

SLC-off Gap-Filled Products Gap-Fill Algorithm Methodology

Background

The U.S. Geological Survey (USGS) Earth Resources Observation Systems (EROS) Data Center (EDC) has developed multi-scene (same path/row) gap-filled products to improve the usability of Enhanced Thematic Mapper Plus (ETM+) data acquired after the May 31, 2003 Scan Line Corrector (SLC) failure. The initial gap-fill capability employs a histogram-based compositing algorithm designed to combine an SLC-off image with one or more SLC-off and/or SLC-on scenes. This method was selected based (at least partly) on its simplicity and ease of rapid implementation.

A gap-fill algorithm trade study, conducted jointly by the USGS EDC and the Biospheric Sciences Branch at the National Aeronautics Space Administration (NASA) Goddard Space Flight Center, evaluated four candidate algorithms prior to the gap-filled product development. The goal of this study was to provide recommendations for the best candidate gap-fill method for implementation in the EDC production system and a description of the preprocessing (e.g., image registration) required to support the selected algorithm. In addition to selecting and defining the gap-fill algorithm, the study team recommended criteria that should be used to select input scenes (e.g., minimum date separation, closest to anniversary date, minimum cloud cover).

The methodology, results, and conclusions of the gap-fill algorithm trade study will be fully documented in a forthcoming journal article. The remainder of this document provides a description of the selected SLC-off gap-fill algorithm, a discussion of known limitations of the algorithm, a description of the process flow used to create multi-scene gap-filled products using the algorithm, and a set of suggested input scene selection criteria and considerations.

SLC-off gap-filled products were released in two phases. Phase 1 involved a local linear histogram matching technique that was chosen for rapid implementation. It allowed the user to choose one SLC-on scene for the fill image and was used for gap-filled products generated between 6/1/2004 to 11/18/2004. Phase 2 is an enhancement of the Phase 1 algorithm that allows users to choose multiple scenes and to combine SLC-off scenes. Phase 2 processing is used on all gap-filled products processed on or after 11/18/2004.

Multi-Scene SLC-off Gap-Filled Product Processing Flow

Figure 1 shows the data processing flow for multi-scene SLC-off gap-filled products. The scene merging process shown in the figure uses the gap-fill algorithm described below. The following subsections provide additional details regarding the image registration (Level 1Gp flow) and gap mask generation processes.

Phase 2 Gap-Fill Algorithm

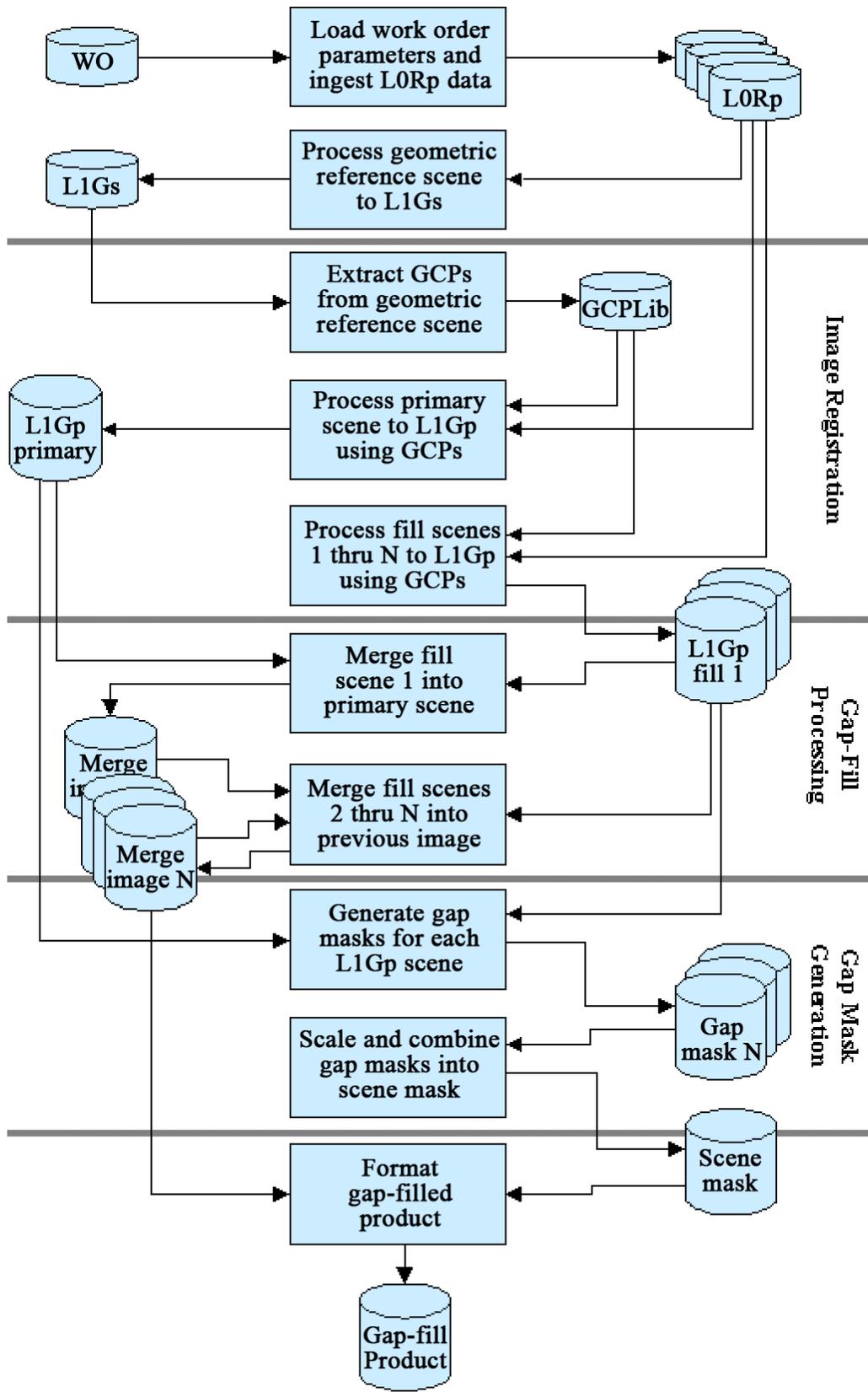


Figure 1: Multi-Scene SLC-off Gap-Filled Product Processing Flow

Image Registration

All of the scenes being integrated into a single gap-filled product must be geometrically registered prior to the radiometric adjustment and merging (gap-fill) step. The initial approach to the image registration problem was to treat the primary SLC-off scene as the geometric reference. Image chips were extracted from the primary scene for use as ground control points in the precision-correction processing of the SLC-on fill scene. Experience with the Phase 1 SLC-off to SLC-on image registration process demonstrated that the control point chip correlation process works better with fully populated (i.e., gapless) image chips. Because of this, the Phase 1 flow was redesigned to use the SLC-on fill image as the geometric reference and, hence, source of control point chips. This provides another motivation (in addition to guaranteeing complete gap fill) for recommending that an SLC-on scene is included at the end of the fill scene list in the multi-scene SLC-off gap-fill process. The SLC-on scene can be used as the geometrical reference, providing the benefits of gap-free control point chips.

If an SLC-on scene is available, it should be used as the reference for control point extraction as noted above. If no SLC-on scene is provided, then the primary SLC-off scene should be used as the reference.

Work order processing starts with the geometric reference scene. This scene is processed to a Band 8 Level 1Gs north-up Universal Transverse Mercator (UTM) image from which control point chips are extracted. These control points are stored in a library that is then used to process the primary scene and each of the fill scenes to Level 1Gp using the user-defined processing parameters. The control point correlation segment of the Level 1Gp process uses the Mutual Information (MI) correlation technique.

Gap-Fill Processing -- Phase 2 Algorithm Description

(For all gap-filled products generated on or after 11/18/2004)

Histogram matching (a simple procedure used to correct striping or banding in imagery) can be applied in a similar fashion to fill the scan gaps in SLC-off data. A transfer function is calculated that converts the radiometric values of one scene into equivalent radiometric values of a second scene. The transformed data are used to fill the gaps.

The scene that the user selects to be used for the bulk of the data is defined as the *primary* scene. All valid data within the primary will be retained. The scenes the user selects to fill in the gaps in the primary scene are referred to as the *fill* scenes. More than one fill scene may be necessary due to the gap geometry; in many cases, the gaps in the primary and fill scene will overlap, resulting in residual gaps in the *filled* product. Multiple fill scenes can be used to completely fill the remaining gaps.

The histogram-matching method requires co-registered L1Gs products for all scenes being used, including primary and all fill scenes.

The methodology for creating a histogram-matched composite is to iterate the following process over all gap pixels in the primary scene:

Phase 2 Gap-Fill Algorithm

1. Extract an $n \times n$ window about the gap pixel in both primary and fill. The Phase I algorithm extracted an $n \times n$ window, with the intent of using the entire window. The Phase II algorithm uses the smallest possible chip within the $n \times n$ window (step 2), but in order to change the code as little as possible, the read logic was maintained from Phase I. The window dimension n has been defined within the code as 31, to ensure that the window will completely span the worst-case gap overlap geometry caused by adjacent scan gaps. Larger windows were found to yield essentially identical results while imposing a significant performance penalty.
2. Find smallest square of pixels between the two chips that contains at least a minimum number of common pixels. Common pixels are those that are valid data in both the primary and fill scenes (see Figure 2). This creates a mask for both the primary and fill chip, distinguishing between valid data and invalid data (low-saturated data such as gaps or data dropouts, and high-saturated data such as clouds). Beginning with the smallest square (1 x 1 pixel), the number of valid common pixels between the primary mask and the fill mask is counted. If the minimum number of common pixels is not met, the square is expanded by one pixel on each side (i.e., 3x3 to 5x5 to 7x7...). The common pixels are counted again. Iterate until the minimum number of common pixels is met. If the minimum number of pixels is not met before reaching the maximum extent of the window, all available common pixels are used, regardless of the total.

The minimum number has been defined within the code as 144, so that the valid data content will be equivalent to the original (Phase 1) 17x17 fixed window with 50% valid data. A new mask is generated with only the common pixels of the chips denoted (for use in step 3).

3. Using the common pixel mask to extract only the common pixels from the chip, regress the primary chip against the fill chip to find the transform between the two scenes. This regression computes the least squares solution for the following over-determined system of linear equations:

$$\begin{bmatrix} 1 & fillDN_1 \\ 1 & fillDN_2 \\ \vdots & \vdots \\ 1 & fillDN_N \end{bmatrix} \begin{bmatrix} bias \\ gain \end{bmatrix} = \begin{bmatrix} primeDN_1 \\ primeDN_2 \\ \vdots \\ primeDN_N \end{bmatrix}$$

The least squares solution is given by:

Phase 2 Gap-Fill Algorithm

$$\begin{bmatrix} bias \\ gain \end{bmatrix} = \begin{bmatrix} N & \sum_{i=1}^N fillDN_i \\ \sum_{i=1}^N fillDN_i & \sum_{i=1}^N (fillDN_i)^2 \end{bmatrix}^{-1} \begin{bmatrix} \sum_{i=1}^N primeDN_i \\ \sum_{i=1}^N (fillDN_i)(primeDN_i) \end{bmatrix}$$

In this adaptive window cartoon illustration, black pixels are gap and white pixels are valid data. The images are geo-registered so that pixel (x,y) is the same location on the ground in both scenes. The gap pixels cover different locations on the ground in this example (a), leaving only 4 common pixels in this 3x3 chip. If this does not meet the minimum number of required common pixels, expand the chip by 1 pixel on each side (b), and count the common pixels again. Continue until the minimum number of common pixels or the maximum size of the window is reached.

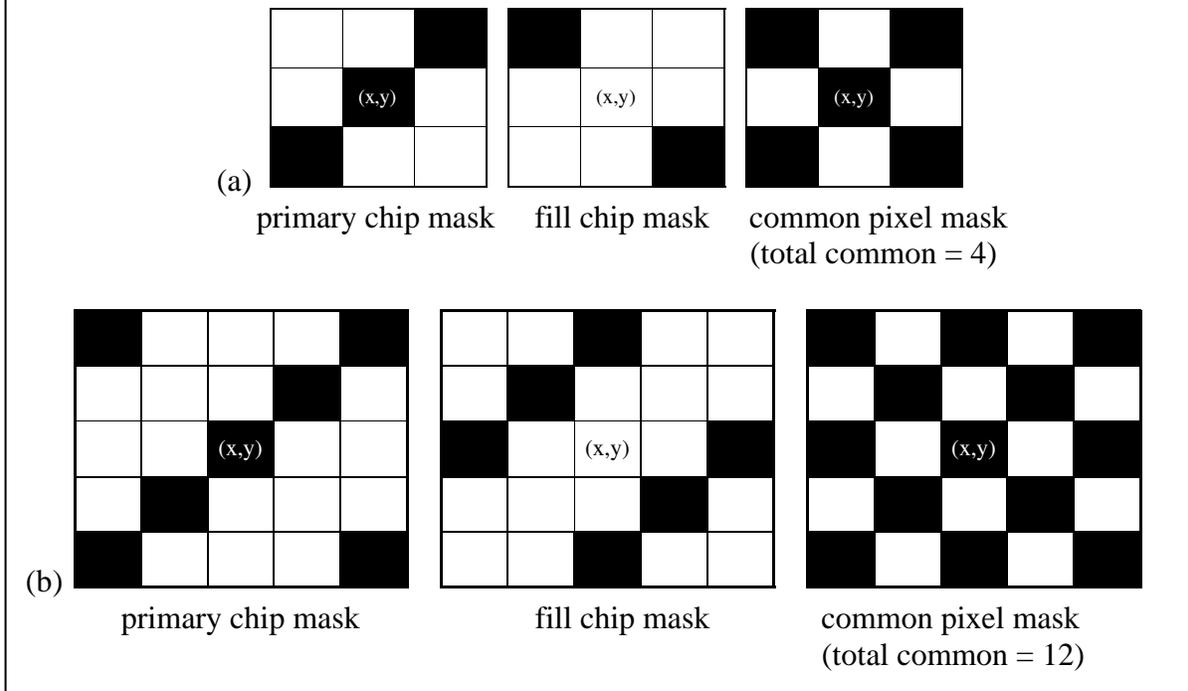


Figure 2: Adaptive Window Logic

Check for a “reasonable” gain to prevent outliers from having a strong effect on the transform. If the calculated gain is greater than the maximum allowable gain ($Mgain$) or less than the inverse ($\frac{1}{Mgain}$) of the maximum allowable gain, then calculate gain and bias using the mean and standard deviation:

$$gain = \frac{\sigma_P}{\sigma_F}$$

$$bias = \mu_P - \mu_F * gain$$

where:

Phase 2 Gap-Fill Algorithm

$$\begin{aligned}\mu_p &= \frac{1}{N} \sum_{i=1}^N \text{primaryDN}_i && \text{the primary image window mean} \\ \sigma_p &= \frac{1}{(N-1)} \sum_{i=1}^N (\text{primaryDN}_i - \mu_p)^2 && \text{the primary image standard deviation} \\ \mu_F &= \frac{1}{N} \sum_{i=1}^N \text{fillDN}_i && \text{the fill image window mean} \\ \sigma_F &= \frac{1}{(N-1)} \sum_{i=1}^N (\text{fillDN}_i - \mu_F)^2 && \text{the fill image standard deviation}\end{aligned}$$

If this gain is also greater than the maximum allowable gain (or less than its inverse), then set:

$$\begin{aligned}\text{gain} &= 1 \\ \text{bias} &= \mu_p - \mu_F\end{aligned}$$

No reasonable gain can be calculated, so perform a bias-only adjustment to the fill scene. The maximum allowable gain is an Object Description Language (ODL) parameter, currently set at 3.0, making the allowable gain range

$$\frac{1}{3} < \text{gain} < 3.$$

4. Calculate the new value for the fill pixel as a 16-bit number.

$$\text{mergedDN} = \text{fillDN} * \text{gain} + \text{bias}$$

Check that the filled value does not overflow the 8-bit scale. If it does, assign a value of 255 to the pixel in the filled scene. Otherwise, cast the filled value as an 8-bit number, and assign the calculated value to the pixel-filled scene.

Complete this procedure (shown as a process flow in Figure 3) for each pixel with a value of 0 in the primary scene. Note that this algorithm will also replace some normal “fill” data around the edges of the scene, where the fill scene coverage shifts relative to the primary scene, thereby expanding the extent of the primary scene. If gap remain in the merged scene, assign the merged scene to the “primary” title, take the second fill scene in the list of available scenes, and repeat the process.

Phase 2 Gap-Fill Algorithm

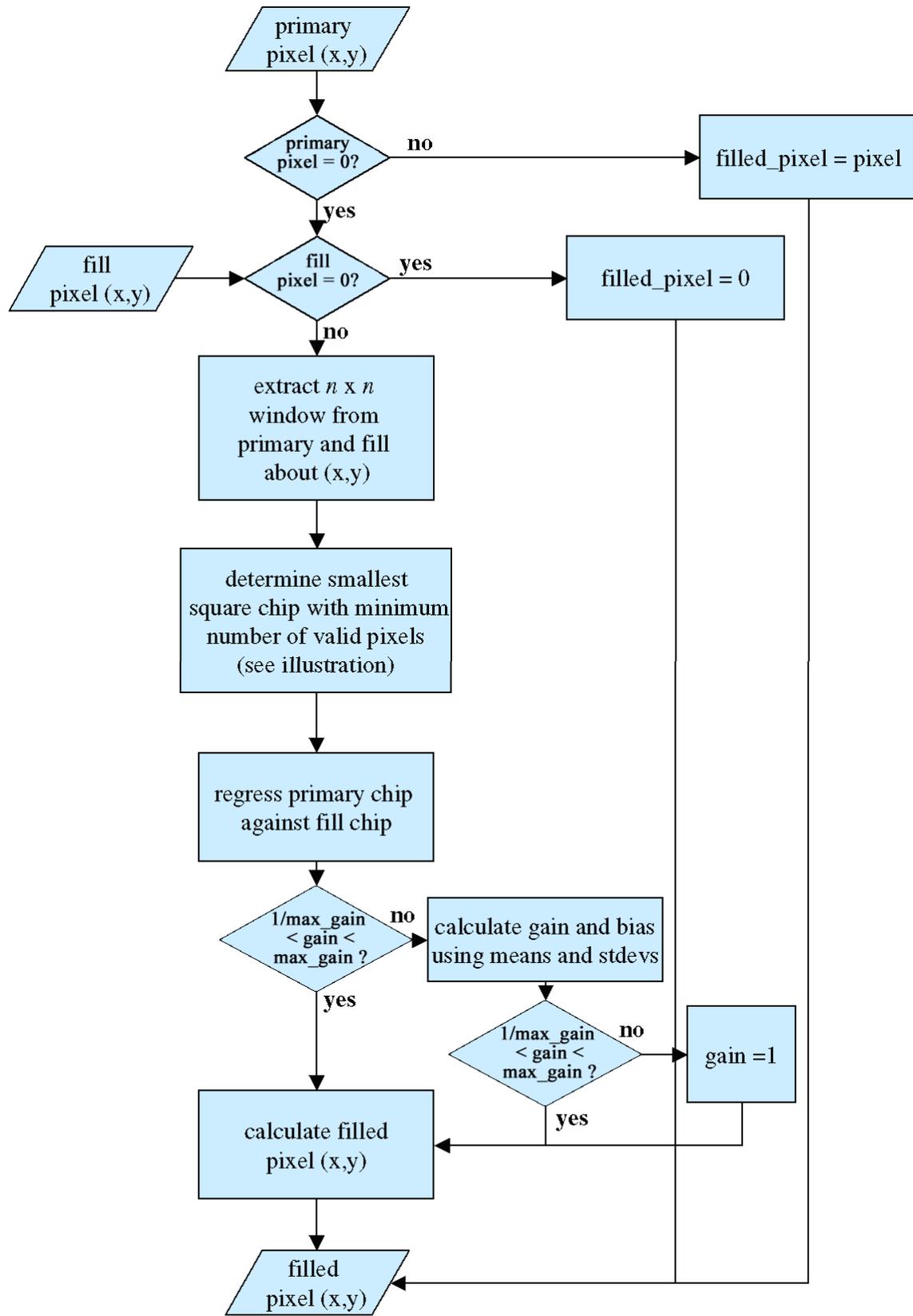


Figure 3: Histogram Matching Process

**Gap-Fill Processing -- Phase 1 Algorithm Description
(For products processed from 6/1/2004 to 10/13/2004)**

From 6/1/2004 to 10/14/2004, gap-filled products were made available from the USGS EDC using a simpler version of the histogram matching algorithm. While functionally similar to the Phase 2 algorithm, the Phase 1 algorithm allowed only a single SLC-on fill scene, restricted to data from before the SLC failure. Since 11/18/2004, all gap-filled products are created using the Phase 2 processing. This section is included for historical interest.

The Phase 1 algorithm required co-registered L1Gs products for both the primary and the fill scene. The Phase 1 methodology was to iterate the following over all gap pixels in the primary scene:

1. Extract an $n \times n$ window about the gap pixel in both the primary and the fill data. The window width was defined as 17 for the 30-meter bands, 11 for the Thermal bands, and 31 for the Pan band. These sizes were chosen to ensure that the window around a pixel in the center of a scan gap spans the gap.
2. Exclude from the common pixels any high or low saturated data. This step masks out the gap pixels as well as any high-saturated clouds in both the primary and fill data sets. Excluding clouds has been shown to significantly improve the gain calculation. After exclusions are made, the common pixels are counted. If there is less than a minimum number of pixels (currently defined as 2), then the fill data value is used to fill the gap pixel without modification.
3. Calculate a gain and bias using the mean and standard deviation of the common pixels:

$$gain = \frac{\sigma_P}{\sigma_F}$$

$$bias = \mu_P - \mu_F * gain$$

where:

$$\mu_P = \frac{1}{N} \sum_{i=1}^N primaryDN_i \quad \text{the primary image window mean}$$

$$\sigma_P = \frac{1}{(N-1)} \sum_{i=1}^N (primaryDN_i - \mu_P)^2 \quad \text{the primary image standard deviation}$$

$$\mu_F = \frac{1}{N} \sum_{i=1}^N fillDN_i \quad \text{the fill image window mean}$$

$$\sigma_F = \frac{1}{(N-1)} \sum_{i=1}^N (fillDN_i - \mu_F)^2 \quad \text{the fill image standard deviation}$$

No check for “reasonable” gain was performed in Phase 1 gap-fill processing.

Phase 2 Gap-Fill Algorithm

4. Calculate the new value for the fill pixel as a 16-bit number.

$$mergedDN = fillDN * gain + bias$$

Check that the filled value is within the unsigned 8-bit scale of 0 to 255. If it is not, assign a value of 0 or 255 to the pixel in the filled scene. Otherwise, cast the filled value as an 8-bit number, and assign the calculated value to the pixel-filled scene.

Gap Mask Generation

The gap masks accompanying gap-filled products to identify the source of each image pixel in the output product require somewhat more sophisticated processing in the case of products created using multiple fill scenes. For the Phase 1 SLC-off to SLC-on gap-filled products, it was sufficient to show those pixels taken from the SLC-off "primary" scene. All other non-fill pixels in the output product could be presumed to originate from the SLC-on "fill" scene. The Phase 1 SLC-off gap mask was able to provide the necessary information because it identified the primary scene pixels with 1s and all other non-populated pixels with 0s.

For the Phase 2 multi-scene SLC-off gap-filled products, the gap mask was modified to identify up to six source images (one primary image plus up to five fill images - limited by the number of fill image dates that can be listed in the gap-filled product metadata). Instead of containing only 0s and 1s, the Phase 2 mask will contain numbers ranging from 0 to 6, with the meanings shown in Table 1:

Value	Meaning
0	No Data
1	Primary Scene
2	Fill Scene #1
3	Fill Scene #2
4	Fill Scene #3
5	Fill Scene #4
6	Fill Scene #5

Table 1: Gap Mask Codes

The chosen method for generating the gap mask is as follows:

1. Generate gap masks for each input scene using the normal SLC-off gap mask generation method (i.e., resample the image a second time with the output DN range clipped to 0 to 1).
2. Multiply the gap masks for the fill scenes by the appropriate scene code from Table 1. (For example, multiply the mask for fill scene #3 by 4.)

Phase 2 Gap-Fill Algorithm

3. Merge the fill scene masks with the primary scene mask, clipping off any imagery falling outside the primary scene frame, in the order in which the scenes were merged:
 - a. Merge the fill scene #1 mask with the primary scene mask by replacing zeros in the primary mask with the value from the fill mask.
 - b. Merge the fill scene #2 mask with the mask resulting from step 3a by replacing zeros in the product mask with the value from the fill mask.
 - c. Repeat step 3b until all fill scene masks have been incorporated.
4. Repeat the above procedure for each spectral band and format the resulting image as a GeoTIFF product. This product should geometrically match the gap-filled image product.
5. Use the GNU zip (gzip) utility to compress the mask files.

This approach has the disadvantage of requiring a second resampling operation on each input image. This was deemed preferable to developing a new application to create gap masks directly from the product image during the SLC-off product development. Because new code will be required in any event to merge multiple gap masks, it may be better in this case to operate directly on the images to produce the scaled gap masks and perform the mask merging.

Known Algorithm Limitations

The adaptive local linear histogram adjustment algorithm can yield poor results if the scenes being combined exhibit radical differences in target radiance due, for example, to the presence of clouds, snow, or sun glint. Although attempts were made to include a cloud detection and masking capability in the Phase 2 gap-fill process, the problems associated with identifying thin clouds and cloud edges as well as cloud shadows proved to be too difficult to overcome in the available time.

Clouds present in the primary SLC-off scene tend to pull nearby fill scene data to high bias levels so that the fill scene data appear "cloud-like" even if the scene is, in fact, cloud-free. This creates a superficially reasonable-looking gap-filled image but discards any benefit to be gained from the cloud-free fill scene data in the primary scene cloud areas. Fill scene clouds also cause problems for the algorithm. In addition to inserting cloudy "wedges" into otherwise cloud-free primary scene imagery, where fill scene clouds overlap valid primary scene data, the gain and bias computations are corrupted for nearby pixels. When these bright fill scene pixels are compared to the corresponding darker primary scene pixels, the resulting gain and bias adjustments will tend to reduce the fill scene image DN value. This leads to dark areas in the fill regions in the neighborhood of fill scene clouds.

Similar effects can be expected wherever large changes in scene content and radiance occur, such as changes in snow cover or specular reflection off water due to changes in solar illumination geometry.

Phase 2 Gap-Fill Algorithm

Although the gap-fill algorithm can perform well when "adjusting" for temporal changes in homogenous regions such as agricultural fields, this algorithm has difficulty when the size of the regions exhibiting this change becomes too small. Fill data are typically better matched in the large agricultural fields typical of the American Midwest and West than in the smaller fields of the Northeast and, presumably, much of Europe and Asia. In general, features smaller than the local window size (nominally 19x19 pixels) are difficult to model if significant change has occurred.

Gap-Fill Scene Selection Criteria

The most important criteria for selecting fill scenes are those related to the similarity of image content between the primary and fill scenes. When selecting fill scenes, cloud cover, snow cover, and significant land cover changes are more important than predicted gap geometry. Experience with sample gap-filled products has shown that even a scene that is predicted to provide relatively little new ground coverage due to the similarity of its predicted gap location with the primary scene should be selected as the first fill scene if it provides the best cloud-free, similar date acquisition. Any new data such a scene provides is likely to be of high quality, which will provide valuable additional information for transforming subsequent fill scenes.

The following scene selection guidelines are roughly in order of priority:

1. Select fill scenes that are as free of clouds as possible and that contain as few obvious changes in scene content (e.g., different snow cover, water body ice cover/sun glint differences) as possible.
2. Select SLC-off scenes that are as close in time to the primary scene as possible to minimize changes in vegetation conditions. Failing this, select fill scenes (SLC-off or SLC-on) that are as close to an anniversary date as possible to take advantage of the similarities in seasonal vegetation cycles.
3. Select scenes that provide good predicted gap coverage based on the gap phase statistics. These statistics predict the locations of the scan gaps in each scene relative to the Worldwide Reference System (WRS) scene center. Computing the difference between the fill scene gap phase and the primary scene gap phase makes it possible to predict (to within a few pixels) the gap overlap geometry.
4. Select SLC-off fill scenes that are within +/- 4 WRS cycles of the primary scene, if possible. Since the ETM+ scan period increases slowly with time, the primary/fill scene relative scan gap phase will vary across the scene if the acquisition dates are too far apart. Thus, even an apparently "perfect" scan gap phase match will not provide complete gap coverage.
5. For Phase 2 products with multiple SLC-off scenes, including an anniversary date SLC-on scene as the final fill scene is recommended as a way of ensuring good image registration performance and providing complete gap coverage.

Phase 2 Gap-Fill Algorithm

Acronym List:

EDC	EROS Data Center
EROS	Earth Resources Observation Systems
ETM+	Enhanced Thematic Mapper Plus
GCP	Geometric Calibration Point
L0Rp	Level-0 Raw Product
L1Gp	Level-1 precision-corrected Product
L1Gs	Level-1 systematically-corrected Product
MI	Mutual Information correlation technique
NASA	National Aeronautics Space Administration
SLC	Scan-Line Corrector
USGS	U.S. Geological Survey
UTM	Universal Transverse Mercator map projection
WO	Work Order
WRS	Worldwide Reference System