal Time – Coordinated (**UTC**).

- Spacecraft Time is the time system applicable to the spacecraft time codes found in the telemetry, or MSS-X header data.
- UTC is the standard reference for civil timekeeping. UTC is adjusted periodically by whole leap seconds to keep within 0.9 seconds of **UTC Corrected (UT1)**. UT1 is based on the actual rotation of the Earth and provides the transformation from stellar-referenced Earth-Centered Inertial (**ECI**) coordinates to terrestrial-referenced **ECR** coordinates.

The significance of these two time systems with respect to the MSS geometric algorithms is described in the following sections.

4.0.1 Spacecraft Time

The spacecraft time is associated with a specific epoch of the instrument. This time reference is instrument specific.

4.1 UTC

As mentioned, UTC is maintained within 0.9 seconds of UT1 by the occasional insertion of leap seconds. A table of leap seconds relating UTC to International Atomic Time (**TAI**) is available from the National Earth Orientation Service (**NEOS**) website at <http://maia.usno.navy.mil>.

5 Algorithm Description

5.1 Common Methods

5.1.1 GCPs

The [GCPs](#) are small image chips (64×64 pixels) with geographic information that are extracted from the reference. The GCPs are correlated with the given (search) image to determine the true geospatial offsets of the given image. The GCPs are extracted from the reference image using the Modified Moravec Interest Operator ([MMIO](#)) algorithm developed to identify well defined interest points from the reference scene [[mmiopaper, 2008](#)]. Using these interest points increases the success of correlation with the search image and provides accurate offsets. For MSS processing, the Global Land Survey ([GLS](#)) 2000 data set is used as reference image for the precision correction process. The USGS validated the GLS 2000 reference data set to have an expected Root Mean Squared Error ([RMSE](#)) of 25m or less [[Rengarajan et al., 2012](#)]. Other GLS data set such as GLS 1975 and GLS 1990 are also used for performing Image-to-Image ([I2I](#)) characterization [[I7g, 2006](#)] in the GPYRAMID algorithm described in (Section 7). The USGS validated the accuracy of the GLS 1975 and GLS 1990 to be accurate to within one pixel from the GLS 2000 data set [[Rengarajan et al., 2012](#)]. The GLS 1975 and GLS 1990 are used for initial approximation in GPYRAMID algorithm to reduce the negative impact on correlation due to temporal changes between MSS data and GLS 2000 data derived from Enhanced Thematic Mapper Plus ([ETM+](#)) scenes.

5.1.2 Mensuration

Unless otherwise specified, normalized cross-correlation measures the spatial differences between two image sources, one image source is considered the

search imagery while the other is considered the reference imagery. For product generation of MSS imagery the reference imagery is the GLS data sets. The search imagery is the corresponding MSS imagery data type (MSS-X, MSS-P, MSS-A). Image windows are extracted and correlation is performed over the windowed area. The correlation process only measures linear distortions over the windowed areas. By choosing windows that are well distributed throughout the imagery, nonlinear differences between the image sources can be found.

Normalized cross-correlations produce a discrete correlation surface. To find a sub-pixel location associated with the offset, fit a polynomial around a 3×3 area centered on the correlation peak. Use the polynomial coefficients to solve for the peak or sub-pixel location. The normalized cross-correlation process reduces any correlation artifacts that may arise from radiometric differences between the two image sources.

If the two image windows of size $N \times M$ are defined by f and g , the measurement steps are:

1. Perform normalized gray scale correlation:

$$R(x, y) = \frac{\sum_{j=-N/2}^{N/2} \sum_{i=-M/2}^{M/2} [(f(j, i) - \bar{f}) (g(x + j, y + i) - \bar{g})]}{\sum_{j=-N/2}^{N/2} \sum_{i=-M/2}^{M/2} [(f(j, i) - \bar{f})^2 (g(x + j, y + i) - \bar{g})^2]^{1/2}}$$

Where:

$$\bar{f} = \frac{1}{(M + 1)(N + 1)} \sum_{j=-N/2}^{N/2} \sum_{i=-M/2}^{M/2} f(j, i)$$

$$\bar{g} = \frac{1}{(M + 1)(N + 1)} \sum_{j=-N/2}^{N/2} \sum_{i=-M/2}^{M/2} g(x + j, y + i)$$

2. Find the peak of the discrete correlation surface.
3. Fit a second order polynomial to a 3×3 area centered on the correlation peak. The polynomial's form:

$$P(x, y) = a_0 + a_1x + a_2y + a_3xy + a_4x^2 + a_5y^2$$

A least squares fit is performed on the points to solve for the polynomial coefficients.

a) Set up matrices:

$$\begin{matrix} [Y] & = & [X] & [a] \\ 9 \times 1 & & 9 \times 6 & 6 \times 1 \end{matrix}$$

Where X = Correlation locations centered around the peak, Y = Correlation values corresponding to X locations, and a = Polynomial coefficients.

b) Solve for polynomial coefficients:

$$[a] = \left([X]^T [X] \right)^{-1} [X]^T [Y]$$

4. Find partial derivatives of polynomial equation in terms of x and y :

$$\begin{aligned} \frac{\partial}{\partial x} P(x, y) &= a_1 + a_3 y + 2a_4 x \\ \frac{\partial}{\partial y} P(x, y) &= a_2 + a_3 x + 2a_5 y \end{aligned}$$

5. Set partial equations equal to zero and solve for x and y :

$$\begin{aligned} x_{\text{offset}} &= \frac{2a_1 a_5 - a_2 a_3}{a_3^2 - 4a_4 a_5} \\ y_{\text{offset}} &= \frac{2a_2 a_4 - a_1 a_3}{a_3^2 - 4a_4 a_5} \end{aligned}$$

Where x_{offset} = Sub-pixel offset in x direction and y_{offset} = Sub-pixel offset in y direction.

6. Combine the sub-pixel offset calculated in step 5 to the peak location from step 2 to calculate the total offset.

5.1.3 Coordinate Axis Rotation

To apply the attitude perturbations, define a nominal, non-perturbed, orthogonal set of coordinate axis. Roll, pitch, and yaw rotations are applied across each coordinate axis. This rotation produces a coordinate transformation to the nominal coordinate axes for each angular rotation. Quaternion operations apply these rotations across an individual axis.

Given a set of three orthogonal vectors representing a three dimensional coordinate system:

$$\begin{aligned}\vec{x} &= \langle x_x, x_y, x_z \rangle \\ \vec{y} &= \langle y_x, y_y, y_z \rangle \\ \vec{z} &= \langle z_x, z_y, z_z \rangle\end{aligned}$$

A rotation around any one axis (the following example is for an angle of θ around the y -axis) gives a new coordinate system defined by three new vectors. Define quaternions for three axis vectors:

$$\begin{aligned}Q_x &= \langle x_x, x_y, x_z, 0.0 \rangle \\ Q_y &= \langle y_x, y_y, y_z, 0.0 \rangle \\ Q_z &= \langle z_x, z_y, z_z, 0.0 \rangle\end{aligned}$$

Rotate around the y -axis by an angle \hat{y} using quaternion math:

$$\begin{aligned}Q_{y\text{new}} &= \langle y_x \sin \frac{\theta}{2}, y_y \sin \frac{\theta}{2}, y_z \sin \frac{\theta}{2}, \cos \frac{\theta}{2} \rangle \\ Q_{x\text{new}} &= Q_{y\text{new}} Q_x Q_{y\text{new}}^* \\ Q_{z\text{new}} &= Q_{y\text{new}} Q_z Q_{y\text{new}}^*\end{aligned}$$

Where $Q_{y\text{new}}^*$ is the complex conjugate of the quaternion $Q_{y\text{new}}$.

5.2 Read MSS-X Header

The MSS-X telemetry data are stored within an Object Description Language (ODL) compliant file. This file is referred to as the MSS-X header file. Within this header file, the following information is retrieved when creating the geometric model:

- Relationship between image line / sample location and Greenwich Mean Time (GMT), given as scene center time.
- Spacecraft attitude and corresponding GMT information.
- Spacecraft ephemeris and corresponding GMT information.

5.3 Read MSS-X Cal Data

Scan line length is stored within the MSS-X calibration data file. This scan line length is read from the MSS-X calibration data file and stored within the L0R Mirror Scan Correction Data (MSCD) file during reformatting for MSS Level 0 Reformatted Product (L0Rp) generation. This reformatting produces a Landsat L0R compliant data set that can be used for processing. The scan line length stored within the MSCD file normalizes or scales the MSS-X mirror profile coefficients and is stored per-scan within the geometric model. This action adjusts individual scan line lengths that vary from the nominal scan length. These parameters normalize the line length of each scan.

5.4 Create MSS-X Geometric Model

The MSS-X geometric model represents information that can be used to map, for a given Earth model, a latitude and longitude coordinate to its corresponding radiometrically corrected but geometrically uncorrected pixel location with an L1R-H data set. This type of mapping is referred to as a “reverse” geometric model. This type of model differs from the Image Assessment System (IAS) Thematic Mapper (TM) and ETM+ “forward” geometric models, which map L0Rp pixels to a location on a given Earth model. This inverse geometric model establishes a relationship between L1R-H pixels, spacecraft position, and orientation to time, specifically GMT. Using this pixel time, spacecraft position, spacecraft attitude, scan mirror angle, and focal plane characteristics, an individual L1R-H pixel can have a geographic location on the Earth model for a specific state of the spacecraft. Under this type of modeling, although a solution is found where spacecraft position, orientation, and scan mirror state matches a given time and L1R-H pixel, this solution does not necessarily represent the true state of the spacecraft or scan mirror. Only one possible solution exists between time and the given parameters.

The geometric model contains the following:

- L0Rp scene center time
- Spacecraft position with respect to GMT
- Spacecraft velocity with respect to GMT
- Spacecraft pointing with respect to GMT

- Mirror scan angle with respect to LORp pixel

LORp scene center is read directly from the MSS-X header file and stored within the MSS-X geometric model. This time is read from the MSS-X header file as an ASCII format, which converts to a floating point format and is stored within the model. All other times present within the inverse geometric model are based upon this scene center time.

Spacecraft position is stored in an ASCII format within the MSS-X header file. Corresponding spacecraft positional times are also stored in the MSS-X header as an ASCII format. Positions are stored as latitude, longitude, and altitude coordinates. These latitude and longitude coordinates are stored in units of degrees. Spacecraft positional times are in GMT and are in units of milliseconds. Each latitude, longitude, and altitude has a corresponding GMT time. These coordinates are converted from this geodetic coordinate system to a Cartesian coordinate system and stored within the geometric model along with the given time for each point. Values are stored as x , y , and z coordinates in units of meters. The nominal number of spacecraft coordinates and times is 11. A GMT time for each coordinate is also stored in the geometric model in units of seconds.

Spacecraft attitude is stored in an ASCII format within the MSS-X header file. Unlike the spacecraft positional elements, no spacecraft attitude times are stored within the MSS-X header file. The nominal number of spacecraft attitude points is nine, where the center value is the attitude at the scene center time, listed in the MSS-X header file, and each point is separated by a sampling time of 13.8030 seconds. These values represent the roll, pitch, and yaw of the spacecraft with respect to the local horizontal. The MSS-X header file also contains the spacecraft scene mean roll, pitch, and yaw and the scene roll, pitch, and yaw rates. Upon investigation of several MSS-X header attitude values, the individual values contained inconsistencies when viewed across multiple scenes containing overlapping points. A better representation, with regards to keeping consistency between consecutive scenes, was present between the image center roll, pitch, and yaw values and the corresponding roll, pitch, and yaw rates. These scene center values are used for processing. Therefore, the MSS-X geometric processing system contains two different spacecraft attitude options for defining the nine roll, pitch, and yaw values within the geometric model:

1. Use each of the individual nine roll, pitch, and yaw as stored within the MSS-X header.

2. Use the roll, pitch, and yaw scene center values along with the roll, pitch, and yaw rates to define nine values with a sampling rate of 13.8030 seconds.

The roll, pitch, and yaw values are stored within the geometric model in units of radians. A GMT time is stored for each attitude coordinate and is given in units of seconds.

The MSS-X mirror polynomial model for the deviations from the ideal linear behavior is treated as first and second half-scan polynomials. The polynomial function is a fifth order polynomial where, currently, all coefficients above the second order are set to zero; the implementation was completed in this manner to be able to use the infrastructure already present with the Level 1 software for the TM and ETM+ mirror. To reuse previous IAS functionality, the historical second order MSS-X mirror polynomials were recast to be represented as Legendre polynomials. The polynomial coefficients are stored for each scan, with each profile normalized in length based on the scan line length read from the MSS-X calibration file. During the creation of the MSS-X geometric model, the scan mirror polynomials are used in conjunction with the scan line length for each scan, which is read from the LORp MSCD file, to create a set of Legendre coefficients that represent the angular motion of the scanning mirror for each scan. These coefficients are stored within the geometric model. See Figure 5.1.

$$\begin{aligned}Ffh &= a_0 + a_1p_i + a_2p_i^2 \\Fsh &= b_0 + b_1p_i + b_2p_i^2\end{aligned}$$

Where Ffh = First half across track mirror angle for LORp pixel p_i ,
Fsh = Second half across track mirror angle for LORp pixel p_i .

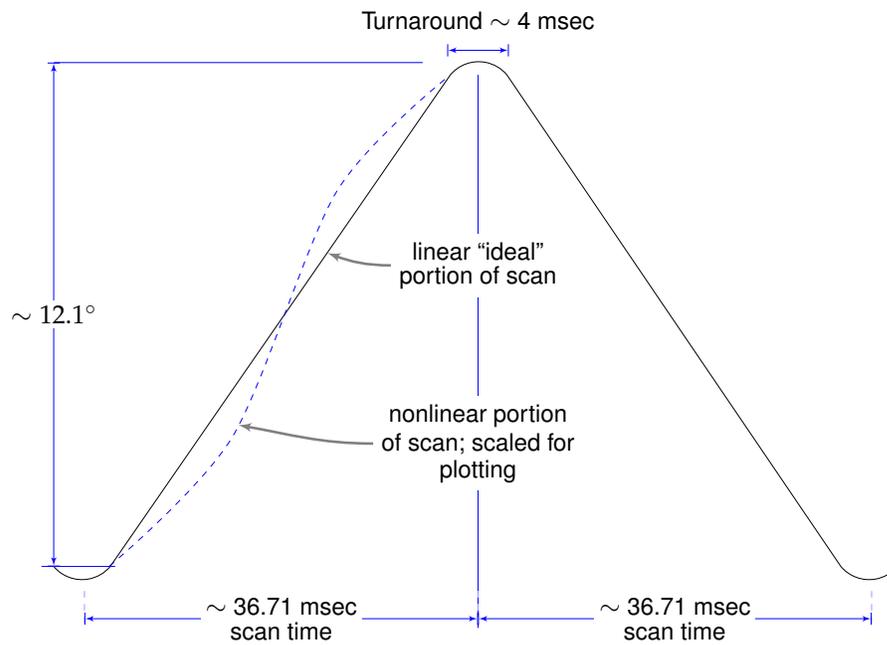


Figure 5.1. MSS Scanning Velocity

The fifth order Legendre coefficients are:

$$Lfh_0 = 0$$

$$Lfh_1 = a_0 \text{norm}$$

$$Lfh_2 = a_1 \text{norm}^2$$

$$Lfh_3 = a_2 \text{norm}^3$$

$$Lfh_4 = 0$$

$$Lfh_5 = 0$$

$$Lsh_0 = -b_0 - b_1 - b_2$$

$$Lsh_1 = (b_0 + 2b_1 + 3b_2) \text{norm}$$

$$Lsh_2 = -(b_1 + 3b_2) \text{norm}^2$$

$$Lsh_3 = b_2 \text{norm}^3$$

$$Lsh_4 = 0$$

$$Lsh_5 = 0$$

$T_s = \text{lendwell}$
 $\text{norm} = 1/T_s$
 $T_s = \text{Total scan time}$
 $\text{dwell} = \text{Detector dwell time}$
 $\text{len} = \text{Number of pixels per scan (line length read from MSS-X calibration file)}$

Across track look angles associated with the MSS-X scan mirror are then calculated from these mirror polynomials as:

if sample < len/2 **then**

z = ...

x-track angle = Ffh = Lfh₀ + Lfh₁z + Lfh₂z² + Lfh₃z³ + Lfh₄z⁴ + Lfh₅z⁵

else

z = ...

x-track angle = Fsh = Lsh₀ + Lsh₁z + Lsh₂z² + Lsh₃z³ + Lsh₄z⁴ + Lsh₅z⁵

end if

The algorithm flow is as follows:

1. Open support files.

- MSS-X header file

2. Read image center times.

The acquisition year is read from the exposure date listed within the MSS-X LORp header file (EXPOSURE_DATE):

Acquisition Year = 1900 + convert to digits (characters 8 and 9
 from the header exposure date)

Image time, in sample and line coordinates of the raw image space, are related to time by the spacecraft exposure time (SPACECRAFT_TIME_OF_EXP) listed in the MSS-X LORp header file. This field is a 16-integer ASCII string with the following format:

XXXDDDHMMSSMMM

Where

DDD = Day of Year **DOY** of acquisition

HH = Hour of day of acquisition

MM = Minutes

SS = Seconds

MMM = Milliseconds

The Seconds of Day (**SOD**) associated with these parameters are:

$$SOD = HH \times 3600 + MM \times 60 + SS + MMM/1000$$

The scene center time stored within the geometric model is then defined as the year, **DOY**, and **SOD**.

Model Scene Center Year = Acquisition year

Model Scene Center Day or Year = DDD

Model Scene Center Seconds of day = SOD

This scene center time is the time associated with the LORp scene center location (a line and sample location).

3. Create spacecraft attitude data.

a) Establish attitude times.

Attitude Time[0] = -13.80300

for count = 1 to 8 **do**

Attitude Time [count] = Attitude Time [count-1] + 3.45075

end for

for count = 0 to 8 **do**

Attitude Time [count] = Attitude Time [count] + Model Center Time

end for

Attitude processing can be handled in one of three methods:

i. Center average

Read the mean roll, pitch, and yaw values from the LORp header file. Store the roll, pitch, and yaw values within the geometric model:

for count = 0 to 8 **do**

Spacecraft roll [count] = Mean roll

Spacecraft pitch [count] = Mean pitch

Spacecraft yaw [count] = Mean yaw

end for

ii. Center the roll, pitch, yaw and roll, pitch, yaw rates.

Read the scene center roll, pitch, and yaw values from the LORp header file. (VEHICLE_ROLL_AT_IMAGE_CTR_TIME, VEHICLE_PITCH_AT_IMAGE_CTR_TIME, VEHICLE_YAW_AT_IMAGE_CTR_TIME)

Read roll, pitch, and yaw rates from the LORp header file (MEAN_ROLL, MEAN_PITCH, and MEAN_YAW).

Delta time = Attitude Time[count]

– Model Center Time SOD

Spacecraft roll[count] = Mean Roll + Roll Rate × delta time

Spacecraft pitch[count] = Mean Pitch + Pitch Rate × delta time

Spacecraft yaw[count] = Mean Yaw + Yaw Rate × delta time

iii. Individual values

Read roll, pitch, and yaw values from the LORp header (ROLL_VALUES, PITCH_VALUES, YAW_VALUES).

4. Read ephemeris data.

Ephemeris processing can be handled in one of two ways; either using the latitude, longitude, and altitude stored within the MSS-X LORp header file or using the North American Aerospace Defense Command ([NORAD](#)) two-line elements.

- Read ephemeris times from the LORp header file.

Read the GMT milliseconds (GMT_MILLISECONDS_OF_DAY) from the LORp header file. Convert the GMT times to seconds and store in the geometric model as ephemeris times.

for count = 0 to 11 **do**

Ephemeris Time [count] = GMT milliseconds / 1000.0

end for

5. Read ephemeris from LORp header file.

Read ephemeris (NADIR_LATITUDE, NADIR_LONGITUDE, NADIR_ALT-

ITUDE) from the LORp header file and convert to Cartesian coordinates. Calculate spacecraft velocity from the spacecraft ephemeris.

for count = 0 to 11 **do**

Convert from geodetic to Cartesian (x, y, z) coordinates (Section 3.5).

(latitude, longitude, altitude) $\Rightarrow (x, y, z)$

Spacecraft x [count] = x

Spacecraft y [count] = y

Spacecraft z [count] = z

end for

for $i = 0 \dots 10$ **do**

Compute the following for the spacecraft location (\vec{r}) and velocity (\vec{v}) based on the ephemeris times (t_i) :

$$\Delta t = t_{i+1} - t_i$$

$$v_x = (r_{x,i+1} - r_{x,i}) / \Delta t$$

$$v_y = (r_{y,i+1} - r_{y,i}) / \Delta t$$

$$v_z = (r_{z,i+1} - r_{z,i}) / \Delta t$$

end for

Ephemeris can also be generated using the NORAD Simplified General Perturbations (SGP4) orbital model. The code is derived from the Pascal version of the SGP4 written by [Kelso, 2007]. Under this scenario, the NORAD two-line elements for the spacecraft are read from an ASCII file, and the SGP4 model interpolates points between the given two-line elements to fill in the spacecraft x , y , and z parameters that would otherwise have been derived from the MSS-X header file.

6. Create along-scan mirror profile polynomials.

Create per scan mirror profiles using mirror polynomial coefficients, along with the line length and scan mirror times stored within the LORp MSCD file.

a) Read scan mirror parameters.

The MSS scan mirror polynomials are a set of second order polynomials that represent the scanning motion of the mirror in angular units with respect to time. The following represent the first and sec-

ond half scan mirror coefficients within the MSS-X geometric model.

$$\begin{aligned}a_0 &= \text{Mirr_First_Half_Coef}[0] \\a_1 &= \text{Mirr_First_Half_Coef}[1] \\a_2 &= \text{Mirr_First_Half_Coef}[2] \\b_0 &= \text{Mirr_Second_Half_Coef}[0] \\b_1 &= \text{Mirr_Second_Half_Coef}[1] \\b_2 &= \text{Mirr_Second_Half_Coef}[2]\end{aligned}$$

Given that the detector dwell time is the amount of time a detector has to collect photons, the coefficients previously listed can be recast into a set of Legendre polynomials as follows:

b) Read MSCD scan times and line lengths.

```
for count = 0 to 389 (390 scans in MSS-X LORp image file) do
  Scan time [count] = MSCD time [count]
  Scan Line Length [count] = MSCD line length [count]
end for
```

c) Create scan profile polynomials.

```
for count = 0 to 389 (390 scans in MSS-X LORp image file) do
  Find the normalized scan length
  Total scan time = Scan Line Length[count] × dwell time
  Norm = 1.0 / Total Scan Time
end for
```

Find the first and second half Legendre coefficients.

$$\begin{aligned}\text{Lfh}_0 &= 0 \\ \text{Lfh}_1 &= a_0 \times \text{norm} \\ \text{Lfh}_2 &= a_1 \times \text{norm}^2 \\ \text{Lfh}_3 &= a_2 \times \text{norm}^3 \\ \text{Lfh}_4 &= 0 \\ \text{Lfh}_5 &= 0\end{aligned}$$

$$\begin{aligned}Lsh_0 &= -b_0 - b_1 - b_2 \\Lsh_1 &= (b_0 + 2b_1 + 3b_2) \times \text{norm} \\Lsh_2 &= -(b_1 + 3b_2) \times \text{norm}^2 \\Lsh_3 &= b_2 \times \text{norm}^3 \\Lsh_4 &= 0 \\Lsh_5 &= 0\end{aligned}$$

5.5 Map Output Coordinates (UTM Projection) to Input L1R-H Pixel Locations

Inverse Model

The geometric model, along with spacecraft and instrument parameters, allows the mapping of geographic locations to input MSS-X L1R-H pixel locations. This process, typically called an inverse geometric model, is performed in an iterative process finding a time (through iteration) that matches a given set of spacecraft state parameters (position and attitude) and a scan mirror profile to a given geographic location. The spacecraft, scan mirror state, and time establishes an L1R-H pixel for a given geographic location.

The inverse model is based on establishing a target, spacecraft, and sensor model. The target vector is a dynamic model based on the Earth's location of a given output pixel. The dynamic model consists of two vectors, a positional vector and a velocity vector for that position. The spacecraft model is based on a position vector and an attitude, roll, pitch, and yaw state or set of angles. The sensor model is based on an along-scan angle whose profile is described by a fifth order polynomial (only three of which are non-zero). The inverse model initializes the target model based on an output pixel location, initializes a spacecraft time and state for that target model, and then calculates errors or delta times associated with these target, spacecraft, and sensor models. The process iterates and adjusts the spacecraft and sensor models based on these delta times. When the delta times reach a small enough state (a value based on an associated sub-pixel adjustment), the model found a solution. The relationship between these input locations to output pixels (or geographic locations) is stored within the resampling grid. During this process, the geographic location stays fixed and the input L1R-H pixel (through time) varies, thus the "inverse" model nomenclature.

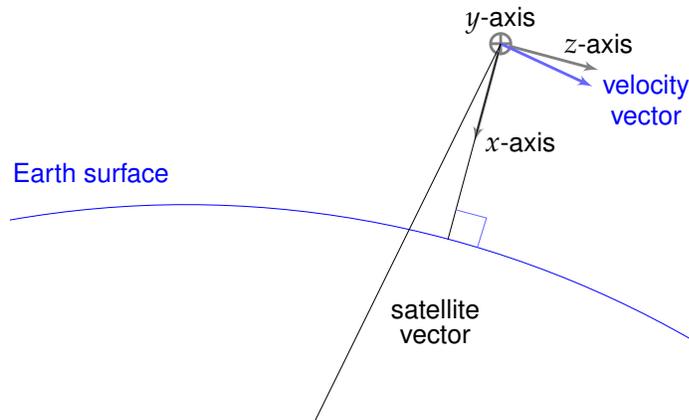


Figure 5.2. Position and Velocity Vector with Respect to Attitude xyz Axis

The steps are described in section 5.5.1.

5.5.1 Defining the MSS-X Processing Parameters

Image_Center_Line = L1R-H center line location for LORp header
 scene center time

Image_Center_Sample = L1R-H center sample location for LORp header
 scene center time

Image_Line_Rate = Time between successive scan sweeps

Image_Yaw = Yaw bias

Image_Sample_Slope = Linear scan time across one active scan sweep

Number_Channels = Number of detectors

5.5.2 Initialize Target Parameters

Given output image location y_{out}, x_{out} in UTM coordinates.
 Convert y_{out}, x_{out} to Geodetic latitude and longitude (see Section 3.5).
 Convert geodetic latitude and longitude to geocentric latitude (see Section 3.5).

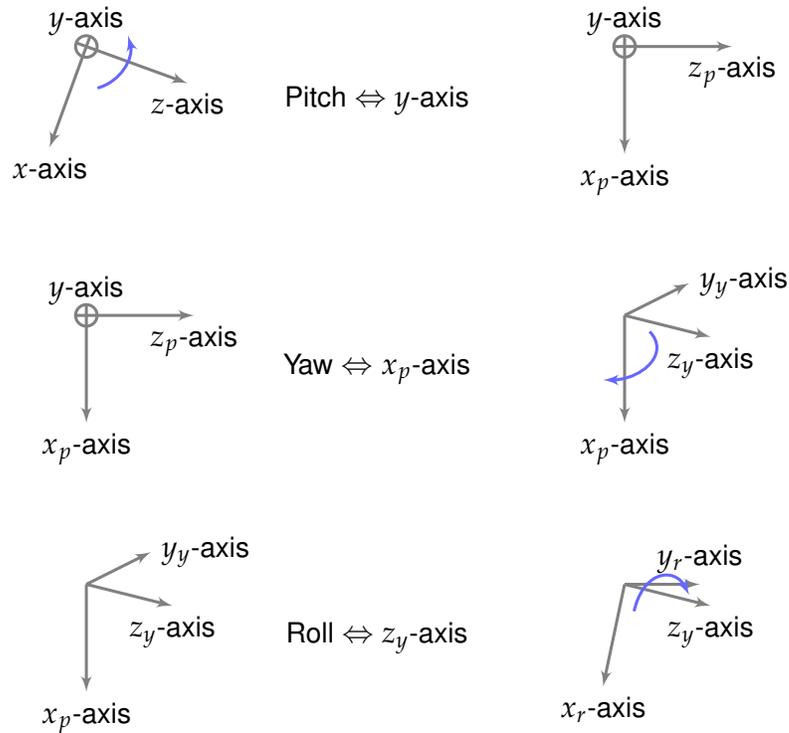


Figure 5.3. Apply Pitch-Yaw-Roll to Attitude xyz Axis

This geocentric coordinate and the Earth's angular rotation rate establish the target model state (position) and rate (velocity).

$$\text{Target position}_i = \text{tp}_i = \langle \text{target}_{x_i}, \text{target}_{y_i}, \text{target}_{z_i} \rangle$$

$$\text{Target velocity}_i = \text{tv}_i = \langle -\omega_e \text{target}_{y_i}, \omega_e \text{target}_{x_i}, 0 \rangle$$

Where ω_e is the Earth's inertial rotation rate.

5.5.3 Initialize Orbital Propagation

Initialize the spacecraft model

From the MSS-X header file and ephemeris calculate:

$$\text{Starting position} \Rightarrow \text{starting vector} = \langle s \rangle = \langle s_x, s_y, s_z \rangle$$

$$\text{Center position} \Rightarrow \text{center vector} = \langle c \rangle = \langle c_x, c_y, c_z \rangle$$

$$\text{End position} \Rightarrow \text{end vector} = \langle e \rangle = \langle e_x, e_y, e_z \rangle$$

Convert the satellite start, mid, and end positional vectors to geocentric latitude, longitude, and radius:

$$\begin{aligned}\langle \text{start}_x, \text{start}_y, \text{start}_z \rangle &\Rightarrow \langle \text{start_lat}, \text{start_lon}, \text{start_rad} \rangle \\ \langle \text{mid}_x, \text{mid}_y, \text{mid}_z \rangle &\Rightarrow \langle \text{mid_lat}, \text{mid_lon}, \text{mid_rad} \rangle \\ \langle \text{end}_x, \text{end}_y, \text{end}_z \rangle &\Rightarrow \langle \text{end_lat}, \text{end_lon}, \text{end_rad} \rangle\end{aligned}$$

Set the times for start, mid, and end vectors:

Start spacecraft positional time: $\text{time_sp_start} = \text{ephem_time0} - \text{ephem_time_mid}$

Mid spacecraft positional time: $\text{time_sp_mid} = 0$

End spacecraft positional time: $\text{time_sp_end} = \text{ephem_time_end} - \text{ephem_time_mid}$

Note the ephemeris center time, read from MSS-X header file, is considered as $\text{time} = 0$.

Calculate change in orbital radius for start-to-mid and mid-to-end positions:

$$\begin{aligned}r_{\text{rate}}_{\text{m-to-s}} &= \frac{\text{mid_rad} - \text{start_rad}}{\text{time_sp_mid} - \text{time_sp_start}} \\ r_{\text{rate}}_{\text{e-to-m}} &= \frac{\text{end_rad} - \text{mid_rad}}{\text{time_sp_end} - \text{time_sp_mid}}\end{aligned}$$

Calculate polynomial for change in orbital radius:

$$\begin{aligned}r_{\text{acc_time}}_{\text{s-to-m}} &= \text{time_sp_start}^2 - \text{time_sp_mid}^2 \\ r_{\text{acc_time}}_{\text{m-to-e}} &= \text{time_sp_end}^2 - \text{time_sp_mid}^2 \\ r_{\text{coef}}_0 &= \frac{r_{\text{rate}}_{\text{e-to-m}} + r_{\text{rate}}_{\text{m-to-s}}}{\frac{r_{\text{acc_time}}_{\text{e-to-m}}}{r_{\text{rate}}_{\text{e-to-m}}} - \frac{r_{\text{acc_time}}_{\text{s-to-m}}}{r_{\text{rate}}_{\text{s-to-m}}}} \\ r_{\text{coef}}_1 &= \frac{\text{start_rad} - r_{\text{coef}}_0 \times r_{\text{acc_time}}_{\text{s-to-m}} - \text{mid_rad}}{r_{\text{rate}}_{\text{s-to-m}}} \\ r_{\text{coef}}_2 &= \text{mid_rad}\end{aligned}$$

Adjust center longitude for the Earth's rotational effect:

$$\begin{aligned}\text{start_lon} &= \text{start_lon} + \omega_e \times \text{time_sp_start} \\ \text{mid_lon} &= \text{mid_lon} + \omega_e \times \text{time_sp_mid} \\ \text{end_lon} &= \text{end_lon} + \omega_e \times \text{time_sp_end}\end{aligned}$$

Note that adjusting the longitude based on established spacecraft accounts for skew effects relative to the scene center.

Find the angular change between start-to-mid and mid-to-end vectors:

$$\text{angle}_{\text{s-to-m}} = \frac{\text{start_vector} \times \text{mid_vector}}{\|\text{start_vector}\| \|\text{mid_vector}\|}$$

$$\text{angle}_{\text{m-to-s}} = \frac{\text{mid_vector} \times \text{end_vector}}{\|\text{mid_vector}\| \|\text{end_vector}\|}$$

Find the angular mean change in orbit between start-to-mid and mid-to-end vectors:

$$\text{mean_orbit}_{\text{s-to-m}} = \frac{\text{angle}_{\text{s-to-m}}}{\text{time_sp_mid} - \text{time_sp_start}}$$

$$\text{mean_orbit}_{\text{m-to-s}} = \frac{\text{angle}_{\text{m-to-s}}}{\text{time_sp_end} - \text{time_sp_mid}}$$

Find the center mean velocity for a given center and end positional vectors:

$$V_c = \mu / \|c\|$$

Where μ = Earth's gravitational parameter.

Calculate the angle between center and end vector:

$$\theta = \cos^{-1} \frac{ce}{\|c\| \|e\|}$$

$$C_1 = \frac{V_c}{\|c\|} \frac{1}{\tan \theta}$$

$$C_2 = \frac{V_c}{\|e\|} \frac{1}{\sin \theta}$$

$$\text{velocity}_x = C_1 c_x + C_2 e_x$$

$$\text{velocity}_y = C_1 c_y + C_2 e_y$$

$$\text{velocity}_z = C_1 c_z + C_2 e_z$$

5.5.4 Initialize Times

Target time = 0 (time within the image or time from scene center time)

Delta time = 1000

Scene time = Model Scene Center Time + target time

5.5.5 Iterate Solution

Break it iteration ≥ 10 or delta time < 0.001

5.5.5.1 Calculate Scene Time

Scene time = Model Scene Center Time + Target Time

5.5.5.2 Calculate Spacecraft Roll, Pitch, Yaw

For a given scene time, use Lagrange interpolation and the spacecraft roll, pitch, yaw stored within the geometric model to find the corresponding roll, pitch, and yaw.

5.5.5.3 Add Yaw Bias

Yaw = yaw + Yaw Bias

From past processing of MSS-X data, using the same type of inverse model previously discussed, a noticeable yaw bias was present within the systematic imagery. To account for this bias, a constant yaw value is applied within the geometric model during processing. This value is added to the yaw value calculated within the MSS-X attitude data derived from the MSS-X header file.

5.5.5.4 Calculate Attitude Perturbed Satellite State

To define a nominal, non-perturbed coordinate system, create an orthogonal set of axes with the x -axis perpendicular to the surface of the Earth in the direction of the satellite and a z -axis in the velocity direction of the satellite. The cross product of these two vectors defines the third axis or the y -axis (Figure 5.2). To define an attitude perturbed coordinate system, rotate this coordinate system by the pitch angle y -axis; rotate this resulting coordinate system around the new x -axis by the yaw angle from, and then rotate this coordinate system around the new z -axis by the roll angle (see Section 5.1.3, Figure 5.2, and Figure 5.3).

1. Determine the non-perturbed satellite coordinate system.
Determine the attitude non-perturbed satellite coordinate system. Calcu-

late the satellite coordinate axis:

$$\begin{aligned}\text{satpos} &= \text{satellite positional vector} \\ \text{satvel} &= \text{satellite velocity vector} \\ \text{vvec} &= \langle \text{satvel}_x, \text{satvel}_y, \text{satvel}_z \rangle\end{aligned}$$

Convert the satellite position to geodetic coordinates. See Section 3.5
Convert the satellite geodetic coordinates to spherical coordinates.

$$\begin{aligned}x\text{vec}_x &= -h \cos(\text{lat}) \cos(\text{lon}) \\ x\text{vec}_y &= -h \cos(\text{lat}) \sin(\text{lon}) \\ x\text{vec}_z &= -h \sin(\text{latd})\end{aligned}$$

Where lon = Geodetic longitude, latd= Geodetic latitude, h = satellite altitude, $y\text{vec} = \overrightarrow{v\text{vec}} \times \overrightarrow{x\text{vec}}$, $z\text{vec} = \overrightarrow{x\text{vec}} \times \overrightarrow{y\text{vec}}$ (\times = Cross product of two vectors).

2. Calculate the perturbed coordinate system.
 - a) Rotate the coordinate system from step 1 around the y -axis by pitch angle from Section 5.5.5.2. See Section 5.1.3.
 - b) Rotate the coordinate system from step 1 around the x -axis by yaw angle from Section 5.5.5.2. See Section 5.1.3.
 - c) Rotate the coordinate system from step 1 around the z -axis by roll angle from Section 5.5.5.2. See Section 5.1.3.

A new set of coordinates are now defined as:

$$\begin{aligned}\langle \text{perturbed } x\text{-axis} \rangle &= \langle a_{xx}, a_{xy}, a_{xz} \rangle \\ \langle \text{perturbed } y\text{-axis} \rangle &= \langle a_{yx}, a_{yy}, a_{yz} \rangle \\ \langle \text{perturbed } z\text{-axis} \rangle &= \langle a_{zx}, a_{zy}, a_{zz} \rangle\end{aligned}$$

5.5.5.5 Move Target to Target Time $\langle \text{target vector} \rangle = \langle t_x, t_y, t_z \rangle$

Determine the angular change due to the Earth rotation:

$$\begin{aligned}\delta &= \omega_e \text{target time} \\ \text{cdl} &= \cos \delta \\ \text{sdl} &= \sin \delta \\ \text{wsdl} &= \omega_e \text{sdl} \\ \text{sdl per } \omega_e &= \text{sdl} / \omega_e \\ \text{target vector}_x &= \text{tpxi} \times \text{cdl} + \text{tvxi} \times \text{sdl per } \omega_e \\ \text{target vector}_y &= \text{tpyi} \times \text{cdl} + \text{tvyi} \times \text{sdl per } \omega_e \\ \text{target vector}_z &= \text{tpzi}\end{aligned}$$

5.5.5.6 Move Satellite to Target Time

Calculate the satellite position for scene time:

$$\begin{aligned}\langle \text{satellite position} \rangle &= \langle s_x, s_y, s_z \rangle \\ \langle \text{satellite velocity} \rangle &= \langle s_{xv}, s_{yv}, s_{zv} \rangle\end{aligned}$$

See Section 3.5.

Calculate the change in orbital radius for start-to-mid and mid-to-end positions:

$$\begin{aligned}r_{\text{rate}}_{\text{m-to-s}} &= \frac{\text{mid_rad} - \text{start_rad}}{\text{time_sp_mid} - \text{time_sp_start}} \\ r_{\text{rate}}_{\text{e-to-m}} &= \frac{\text{end_rad} - \text{mid_rad}}{\text{time_sp_end} - \text{time_sp_mid}} \\ r_{\text{coef}}_0 &= \frac{r_{\text{rate}}_{\text{e-to-m}} + r_{\text{rate}}_{\text{m-to-s}}}{\frac{r_{\text{acc_time}}_{\text{e-to-m}}}{r_{\text{rate}}_{\text{e-to-m}}} - \frac{r_{\text{acc_time}}_{\text{s-to-m}}}{r_{\text{rate}}_{\text{s-to-m}}}} \\ r_{\text{coef}}_1 &= \frac{\text{start_rad} - r_{\text{coef}}_0 \times r_{\text{acc_time}}_{\text{s-to-m}} - \text{mid_rad}}{r_{\text{rate}}_{\text{s-to-m}}} \\ r_{\text{coef}}_2 &= \text{mid_rad}\end{aligned}$$

Find the angular change between the start-to-mid and mid-to-end vectors:

$$\text{angle}_{\text{s-to-m}} = \frac{\text{start_vector} \cdot \text{mid_vector}}{\|\text{start_vector}\| \|\text{mid_vector}\|}$$
$$\text{angle}_{\text{m-to-e}} = \frac{\text{mid_vector} \cdot \text{end_vector}}{\|\text{mid_vector}\| \|\text{end_vector}\|}$$

Find the angular mean change in orbit between the start-to-mid and mid-to-end vectors:

$$\text{mean_orbit}_{\text{s-to-m}} = \frac{\text{angle}_{\text{s-to-m}}}{\text{time_sp_mid} - \text{time_sp_start}}$$
$$\text{mean_orbit}_{\text{m-to-e}} = \frac{\text{angle}_{\text{m-to-e}}}{\text{time_sp_end} - \text{time_sp_mid}}$$

Calculate the relative time:

$$t_r = \text{Scene Time} - \text{ephemeris_center_time}$$

Calculate the current radius:

$$r_c = \text{rcoef}_2 + \text{rcoef}_1 \times t_r + \text{rcoef}_0 \times t_r^2$$

if $t_r < 0$ **then**

$$d_m = \text{mean_orbit}_{\text{s-to-m}} \times t_r$$

else

$$d_m = \text{mean_orbit}_{\text{m-to-s}} \times t_r$$

end if

Calculate the satellite state vector for the current target time (see Sec-

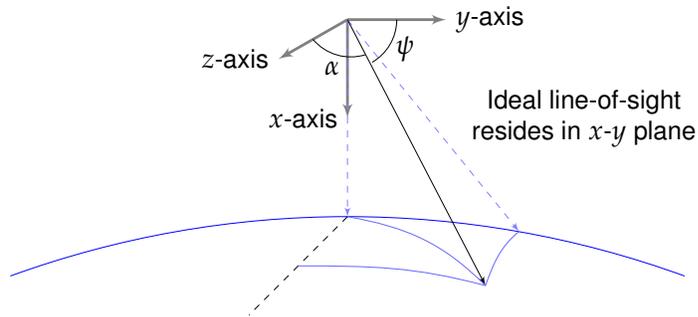


Figure 5.4. LOS Vector Within xyz Coordinate Axis

tion 3.5):

$$V_r = \sqrt{\frac{\mu}{r_c}}$$

$$C_3 = \frac{r_c}{\|c\|} \cos d_m$$

$$C_4 = \frac{r_c}{\|\text{velocity}\|} \sin d_m$$

$$C_5 = \frac{V_r}{\|\text{velocity}\|} \cos d_m$$

$$C_6 = \frac{V_r}{\|c\|} \sin d_m$$

$$\text{satellite position}_x = C_3 c_x + \text{velocity}_x \times C_4$$

$$\text{satellite position}_y = C_3 c_y + \text{velocity}_y \times C_4$$

$$\text{satellite position}_z = C_3 c_z + \text{velocity}_z \times C_4$$

$$\text{satellite velocity}_x = C_5 \text{velocity}_x - C_6 c_x$$

$$\text{satellite velocity}_y = C_5 \text{velocity}_y - C_6 c_y$$

$$\text{satellite velocity}_z = C_5 \text{velocity}_z - C_6 c_z$$

5.5.5.7 Calculate LOS Vector for Target Time

$$\text{LOS} = \langle \text{target vector} \rangle - \langle \text{satellite position} \rangle$$

5.5.5.8 Calculate Angular Errors for LOS Vector

The LOS vector should reside within the x - y plane of the perturbed coordinate system. Angular errors are calculated for the LOS and transformed into delta time adjustments (see Figure 5.4, Figure 5.5, and Figure 5.6).

$$\alpha = \cos^{-1} \frac{\overrightarrow{\text{perturbed } z \text{ axis}} \times \overrightarrow{\text{LOS}}}{\|\overrightarrow{\text{LOS}}\| \|\overrightarrow{\text{perturbed } z \text{ axis}}\|}$$
$$\psi = \cos^{-1} \frac{\overrightarrow{\text{perturbed } y \text{ axis}} \times \overrightarrow{\text{LOS}}}{\|\overrightarrow{\text{LOS}}\| \|\overrightarrow{\text{perturbed } y \text{ axis}}\|}$$
$$q = \cos^{-1}(\cos \alpha \sin \psi)$$
$$\eta = \frac{\pi}{2} - q$$
$$z = \sin^{-1} \left(\frac{q}{R_{\text{earth}}} \|\text{satpos}\| \right)$$
$$\Delta = z - \eta$$

5.5.5.9 Calculate Delta Time

if delta time ≤ 0 **then**

$$\text{delta time} = \frac{\Delta}{\text{mean_orbit}_{\text{s-to-m}}}$$

else

$$\text{delta time} = \frac{\Delta}{\text{mean_orbit}_{\text{m-to-e}}}$$

end if

if delta time > 0.001 **then**

Calculate a new target time and return to Section 5.5.5.

$$\text{target time} = \text{target time} + \text{delta time}$$

else

Calculate the input line and sample location (Sections 5.5.5.10 and 5.5.5.11).

end if

5.5.5.10 Calculate Sample Location

Determine the linear sample adjustment:

$$\lambda = \frac{\pi}{2} - \psi$$

$$\text{linear sample} = \text{center sample} + \lambda \text{ samples per radian}$$

Calculate the mirror polynomial normalized sample location:

$$z = \frac{\text{linear sample}}{\text{scan line length}}$$

if $z \leq \text{center sample}$ **then**

$$\text{nonlinear sample} = \text{Lfh}_0 + \text{Lfh}_1z + \text{Lfh}_2z^2 + \text{Lfh}_3z^3 + \text{Lfh}_4z^4 + \text{Lfh}_5z^5$$

else

$$z = 1 - z$$

$$\text{nonlinear sample} = \text{Lsh}_0 + \text{Lsh}_1z + \text{Lsh}_2z^2 + \text{Lsh}_3z^3 + \text{Lsh}_4z^4 + \text{Lsh}_5z^5$$

end if

Determine the overall sample location:

$$\text{sample} = \text{linear sample} - \text{nonlinear sample}$$

5.5.5.11 Calculate Line Location

$$\text{line} = \text{center line} + \text{along track sample rate (lines-per-second)} \times \text{scene time}$$

5.6 Create Hybrid Imagery – Along Scan Line Adjustments

Along scan detector offsets and scan length adjustments are completed using a one-dimensional resampling in the sample direction of the L1R imagery. Cubic convolution is used as the resampling method. This hybrid image file will be used, along with the geometric resampling grid, to create a geometrically corrected MSS-X data set. To improve efficiency, the cubic convolution weights are calculated as an LUT set of weights with each set representing a 1/32 of pixel shift.

5.6.1 Define MSS-X Detector Offsets

Detector offsets [0..number of detectors]

5.6.2 Read Scan Line Length from L0R MSCD

From MSCD, read \Rightarrow measured line length[scan number]

5.6.3 Build a Set of Cubic Convolution Resampling Weights

Resampling weights are calculated from the cubic convolution function:

$$f(x) = \begin{cases} (\alpha + 2)|x|^3 - (\alpha + 3)|x|^2 + 1 & 0 \leq |x| < 1, \\ \alpha|x|^3 - 5\alpha|x|^2 + 8\alpha|x| - 4\alpha & 1 \leq |x| < 2, \\ 0 & |x| > 2 \end{cases}$$

A set of 33 weights are calculated, representing sample adjustments in 1/32 of pixel increments bounded from [0.0 ... 1.0].

For n=0 to 32

$$\begin{aligned} x &= 1 - \frac{1}{32}n \\ \text{CCW}_{n,0} &= f(-1 - x) \\ \text{CCW}_{n,1} &= f(-x) \\ \text{CCW}_{n,2} &= f(1 - x) \\ \text{CCW}_{n,3} &= f(2 - x) \end{aligned}$$

5.6.4 Resample L1R Imagery

For line=0, number lines - 1

5.6.4.1 Determine Nominal Line Adjustment

$$\begin{aligned} \text{scan number} &= \frac{\text{line}}{\text{lines per scan}} \\ \text{line length adjustment} &= \frac{\text{measured line length}[\text{scan number}]}{\text{nominal scan length}} \end{aligned}$$

For sample=0, number samples - 1

5.6.4.2 Calculate Total Along Line Adjustment

total adjustment = line length adjustment × sample
+ detector offset[detector number]

5.6.4.3 Determine Resampling Weights Index

Sub-pixel = sample - [sample]
index = integer × 32 × $\left(\frac{1}{64} + \text{subpixel}\right)$

5.6.4.4 Apply Cubic Convolution Weights

sum = 0
for count = 0 to 3 **do**
 sum = sum + Level0r [line][sample+count-1] × ccw[index][count]
end for
Hybrid0r[line][sample] = sum

6 Overview of MSS-A and MSS-P Processing

Two levels of partially processed MSS data reside in the Earth Resource Observation and Science (EROS) archives: MSS-A and MSS-P products. MSS-A products have radiometric corrections applied; geometric corrections for two different map projections have been supplied but not applied. These corrections are described in a set of geometric resampling grids; information about the satellites' location and attitude during imaging were used in calculating these grids, but the satellite location information and scan-based time codes were not saved and stored in the MSS-A archive files. MSS-P products are derived from the MSS-A products, and have the geometric corrections applied. MSS-P products are therefore already resampled to one of two path-oriented map projections.¹

An effort was made to store a given Landsat MSS scene in as raw a format as possible; however, some image acquisitions only exist in the already-processed MSS-P format. While limited to a subset of Landsats 2-3 MSS acquisitions, these images were large enough in number to justify a software implementation to handle this partially processed, already map-projected data. MSS-A scenes are more numerous in the archive. Many Landsat 2-3 scenes are available only in this format and all current U.S. holdings of Landsat 4-5 MSS data are in MSS-A format.

¹Two map projection resampling grids are stored with MSS-A products. Landsats 2-3 MSS-A products contain grids for creating products in a path-oriented Hotine Oblique Mercator (HOM) projection, and another grid for creating products in UTM / Universal Polar Stereographic (UPS) (UPS being used above 84N and below 80S). Landsats 4-5 MSS-A products contain grids for creating products in a path-oriented SOM projection and another grid for creating products in UTM / UPS.

6.1 MSS-P Geometric Processing

MSS-P data are radiometrically corrected and processed to a path-oriented, systematic product. Control points may have refined the geometric accuracy of the scene, but this is not always the case. The majority of these Landsats 2-3 MSS scenes were processed to the [HOM](#) map projection, but a handful of scenes exist in the [UTM](#) and [UPS](#) map projections. These algorithms describe the processing required to work with the HOM projection, UTM, and/or [UPS](#) implementations.

The goal of the following algorithm is to create a north-up systematic product in the UTM projection for use in the pyramid-based hierarchical matching algorithms. The creation of the north-up systematic product is a straight-forward reprojection after understanding the HOM projection, the projection parameters used for a given WRS-1 path / row, and has knowledge of the projection frame of reference. To update this transformation with ground truth ([I2I](#) tie points), the preferred method generates a geometric mapping grid and then uses a general-purpose image warp / resampling tool. The MSSRefine process retains the systematic mapping grid that maps from path-oriented HOM to north-up UTM.

6.2 HOM Projection

The HOM, as used in Landsat 2-3 MSS-P and MSS-A products, is described in [\[Snyder, 1987\]](#) (in the Oblique Mercator chapter text (pp. 68–75), as well as the “numerical examples” Section (pp. 276–278) for both Oblique Mercator and in the SOM sections), and in Appendix D of [\[USGS, 1979\]](#). Neither of these references contains complete parameter information; however some insights to the parameters used were obtained in [\[Claire, 1973\]](#), as well as experimentally.

A set of tic-marks contained in the MSS-A and MSS-P ancillary files describe the projection image frame and the pixel-grid location within the projection. The set of tic-marks is given in the HOM projection u,v skew coordinates. Common projection packages (GCTP, Proj4) use an x,y projection coordinate system—which, in the case of HOM, is a rotation of coordinates.

The HOM projection requires the following parameters:

1. The ellipsoid upon which the projection is based
2. The latitude of the projection center
3. The longitude of the projection center

4. The azimuth at the projection center (of the u,v skew coordinates to north)

The HOM projection is divided into five zones based on the latitude of the scene center point. Based on the path and the latitude of the scene center, calculate the HOM zone number and set projection parameters accordingly. This information is based on table AD-2 in the HOM documentation in [USGS, 1979], Appendix D. Oblique Mercator Projections. That table specifies zones based on row numbers and is ambiguous in that the row number breaks are included in both ranges; that is, zone 1 is from row 1 – 25, zone 2 is from 25 – 44, etc. Given that information, which zone is row 25 in? A similar problem occurs for the other zones as well. This issue has been determined experimentally; tic-mark coordinates and the registration accuracy are off if the incorrect HOM zone is used.

```
int HOMzone;
if (row <= 25) HOMzone = 1;
else if (row <= 44) HOMzone = 2;
else if (row <= 76) HOMzone = 3;
else if (row <= 94) HOMzone = 4;
else if (row <= 122) HOMzone = 5;
else HOMzone = -1; // This is an ascending pass...
```

From table AD-2 from [USGS, 1979], Appendix D. Oblique Mercator Projections, the centerLon computations are derived from the table on page AD-6, based on those given for path 31.

```
double HOMcenterLat;
double HOMcenterLon;
double HOMangleDeg;

if (HOMzone == 1) {
    HOMcenterLat = 67.0983333333;
    HOMcenterLon = -(((path - 31) * (360.0/251.0)) + 81.9700);
    HOMangleDeg = 24.7708181;
}
if (HOMzone == 2) {
    HOMcenterLat = 36.0;
    HOMcenterLon = -(((path - 31) * (360.0/251.0)) + 99.2750);
    HOMangleDeg = 14.3394883;
}
```

```

if (HOMzone == 3) {
    HOMcenterLat = 0.000277777; // this is 1"N
    HOMcenterLon = -(((path - 31) * (360.0/251.0)) + 108.5069);
    HOMangleDeg = 13.001443;
}
if (HOMzone == 4) {
    HOMcenterLat = -36.0;
    HOMcenterLon = -(((path - 31) * (360.0/251.0)) + 117.7388);
    HOMangleDeg = 14.3394883;
}
if (HOMzone == 5) {
    HOMcenterLat = -67.0983333333;
    HOMcenterLon = -(((path - 31) * (360.0/251.0)) + 135.0438);
    HOMangleDeg = 24.7708181;
}
if (HOMcenterLon < -180.0) HOMcenterLon += 360.0;

```

Rotate the u,v skew coordinates into HOM x,y coordinates:

$$\begin{aligned}
 x &= v \cos \text{HOMangle} + u \sin \text{HOMangle} \\
 y &= u \cos \text{HOMangle} - v \sin \text{HOMangle}
 \end{aligned}$$

6.3 Determine Image Frame

A theoretical set of horizontal and vertical tic-marks describe the location of image pixels cast into the HOM projection space. These tic-mark projection locations are given in the MSS-A and MSS-P annotation files. The actual tic-marks were to be added during (historical) film processing; their locations are in the documentation of an old processing routine PixGeo (author and source unknown). Much of that documentation is duplicated in this document.

Ideally, the tic-marks are arranged in the image coordinate system, as shown in Figure 6.1.

The number of tic-marks varies. Usually three or four in-common coordinates per pair of sides (top / bottom, right / left) exist; however, in some instances only two in-common coordinates exist per pair of sides. Scenes processed early in the mission had only projection-based tic-marks, whereas later scenes often had a middle tic in latitude and longitude coordinates. These variances make parsing the annotation files a complex task.

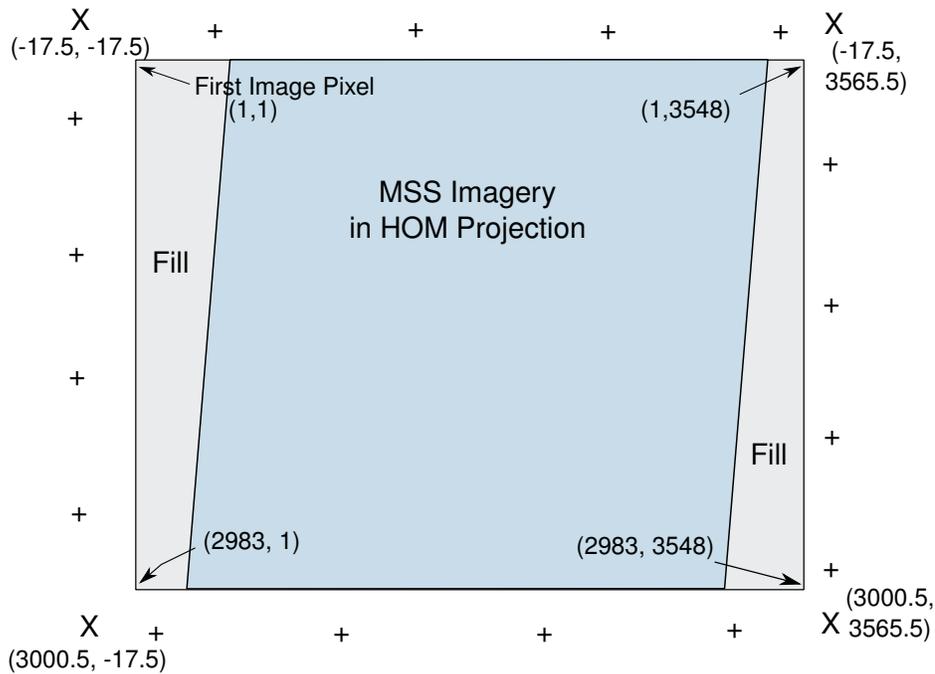


Figure 6.1. Arrangement of Tic-Mark locations on the Image Pixel Grid

As an example, the following listing is a formatted dump of an annotation file from an MSS-P scene cast on the HOM projection:

```
mssP
rec1: 22OCT80 C N34-33/W115-39 D042-036 SUN EL37 A145 cor:G 3 proj:H
res:C P N E-22100-17374-4
Top tics: (324:V040 ) (1202: V045) (2081: V050) (2959: V055)
Left tics: (130:U490 ) (1008: U485) (1887: U480) (2765: U475)
Right tics: (320:U490 ) (1199: U485) (2077: U480) (2956: U475)
Bottom tics: (163:V040 ) (1042: V045) (1920: V050) (2799: V055)
```

This information maps into the tic-mark geometry space, as shown in Figure 6.2. For the top and bottom tics, the line coordinates are set at -17.5 (top) and 3000.5 (bottom), respectively. The sample coordinates are obtained from the annotation file. Likewise, the sample coordinates for the left and right tics are set at -17.5 (left) and 3565.5 (right), respectively.

At this point, the tic location information is in line and the sample coordinates are along the outside of the image, but have only partial projection coordinates

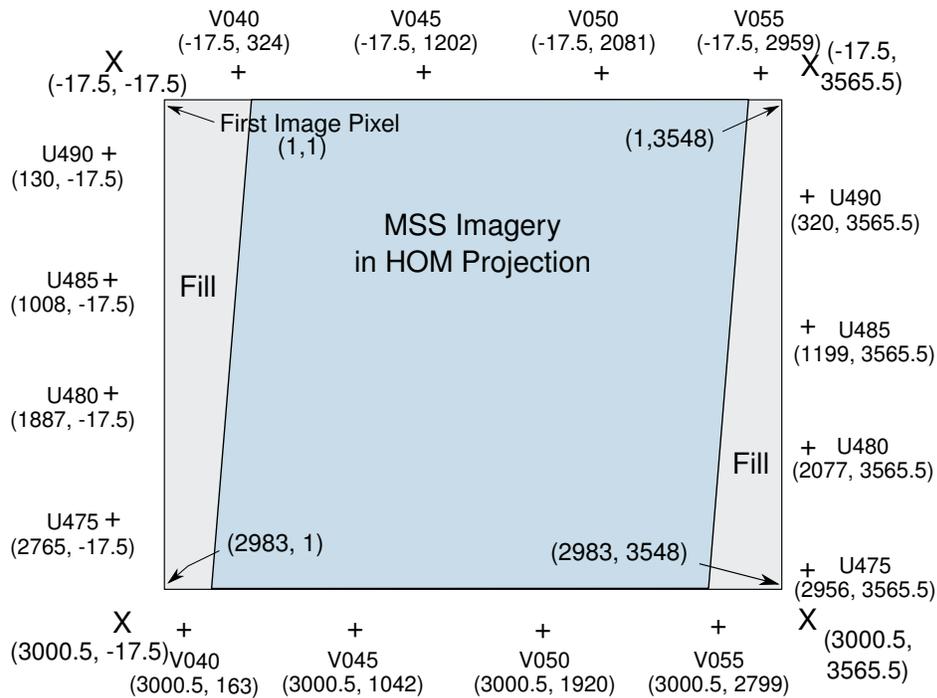


Figure 6.2. Tic-Mark Location Information on the Image Grid

for a given tic. (For example, the upper left tic-mark in Figure 6.2 is at image line -17.5, sample 324 with a v coordinate of 040 and an unknown u coordinate.) Likewise, the v coordinates are unknown on the left and right sides. v and u coordinates are both multiples of 10,000. In addition, an offset of 500,000 meters is applied to v coordinates and an offset of 5,000,000 applied to the u coordinates. This is similar to false northings / eastings applied in other projections to ensure positive coordinate mapping systems. To obtain u, v coordinate pairs, calculate the line and sample the locations of the crossing points of the tic-mark pairs, as shown in Figure 6.3.

In Figure 6.3, the 16 line / sample, U/V coordinate pairs at the HOM projection grid line intersect to create a first-order mapping polynomial to map between image pixel coordinate space and HOM projection space. This transformation, along with output projection space information, creates a geometric mapping grid for application with a generic resampler.

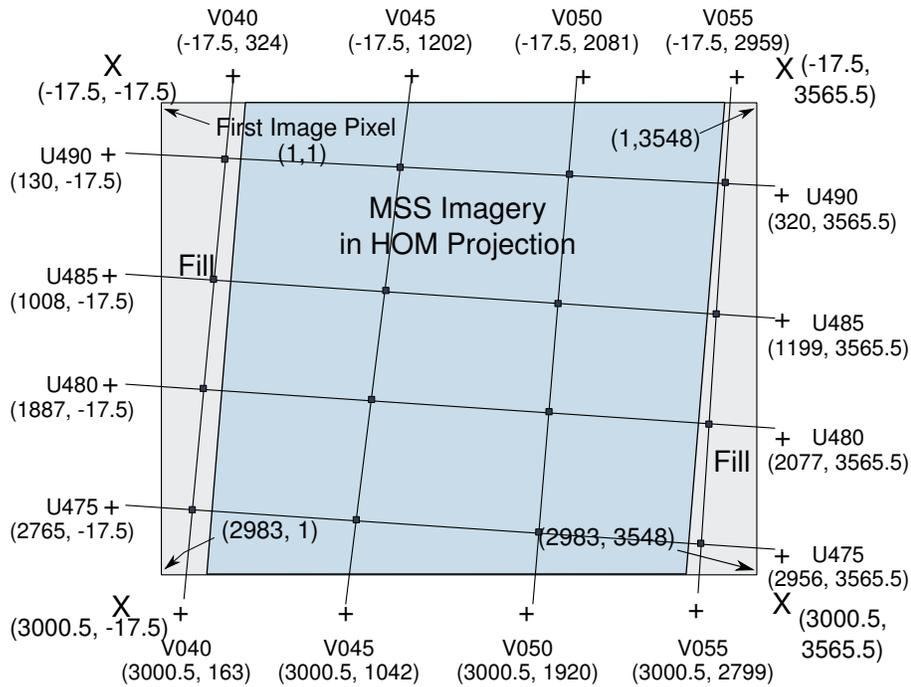


Figure 6.3. Image Space Grid with Tic-Mark Intersections Displayed

6.4 MSS-A Geometric Processing

MSS-A data contains two sets of annotation data and two sets of grids in the ancillary file. The first set of annotation data and the first set of mapping grids describe the mapping of the MSS-A data to an MSS-P path-oriented geometry in a UTM projection. The second set of annotation and mapping grids in the ancillary file describes the mapping to MSS-P geometry in the HOM (Landsats 2-3) or the SOM (Landsats 4-5). This second set of mappings was historically used to create a path-oriented HOM or SOM projected product. The IAS / Level 1 Product Generation System (LPGS) system does not create these products but uses the mapping grids and associated annotation data to create a north-up systematic product for use in registration refinement. Use of the HOM / SOM in this regard has historical precedence, and it produces an intermediate mapping space that is more aligned with the original scan lines, which is desirable for along-scan terrain correction algorithms applied in subsequent processing.

Reading the tic-mark information from the annotation file is similar to the

process described previously for the Landsats 2-3 MSS-P products. In the case of MSS-A, two annotation records exist; the second record is used with the HOM / SOM projection. Refer to the MSS-P section (above) for the algorithm to locate and frame the scene. The output space frame's line / sample coordinates align with the line / sample coordinates in the horizontal and vertical resampling grids contained with the MSS-A products.

```
mssA
rec1: 15SEP92 C N40-20/W118-14 D042-032 SUN EL44 A138 cor:R proj:U
res: P N E-53120-17550-2
  Top tics:   (651: E035) (1456:W118-15) (1545:E040 ) (2440: E045)
              (3335: E050)
  Left tics:  (353: N455) (1248: N450) (1991:N040-15) (2144:N445 )
  Right tics: (525: N450) (1320:N040-15) (1420:N445 ) (2316: N440)
  Bottom tics: (363: E030) (1258: E035) (2021:W118-15) (2153:E040 )
(3048: E045)
rec2: 15SEP92 C N40-20/W118-14 D042-032 SUN EL44 A138 cor:R proj:S res:
P N E-53120-17550-2
  Top tics:   (512: V+0050) (1389: V+0055) (1456:W118-15) (2268: V+0060)
              (3146: V+0065)
  Left tics:  (873: U+1555) (1752: U+1560) (1991:N040-15) (2630: U+1565)
  Right tics: (687: U+1555) (1320:N040-15) (1565:U+1560 ) (2444: U+1565)
  Bottom tics: (669: V+0050) (1546: V+0055) (2020:W118-15) (2424: V+0060)
(3303: V+0065)
```

For Landsat 2-3 MSS-A products, parse the second record in the annotation file the same way as the annotation record was parsed for the case of MSS-P products. The HOM has the same parameters as the MSS-P application.

For Landsats 4-5 MSS-A products, use the SOM projection using the following algorithm:

1. Parse the tic coordinate information (as before)
2. Multiply each SOM u,v coordinate by 10,000
3. Assign to SOM x,y coordinates as follows:
 - a) $x = v$
 - b) $y = -u$
4. Perform the SOM map projection transformation to latitude, longitude

The SOM tics should look similar to the following x,y coordinates:

500000.0 -15550000.0
550000.0 -15550000.0
600000.0 -15550000.0
650000.0 -15550000.0
500000.0 -15600000.0
550000.0 -15600000.0
600000.0 -15600000.0
650000.0 -15600000.0
500000.0 -15650000.0
550000.0 -15650000.0
600000.0 -15650000.0
650000.0 -15650000.0

This results in longitude, latitude coordinates similar to the following:

-118.953844	40.780014
-118.366801	40.721788
-117.780787	40.660593
-117.195887	40.596441
-119.027989	40.333354
-118.444765	40.275484
-117.862544	40.214689
-117.281408	40.150984
-119.101214	39.886613
-118.521725	39.829093
-117.943214	39.768693
-117.365762	39.705428

6.5 Assumptions Made in the Algorithm Development

The tic-mark information provided in the annotation record is the only location information available. Several assumptions were made because the original algorithms were either not adequately documented or the documentation has not survived.

1. For the SOM projected products, the ellipsoid used for the SOM processing is not exactly known. Some documents refer to the International Ellipsoid of 1924 while other documents state using the same ellipsoid that the control points were in, which suggests the Clarke 1866 ellipsoid for data located over the U.S. during that time period, and the International 1924 for other parts of the globe. This is also an unknown for the UTM tic-marks (although they are not used in the IAS / LPGS). Because these systematic products are later refined with control points, the error in selecting one of these two ellipsoids is largely a first-order error, well below the gross errors due to the systematic nature of the original processing.
2. A constant internal to the SOM projection subroutines that the USGS and other agencies developed and routinely used has changed over the years. This constant determines the nominal longitude of the ascending node from the WRS path system and can result in an 800m uncertainty if not chosen correctly. This constant is thought to be accurate for the time period in which these A products were generated.

Landsat 5 path number: 42

Space Oblique Mercator Projection Parameters:

Semi-Major Axis of Ellipsoid:	6378388.000000 meters
Semi-Minor Axis of Ellipsoid:	0.006723 meters
Inclination of orbit	98.200000
Longitude of ascending orbit	64.412873
Landsat Ratio	0.520161

6.6 MSS-A Geometric Processing

MSS-A data are not geometrically corrected. The A-to-P resampling grids are contained in the MSS-A ancillary file. As stated, projection location information is contained in the MSS-A annotation file and is processed the same as it is in MSS-P. The image coordinate space that the tic-mark mapping described applies to the yet-to-be-constructed HOM / SOM path-oriented image geometry. This geometry is described by the A-to-P mappings contained in the horizontal and vertical resampling grids in the ancillary file.

Imagery contained in the MSS-A format have radiometric corrections applied, and geometric corrections calculated but not applied. The geometric corrections are described by a set of resampling grids, a Horizontal Resampling Space (HRS) grid, and a Vertical Resampling Space (VRS) grid. The geometric corrections were historically applied to create the MSS-P products. The satellite attitude and ephemeris information needed to create a full-up satellite geometry model no longer exists; therefore, the HRS and VRS are the best path to a systematic product (although more satellite location information is present for Landsats 4-5 MSS-A products than for Landsats 2-3 MSS-A products).

The HRS and VRS grids are well described in Appendix H of [USGS, 1979]; the diagrams showing the horizontal-only mapping from Input (Raw) Space to Hybrid Space (using the HRS) and then the vertical-only mapping to Output Space (using the VRS) are not repeated in this document. The A-to-P mapping process, using these HRS and VRS, are described in detail in the document that accompanies USGS-EROS memo OCS 8-5, dated August 6, 1981. The document describes many offsets and constants used in the process that are not documented elsewhere. These two references are required reading for a full understanding of the A-to-P mapping process. This is a forward-mapping process and results in the path-oriented, HOM projected space of the MSS-P products. The IAS / LPGS geometry algorithms use the HRS and the VRS grids, combined with the MSS-P geometric processing, to perform an inverse mapping process from the final, north-up map projected space, through the HOM, path-oriented MSS-P space, through the VRS and HRS grids to get to raw MSS-A geometry space. The IAS / LPGS geometry algorithms for MSS-A are similar in concept to the MSS-P geometry processes, except that the output MSS-P geometric space is intermediate; this information creates an inverse mapping to the MSS-A imagery by stepping through the VRS and HRS grids in the following algorithm.

Given the HRS and VRS grids and an output (path-oriented MSS-P HOM-projection space) line, sample coordinate, find the input space (MSS-A) line, sample coordinate. This “output space” is combined with the MSS-P conversions above into one mapping grid that defines the mapping from the MSS-A space to the north-up output space.

Given: HOM-projection space line, sample coordinates
HRS and VRS grids and other constants from the ancillary file

1. Find the VRS grid column that contains the HOM-space sample coordinate:

```

VRSCol = (int)(HOM\_outSamp/60); (60 is the number of
                                samples in a VRS grid cell)
sampInVRSCol = HOM\_outSamp - ( VRSCol * 60);
if (VRSCol < 0) failure; (These are checks for being
                          outside the defined grid)
if (VRSCol > 60) failure; (60 is the number of VRS
                          gridcell columns)

```

2. Find the VRS grid row that contains the HOM-space line coordinate:

```

VRSRow = (int)(HOM\_outLine/70); (70 is the number of
                                lines in a VRS grid cell)
lineInVRSRow = HOM\_outLine - ( VRSRow * 70);
if (VRSRow < 0) failure;
if (VRSRow > 43) failure; (43 is the number of VRS
                          gridcell rows)

```

3. Map the HOM-space output line coordinate into the hybrid space line coordinate. For line coordinates, the hybrid space = input space:

```

VRt = vrs.row[VRSRow].col[VRSCol] + sampInVRSCol *
      ((vrs.row[VRSRow].col[VRSCol+1]
        - vrs.row[VRSRow].col[VRSCol])/60.0);
VRb = vrs.row[VRSRow+1].col[VRSCol] + sampInVRSCol *
      ((vrs.row[VRSRow+1].col[VRSCol+1]
        - vrs.row[VRSRow+1].col[VRSCol])/60.0);
hybLine = VRt+lineInRow * ((VRb-VRt)/70.0); (70 is the
                                             number of lines in each VRS grid cell)

```

4. Using the HRS, find the input space sample locations (that is find the HRS row and column that contains the hybrid space coordinates). For sample coordinates, the hybrid space = output space:

```

HRSRow = (int)(hybLine/48); (48 is the number of
                            lines in a HRS grid cell)
lineInHRSRow = hybLine - (HRSRow * 48);
if (HRSRow < 0) failure;

```

```
if (HRSRow > 50) failure; (50 is the number of
                           HRS gridcell rows)
```

5. Once the HRS cell containing the hybrid coordinates is available, calculate the input space coordinates:

```
HRl = hrs.row[HRSRow].col[VRSCol] + lineInHRSRow *
      ((hrs.row[HRSRow+1].col[VRSCol]
        - hrs.row[HRSRow].col[VRSCol])/48.0);

HRr = hrs.row[HRSRow].col[VRSCol+1] + lineInHRSRow *
      ((hrs.row[HRSRow+1].col[VRSCol+1]
        - hrs.row[HRSRow].col[VRSCol+1])/48.0);

inSamp = (HRl + sampInVRSCol * ((HRr - HRl) / 60.0))
          + (numPixelsPerScanLine/2.0);

inLine = hybLine + (centerScanLinePart
                   - ((numSweeps * 6.0) + 2.5))
```

Where the HRS and VRS resampling grids and the constants numPixelsPerScanLine, centerScanLinePart, and numSweeps are read from the MSS-A ancillary file.

7 The GPYRAMID Algorithm

The GPYRAMID algorithm registers two images that have potentially significant relative offsets. The conventional registration algorithm using grey-scale correlation techniques have limitations when the relative offsets between the two images are high. To start the GPYRAMID algorithm, create undersampled images using the Gaussian resampling technique at multiple resolutions, also called pyramid levels. The first (lowest) level is the original image, and successive levels are images undersampled by a factor of two from the previous levels. The gray-scale correlation based registration algorithm runs at each level using the cumulative, relative offset approximations derived from previous levels. The algorithm first runs at the highest pyramid level, where the image size is smaller and the resolution is coarser. The first level, being a low resolution image, helps register images that have significant offsets at a coarser resolution. The number of pixels required to search for the correct matching is smaller; a bigger correlation window ensures more ground area is considered during matching, which avoids any false matches between the two images due to temporal changes.

The GPYRAMID algorithm in IAS / LPGS is used in the precision correction process for MSS and TM sensors. The Level 1 Geometrically Systematically-Corrected (L1G) are registered to the ortho-rectified reference scene (GLS2000) using the GPYRAMID algorithm. The GPYRAMID algorithm uses I2I characterization algorithm [I7g, 2006] for correlation of the systematic images and reference images at different pyramid levels. The last step in the GPYRAMID algorithm, the GLS2000 GCP chips from MMIO algorithm [mmiopaper, 2008] registers the systematic image using the cumulative relative offsets measured at the final pyramid level. The measured residuals at the final iteration are used as input to the precision solution and precision correction process.

The algorithm as implemented in the IAS / LPGS system:

1. If TM, Band 5 of the systematic image is used for correlation; otherwise, (MSS sensor) Band 7 of the systematic image is used for correlation.

2. If the MSS sensor and WRS1 acquire the systematic scene, then the GLS1975 data set are used as a reference scene for the GPYRAMID correlation. If the systematic scene is MSS and in the WRS2 reference, then the GLS1990 dataset is used, which are resampled in the GPYRAMID algorithm to match the MSS product resolution.
3. If the WRS1 GLS1975 data are unavailable for a given path / row, then the mosaic of GLS2000 data in the WRS1 reference system are used.
4. If the WRS2 GLS1990 data are unavailable for a given path / row, then the GLS2000 data are used.
5. If a systematic scene is from the TM sensor and the date of acquisition is less than the threshold date (1-1-1995), then GLS1990 is used as a reference. If the date of acquisition is on or after the threshold date (1-1-1995), then the GLS2000 reference data set is used for the GPYRAMID correlation.
6. Gaussian Pyramid Levels – To generate the Gaussian image pyramids, downsample the systematic image (Band 7 / Band 5); to reference GLS data, use the Gaussian resampling technique by a factor of 2 for each pyramid level. In general, four pyramid levels exist; the first level is at the original resolution of the systematic image (i.e., for MSS, the first level is at 60m) (See Figure 7.1.). The second level (X2) is generated by resampling the first (X1) pyramid level by a factor of 2. For MSS, the (X2) level produces a pyramid level with resolution at 120m. The third level (X3) is at four times the original resolution or a factor of 2 from the second level (X2) and the fourth level (X4) is eight times the original resolution or a factor of 2 from the third level (X3). Similar image pyramid levels are generated in the same way for reference data. The algorithm can have any number of pyramid levels; user-defined ODL parameters can specify this information.
7. Generate evenly spaced GCP locations from the search image (Systematic MSS / TM scene) (see Figure 7.2).
8. Using the geolocation information from the reference and search image and the search image GCP locations, the corresponding GCP locations in the reference image are determined. The GCP text file containing the

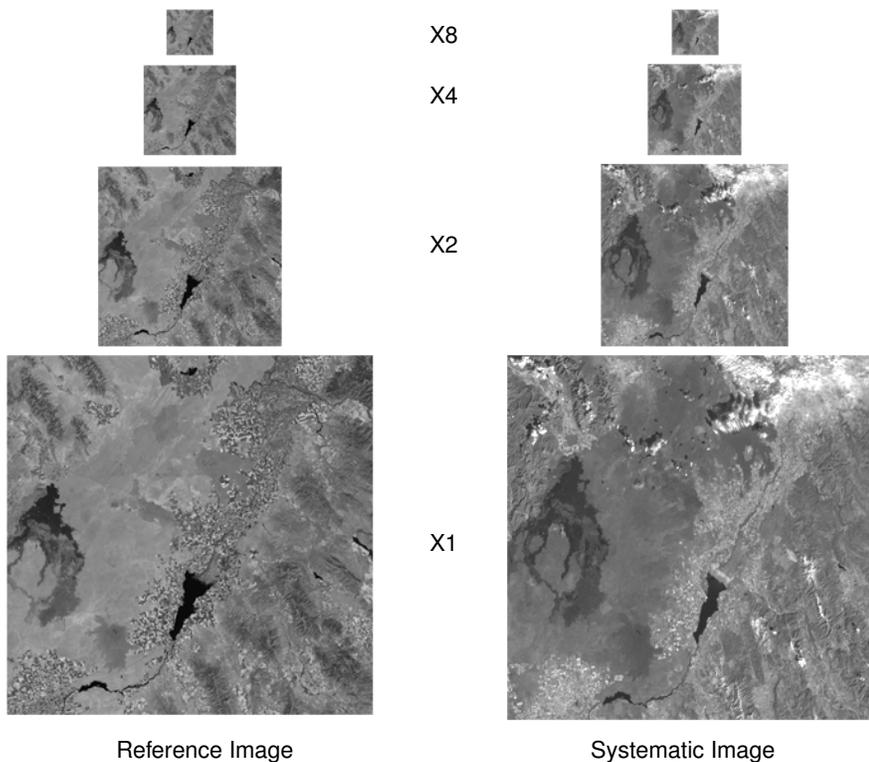


Figure 7.1. Gaussian Image Pyramid Levels

geographic locations of the GCPs and their attributes such as pixel size, line, and sample locations, etc. are created.

$$\begin{aligned} \text{line_offset} &= \text{round}(-(\text{search_UL}_y - \text{ref_UL}_y) / \text{ref_proj_dist}_y) \\ \text{sample_offset} &= \text{round}((\text{search_UL}_x - \text{ref_UL}_x) / \text{ref_proj_dist}_x) \\ \text{GCP_ref_loc}_x &= \text{GCP_search_loc}_x + \text{sample_offset} \\ \text{GCP_ref_loc}_y &= \text{GCP_search_loc}_y + \text{line_offset} \end{aligned}$$

Where search_UL_y is the upper left line coordinate of the search image, search_UL_x is the upper left sample coordinate of the search image, ref_UL_y is the upper left line coordinate of the reference image, ref_UL_x is the upper left sample coordinate of the reference image, ref_proj_dist_x and ref_proj_dist_y are projection distances for the reference image in the sample and line direction, GCP_search_loc_x is the GCP sample location

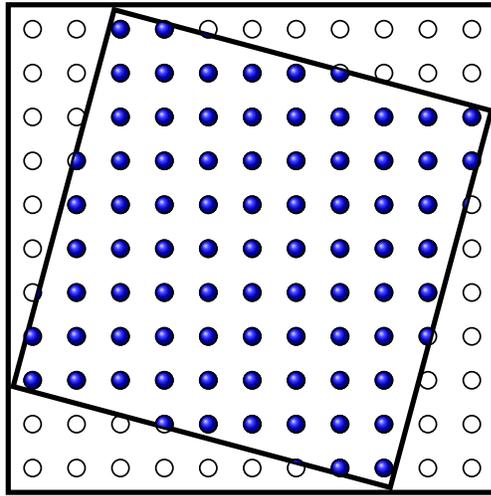


Figure 7.2. UTM Projected Scene with Evenly Spaced GCP Points

in the search image, $GCP_search_loc_y$ is the GCP line location in the search image, $GCP_ref_loc_x$ is the GCP sample location in the reference image, and $GCP_ref_loc_y$ is the GCP line location in the reference image.

9. Identify the location of the same GCPs on all image pyramid levels and generate the GCP text file containing the geographic locations of the GCPs and their attributes for each pyramid level. The same GCPs across all image pyramid levels have the same GCP identification numbers.
10. The I2I characterization [I7g, 2006] is performed using the GCP text file for the highest pyramid level. The residuals are stored in a file for the highest image pyramid level.
11. The results from the correlations go through an outlier test, as described in the following steps:
 - a) Only correlated GCPs from I2I characterization are kept and other GCPs are deleted for this pyramid level. If no GCPs are correlated, then the scene cannot be registered and falls back to the systematic product.
 - b) Count the number of GCPs whose offsets are positive against the number of GCPs whose offsets are negative in both line and sample direction.

- c) If the ratio of the minimum of the two counts to the total GCPs is less than the threshold (derived by empirical methods), then all the GCPs that belong to the minimum count set are deleted for further consideration. This step is performed for both line and sample direction.
- d) The mean, median, and standard deviation for the remaining valid GCPs are calculated.

$$\mu = \sum_{k=1}^n \text{RMSE}_{rk}$$

$$\sigma = \frac{1}{n-1} \sum_{k=1}^n (\text{RMSE}_{rk} - \mu)^2$$

$$\mu' = \begin{cases} \frac{1}{2} \left(\text{RMSE}_{rs(n/2)} + \text{RMSE}_{rs(\frac{n}{2}+1)} \right) & n \text{ is even} \\ \text{RMSE}_{rs(\frac{n+1}{2})} & n \text{ is odd} \end{cases}$$

Where n is the number of points, r is the line or sample direction, μ is the mean value of RMSE_r for all n points, σ is the standard deviation of RMSE_r for all n points, μ' is the median value for the RMSE_r , RMSE_{rk} is the k th index of RMSE_r , RMSE_{rs} is RMSE_r sorted by residuals, and $\text{RMSE}_{rs(n/2)}$ is the $n/2$ index of RMSE_{rs} .

- i. Any GCPs whose line or sample offsets are more than three times the standard deviation from the mean in line or sample direction are considered outliers and removed.
- ii. All the GCPs considered outliers are removed from the GCP text file for the next image pyramid level (i.e., for outliers detected in X8 level, those GCP IDs are removed from the X4 GCP text file and are not included in the remaining GCP text files for subsequent levels).
- iii. The GCP text file for the next pyramid level is adjusted for their coordinate locations using the offset in the line and sample direction measured from the current pyramid level.
- iv. The updated northing and easting coordinates are converted to the latitude / longitude coordinates using the UTM zone; and all the geographic information are updated in the GCP text file for

all the valid GCPs.

$$\begin{aligned}\text{Updated_Northing_next_level} &= \text{Northing} \\ &\quad - \text{pixel_size}_y \times \text{offset}_y \\ \text{Updated_Easting_next_level} &= \text{Easting} \\ &\quad + \text{pixel_size}_x \times \text{offset}_x\end{aligned}$$

Where Northing is the UTM Northing coordinate for the particular GCP, Easting is the UTM Easting coordinate for the particular GCP, pixel_size_x is the projection distance in sample direction for the pyramid level, pixel_size_y is the projection distance in line direction for the pyramid level, offset_x is the offset measured by I2I characterization (in pixels) in sample direction, and offset_y is the offset measured by I2I characterization (in pixels) in line direction.

12. The consistency check is performed for each GCP across multiple pyramid levels except for the first image pyramid level.
 - a) Because the GCP offsets derived from the previous pyramid level in correlation are updated with the next pyramid level, the I2I correlation should not produce any more than 1.5 pixel offsets in the next pyramid level. If the offsets are more than 1.5 pixels, then the points were either wrongly matched in the previous or current pyramid level. The threshold is selected as 1.5 pixels instead of 1 pixel due to the possibility of variations during Gaussian pyramid image generation that may cause an additional uncertainty in matching. The primary purpose of the consistency check is to remove any blunders from matching.
 - b) The GCP points whose offsets are beyond a threshold of 1.5 pixels in either line or sample direction are considered outliers and removed as part of the consistency check.
13. Once the highest pyramid level (X8) correlation and outlier rejection process is complete, the same procedure is repeated for other pyramid levels.
 - a) Loop back to Step 10 and continue until the lowest level (X1) pyramid is complete with outliers identified and removed.

14. Determine the average offsets between systematic scene and GLS reference.
 - a) After the last pyramid level (X1) is complete, the valid GCP's line and sample offsets are used to compute the mean offsets for line and sample direction.
15. The gpcorrelate algorithm [[Park and Schowengerdt, 1983](#)] uses the GLS 2000 MMIO GCP chips to register the systematic image using the relative offsets measured at the final pyramid level (X1). The measured residuals at the final iteration are used as input to the precision solution and precision correction process (Section [10](#)).

8 Create Resampling Grid

The MSS-X geometric resampling grid defines the piece-wise (gridded) geometric transformation of imagery in satellite perspective space to geometrically corrected imagery in a user specified output projection space. The geometric transformation function (the Resampler) uses the grid to rectify the [L1R-H](#) image data to the map projected output space. This algorithm assumes that the MSS-X satellite model has been initialized using the input image's associated MSS-X LORp files and that a hybrid image file (Section [5.5](#)) has been created. The MSS-X resampling grid consists of a set of evenly spaced output locations that are at integer sample and line locations. These integer and line locations map to a set of floating point, non-integer, input space sample and line coordinates. A set of four grid coordinates, two consecutive in the sample direction and two consecutive in the line direction, defines one grid cell. A group of bilinear polynomials, determined from the grid cell corner coordinates, map line and sample coordinates between input ([L1R-H](#)) space to output (map projected) space.

8.1 Read MSS-X Geometric Model File

8.2 Read MSS-X CPF

8.3 Determine Output Map Projected Image Frame

Because the MSS-X geometric model works as an inverse mapping function, the output image frame is determined through an iterative process. An initial set of projection coordinates is chosen, mapped to the MSS-X input space and checked for “quality-of-fit”. This quality-of-fit is a set of map projection cor-

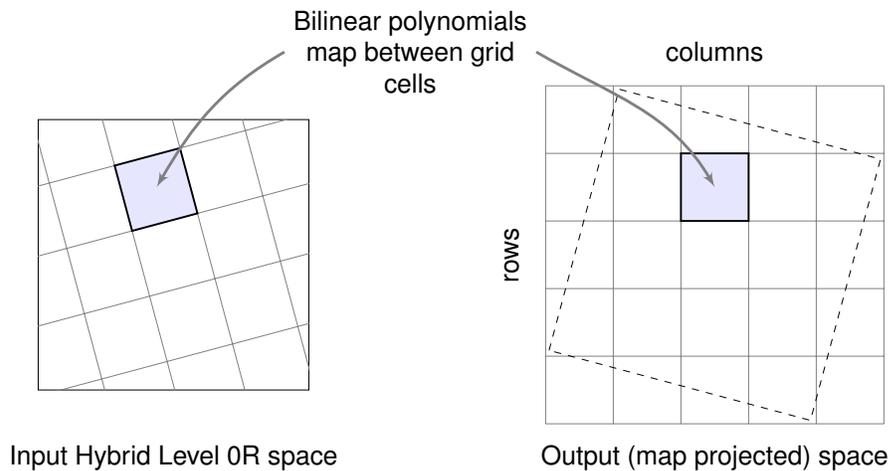


Figure 8.1. Geometric Resampling Grid

ners that bound the imagery within a reasonable range by not allowing an objectionably large amount of image fill. The GNU Scientific Library (GSL) [GSL Contributors, 2010] is used in the fitting process. The GSL nonlinear multi-dimensional routines are used, along with the MSS-X inverse model listed in section 5.4, to find a set of output LIG image projection corners (target corners) to fit the corresponding MSS-X geometric model while maintaining a few pixels of fill outside the outer corners of the final MSS-X image product.

The following is the algorithm:

1. Initialize the target coordinates.
Target corners = LOR Data Descriptor Records map projection corners.
2. Initialize GSL root solver routines.
3. Call GSL solve routines.
Determine upper left, upper right, lower right, and lower left corners: GSL upper left target(x, y), GSL upper right target(x, y), GSL lower right target(x, y), GSL lower left target(x, y).

- Determine the bounding rectangle from the GSL corners.

$$\begin{aligned} \text{upper left } x &= \text{lower left } x = \min(\text{GSL corners}) \\ \text{upper left } y &= \text{upper right } y = \max(\text{GSL corners}) \\ \text{upper right } x &= \text{lower right } x = \max(\text{GSL corners}) \\ \text{lower left } y &= \text{lower right } y = \min(\text{GSL corners}) \end{aligned}$$

8.4 Map Output Grid Cell Locations to L1R-H Pixels

```

for col = 0 to number of grid columns do
  for row = 0 to number of grid rows do
    Output sample = Grid Cell Output Sample[col][row]
    Output line = Grid Cell Output Line[col][row]
  end for
end for

```

- Calculate the output projection coordinate.

$$\begin{aligned} x_{\text{utm}} &= \text{upper left target } x + (\text{output line} - 1) \text{grid pixel size } x \\ y_{\text{utm}} &= \text{upper left target } y - (\text{output sample} - 1) \text{grid pixel size } y \end{aligned}$$

- Convert the map projection coordinates.
Convert the UTM map projection coordinates from meters to radians. The [GCTP](#) converts from the UTM output projection to a geographic coordinate system $(x_{\text{geo}}, y_{\text{geo}})$. See Section [3.4](#).

If X_{geo} is less than 0, add 2π .

- Call the MSS-X inverse model.
See Section [5.4](#) for more information.

$$\begin{aligned} \text{input sample} &= \text{MSS-X inverse}(x_{\text{geo}}, y_{\text{geo}}) \\ \text{input line} &= \text{MSS-X inverse}(x_{\text{geo}}, y_{\text{geo}}) \\ \text{grid cell input sample}[col][row] &= \text{input sample} \\ \text{grid cell input line}[col][row] &= \text{input line} \end{aligned}$$

4. Determine individual grid cell mapping coefficients.

Bilinear mapping coefficients for each grid cell are calculated for mapping from input location to output location (forward mapping) and for mapping from output location to input location (inverse mapping). A separate mapping function is used for lines and samples, which equates to four mapping functions. A set of four mapping functions is calculated for each grid cell and for every band stored in the grid.

The following methodology calculates one set of four bilinear mapping equations:

A 9×4 matrix fits nine points within a grid cell. The matrix equation takes the form of:

$$[A][\text{coeff}] = [b]$$

In this equation, matrix A is 9×4 , vector b is 9×1 , and the coefficient matrix is 4×1 . The coefficient matrix, coeff , can be solved to obtain the mapping coefficients as:

$$[\text{coeff}] = [A^T A]^{-1} [A^T b]$$

In the case of solving for an equation to map an input line and sample location to an output sample location belonging to one grid cell, the matrices can be defined as:

$$A_{n,0} = 1 \quad \text{where } n = 0, 8$$

$$A_{0,1} = \text{upper left input sample location for current grid cell}$$

$$A_{1,1} = \text{upper right input sample location for current grid cell}$$

$$A_{2,1} = \text{lower left input sample location for current grid cell}$$

$$A_{3,1} = \text{lower right input sample location for current grid cell}$$

$$A_{4,1} = (A_{0,1} + A_{1,1} + A_{2,1} + A_{3,1})/4$$

$$A_{5,1} = (A_{0,1} + A_{1,1})/2$$

$$A_{6,1} = (A_{1,1} + A_{3,1})/2$$

$$A_{7,1} = (A_{2,1} + A_{3,1})/2$$

$$A_{8,1} = (A_{2,1} + A_{0,1})/2$$

$$A_{0,2} = \text{upper left input line location for current grid cell}$$

$$A_{1,2} = \text{upper right input line location for current grid cell}$$

$$A_{2,2} = \text{lower left input line location for current grid cell}$$

$$A_{3,2} = \text{lower right input line location for current grid cell}$$

$$A_{4,2} = (A_{0,2} + A_{1,2} + A_{2,2} + A_{3,2})/4$$

$$A_{5,2} = (A_{0,2} + A_{1,2})/2$$

$$A_{6,2} = (A_{1,2} + A_{3,2})/2$$

$$A_{7,2} = (A_{2,2} + A_{3,2})/2$$

$$A_{8,2} = (A_{2,2} + A_{0,2})/2$$

$$A_{n,3} = A_{n,1}A_{n,2} \quad \text{where } n = 0 \dots 8$$

b_0 = upper left output sample location for current grid cell

b_1 = upper right output sample location for current grid cell

b_2 = lower left output sample location for current grid cell

b_3 = lower right output sample location for current grid cell

$$b_4 = (b_0 + b_1 + b_2 + b_3)/4$$

$$b_5 = (b_0 + b_1)/2$$

$$b_6 = (b_1 + b_3)/2$$

$$b_7 = (b_2 + b_3)/2$$

$$b_8 = (b_2 + b_0)/2$$

The line and sample locations listed are defined at the grid cell corner coordinates. The points interpolated in between the grid cell line segments provide stability for what could be, most notably, a mapping that involves a 45° rotation, an ill-defined solution if only four points were used in the calculation. The set of coefficients define a bilinear mapping equation of the form:

$$\begin{aligned} \text{sample}_o = & \text{coeff}_0 + \text{coeff}_1 \times \text{sample}_i + \text{coeff}_2 \times \text{line}_i \\ & + \text{coeff}_3 \times \text{sample}_i \times \text{line}_i \end{aligned}$$

where sample_o = Output sample location, sample_i = Input sample location, and line_i = Input line location.

A set of sample and line coefficients map the output projection space to the L1R-H space and a set of sample and line coefficients map the L1R-H

space to the output projection sample.

$$\text{sample}_o = \text{coeff}_0 + \text{coeff}_1 \times \text{sample}_i + \text{coeff}_2 \times \text{line}_i \\ + \text{coeff}_3 \times \text{sample}_i \times \text{line}_i$$

$$\text{line}_o = \text{coeff}_4 + \text{coeff}_5 \times \text{sample}_i + \text{coeff}_6 \times \text{line}_i \\ + \text{coeff}_7 \times \text{sample}_i \times \text{line}_i$$

$$\text{sample}_i = \text{coeff}_8 + \text{coeff}_9 \times \text{sample}_o + \text{coeff}_{10} \times \text{line}_o \\ + \text{coeff}_3 \times \text{sample}_o \times \text{line}_o$$

$$\text{line}_i = \text{coeff}_{12} + \text{coeff}_{11} \times \text{sample}_o + \text{coeff}_{12} \times \text{line}_o \\ + \text{coeff}_{13} \times \text{sample}_o \times \text{line}_o$$

The forward mapping equations, mapping input line, and sample locations to output line locations can be solved by swapping output line locations for output sample locations in the matrix $[b]$. The reverse mapping equations, mapping output locations to input line and sample, can be found using output line and sample locations in the $[A]$ matrix and the corresponding input sample and then line locations in the $[b]$ matrix.

5. Calculate inverse mapping coefficients.
6. Calculate forward mapping coefficients.
7. Calculate the rough transform polynomial.

Calculate the rough mapping coefficients for the grid. The rough polynomial is a set of polynomials that map output line and sample locations to input line and sample locations. The rough polynomial is generated using a large number of points distributed over the entire scene, and by calculating a polynomial equation that maps an output location to an input location. The rough polynomial is only meant to get a “close” approximation to the input line and sample location for a corresponding output line and sample location. Once this approximation is made, the value can be refined for a more accurate solution. A rough mapping polynomial is found for every band.

A bilinear polynomial is used for rough mapping. The mapping function therefore looks like the functions used for each individual grid cell; however, the set up of the matrices to solve for the mapping coefficients are different:

$$\begin{matrix} [A] & [\text{coeff}] & = & [b] \\ N \times 4 & 4 \times 1 & & N \times 1 \end{matrix}$$

Where the matrix $[A]$ is defined by the output line and sample locations, matrix $[b]$ is defined by either the input lines or input samples, and N is equal to the total number of points stored in the grid for one band of imagery. The rough polynomial is therefore found by using all the point locations stored in the grid for a given band. One mapping is available for output line and sample location to input sample location and one mapping is available for output line and sample location to input line location.

9 Resample L1R-H Imagery

The following steps are applied to the L1R-H imagery. Use these steps for the creation of a Level 1 Geometrically Precision/Terrain-Corrected (L1T).

1. Read the resampling grid.
2. Read the L1R-H imagery.
 - a) Build the terrain table.

To account for relief or terrain effects, adjust the input sample location during the inverse mapping calculation (output projection pixel) \Rightarrow (input pixel) relationship. The sample location is adjusted for relief based on the off-nadir across track viewing angle and elevation height associated with the output projection coordinate of the L1R-H pixel. To avoid performing the calculations for each output pixel, an LUT based on off-nadir viewing angle and height, or elevation, is created. For any given elevation and angle associated with the MSS-X data, a delta pixel adjusts the sample location of the L1R-H.
 - b) Determine the bounds on terrain table parameters.
 - i. Given a nominal satellite orbital radius, set this value as the magnitude of the satellite positional vector.
 - ii. If building L1T, read DEM and determine the minimum and maximum elevation values.

$$\text{number of elevation values} = \frac{\text{maximum elevation} - \text{minimum elevation}}{\text{elevation increment}} + 1$$

The elevation increment is set to 1 meter.

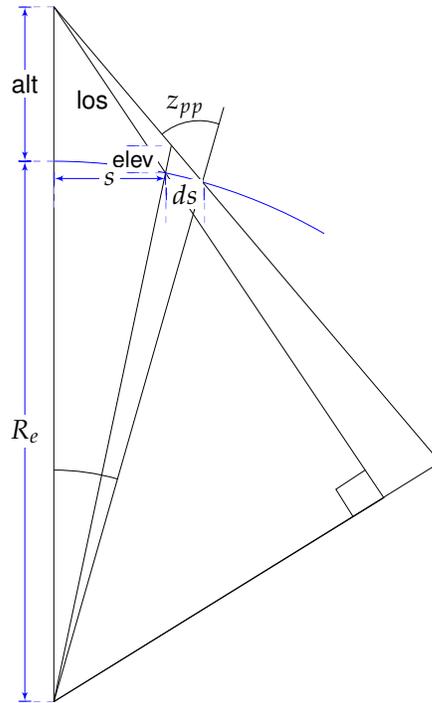


Figure 9.1. Terrain Table Calculations

- iii. If producing L1T, build the nadir sample location on a per-input line basis. Read L0R Scan Line Offset (SLO) and determine the nadir location:

maximum nadir = 0

for i = 0 to number of lines in Level0R **do**

$nadir_i = \frac{1}{2}(\text{number of samples in L0R} - \text{Right Hand SLO} + \text{Left Hand SLO})$

if $nadir_i > \text{maximum nadir}$ **then**

maximum nadir = $nadir_i$

end if

end for

number off-nadir samples = maximum nadir

- iv. If producing L1T, calculate the terrain LUT (see Figure 9.1).

for sample=0 to number of off-nadir samples **do**

for elev=minimum elevation to maximum elevation **do**
ground range = $g_r = (\text{pixel size}/\text{sample})$

$$\text{los} = \sqrt{R_e^2 + (R_e + \text{alt})^2 + 2R_e(R_e + \text{alt}) \cos s}$$

$$\beta = \sin^{-1} \left(\frac{R_e}{\text{los}} \sin s \right)$$

$$d\theta = \tan^{-1} \left(\frac{-\text{elev}(R_e + \text{alt}) \sin \beta / R_e}{(R_e + \text{elev}) \sqrt{1 - (R_e + \text{alt})^2 \frac{\sin^2 \beta}{R_e^2}} - (R_e + \text{alt}) \cos \beta} \right)$$

$$z_{pp} = \sin^{-1} ((R_e + \text{alt}) \sin ((\beta + d\theta) / R_e))$$

$$ds = z_{pp} - s - \beta - d\theta$$

$$dg_r = R_e ds$$

$$\text{terrain offset}[\text{sample}][\text{elev}] = \frac{dg_r}{\text{pixel size}}.$$

end for
end for

3. Build the cubic convolution resampling table.

To save time and for efficiency, a table of two-dimensional cubic convolution weights is created. The cubic convolution equation generates the resampling weights. The two-dimensional cubic convolution is a separable function with each dimensional defined by the one-dimensional function. The two-dimensional function is represented by a 4×4 kernel with each set of weights representing $1/32$ of a pixel shift in either the sample or line direction.

4. Resample the L1R-H imagery.

Resample the L1R-H imagery using the resampling grid (and terrain offset if producing L1T).

Round = $1/64$

for line=0 to number of output lines **do**

for sample=0 to number of output samples **do**

if producing L1T **then**

Read the co-registered elevation for the output pixel.

Read L1R-H input sample offset from terrain offset table.

end if

Determine grid cell number for output pixel location:

row = oline / number of grid rows

col = osamp / number of grid columns

grid cell index = row * number grid columns + col

Read the grid cell inverse mapping coefficients for the grid cell number.

Calculate the input line and sample location:

$$\text{sample}_i = \text{coeff}_8 + \text{coeff}_9 \times \text{sample}_o + \text{coeff}_2 \times \text{line}_o \\ + \text{coeff}_3 \times \text{sample}_o \times \text{line}_i$$

$$\text{line}_i = \text{coeff}_{12} + \text{coeff}_{11} \times \text{sample}_o + \text{coeff}_{12} \times \text{line}_o \\ + \text{coeff}_{13} \times \text{sample}_o \times \text{line}_i$$

if producing L1T then

sample_i = sample_i + terrain offset[sample][elev]

end if

Determine the integer and sub-pixel line and sample locations:

$$\text{isample} = \lfloor \text{sample}_i \rfloor$$

$$\text{iline} = \lfloor \text{line}_i \rfloor$$

$$ds = \text{sample}_i - \text{isample}$$

$$dl = \text{line}_i - \text{iline}$$

Determine the cubic convolution weights (or LUT indexes):

$$ll = (dl + \text{round}) \times 32$$

$$ss = (ds + \text{round}) \times 32$$

$$\text{ccw} = \text{cubic convolution LUT}[ll][ss]$$

Apply resampling weights to the input pixel location:

total = 0.0

for ii=0 to 3 do

for nn=0 to 3 do

Total = total + ccw[ii][nn] * L1R-H[iline+ii-1][isample+nn-1]

end for

end for

```
        Output pixel [line][sample] = total
    end for
end for
```


10 MSS REFINE

The MSSREFINE algorithm generates a precision grid from a systematic grid. The MSSREFINE algorithm uses the registration information, such as GCP residuals, that the GPYRAMID algorithm provided for MSS. For ETM+ and TM sensors, the task of precision modeling is to correct the satellite's model parameters, such as ephemeris and attitude. For MSS, the ancillary information, such as time, attitude, and ephemeris may not be accurate or does not have a higher degree of precision to implement a similar precision correction. Hence, a polynomial based precision correction is implemented for MSS sensors for all formats.

The MSSREFINE algorithm starts with using the GCPs that the GPYRAMID process correlated. Each of those GCPs is adjusted for relief displacement in the input image (radiometrically corrected input image) using the systematic grid. The adjusted GCPs in input image are projected back to projection space (output grid space) using the same systematic grid. The systematic image location of each GCP and its relief adjusted correlated locations are used to fit the polynomial of either first or second order using the least squares fit method. To remove the outliers, compare the residuals in the fit to its weighted standard deviation. The systematic grid is adjusted with the polynomial coefficients to generate a precision grid, which relates the output projection location to the input line and sample location from the ALIGNLOR image. The geometric resampling algorithm uses the precision grid to create a precision terrain corrected product. By default, the second order polynomial fit is used for precision correction. If significant warping occurs from the second order polynomial fit, then the first order polynomial fit is used for precision correction. To determine warping on the precision corrected image, check if the set of points along each edge of the precision corrected image lies in a straight line to within certain specified tolerance.

The following sections explain the MSSREFINE algorithm as implemented

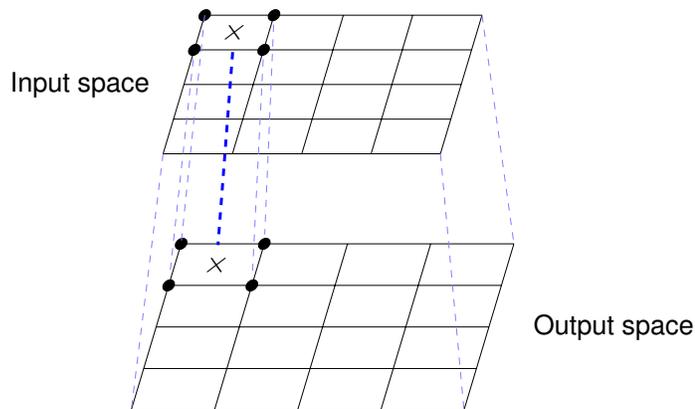


Figure 10.1. Grid Cell Mapping

in the IAS / LPGS.

1. Read the input data.
The necessary input to this algorithm, such as the systematic image's projection and geographic corner information, the systematic grid file, etc., are read along with any other required ancillary information depending on the formats. For example, on the MSS-P format, the tick marks from annotation files are required.
2. Find the line and sample locations for all GCPs in the input grid space.
The systematic grid relates the input line and sample location from the ALIGNLOR image to the output projection line and sample location. The grid structure stores the corresponding location of grid points in input space and output space and the polynomial for each grid cell. For all GCPs that the GPYRAMID algorithm considers valid, the location of the GCPs in the input space is determined by finding the location of the GCPs in the appropriate grid cells and by using the inverse mapping coefficients as described in step 4 of Section 8.4.
3. Initialize the nadir line and sample for each line.
To establish the nadir line for the scene, assume the center of the image as nadir line for all formats. To determine this information, use the Right SLO and Left SLO values. Use the following formula to determine the

Nadir sample for each line:

$$\text{Nadir}_i = \frac{1}{2}(\text{number of samples in the image} - R_SLO_i + L_SLO_i)$$

Where R_SLO_i is the right SLO values for i th line, L_SLO_i is the left SLO values for i th line, and Nadir_i is the nadir sample for i th line.

4. Apply relief adjustment to GCPs.

The GCPs used for registration were derived from an ortho-rectified product. To account for the parallax error due to terrain or relief, an adjustment in the LOS is required. To account for the relief adjustments, adjust the input sample location. The GCPs in the input space locations are adjusted based on the off-nadir across track viewing angle and its corresponding elevation. The nadir sample from the previous step determines the off-nadir across track viewing angle. Use the following formula to determine the sample offset for adjustment due to relief:

$$\text{ground range} = g_r = \text{PixelSize} \times (\text{sample} - \text{nadir sample})$$

$$\text{CentralAngle} = \alpha = \frac{g_r}{R_e}$$

$$\text{los} = \sqrt{R_e^2 + (R_e + \text{alt})^2 + 2R_e(R_e + \text{alt}) \cos s}$$

$$\beta = \sin^{-1} \left(\frac{R_e}{\text{los}} \sin s \right)$$

$$d\theta = \tan^{-1} \left(\frac{-\text{elev}(R_e + \text{alt}) \sin \beta / R_e}{(R_e + \text{elev}) \sqrt{1 - (R_e + \text{alt})^2 \frac{\sin^2 \beta}{R_e^2} - (R_e + \text{alt}) \cos \beta}} \right)$$

$$z_{pp} = \sin^{-1} ((R_e + \text{alt}) \sin ((\beta + d\theta) / R_e))$$

$$ds = z_{pp} - s - \beta - d\theta$$

$$dg_r = R_e ds$$

$$\text{sample}_{\text{adj}} = \text{sample} + \text{dir} \frac{dg_r}{\text{PixelSize}}$$

Where $\text{dir} = 1$ if $(\text{sample} < \text{nadir sample})$, else $\text{dir} = -1$, R_e = radius of Earth, alt = altitude of the spacecraft, and elev = elevation of the GCP.

5. Find the line and sample locations for all relief adjusted GCPs in the output grid space.

The same systematic grid structure finds the line and sample locations for all relief adjusted GCPs in the output projection space. To determine this information, use the forward mapping coefficients, as described in step 4 of Section 8.4.

6. Compute the polynomial fit coefficients.

The MSS scenes are precision corrected using either the first order or second order polynomial function. The first order term includes the cross term, and is not truly a function with degree 1. The polynomial fit coefficients are computed between the systematic image location of each GCP to its relief adjusted correlated locations in the output projection space using the least squares fit method. The following methodology determines the fit coefficients.

The second order fit equation is expressed as:

$$\begin{aligned} \text{sample}_{\text{adj}} &= a_0 + a_1\text{sample} + a_2\text{line} + a_3\text{line} \cdot \text{sample} \\ &\quad + a_4\text{sample}^2 + a_5\text{line}^2 \\ \text{line}_{\text{adj}} &= b_0 + b_1\text{sample} + b_2\text{line} + b_3\text{line} \cdot \text{sample} \\ &\quad + b_4\text{sample}^2 + b_5\text{line}^2 \end{aligned}$$

The first order fit equation is expressed as:

$$\begin{aligned} \text{sample}_{\text{adj}} &= a_0 + a_1\text{sample} + a_2\text{line} + a_3\text{line} \cdot \text{sample} \\ \text{line}_{\text{adj}} &= b_0 + b_1\text{sample} + b_2\text{line} + b_3\text{line} \cdot \text{sample} \end{aligned}$$

Where $\text{sample}_{\text{adj}}$ and line_{adj} are the GCPs correlated and relief adjusted line and sample location in the output projection space, sample and line are the GCPs location in the output projection space of the systematic grid, a_0 – a_5 and b_0 – b_5 are the second order polynomial fit coefficients, and a_0 – a_3 and b_0 – b_3 are the first order polynomial fit coefficients.

These equations are solved using least squares adjustment in the matrix form:

$$[\text{Predicted_line_sample}][\text{fit_coeff}] = [\text{true_line_sample}]$$

Or:

$$\begin{aligned} [A_1][\text{coeff}_s] &= [b_s] \\ [A_2][\text{coeff}_l] &= [b_l] \end{aligned}$$

Where:

$$\begin{aligned}
 A_{1n,0} &= 1 \\
 A_{1n,1} &= \text{sample}_n \\
 A_{1n,2} &= \text{line}_n \\
 A_{1n,3} &= \text{sample}_n \text{line}_n \\
 A_{1n,4} &= \text{sample}_n^2 \quad (\text{if fit order} = 2) \\
 A_{1n,5} &= \text{line}_n^2 \quad (\text{if fit order} = 2)
 \end{aligned}$$

$$\begin{aligned}
 A_{2n,0} &= 1 \\
 A_{2n,1} &= \text{sample}_n \\
 A_{2n,2} &= \text{line}_n \\
 A_{2n,3} &= \text{sample}_n \text{line}_n \\
 A_{2n,4} &= \text{sample}_n^2 \quad (\text{if fit order} = 2) \\
 A_{2n,5} &= \text{line}_n^2 \quad (\text{if fit order} = 2)
 \end{aligned}$$

$$\begin{aligned}
 b_{sn} &= [\text{sample}_{\text{adj}}]_n \\
 b_{ln} &= [\text{line}_{\text{adj}}]_n \\
 \text{coeff}_s &= \begin{cases} [a_0, a_1, a_2, a_3, a_4, a_5]^T & (\text{if fit order} = 2) \\ [a_0, a_1, a_2, a_3]^T & (\text{if fit order} = 1) \end{cases} \\
 \text{coeff}_l &= \begin{cases} [b_0, b_1, b_2, b_3, b_4, b_5]^T & (\text{if fit order} = 2) \\ [b_0, b_1, b_2, b_3]^T & (\text{if fit order} = 1) \end{cases}
 \end{aligned}$$

n is the index of the GCPs, $0 \leq n < k$ where k is the number of GCPs.

7. Outlier detection and removal.

The polynomial fit is iterated several times until the number of iterations reaches the maximum number of allowed iterations or no outliers are found. The outliers are detected in each iteration and removed for determining the polynomial fit. A GCP is detected as an outlier if its residuals after applying the polynomial fit are greater than the fit threshold. To determine the fit thresholds, sum the squares of the residuals obtained during polynomial fit in the line and sample direction. These sum squared residuals are weighted by a tolerance to calculate the fit threshold. Points

whose residuals after polynomial fit are greater than the fit thresholds are marked as outliers and removed; the remaining points are reiterated for new polynomial fit coefficients.

$$\begin{aligned}
 [\text{residual}_{\text{sample}}] &= [A_1][\text{coeff}_s] - [b_1] \\
 [\text{residual}_{\text{line}}] &= [A_2][\text{coeff}_l] - [b_2] \\
 \text{res}_s^2 &= \sum \text{residual}_{\text{sample}}^2 \\
 \text{res}_l^2 &= \sum \text{residual}_{\text{line}}^2 \\
 \text{threshold}_{\text{fit}} &= \text{tolerance} \sqrt{\frac{\text{res}_s^2 + \text{res}_l^2}{\text{num}_{\text{gcp}}}} \\
 \text{residuals}_i &= \sqrt{\text{residual}_{\text{sample}_i}^2 + \text{residual}_{\text{line}_i}^2}
 \end{aligned}$$

If $(\text{residuals}_i > \text{threshold}_{\text{fit}})$ then GCP_i is an outlier where i is the index of the GCP, $\text{residual}_{\text{sample}_i}$ is the i th index of $[\text{residual}_{\text{sample}}]$ array, $\text{residual}_{\text{line}_i}$ is the i th index of $[\text{residual}_{\text{line}}]$ array, tolerance is the userdefined number, and num_{gcp} is the number of GCPs.

8. Generate the precision grid.

a) Adjust the grid framing.

As an initial approximation, the systematic grid is copied as a precision grid. The four corner coordinates of the grids are adjusted with the fit order coefficients of the zero order. This step ensures that the precision grid frame covers the whole image and is not clipped due to the systematic scene's offsets from their true geographic location. Because the grid framing coordinates were adjusted, the line and sample values need to be adjusted also. To adjust these values, add the line and sample offsets to each of the line and sample location coordinates in the output space of the grid. In general, these values are indexed such that the first pixel has the coordinate (0,0) in the output space. At this stage, the precision grid is adjusted to account for the frame shift, which prevents any image from being

clipped in the final product.

$$S_{\text{offset}} = -[a_0]$$

$$L_{\text{offset}} = -[b_0]$$

$$X_{\text{offset}} = S_{\text{offset}} \cdot \text{PixelSize}$$

$$Y_{\text{offset}} = -L_{\text{offset}} \cdot \text{PixelSize}$$

$$\text{precUL}_x = \text{sysUL}_x + X_{\text{offset}}$$

$$\text{precUL}_y = \text{sysUL}_y + y_{\text{offset}}$$

$$\text{precUR}_x = \text{sysUR}_x + X_{\text{offset}}$$

$$\text{precUR}_y = \text{sysUR}_y + y_{\text{offset}}$$

$$\text{precLL}_x = \text{sysLL}_x + X_{\text{offset}}$$

$$\text{precLL}_y = \text{sysLL}_y + y_{\text{offset}}$$

$$\text{precLR}_x = \text{sysLR}_x + X_{\text{offset}}$$

$$\text{precLR}_y = \text{sysLR}_y + y_{\text{offset}}$$

$$\text{preOUT}_s = \text{sysOUT}_s + S_{\text{offset}}$$

$$\text{preOUT}_l = \text{sysOUT}_l + L_{\text{offset}}$$

Where: preOUT_s is the sample location coordinates of the precision grid in the output space, preOUT_l is the line location coordinates of the precision grid in the output space, sysOUT_s is the sample location coordinates of the systematic grid in the output space, and sysOUT_l is the line location coordinates of the systematic grid in the output space.

- b) Calculate the precision grid coefficients for each grid cell.

The precision grid from the previous step is not accounted for the actual precision coefficients but only adjusted for framing. To adjust each output line and sample in the output space of the grid, use the fit coefficient equations to determine their output space coordinates. Once the grid point's output space coordinate is determined, use the systematic grid's input output relationship (grid coefficients) to determine the corresponding input line and sample for the given

grid point.

$$\begin{aligned} \text{new}_{\text{sample}} &= a_0 + a_1 \text{preOUT}_s + a_2 \text{preOUT}_l + a_3 \text{preOUT}_s \cdot \text{preOUT}_l \\ &\quad + a_4 \text{preOUT}_s^2 + a_5 \text{preOUT}_l^2 \\ \text{new}_{\text{line}} &= b_0 + b_1 \text{preOUT}_s + b_2 \text{preOUT}_l + b_3 \text{preOUT}_s \cdot \text{preOUT}_l \\ &\quad + b_4 \text{preOUT}_s^2 + b_5 \text{preOUT}_l^2 \end{aligned}$$

Where $\text{new}_{\text{sample}}$ is the new sample coordinates of the precision grid in the output grid space and new_{line} is the new line coordinates of the precision grid in the output grid space.

The systematic grid cell coefficients determine the input line and sample location for the corresponding precision line and sample grid coordinates (newsample and newline). For each grid cell point in the output space, their corresponding input line and sample locations are determined. Each grid cell's grid corner and central points determine bilinear fit coefficients, as explained in section 8.4.

c) Check for scene warping.

The precision correction process uses second order polynomial functions, which could cause skew or warping in the image if the points used for fitting the polynomials are not accurate and/or if the systematic models are not accurate. Due to the nature of the second order fit, testing is required to ensure that there are no skew or warping observed in the precision corrected image. If skew is detected in the scene, then the MSSREFINE algorithm uses first order polynomial fit for precision correction. If a scene is precision corrected using the first order polynomial fit, then the scene curvature tests are skipped and the results from the first order fit generate the precision grid.

i. Determine the first and last image lines.

The SLO values on the left and right side of the image determine the first and last image lines. This determination is necessary for the MSS-A scenes as there are fill data at the top and bottom of the radiometrically corrected images, which are used for geometry. Depending on the MSS data format, the size of the top and bottom fill regions can be zero or a positive number.

```
for  $i = \text{start\_line}$  to  $\text{end\_line}$  do  
    if  $L_{\text{SLO}i} + R_{\text{SLO}i} < n_{\text{sample}} - \text{tol}$  then
```

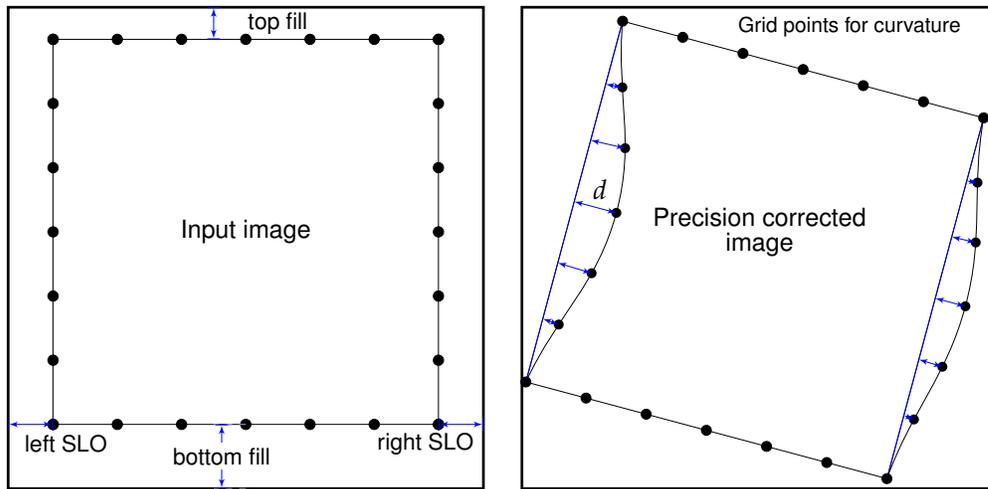


Figure 10.2. Curvature Test for Precision Correction Process

The first image line is detected.

end if

end for

for $j = \text{end_line}$ to first_image_line **do**

if $L_{\text{SLO}j} + R_{\text{SLO}j} < n_{\text{sample}} - \text{tol}$ **then**

The last image line is detected.

end if

end for

ii. Find the input and output grid points.

To validate the curvature of the scene edges, select a few grid points on the top, bottom, left, and right edges of the image (from grid). The distance of these grid points to the straight line connecting the first and last grid point for each of the four directions (top, left, right, and bottom) are determined (d as shown in Figure 10.2 if the distance measured d exceeds the tolerance, then the image is assumed to have warping, and hence falls back to first order polynomial fit correction.

The grid points are selected in the input space of the precision grid such that the grid points are equidistant from each other along the edges of the image. In Figure 10.2, seven grid points are used for the curvature test of which two points are

the first and last points along the edge. To determine each of these point's corresponding location in the output space of the precision grid, use the precision grid coefficients and the corresponding grid cells. To construct a straight line equation, use the first and last point's location in the output grid space for each edge (top, left, bottom, and right). The distance between each curvature points to its corresponding straight line that it calculated. If any of the distance (d_i) is beyond the curvature tolerance (t_i), then the precision corrected scene fails the curvature test and the precision correction process continues with the first order polynomial fit correction.

for all curvature test points **do**

if $d_i > t_i$ **then**

Precision corrected scene likely to have warping.

Scene corrected with first order polynomial fit.

end if

end for

The generated precision grid produces a terrain corrected product.

9. Update the grid.

- a) Read GCPs.
- b) Map adjusted (predicted + offset) output space pixels to input space (x1R).
- c) Compute nadir for every input line.
- d) Modify the adjusted input space pixels according to effects due to relief/terrain.
- e) $x1R \Rightarrow x1R_terr$.
- f) Map the original L1R (x1R) hybrid space pixels and terrain adjusted (x1R_terr) to the output.
- g) Find the line and sample difference between output pixel locations calculated in 9f.
- h) Correct adjusted (predicted + offset) output pixels by 9g.
- i) Calculate line and sample polynomial coefficients.

- i. Until no outliers are present.
 - ii. For each GCP:
 - A. Build an observation matrix (summing individual observations)
 - B. Build a normal matrix (summing normal matrix / equations)
 - iii. Solve for polynomial coefficients (LSQ fit).
 - iv. Calculate line and sample residuals after the polynomial update.
 - v. Check for outliers; if no outliers are present, exit; otherwise, return to [9b](#).
- j) Create the new grid by adjusting the original grid cell locations by polynomial adjustment calculated in [9i](#).

11 GVERIFY Algorithm Description

The geometric verification algorithm, also referred to as GVERIFY, runs on precision terrain corrected products. The main purpose of this algorithm is to independently determine the relative accuracy of the terrain corrected product with respect to the corresponding reference ortho-rectified product (GLS2000).

The GVERIFY algorithm is required for checking MSS products to provide a level of confidence on the geometric accuracy of the terrain corrected products (L1T). Because the GVERIFY algorithm provides an independent estimate of the L1T scene accuracy, this algorithm is also used on the TM sensor from Landsats 4-5.

The GVERIFY algorithm uses an I2I characterization procedure (cross-correlation) along with a simple outlier detection algorithm to determine the relative offsets of terrain corrected scene to the reference GLS2000, which are accurate to 25 m positional accuracy [Rengarajan et al., 2012]. The expectation of this algorithm is to provide a relative error estimate for four quadrants of the scene, overall relative error estimate of the full scene, and to provide a color-coded browse image showing the relative offsets at different geographic locations on the scene. Due to the inherent nature of the cross-correlation process and scene content issues within the MSS scene, limitations exist on this algorithm, which are described at the end of this section.

The input to this algorithm is a precision terrain corrected Landsat scene and the corresponding reference ortho-rectified products (GLS2000). A dense set of evenly spaced GCP locations and corresponding data are extracted from the search image (terrain corrected product); these geographic locations are transferred to the reference image (GLS2000), where a second set of reference GCPs are extracted. The reference image is correlated with the search image and the correlation statistics, just as correlation peak and correlation strength are compared against their corresponding thresholds, which were derived by empirical methods. All correlated points that pass the minimum thresholds are

used for further calculation and outlier rejection. For all points, radial offsets are calculated by the combined **RMSE** of the offsets in line and sample direction.

The outlier detection algorithm uses a neighborhood analysis only for points whose radial offsets are greater than two pixels (suspect point). For each suspect point, a bounding rectangular region is selected based on the user specified parameters. The ratio for the number of points within the bounding box whose radial offsets are less than two pixel offsets to the total number of points is found and compared against a threshold derived from empirical analysis. If the ratio is greater than the threshold, then the suspect point is marked as outlier. Irrespective of whether the point is outlier or not, the ratios are used for other suspect point neighborhood analysis, and once all the outliers are detected, they are removed from the point set.

The last outlier detection algorithm uses a simple standard deviation based test. Points whose radial offsets are outside the 'n' number of standard deviation from mean are considered as outliers where 'n' is a user specified standard deviation scaling parameter. The final set of points after the standard deviation test are considered the set of good GCPs. A user specified rank system is used and each point is ranked depending on its offsets.

For example:

- Rank 1 Offsets less than 0.5 pixel
- Rank 2 Offsets between 0.5 to 1 pixel
- Rank 3 Offsets between 1 to 2 pixel
- Rank 4 Offsets between 2 to 3 pixel
- Rank 5 Offsets greater than 3 pixel

Simple statistics, such as mean, median, standard deviation, and RMSE for the good GCPs, are calculated for the entire scene, by dividing the scene into four regions.

The set of good GCP points are color-coded based on their ranks, overlaid on the terrain corrected product, and generates a browse image. The quadrant RMSE and full scene RMSE are provided in the Level 1 metadata files for the end user.

GVERIFY uses all the required parameters from the ODL file. The algorithm and the implementation flow are provided in the following:

1. If WRS-2 and MSS, resample GLS2000 Band 5 (30 m) to 60 m (reference image). The WRS-2 GLS 2000 reference data for WRS-1 reference system is created by a mosaic of overlapping GLS2000 band files (resampled

to 60 m) for each land WRS-1 path / rows. Because all MSS products are resampled to 60 m, for the WRS-2 reference system, the reference data (GLS2000 band file) is resampled to 60 m using the Gaussian Pyramid resampling techniques [Adelson et al., 1984]. All TM products are resampled to 30 m and hence no resampling of reference data is required.

2. Generate an evenly spaced GCP chip location from the search image (MSS / TM scene).
3. Using the geolocation information from the reference and search image and the search image GCP locations, the corresponding GCP locations in the reference image are determined. The GCP text file containing the geographic locations of the GCPs and their attributes such as pixel size, line, and sample locations, etc. are created.

$$\text{line}_{\text{offset}} = \text{round} \left(\frac{-(\text{search_UL}_y - \text{ref_UL}_y)}{\text{ref_proj_dist}_y} \right)$$

$$\text{sample}_{\text{offset}} = \text{round} \left(\frac{\text{search_UL}_x - \text{ref_UL}_x}{\text{ref_proj_dist}_x} \right)$$

$$\text{GCP_ref_loc}_x = \text{GCP_search_loc}_x + \text{sample}_{\text{offset}}$$

$$\text{GCP_ref_loc}_y = \text{GCP_search_loc}_y + \text{line}_{\text{offset}}$$

Where

search_UL _y	The Upper Left line coordinate of the search image
search_UL _x	The Upper Left sample coordinate of the search image
ref_UL _y	The Upper Left line coordinate of the reference image
ref_UL _x	The Upper Left sample coordinate of the reference image
ref_proj_dist _x , ref_proj_dist _y	Projection distance for the reference image in sample and line direction
GCP_search_loc _x	The GCP chip sample location in the search image
GCP_search_loc _y	The GCP chip line location in the search image
GCP_ref_loc _x	The GCP chip sample location in the reference image
GCP_ref_loc _y	The GCP chip line location in the reference image

4. I2I characterization is performed using the GCP text file and the residuals are stored in a file [I7g, 2006].
5. For each correlated point from step 4, check if the correlation peak value is greater than the correlation peak threshold and correlation strength is greater than the correlation strength threshold (both thresholds are provided in the ODL file).
6. Points that failed step 5 are marked as outliers and removed.
7. Use the following formula to calculate a combined offset (radial) for each good point:

$$\text{RMSE}_r = \sqrt{\text{res}_l^2 + \text{res}_s^2}$$

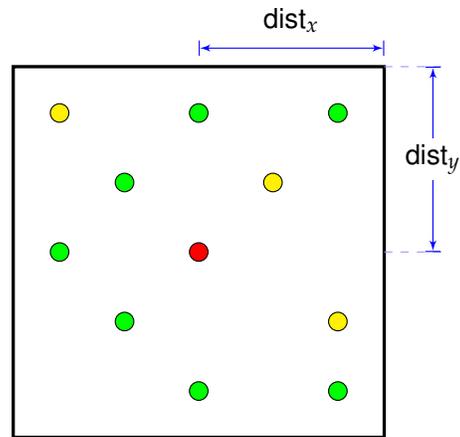
Where res_l is the measured residuals in line direction, res_s is the measured residuals in the sample direction, and RMSE_r is the combined offset in line and sample direction.

8. Identify points as suspects if their RMSE_r is greater than the threshold for suspects as listed in the ODL.
9. If the number of suspect points identified in step 8 is more than 50 percent of all points in step 7, then calculate the mean, median, standard deviation, RMSE and Median Absolute Deviation (MAD) statistics for all points and continue with the neighbor analysis in step 12.
10. If number of suspect points are less than 50 percent of all points in step 7, then calculate the MAD, ZMAD, and median for all the points from step 7.

$$\text{MAD} = \text{med } |x_i - \text{med } x|$$

$$\text{ZMAD}_i = \frac{|x_i - \text{med } x|}{\text{MAD}}$$

11. Identify outlier points using ZMAD statistics and remove the outliers. Calculate the mean, median, standard deviation, RMSE, and MAD statistics for the good points (excluding outliers) $\text{ZMAD}_i > n$ where n is a ZMAD outlier threshold from the ODL file.
12. Neighborhood analysis – For each suspect point (any points whose RMSE_r is greater than or equal to two pixels), construct a bounding box with the



- Suspect point under consideration for outlier
- Suspect point (RMSE_LS > 2 pixels)
- Suspect point (RMSE_LS < 2 pixels)

Figure 11.1. Neighborhood Analysis

suspect point at the center.

$$UL_{\text{box}} = [GCP_s - dist_x, GCP_l - dist_y]$$

$$LR_{\text{box}} = [GCP_s + dist_x, GCP_l + dist_y]$$

Where GCP_s is the sample location of the GCP in the search image, GCP_l is the line location of the GCP in the search image, $dist_x$ is the distance in sample direction for neighborhood analysis (pixels), and $dist_y$ is the distance in line direction for neighborhood analysis (pixels).

Figure 11.1 displays the sample distribution in the neighbor of a suspect point.

- a) In the first iteration, the bounding box region is constructed with 200 pixels on either side of the suspect point.
- b) Identify the number of points in this neighborhood (bounding box) that are considered valid from step 11.
- c) If the number of points found in step 12b is greater than or equal to 10, then proceed to step 13.

- d) If the number of points is less than 10, then increase the region size by a step size of 50 pixels. Repeat the step increase until either 10 points are found in the neighbor region or a maximum window size mentioned in the ODL, whichever happens first.
- e) If the number of points from step 12d is less than five, the suspect point under consideration is marked as an outlier but still considered for other suspect point's neighborhood analysis.
13. Find the number of correlated points in the bounding box; the number of points whose $RMSE_r < \text{two pixels}$ and the ratio between the two. To determine the number of points inside the bounding box, compare each point's line and sample location to check if they are inside the bounding box line and sample locations. For each of those points inside the bounding box, the points whose $RMSE_r$ is less than two pixels are considered good points. Figure 11.1, displays seven good points, three suspect points, and has a ratio of 0.7.
14. If the ratio of good points to total points inside the bounding box is greater than the neighbor threshold (from ODL file), then the suspect point is flagged as outliers. If the ratio of good points is less than the threshold, then the suspect point is not considered an outlier. When the suspect points are flagged as outliers, they are still considered a suspect point for a different suspect point's neighborhood analysis. After running the neighborhood analysis for all suspect points in the scene, all points flagged as outliers are removed.
15. Calculate the standard deviation, mean, median, RMSE, and MAD statistics for all non-outlier (valid) points from the previous step:

$$\mu = \frac{1}{n} \sum_{k=1}^n RMSE_{rk}$$

$$\sigma = \frac{1}{n-1} \sum_{k=1}^n (RMSE_{rk} - \mu)^2$$

$$\text{med } RMSE_r = \begin{cases} \frac{1}{2} (RMSE_{rs(n/2)} + RMSE_{rs(n/2+1)}) & \text{if } n \text{ is even} \\ RMSE_{rs((n+1)/2)} & \text{if } n \text{ is odd} \end{cases}$$

$$MAD = \text{med } |RMSE_{rk} - \text{med } RMSE_r|$$

Where n is the number of points, μ is the mean value of $RMSE_r$ for all n points, σ is the standard deviation of $RMSE_r$ for all n points, $\text{med } RMSE_r$ is the median value for the $RMSE_r$, $RMSE_{rk}$ is the k th index of $RMSE_r$, $RMSE_{rs}$ is $RMSE_r$ sorted by residuals, and $RMSE_{rs(n/2)}$ is the $n/2$ index of $RMSE_{rs}$.

16. For each point, check if the ZMAD of that point is greater than the ZMAD threshold after the neighborhood analysis (comes from the ODL). If the ZMAD is greater, the identified points are considered an outlier.
17. Generate the final statistics such as mean, median, RMSE, standard deviation, and MAD for all valid points after step 16. This is the full scene statistics.
18. The valid GCPs are ranked based on $RMSE_r$ depending on the ranks defined in the ODL. For example (in pixels):

$RMSE_r \leq 0.5$	Rank 1
$0.5 < RMSE_r \leq 1$	Rank 2
$1 < RMSE_r \leq 2$	Rank 3
$2 < RMSE_r \leq 3$	Rank 4
$RMSE_r > 3$	Rank 5

19. The scene is divided into rectangular regions. The number of regions are defined in the ODL parameter for the GVERIFY algorithm. Figure 11.2 shows a scene divided by three regions along the line and three regions along the sample direction.
 - a) Determine the four vertices of the image location in the product frame.
 - b) Construct the equation of the line for all four corners using the vertices.
 - c) Determine the spacing distance in line and sample direction by dividing the line and sample distance by the number of divisions along the line and sample direction.
 - d) Determine the vertices for each region using the starting coordinate (UL,LL), ending coordinate (UR,LR), slope, and intercept of the line from the left edge, right edge, top, and bottom edge of the image along with sample and line distances.

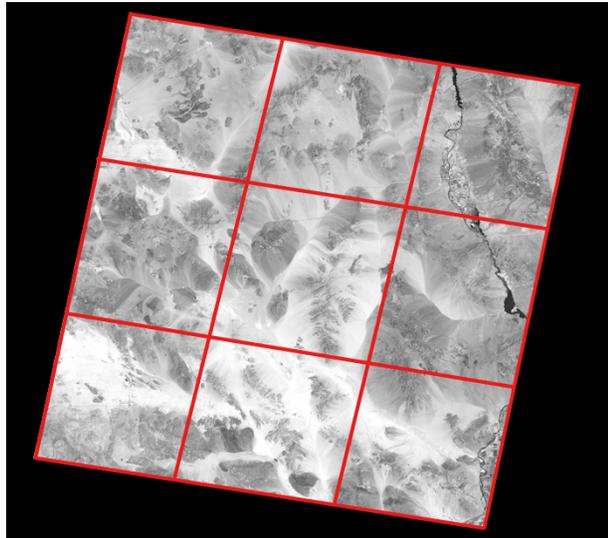


Figure 11.2. Scene Divided into 3×3 Regions

20. For each region, find all points that fall within the region and their statistics such as mean, standard deviation, RMSE, median for line and sample, and radial directions, number of points, and histogram of ranks.
 - a) To determine the points inside the region, check whether the line and sample coordinate falls between the region's corner line and sample coordinates.
 - b) The point statistics such as mean, standard deviation, and RMSE are all calculated by the formula as shown in step 15.
21. To determine the quadrant statistics and full scene statistics, follow the same procedure as listed in steps 19 and 20.

For the quadrant statistics, the regions are generated such that there are two regions in the line and two regions in the sample direction. This method generates a quadrant and the statistics within the quadrant are determined as in step 20.
22. Each valid GCP is assigned a color based on the rank, as in step 18 using a color LUT. These points are overlaid on the single band terrain corrected image to show the distribution of valid GCPs and their color-coded radial offsets across the image.

23. To create a browse jpeg image of the valid GCP overlaid on an L1T single band image, a three band Hierarchical Data Format (HDF) data set (input from step 22) is generated with the GCPs overlain. This HDF dataset is converted into a Portable Network Graphics (PNG) file, which is then reduced in size and converted to a JPG.

11.0.1 Known Limitations to GVERIFY Algorithm

The GVERIFY algorithm has limitations in accurately correlating the points from two images, particularly for MSS.

1. The GLS2000 Band 5 (1.55 - 1.75 μm) is used as a reference, whereas Band 7 of the MSS scene (0.8 - 1.1 μm) is used as a search image for correlation. These two spectral bands do not have any overlap and as a result may not provide accurate and consistent correlation peaks for different land features.
2. The reference and search image have varying native radiometric resolutions. Sensors collected the MSS scenes as 6-bit data, whereas GLS2000 (from ETM+) collected the scenes as 8-bit data.
3. The GLS2000 Band 5 scene has a native spatial resolution of 30 m, whereas the MSS Band 7 data have a native spatial resolution of approximately 80 m. The reference GLS2000 images are down sampled to 60 m to match the resolution of MSS products, which are also at 60 m. The varying native spatial resolution and radiometric resolution may lead to poor contrast on the MSS scene to provide an accurate correlation match with the GLS2000 scene.
4. The MSS scenes were acquired from different satellites (1972–1992), whereas GLS2000 data set was generated from ETM+ sensor scenes (1999–2003). The temporal variations between the two data sets could potentially cause misregistration or could detect temporal variations as true registration offsets.
5. Due to the radiometric resolution and spatial resolution of the MSS sensor, features (particularly thin clouds over land) are correlated incorrectly with the good contrast GLS2000 scene without clouds and thus may produce false registration offsets.

Appendices

Acronyms

ACS	Attitude Control System
ASCII	American Standard Code for Information Interchange
CPF	Calibration Parameter File
DEM	Digital Elevation Model
DOY	Day of Year
DoD	Department of Defense
ECI	Earth-Centered Inertial
ECR	Earth-Centered Rotating
EROS	Earth Resource Observation and Science
ETM+	Enhanced Thematic Mapper Plus
GCP	Ground Control Point
GCTP	General Cartographic Transformation Package
GLS	Global Land Survey
GMT	Greenwich Mean Time
GSL	GNU Scientific Library
HOM	Hotine Oblique Mercator
HRS	Horizontal Resampling Space

I2I	Image-to-Image
IAS	Image Assessment System
L0R	Level 0 Reformatted
L0Rp	Level 0 Reformatted Product
L1G	Level 1 Geometrically Systematically-Corrected
L1R	Level 1 Radiometrically Corrected
L1R-H	Hybrid Level 1 Radiometric
L1T	Level 1 Geometrically Precision/Terrain-Corrected
LOS	Line-of-Sight
LPGS	Level 1 Product Generation System
LUT	Lookup Table
MAD	Median Absolute Deviation
MMIO	Modified Moravec Interest Operator
MSCD	Mirror Scan Correction Data
MSS	Multispectral Scanner
NEOS	National Earth Orientation Service
NORAD	North American Aerospace Defense Command
ODL	Object Description Language
PNG	Portable Network Graphics
RMSE	Root Mean Squared Error
SGP4	Simplified General Perturbations
SLO	Scan Line Offset
SOD	Seconds of Day

SOM	Space Oblique Mercator
TAI	International Atomic Time
TM	Thematic Mapper
UPS	Universal Polar Stereographic
USGS	United States Geological Survey
UT1	UTC Corrected
UTC	Universal Time – Coordinated
UTM	Universal Transverse Mercator
VRS	Vertical Resampling Space
WGS84	World Geodetic System 1984
WRS	Worldwide Reference System

References

- [I7g, 2006] (2006). Landsat 7 Image Assessment System (IAS) geometric algorithm theoretical basis document. Technical Report LS-IAS-01. Version 1.0.
- [Adelson et al., 1984] Adelson, E. H., Anderson, C. H., Bergen, J. R., Burt, P. J., and Ogden, J. M. (1984). Pyramid methods in image processing. *RCA Engineer*, 29(6):33–41.
- [Claire, 1973] Claire, C. N. (1973). *State Plane Coordinates by Automatic Data Processing*. Number Publication 62-4. U.S. Department of Commerce Environmental Sciences Services Administration, Coast and Geodetic Survey. reprinted 1973.
- [GSL Contributors, 2010] GSL Contributors, G. P. (2010). GSL - GNU scientific library - GNU project - free software foundation (FSF). <http://www.gnu.org/software/gsl/>.
- [Kelso, 2007] Kelso, T. (2007). Validation of SGP4 and IS-GPS-200D against GPS precision ephemerides. Presentation. 17th AAS/AIAA Space Flight Mechanics Conference, Sedona, AZ.
- [mmiopaper, 2008] mmiopaper (2008). Extraction of GCP Chips from Geo-Cover Using Modified Moravec Interest Operator (MMIO) Algorithm. Technical report, USGS/EROS.
- [Park and Schowengerdt, 1983] Park, S. and Schowengerdt, R. (1983). Image reconstruction by parametric cubic convolution. *Computer Vision, Graphics and Image Processing*, 23(3):14.
- [Rengarajan et al., 2012] Rengarajan, R., Sampath, A., Storey, J. C., and Choat, M. J. (2012). Validation of geometric accuracy of global land survey (gls) dataset. Technical report.

[Snyder, 1981] Snyder, J. P. (1981). Space oblique mercator projection mathematical development. Geological Survey Bulletin, page 120. USGS, Washington, D.C., 1518 edition.

[Snyder, 1987] Snyder, J. P. (1987). *Map Projections-A Working Manual*. U.S. Government Printing Office, Washington, D.C.

[USGS, 1979] USGS (1979). Landsat data users handbook. Technical report. This binder contains notes pertaining to the revision of the Landsat Data Users Handbook, which are not included in the scanned image. The appendices (J and K) are not included in the scanned image, as they are recorded as separate records. The scanned images of these appendices are available in their individual records.