

Department of the Interior  
U.S. Geological Survey

# **LANDSAT 7 (L7) GAP PHASE STATISTICS ALGORITHM THEORETICAL BASIS DOCUMENT (ATBD)**

**Version 1.0**

**October 2007**





## **Executive Summary**

---

Due the failure in the Landsat 7 Enhanced Thematic Mapper Plus (ETM+) scan line corrector (SLC) mechanism, the ETM+ can no longer collect complete ground coverage for the entire Landsat 7 swath. Data processing techniques which use multiple SLC-off scenes to fill the coverage lost by the SLC failure have been developed. These techniques lead to a need for predicting how multi-scene scan gap interactions will increase the likelihood of complete gap coverage during processing. The gap phase statistic was developed to help measure this interaction.

# Document History

---

Document Number	Document Version	Publication Date	Change Number
LS-IAS-05	1.0	October 2007	5083

# Contents

---

<b>Executive Summary .....</b>	<b>iii</b>
<b>Document History .....</b>	<b>iv</b>
<b>Contents.....</b>	<b>v</b>
<b>List of Figures .....</b>	<b>vi</b>
<b>List of Tables .....</b>	<b>vi</b>
<b>Section 1 Introduction.....</b>	<b>1</b>
<b>Section 2 Gap Phased Statistics .....</b>	<b>2</b>
2.1 Background.....	2
2.2 Gap Phase Statistic Definition.....	3
2.3 Gap Phase Statistic Computation .....	3
<b>Section 3 Conclusions and Recommendations .....</b>	<b>22</b>
<b>Appendix A Glossary.....</b>	<b>23</b>
<b>References.....</b>	<b>24</b>

## List of Figures

---

Figure 2-1. Gap Phase Metric .....	2
Figure 2-2. Geometry of the Gap Phase Metric.....	3
Figure 2-3. "Fuzzy" Gap Probabilities for One Scene.....	18
Figure 2-4. "Fuzzy" Five-Scene Residual Gap Example .....	19
Figure 2-5. Residual Gap Phase Computation Example .....	21

## List of Tables

---

Table 2-1. DID Code Reuse .....	13
Table 2-2. gxxLib Code Reuse.....	14
Table 2-3. xxxLib Code Reuse .....	15
Table 2-4. New Gap Phase Modules .....	16

## Section 1 Introduction

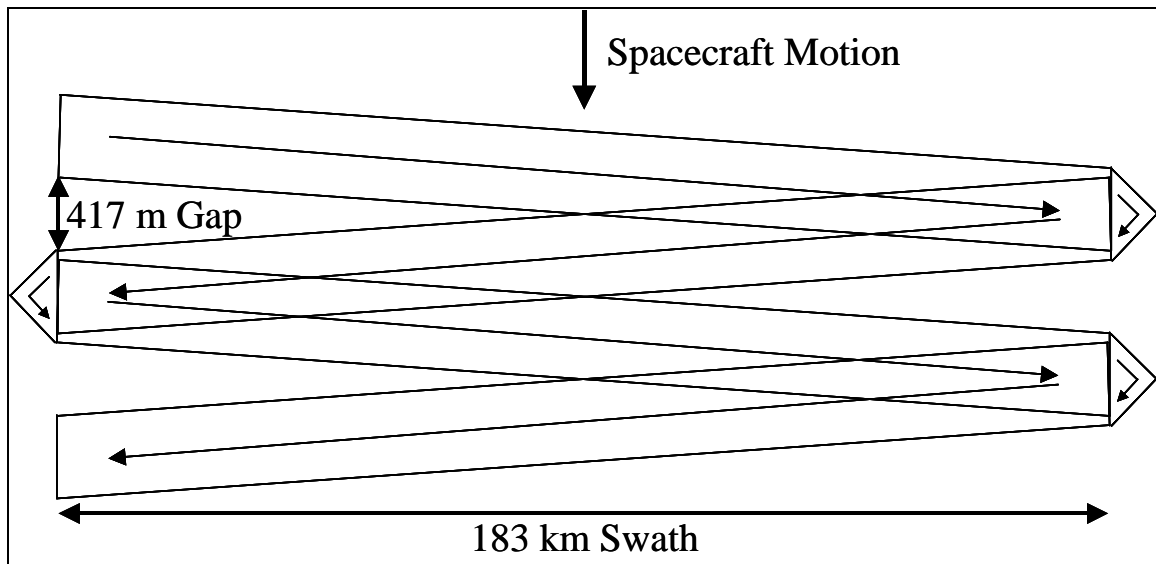
---

The Landsat 7 Enhanced Thematic Mapper Plus (ETM+) scan line corrector (SLC) mechanism was designed to compensate for along-track spacecraft motion during each active cross-track scan. The failure of the SLC on May 31, 2003 has made it impossible for the ETM+ to collect complete ground coverage for the entire Landsat 7 swath. Data processing techniques that fill the lost coverage by combining the data from multiple SLC-off scenes have been developed to improve the usability of ETM+ data acquired since the SLC failure [1]. When selecting the input scenes to include in this gap-filling procedure it is advantageous to have an indication of how the scan gaps in two or more scenes are likely to overlay. A method for predicting the multi-scene scan gap interactions were developed to enable users to more intelligently decide how many and which fill scenes to include in a gap-filled product. The term gap phase statistic was given this measure of the interaction between scenes. This paper describes the gap phase statistic definition, calculation, and implementation.

## Section 2 Gap Phased Statistics

### 2.1 Background

The Landsat 7 Enhanced Thematic Mapper Plus (ETM+) scan line corrector (SLC) mechanism was designed to compensate for along-track spacecraft motion during each active cross-track scan. The failure of the SLC on May 31, 2003 has made it impossible for the ETM+ to collect complete ground coverage for the entire Landsat 7 swath. Instead, the lack of along-track motion compensation causes the ETM+ to collect data in a zigzag pattern, with overlapping data at one end of the scan and scan-to-scan gaps at the other. This pattern is shown, with the vertical scale exaggerated, in Figure 2-1. These wedge-shaped gaps, varying in width from approximately 420 meters at the edge of a scene to zero at nadir, appear on alternate sides of the image. The gap between each forward scan and the following reverse scan appears on the west side (descending mode) of the image while the gap between each reverse scan and the following forward scan is on the east side.



**Figure 2-1. Gap Phase Metric**

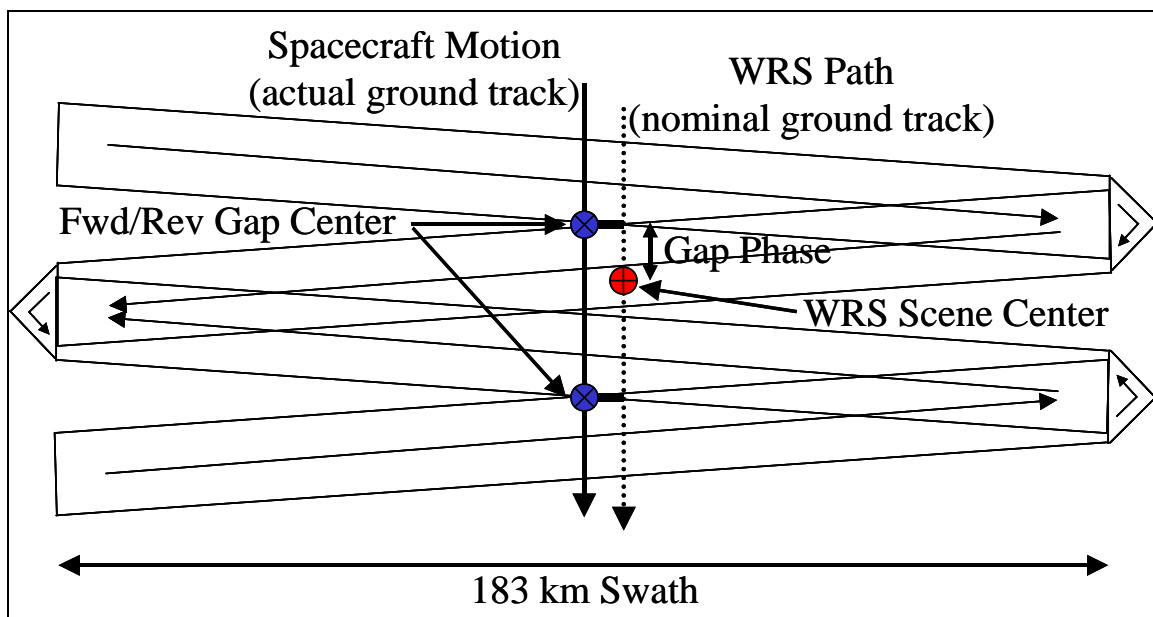
Data processing techniques that fill these scan gaps by combining the data from multiple SLC-off scenes have been developed to improve the usability of ETM+ data acquired since the SLC failure [1]. When selecting the input scenes to include in this gap-filling procedure it is advantageous to have an indication of how the scan gaps in two or more scenes are likely to overlay. Users will want to know whether the fill scenes they have selected can be expected to provide complete gap coverage in their primary scene. A method for predicting these multi-scene scan gap interactions will enable users to more intelligently decide how many and which fill scenes to include in a gap-filled product to increase the likelihood of complete gap coverage. The remainder of this paper describes a method for computing a gap phase statistic that defines a predicted reference location for each ETM+ SLC-off scene's scan gap pattern, and a procedure



for using the gap phase statistics from multiple scenes to predict the residual gap magnitude that would be expected in the corresponding gap-filled product.

## 2.2 Gap Phase Statistic Definition

The gap phase statistic for a Landsat 7 ETM+ SLC-off scene is defined as the along-track distance, in units of 30-meter pixels, from the center of the closest forward-to-reverse scan gap to the nominal worldwide reference system (WRS) scene center. Figure 2-2 shows the geometry of the gap phase metric. Note that gap phase is measured in the plane of the nominal WRS orbit. By defining the gap phase relative to a fixed reference point (the WRS scene center), the distance between gap centers for a pair of scenes can be computed by simply taking the difference of the scene gap phase statistics. This calculation becomes somewhat more complex when more than two scenes are involved. Residual gap prediction using more than two scenes is discussed in detail below.



**Figure 2-2. Geometry of the Gap Phase Metric**

## 2.3 Gap Phase Statistic Computation

In order to predict the location of each scene's scan gaps, instrument timing information and spacecraft position and pointing data are required. The accuracy of the predicted gap phase will be limited by errors in this supporting spacecraft data. Anything that improves the reliability of the input spacecraft ephemeris (position/velocity) and attitude (pointing) data will directly benefit the utility of the computed gap phase statistics. This has several implications for the computation and use of the gap phase statistics:

1. To ensure that valid inputs are used, the supporting instrument and spacecraft telemetry data must be quality checked, outliers must be detected

- and removed, and missing data due to outliers or data gaps must be filled with plausible replacement values.
2. When more accurate sources of supporting information are available (e.g., definitive ephemeris) they should be used in the gap phase computations in order to achieve the most accurate gap location predictions possible.
  3. Even with clean data and definitive ephemeris, residual errors, particularly in the spacecraft attitude knowledge, will lead to significant (~2-3 pixel) uncertainty in the gap phase statistics. Subsequent predictions of multi-scene gap coverage must take this uncertainty into account.

These considerations guided the development of the gap phase statistic computation method described in the following sections. Note that the data preparation operations implied by item #1 are a significant component of the overall algorithm.

#### *Required Input Data*

The inputs required to perform the gap phase statistic computation include the primary non-image supporting data components contained in each Level 0Ra (Level 0 archive) data set [2]. These include data contained in the wideband data stream received from the Landsat 7 spacecraft - the Level 0Ra payload correction data (PCD) and mirror scan correction data (MSD) files; calibration parameters generated by the Landsat 7 Image Assessment System (IAS) and used by the other elements of the Landsat 7 ground system - the calibration parameter file (CPF) [3]; and data generated during Level 0 processing on the Landsat Processing System (LPS) - the Level 0Ra metadata file (MTA). Although these four data inputs: PCD, MSD, MTA, and CPF, are sufficient to compute the gap phase for each full scene in a Level 0Ra subinterval, improved gap phase reliability can be realized if externally provided definitive ephemeris data are also used. Unfortunately, the Landsat 7 Mission Operations Center (MOC) provides these data 24-36 hours after the acquisition of real-time ETM+ data at the Landsat Ground Station (LGS). Thus, definitive ephemeris will not normally be available at Level 0 data processing time for real-time and recent recorder playback data. This will introduce a time lag between the generation of the Level 0Ra inputs and the computation of the corresponding gap phase statistics. Fortunately, only the low volume PCD, MSD, MTA, and CPF files from one of the two ETM+ data formats must be retained for gap phase computation.

#### *Data Preparation*

The LPS generates Level 0Ra PCD and MSD files from the raw Landsat 7 data stream. These data, particularly the PCD, contain numerous artifacts that prevent their direct use in a geometric instrument/spacecraft pointing model. Examples include:

- Fill data, inserted at the beginning and end of the PCD interval to form complete PCD major frame cycles, which must be replaced with reconstructed values to avoid discontinuities in the ephemeris and/or attitude data.
- Gyro resets that, if uncompensated, generate large apparent discontinuities in the attitude data record.

- Transmission errors that corrupt the content of PCD and/or MSD data fields, requiring outlier detection and correction.
- Clock drift errors that must be compensated in the PCD and MSD using the clock correction parameters contained in the PCD.

Fortunately, the algorithms and software required to check and clean the PCD and MSD have already been implemented as part of the IAS and the Landsat Product Generation System (LPGS) Level 0Rp data ingest logic. Since the algorithms applied to detect and correct problems in the PCD and MSD data are described elsewhere [4] only a summary of the data preparation checks applied to the incoming Level 0Ra is provided here. It should be noted that, in addition to correcting PCD and MSD errors where possible, the IAS quality checking software also creates a "report" structure that documents the locations of missing or erroneous data. Subsequent processes, such as the geometric model creation procedure, use this report structure to identify and repair or avoid problem data.

### PCD Processing

The PCD error detection and correction logic was adapted from the IAS ingest application (DID). The PCD for the entire Level 0Ra subinterval is loaded into a data structure, where the following operations are applied:

1. Check and, if necessary, fix the major frame IDs for all PCD fill major frames.
2. Range check the PCD data fields based on a predefined table of nominal values and thresholds. These nominal values and thresholds are read from a database on the IAS but have been hard coded in the prototype gap phase implementation.
3. Validate the ephemeris data - for each ephemeris state vector (position and velocity) compute the orbital semi-major axis, inclination, angular momentum, and eccentricity. If these orbital elements differ from their nominal values by more than the specified threshold, mark the ephemeris point as an outlier, using the report structure, for subsequent replacement by Lagrange interpolation.
4. Validate the attitude estimates (quaternions) - verify that each attitude quaternion estimate is a four-element unit vector by computing the root-sum-square of the quaternion elements. Replace bad quaternions with values linearly interpolated from the surrounding valid points.
5. Validate the gyro data -range check the gyro samples, check for and adjust gyro resets if any, and perform forward and backward predictions for each gyro value from its neighboring points. Replace points flagged as bad by interpolating from surrounding valid data.
6. Validate the gyro drift estimates - check for changes in the gyro drift values that exceed the predefined threshold. Replace bad points with the closest valid value.
7. Validate the angular displacement sensor (ADS) jitter data - range check the ADS samples and perform a second difference check on all

points. Replace bad points by linearly interpolating from surrounding valid data.

8. Validate the clock correction parameters - compute the magnitude of the clock correction and compare it to the predefined threshold. If the computed correction exceeds the threshold set the clock correction to zero.
9. Apply the clock correction to the PCD major frame time codes.

Other PCD checks, present in the IAS, that relate only to radiometric performance and/or trending data were omitted. The resulting corrected PCD structure is used for all subsequent computations.

### MSD Processing

The MSD for the full Level 0Ra subinterval is loaded and checked using logic adapted from the IAS ingest application. New logic was required to read the Level 0Ra HDF-EOS point-structured MSD into a data structure for subsequent processing. In the IAS context the Level 0Rp MSD are stored as an HDF Vdata structure and only the MSD records for the current product (up to a 3-scene interval) are provided. Fortunately, the quality checks developed for Level 0Rp data are equally applicable to a full Level 0Ra subinterval. These checks include:

1. Validate the scan direction - ensure that the scan directions alternate forward and reverse.
2. Validate the first half/second half scan error data - range check the first half and second half scan error values and compare both to the nominal values for the current scan direction, using the predefined thresholds to flag outliers. Replace bad points with the previous value from a scan of the same direction.
3. Validate the scan start time - compute the differences between adjacent scan start times and compare to the nominal total scan time, using the predefined threshold. Replace bad start times with the value predicted using the previous scan start time and the nominal total scan time.
4. Validate the counted line length (active scan length) data - range check the counted line length (the detected number of active scan minor frames) against the nominal line length and a predefined threshold. If the line length is bad, replace it with the theoretical line length computed from the first half and second half scan errors.
5. Apply the clock correction to the scan start times.

Other MSD checks, present in the IAS, that relate only to trending data were omitted. The resulting corrected MSD structure is used for all subsequent computations.

### Metadata and Calibration Parameter Processing

A subset of the Level 0Ra Object Definition Language (ODL) format metadata (MTA) file is loaded into a data structure for subsequent processing. Not all of the metadata fields are extracted, as only those fields necessary to identify and frame the WRS scenes in the subinterval are required. The key metadata fields include:

1. The total number of scenes in the subinterval.
2. The total number of ETM+ scans in the subinterval.
3. The WRS path and row location for each scene.
4. The scene-center scan number for each scene.

Since these data are generated by the LPS and are not subject to transmission errors, no outlier detection and correction algorithms are applied. Only ODL structural integrity checks are performed.

Similarly, the ODL calibration parameter file applicable to the Level 0Ra data acquisition date is loaded into a data structure. As is the case with the metadata file, only ODL integrity checks are performed. The CPF is not a component of the Level 0Ra data set but elements of the CPF are used during Level 0 processing. These files are therefore already being provided to the LPS.

### *Scene Processing*

The information read from the LPS metadata file is used to control a processing loop that cycles through each full scene in the subinterval to compute the corresponding gap phase statistic. The scene loop logic is as follows:

For each scene in the subinterval:

Compute the full scene start and stop scan numbers using the scene-center scan number and the size of a full scene (375 scans).

If the full range of scene scans is available in the MSD structure:

- Window the scene MSD records out into a separate scene MSD structure.
- Make a temporary copy of the full subinterval PCD structure.
- Use the scene MSD structure, the temporary PCD structure, the MSD/PCD quality flags, the CPF structure, and the user-provided ephemeris type (PCD or definitive) flag to construct a geometric model for this scene. This procedure is described in more detail below.
- Use the geometric model to project lines of sight for the ETM+ optical axis at the mid-scan time for the six central scans in the scene, to intersect the Earth's surface. The scan line of sight projection procedure is described in more detail below.
- Use the scan projections and the WRS path/row location to compute the scene gap phase statistic. The gap phase statistic computation procedure is described in more detail below.

- Report the scene WRS path, row, scene center time code, and the computed gap phase.

Else skip this partial scene and go to the next scene.

As noted above, the core components of this scene processing logic are described in the following sections.

### Geometric Model Construction

The geometric model construction logic is reused from the IAS model initialization (TMINIT) application. The model creation algorithms are described in detail elsewhere ([4] and [5]) but are summarized briefly here.

1. Initialize the geometric model data structure.
2. Extract valid ephemeris data from either the PCD structure or definitive ephemeris database, according to the user provided ephemeris type flag. Fill any missing ephemeris points using Lagrange interpolation.
3. Verify that the input CPF validity dates match the scene data.
4. Retrieve the Earth Centered Inertial (ECI) to Earth Centered Earth Fixed (ECEF) conversion parameters from the CPF.
5. Convert the ECI ephemeris data to ECEF and fit cubic interpolation polynomials to both the ECI and ECEF versions.
6. Load the MSD and the PCD sensor settings and attitude data applicable to the current scene into the geometric model structure.
7. Integrate the low frequency attitude estimates (quaternions), the intermediate frequency gyro data, and high frequency ADS data into a single high frequency attitude model.
8. Use the MSD to construct scan mirror profiles for each ETM+ scan in the scene.

The geometric model, when coupled with the CPF, now contains all of the information necessary to generate a sensor line of sight for any ETM+ detector as a function of time, convert this sensor line of sight to an Earth-fixed line of sight using the corresponding spacecraft position and orientation, and project that line of sight to find its intersection with the WGS84 Earth ellipsoid surface.

### Scan Projection

The scan line of sight projection logic is adapted from the IAS "forward model" library routines (gxx\_forward\_etm and gxx\_getlos). These computations use many of the ETM+ geometric instrument model routines in the IAS gxxETModel library. In this application, the imaging time and band/detector numbers are predetermined. The scene-center scan (SC) is initially set to scan index 187, the mid-point of the 375 scans in a scene. The scan index is incremented by 1 (to 188) if scan 187 is a reverse scan so that scan SC is always a forward scan.

The scan projection logic loops through the six scans surrounding the scene-center scan starting at scan SC-2 and continuing through scan SC+3. Due to the forward scan adjustment logic, these scans will always include three forward/reverse scan pairs. The following computations are performed on each of the six scans:

1. Compute the mid-scan time using the scan time code and first half scan time from the MSD (included in the geometric model). This is done using logic extracted from the `gxx_mirror_correct` library routine.
2. Set the detector number to 8.5 (the midpoint between detectors 8 and 9, the center of the array) and the band number to the code for the optical axis.
3. Compute the instrument along- and across-scan angles for this detector, band, and time combination using a modified version of the `gxx_find_angles` library routine that works with non-integer detector numbers.
4. Compute the spacecraft attitude at the mid-scan time using the `gxx_findatt` library routine.
5. Compute the spacecraft position and velocity at the mid-scan time using the `gxx_movesat` library routine.
6. Compute the instrument line of sight from the along- and across-scan angles using the `gxx_findlos` library routine.
7. Convert the instrument line of sight to an Earth-fixed line of sight using the spacecraft position and attitude, the CPF alignment matrices, and the `gxx_attitude` library routine.
8. Project the line of sight to find its intersection with the WGS84 Earth ellipsoid using a modified version of the `gxx_findtarpos` library routine. This modified version, effectively a complete rewrite of `gxx_findtarpos`, does not accept elevation offsets from the ellipsoid surface and returns a geocentric position vector instead of latitude, longitude, and distance from the geocenter.

The ground intersection point ECEF vector, the spacecraft ECEF position vector, and the imaging time for each of the six central scans are provided as inputs to the gap phase computation logic. The spacecraft position and imaging time are not required for the actual gap phase computations but were included in the prototype implementation for diagnostic purposes.

#### Gap Phase Computation

The gap phase computation procedure first calculates the reference location using the scene's WRS path and row. This computation uses the algorithm for computing the WRS scene center, given the path and row number, used by the IAS [4]. This algorithm was extended to express the WRS scene center as an ECEF position vector (**wrspos**) and to also calculate the nominal WRS angular momentum vector. This is accomplished as follows:

1. Compute the longitude of the nominal angular momentum vector by adding 90 degrees to the instantaneous descending node for the current path. This calculation uses the WRS\_Long and Delta\_Long variables defined in section 3.1.5.3.5 of reference 4:

$$AM\_Long = WRS\_Long + Delta\_Long + \pi/2$$

where:

AM\_Long is the angular momentum longitude,

WRS\_Long is the WRS scene center longitude,

Delta\_Long is the correction for scene center latitude.

2. Compute the geocentric latitude of the nominal angular momentum vector from the orbital inclination:

$$AM\_Lat = \pi/2 - \text{inclination}$$

3. Construct the ECEF angular momentum unit vector from the geocentric latitude and longitude:

$$\mathbf{angmo} = \begin{bmatrix} \cos(AM\_Long) \cos(AM\_Lat) \\ \sin(AM\_Long) \cos(AM\_Lat) \\ \sin(AM\_Lat) \end{bmatrix}$$

The **angmo** vector is used to project each scan center position vector (**tpos**) into the nominal WRS orbital plane as follows:

4. Subtract the component of **tpos** that is in the direction of the angular momentum vector:

$$\mathbf{tvec} = \mathbf{tpos} - (\mathbf{tpos} \bullet \mathbf{angmo}) \mathbf{angmo}$$

5. Adjust the length of **tvec** to make it a position on the WGS84 ellipsoid surface. First, compute:

$$mag = \left\| \begin{bmatrix} \frac{1}{a} & 0 & 0 \\ 0 & \frac{1}{a} & 0 \\ 0 & 0 & \frac{1}{b} \end{bmatrix} \begin{bmatrix} tvec.x \\ tvec.y \\ tvec.z \end{bmatrix} \right\|$$

where:

a and b are the WGS84 ellipsoid semimajor and semiminor axes.

Then:

$$\mathbf{tpos}' = \mathbf{tvec} / mag$$

Compute the distance from the adjusted **tpos'** to the WRS scene center vector **wrspos**:

6. Compute the angle between the vectors:

$$angle = \arcsin \left( \frac{\|\mathbf{tpos}' \times \mathbf{wrspos}\|}{\|\mathbf{tpos}'\| \|\mathbf{wrspos}\|} \right) \quad \text{in radians.}$$

7. Compute the arc distance between the vectors:



$$dist = angle * \|\mathbf{wrspos}\|$$

8. Use the angular momentum vector to adjust the sign so that *dist* is positive if **tpos'** is "before" **wrspos** along-track and *dist* is negative if **tpos'** is "after" **wrspos** along-track:

$$if((\mathbf{tpos}' \times \mathbf{wrspos}) \bullet \mathbf{angmo} < 0) dist = -dist$$

The along-track distances for each of the three forward/reverse scan pairs are averaged to compute the along-track distances to the centers of the three left-side scan gaps:

9. Three gap distances are computed:

$$gap\_dist[k] = ( dist[2*k] + dist[2*k+1] ) / 2 \text{ for } k=0,1,2$$

The gap distance with the smallest absolute magnitude is selected as the gap phase statistic, and is scaled from meters to pixels by dividing by 30:

10. Select the value with the smallest absolute magnitude:

$$min\_dist = gap\_dist[0]$$

$$if ( |gap\_dist[1]| < |min\_dist| ) min\_dist = gap\_dist[1]$$

$$if ( |gap\_dist[2]| < |min\_dist| ) min\_dist = gap\_dist[2]$$

11. Scale to pixels:

$$gap\_phase = min\_dist / 30$$

The gap phase, in pixels, is reported for each full scene in the interval.

### *Prototype Software*

The gap phase computation algorithm described above has been implemented in C language prototype software. Most of the code included in the prototype was derived or taken directly from the IAS ingest function (DID) or the IAS geometric (gxxLib) and general-purpose (xxxLib) libraries, with a limited amount of new development. The sources and extent of reused code will be described in more detail below as will the required supporting software environment.

### Environment

The gap phase prototype was built and run on a Silicon Graphics Origin 2000 server running the IRIX (version 6.4) operating system. In addition to the standard C libraries, the code requires the Oracle Pro\*C libraries to support an interface to the definitive ephemeris database tables. The Level 0Ra input functions also require several COTS (or COTS-like) external libraries, including:

1. HDF library - to read the HDF VData-structured PCD file. This library is used by the IAS.
2. HDF-EOS library - to read the HDFEOS Point-structured MSD file. Note that this is not a normal component of the IAS software environment, but should be available on the LPS.

3. ODL library - to read the LORa MTA file and the CPF. This library is used by the IAS.

The IAS general-purpose and geometric libraries are also used extensively by the prototype software:

1. xxxLib library - provides basic data types, data structures (e.g, PCD and MSD), and functions such as error reporting and time format conversions. The xxxLib also provides a Landsat-specific layer of code that simplifies access to the underlying HDF libraries. The xxxLib PCD, MSD, and MTA input routines were modified to accommodate LORa data.
2. gxxLib library - provides the geometric data structures (geometric model and calibration parameter file), and extensive geometric model computation functionality.

A complete analysis of those xxxLib and gxxLib routines invoked by the prototype has not been undertaken.

The prototype application is invoked from the command line with three command line parameters, as follows:

```
gap_phase <Root LORa File Name> <CPF Name> <Ephem Type>
```

where:

<Root LORa File Name> is the root (without .PCD or .MSD file extensions) path and file name of the input LORa files.

<CPF Name> is the input CPF file name.

<Ephem Type> is D for definitive and P for PCD ephemeris.

### Code Reuse Sources

In general, the components reused from the DID application were modified only to remove unneeded (e.g., radiometric) functions and to strip away database interactions (trending and error/event logging). These modified routines are included in the prototype source directory with their original names. In contrast, several gxxLib routines required functional modifications to work around Level 0Rp specific features or to change the routine interfaces. Due to the more extensive functional changes to these routines, they were renamed in the prototype source, replacing the "gxx" prefix with "gap". The reused code units derived from the DID application, the gxxLib, and the xxxLib are listed in Tables 1, 2, and 3, respectively.

Unit Name	Summary of Changes
did_AllocateReportMemory.c	Deleted database error logging.
did_CheckPCDRanges.c	Deleted ingest statistics gathering structure and logic.
did_FixMSCDTime.c	Deleted ingest statistics gathering and reporting logic and database error logging.
did_FixMajfldFill.c	Deleted database error logging.
did_FixPCDTime.c	Deleted ingest statistics gathering and reporting logic and database error logging.
did_FixPCDTimeFill.c	Deleted ingest statistics gathering and reporting logic and database error logging.
did_GetNominalStats.c	Rewrite of did_db_GetNominalStats.pc that hard codes nominal value and range check thresholds from the NOMINAL_LOR_STATS database table.
did_IngestStatus.c	Deleted DID-Log file generation logic.
did_LoadLORaData.c	Mutated (and renamed) version of did_PreProcess that serves as a driver to allocate the LORa data structures and read the LORa data files.
did_Math.c	Unchanged.
did_ModelAttitude.c	Unchanged.
did_ModelGyro.c	Unchanged.
did_TimeCorrection.c	Deleted ingest statistics gathering structure and logic and database error logging.
did_ValidateADS.c	Deleted trending data gathering logic.
did_ValidateAttitude.c	Deleted trending data gathering logic.
did_ValidateEphemeris.c	Deleted trending data gathering logic.
did_ValidateFHS_SHSError.c	Deleted trending data gathering logic.
did_ValidateGyro.c	Deleted trending data gathering logic.
did_ValidateGyroDrift.c	Deleted trending data gathering logic.
did_ValidateLineLength.c	Deleted trending data gathering logic.
did_ValidateMSCD.c	Deleted validation of number of minor frames received field. Added did_FixMSCDTime call originally found in did_IngestMSCD.
did_ValidatePCD.c	Deleted validation for instrument on time, temperature, and current telemetry. Added did_FixPCDTime call originally found in did_FindSceneBoundaries.
did_ValidateScanDirection.c	Deleted trending data gathering logic.
did_ValidateScanStartTime.c	Deleted trending data gathering logic.
did_ValidateTimeCorrection.c	Deleted trending data gathering logic.

**Table 2-1. DID Code Reuse**

The DID units did\_LoadLORaData.c, did\_ValidatePCD.c, and did\_ValidateMSCD.c are called from the gap phase main procedure. Together they comprise the interface to a Level 0Ra ingest, checking, and cleaning "miniDID" capability that could have application elsewhere.

<b>Prototype Unit Name</b>	<b>gxxLib Unit Name</b>	<b>Summary of Changes</b>
gap_align_attitude.c	gxx_align_attitude.c	Changed interface to pass in Report structure instead of LORp path. Deleted LOR Report file read logic. Deleted ADS trending statistics.
gap_findangles.c	gxx_findangles.c	Changed interface and added logic to allow for non-integer input detector number.
gap_findtarpos.c	gxx_findtarpos.c	Complete rewrite to operate only on ellipsoid surface intersections (allowing for direct solution instead of iteration) and to return ECEF vectors instead of lat/long/radius.
gap_init_gyro.c	gxx_init_gyro.c	Changed interface to pass in Report structure instead of LORp path. Deleted LOR Report file read logic.
gap_process_etm_model.c	gxx_process_etm_model.c	Changed interface to pass in Report structure instead of LORp path. Deleted ADS trending statistics.
gap_run_etmodel.c	gxx_run_etmodel.c	Changed interface to pass in PCD, MSCD, CPF, and Report structures directly, to bypass access via LORp HDF header file. Removed LORp read logic.
gap_valid_ephem.c	gxx_valid_ephem.c	Changed interface to pass in Report structure instead of LORp path. Deleted LOR Report file read logic.

**Table 2-2. gxxLib Code Reuse**

<b>Prototype Unit Name</b>	<b>xxxLib Unit Name</b>	<b>Summary of Changes</b>
gap_OpenEphem.c	xxx_OpenEphem.c	Changed interface to pass in ephemeris type, PCD, CPF, and Report structures directly, to bypass access via LORp HDF header file. Removed LORp read logic.
gap_db_ReadDefEphem.c	xxx_db_ReadDefEphem.c	Changed interface to pass in ephemeris start and stop time instead of LORp path.
xxx_OpenODL.c	xxx_OpenODL.c	Added the LPS MTA file name templates to the list of allowable ODL file types.
xxx_ReadMSCDlps.c	xxx_ReadMSCD.c	Rewrite of LORp VData reading version to implement LORa HDFEOS Point data reading logic.
xxx_ReadPCDlps.c	xxx_ReadPCD.c	Modified to read PCD VData from standalone input file instead of through HDF header file.
did_ReadMTA.c (derived from xxx_ReadMTA.c which reads all LORa metadata but renamed to preserve the original)	xxx_ReadMetadata.c	Modified to read the metadata from a standalone input file instead of through the HDF header file. Reads only the subset of the metadata fields needed for the gap phase procedure.

**Table 2-3. xxxLib Code Reuse**

New Development

Only a few entirely new modules were required to implement the gap phase prototype. These fall into three general groups:

1. Diagnostic routines - units borrowed from IAS utilities to dump the contents of the PCD, MSCD, and geometric model data structures for diagnostic purposes. These units are not required for the production version of the software.
2. Gap phase algorithm routines - new units that perform the WRS scene center processing, scan projection, and gap phase computation functions.
3. Main procedure - the main driver that controls processing and invokes the "miniDID", geometric model creation, and gap phase algorithm routines.

The new modules are listed in Table 2-4.

Prototype Unit Name	Summary of Purpose
dump_etmodel.c	Generate a text dump of the contents of the geometric model data structure.
dump_mscd.c	Generate a text dump of the contents of the MSCD data structure.
dump_pcd.c	Generate a text dump of the contents of the PCD data structure.
gap_phase.c	Main interface to the gap phase algorithm routines. Invokes the WRS scene center processing and scan projection units and implements the logic to compute the gap phase statistic from the projected scan center vectors.
gap_project_scan.	Implements the scan projection algorithm.
gap_wrsys.c	Implements the WRS scene center processing (scene center and angular momentum vector computation) logic.
driver.c	Main procedure that invokes the "miniDID", geometric model, and gap phase packages.

**Table 2-4. New Gap Phase Modules**

### Gap Phase Statistic Exploitation

The utility of the computed gap phase statistics depends upon their being available for and used in the gap filled product-ordering interface. Algorithms that show how the gap phase statistics can be used to predict the residual gap magnitude resulting from the combination of two or more SLC-off scenes are presented next to facilitate the effective exploitation of these data.

#### *Residual Gap Prediction*

Ignoring for the time being the uncertainty inherent in each scene's calculated gap phase statistic, it is straightforward to use these data to predict how the scan gaps from two SLC-off scenes will interact. If we assume that the selected primary scene has a gap phase of  $g_p$  and the first fill scene has a gap phase of  $g_{f1}$ , then the predicted residual (edge of scene) gap magnitude can be calculated as follows:

1. Adjust the fill scene gap phase based on the primary scene gap phase:
  - if  $(g_p - g_{f1} > 16)$   $g_{f1} = g_{f1} + 32$
  - if  $(g_p - g_{f1} < -16)$   $g_{f1} = g_{f1} - 32$
2. Compute the residual gap magnitude using the adjusted fill scene gap phase:
 
$$g_r = \text{MAX}( 0, (\text{MIN}( g_p+7, g_{f1}+7 ) - \text{MAX}( g_p-7, g_{f1}-7)) )$$

This calculation subtracts the largest gap "bottom" from the smallest gap "top" to compute the residual two-scene gap. It is straightforward to extend this calculation to multiple scenes. For a primary scene ( $g_p$ ) and N fill scenes ( $g_{f1}$  to  $g_{fN}$ ) the corresponding residual gap computation procedure is:

1. Adjust the fill scene gap phases based on the primary scene gap phase:
  - if  $(g_p - g_{fi} > 16)$   $g_{fi} = g_{fi} + 32$
  - if  $(g_p - g_{fi} < -16)$   $g_{fi} = g_{fi} - 32$
 for  $i=1..N$ .
2. Compute the residual gap magnitude using the adjusted fill scene gap phase values:

$$g_r = \text{MAX}( 0, (\text{MIN}( g_p+7, g_{f1}+7, g_{f2}+7, \dots, g_{fN}+7 ) - \text{MAX}( g_p-7, g_{f1}-7, g_{f2}-7, \dots, g_{fN}-7)) )$$

Unfortunately, the computed gap phase statistics are of limited accuracy. The same spacecraft position and pointing knowledge errors that limit Landsat 7 Level 1Gs product accuracy are applicable to the gap phase performance. Based on the recent (SLC-off) history of along-track L1Gs product geodetic accuracy measurements compiled and documented by the IAS, we would expect the gap phase estimates to be accurate to 75-80 meters when generated using definitive ephemeris data. The gap phase statistics, in units of 30-meter pixels, should therefore be thought of as having 2-3 pixel error bars. This uncertainty should be taken into account in the residual gap prediction computations to avoid presenting data users with an unrealistically optimistic impression of the likelihood of a gap-free final product.

### "Fuzzy" Gap Prediction

Converting the statistical error bounds on the individual gap phase values into corresponding bounds on the residual gap prediction is complicated by the non-linear modulo-32 and MIN/MAX functions employed in the residual gap calculation. A simpler approach treats each scene's scan gap as a probability distribution instead of a fixed interval. This approach effectively "smears out" the predicted scan gaps to create a set of "fuzzy" gaps. It is relatively simple to combine multiple "fuzzy" gaps to derive a prediction for the residual scan gap that accounts for the uncertainty in the gap phase estimates.

### Theory

The fuzzy gap model treats the scene gap phase values as a set of Gaussian random variables with means equal to the gap phase estimates and standard deviations equal to the uncertainty in the estimates. In initial testing a value of 3 pixels has been used for the standard deviation but this should be implemented as a tunable system parameter. For each input scene, the gap phase probability distribution can be used to compute the probability that any given along-track location falls within the scan gap. For scene  $i$ , with gap phase  $g_i$  and uncertainty  $\sigma$ , the probability that along-track point  $x$  (relative to the WRS scene center) falls in the scan gap is the cumulative probability that the actual gap center is within 7 pixels of  $x$ :

$$p_i(x) = \int_{x-7}^{x+7} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(u-g_i)^2}{2\sigma^2}} du$$

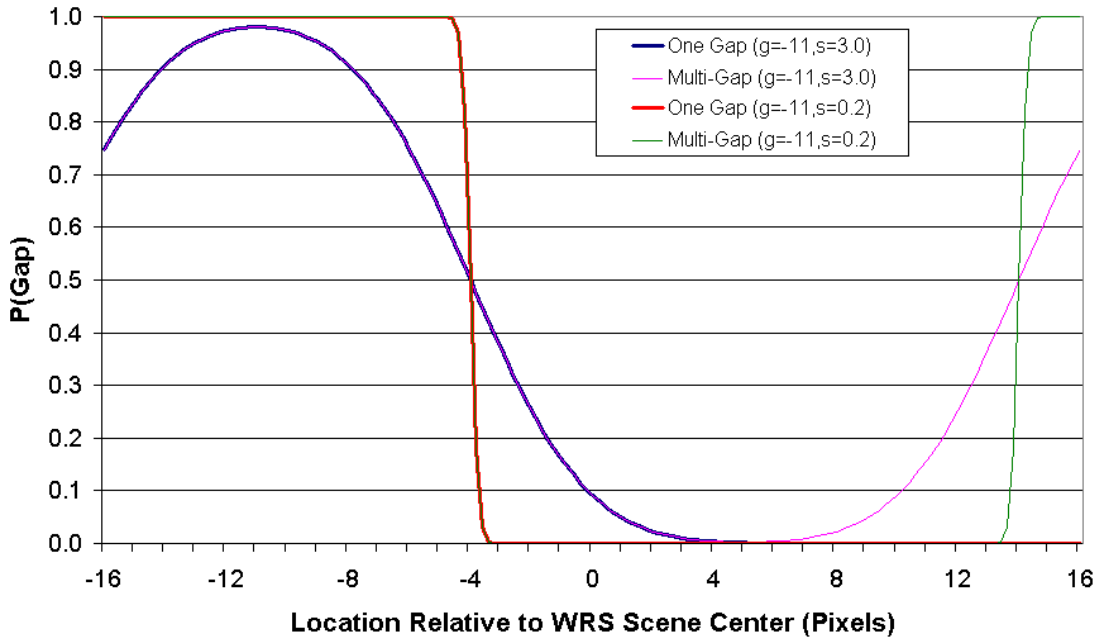
Actually, this is not quite correct because there is more than one scan gap in the scene. In fact, they repeat every two scans, nominally every 32 pixels. We therefore have to sum up the probabilities from all the scan gaps:

$$p_i(x) = \sum_j \int_{x-7}^{x+7} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(u-g_i+32j)^2}{2\sigma^2}} du$$

As a practical matter,  $\sigma$  is much less than 32 so we need only compute the summation for values of  $j=-1, 0, \text{ and } 1$ . Figure 3 shows both the single gap and

multi-gap probabilities for two cases: 1) a gap phase of -11 pixels and an uncertainty of 3 pixels; and 2) a gap phase of -11 pixels and an uncertainty of 0.2 pixels. The second case corresponds to the situation in which the gap phase is known almost exactly.

### "Fuzzy" Gap



**Figure 2-3. "Fuzzy" Gap Probabilities for One Scene**

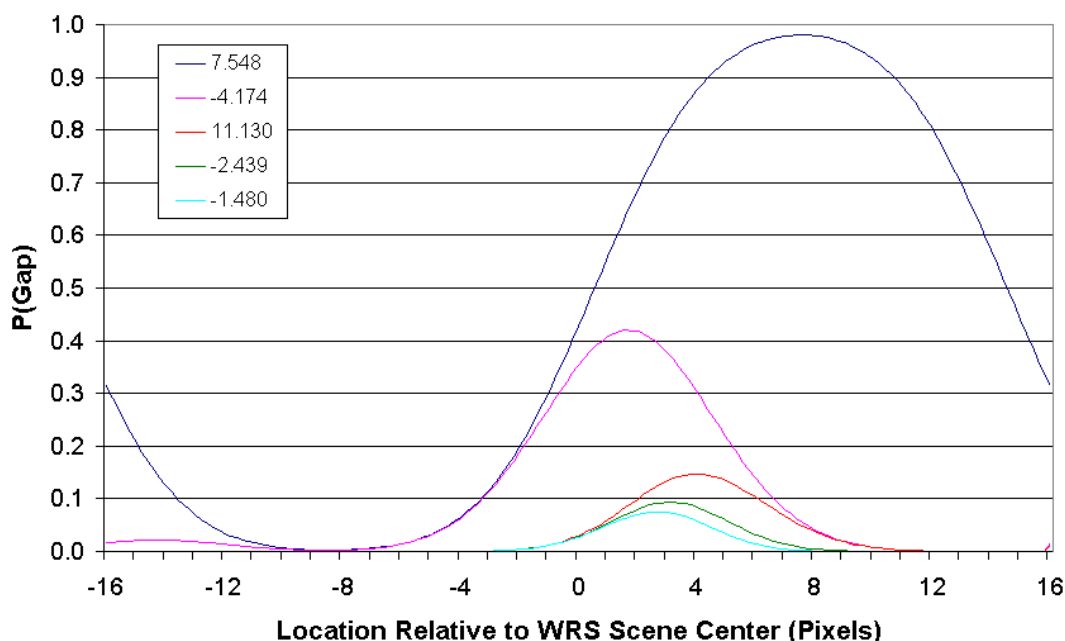
Treating the scan gap as a probability distribution makes it easy to combine the gaps from multiple scenes. Since we will only have a residual scan gap at a given point if all the input scenes have gaps at that point, the residual gap probability for N input scenes is the product of the individual scene probabilities:

$$p_r(x) = p_1(x) p_2(x) \dots p_N(x)$$

An example five-scene combination is shown in figure 4. The plotted lines show the impact on the predicted residual gap as each scene is added. The legend shows the gap phase values used for each scene in this example. Two noteworthy features of this figure are: 1) the gap probability does not go to zero even after all five scenes have been added; and 2) the predicted residual gap probability decreases even though the gap phase values for the 4th and 5th scenes do not extend the total range of gap phase values (they are both between the 1st and 2nd scene values). These are consequences of the "fuzzy" behavior of the residual scan gap model.



## "Fuzzy" Composite Gap



**Figure 2-4. "Fuzzy" Five-Scene Residual Gap Example**

These fuzzy gap distributions are based on the raw gap phase statistics. To provide a more consistent presentation of the gap geometry it is better to use the primary scene gap phase as the reference instead of the WRS scene center. We therefore subtract the primary scene gap phase,  $g_p$ , from each of the fill scene gap phase values ( $g_{fi}$ ) to compute a gap offset ( $g_i$ ) for each fill scene, taking care to keep the fill scene gap offset values within the range -16 to +16 by applying a modulo-32 correction:

$$g_i = \text{MOD}(g_{fi} - g_p + 16, 32) - 16$$

Since the primary scene is the reference for this offset calculation, the primary scene gap offset is zero, by definition.

To apply the fuzzy gap logic in the gap filled product user interface we want to reduce the information in these probability distributions to a single value that characterizes the magnitude of the predicted residual gap. This is accomplished by integrating the fuzzy residual gap probability distribution function over the two-scan (-16 pixels to +16 pixels) interval:

$$g_r = \int_{-16}^{16} p_r(x) dx = \int_{-16}^{16} \left\{ p_p(x) \prod_{i=1}^N p_i(x) \right\} dx$$

$$= \int_{-16}^{16} \left\{ \int_{x-7}^{x+7} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{u^2}{2\sigma^2}} du \prod_{i=1}^N \left( \sum_{j=1}^1 \int_{x-7}^{x+7} \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(u-g_i+32j)^2}{2\sigma^2}} du \right) \right\} dx$$

where:

- $p_r(x)$  is the residual gap distribution,
- $p_p(x)$  is the primary scene distribution (centered at 0),
- $p_i(x)$  is the distribution for fill scene  $i$ .

This calculation yields the expected value of the residual scan gap. For example, if we integrate the gap probability distribution for a single scene we obtain a value of 14 pixels. For the example shown in figure 4, the predicted five-scene residual gap is 0.3 pixels.

### Residual Gap Example

An Excel spreadsheet prototype of the fuzzy gap prediction algorithm has been implemented. This spreadsheet was used to generate the following example of residual gap prediction using a set of gap phase estimates computed for a sequence of SLC-off scenes acquired over WRS path 39 and row 37. Figure 5 reproduces a window extracted from the prototype spreadsheet.

A possible flow for the fill scene selection process would be for the user to initially select a single SLC-off scene as the primary input to a gap filled product. A query would then retrieve and present candidate fill scenes for that site. Preference would be given to scenes acquired within 4 WRS cycles of the primary scene, so that the fill scene gap frequencies would be relatively similar to the primary scene. Having selected a primary scene and retrieving the 8 closest-date acquisitions, figure 5 shows how the gap phase statistics could be employed.

The Date and Gap Phase columns show the input scene date and gap phase statistic values provided for each scene. The Selection column shows the scenes designated as primary and fill scenes. Initially only the Primary scene would be selected. The Gap Offset column is populated by subtracting the candidate fill scene gap phases from the primary scene gap phase, as described above. The gap offset for each candidate scene can then be used to compute the predicted 2-scene residual gap using the fuzzy gap algorithm. These results are presented in the 2-Scene Gap column. Having selected the 11/04/03 acquisition as the first fill scene, based on date similarity and cloud cover, we establish the predicted 2-scene residual gap as 3.3 pixels. The effect of adding each of the remaining candidate scenes to the scene pair is next calculated and presented in the 3-Scene Gap column. In this case, selecting the 10/03/03 scene only reduced the predicted residual gap to 2.6 pixels, due to the fact that this scene's gap offset is between that of the first fill scene and zero. Adding the 12/22/03 scene as the

third fill scene leads to a predicted residual gap of zero, as shown in the 4-Scene Gap column. The 08/16/03 scene would have achieved the same result.

Selection	Date	Gap Phase	Gap Offset	2-Scene Gap	3-Scene Gap	4-Scene Gap
	08/16/03	0.9	-12.8	2.3	0.0	0.0
	09/01/03	-9.0	9.3	5.0	2.6	2.0
	09/17/03	12.4	-1.4	10.4	1.6	1.5
<b>F2</b>	10/03/03	-16.1	2.2	10.2	<b>2.6</b>	2.6
<b>Primary</b>	10/19/03	13.8	0.0	14.0	3.3	2.6
<b>F1</b>	11/04/03	-6.8	11.4	<b>3.3</b>	3.3	2.6
	11/20/03	-10.1	8.2	5.9	2.8	2.2
	12/06/03	6.2	-7.6	6.4	0.2	0.2
<b>F3</b>	12/22/03	-2.2	-16.0	0.9	0.0	<b>0.0</b>

**Figure 2-5. Residual Gap Phase Computation Example**

As a general principle, fill scenes should be prioritized based on cloud cover and similarity of scene content relative to the primary scene. The predicted residual gap information is best used to decide how many, rather than which, fill scenes to use. It is also a good idea to always include a final SLC-on fill scene to ensure complete gap coverage.

## Section 3 Conclusions and Recommendations

---

The gap phase statistic will be a useful tool for selecting the input data for SLC-off to SLC-off (phase 2) gap filled products [1]. Although the gap location predictions are not perfect, when definitive ephemeris data are used, the results are accurate enough to provide useful guidance. The information provided by the fuzzy gap prediction algorithm should enable users to have a realistic expectation of the degree to which any given set of SLC-off scenes will provide complete ground coverage.

To reiterate a point made above, experience with the gap fill algorithm has shown that when setting the fill scene processing order it is always best to put the "best" scenes first, regardless of gap geometry. It is better to fill as much of the image as possible with the best available data than to immediately seek complete coverage using poorer (e.g., cloudier or older) data. The residual gap prediction will indicate the importance of selecting additional SLC-off or SLC-on scenes to provide complete coverage.

Several implementation recommendations follow from the experience gained in this algorithm development:

1. Definitive ephemeris (DE) data should be used to ensure that the gap phase statistics are as accurate as possible. This will introduce operational complications since the DE is not available at Level 0Ra generation time, but the improvement in gap phase accuracy should be worth it. Although the PCD ephemeris is often nearly as accurate as the DE, large (hundreds of meters) errors that would introduce blunders into the gap phase statistics are not uncommon.
2. The uncertainty parameter in the fuzzy gap phase algorithm should be implemented as a tunable system parameter, initially set to 3 pixels, so that it can be adjusted based on the performance of the residual gap predictions, if necessary.

## **Appendix A Glossary**

---

**Glossary Item:** <http://landsat.usgs.gov/resources/acronyms.php>

## References

---

Please see <http://landsat.usgs.gov/resources/acronyms.php> for a list of acronyms.

1. USGS EROS Data Center and NASA Goddard Space Flight Center, *SLC-Off Gap-Filled Products Algorithm Description*, Version 1.0, Greenbelt, Maryland, May 2004.
2. USGS EROS Data Center, *Landsat 7 Processing System (LPS) Output Data Files Data Format Control Book (DFCB)*, Revision 8, L7-DFCB-2, Sioux Falls, South Dakota, August 2002.
3. NASA Goddard Space Flight Center, *Landsat 7 System Calibration Parameter File Definition*, Revision 3, 430-15-01-002-3, Greenbelt, Maryland, April 1999.
4. USGS EROS Data Center, *Landsat 7 Image Assessment System (IAS) Geometric Algorithm Theoretical Basis Document*, Version 3.2, Sioux Falls, South Dakota, July 1998.
5. USGS EROS Data Center, *Enhanced Thematic Mapper Plus Scan Line Corrector Geometric Processing Algorithm Theoretical Basis*, Version 1.1, Sioux Falls, South Dakota, September 2003.