1.

### Ground Control Point Correlation Algorithm

#### Background/Introduction

The Ground Control Point (GCP) correlation algorithm applies standard image matching techniques to measure the locations of a set of GCPs, each consisting of positional information and an image chip, within a Level 1GT OLI/TIRS image. For each measured GCP, the correlation status (success or failure) and the location within the image where the GCP was expected and where it was actually measured are reported.

The GCP correlation algorithm will be used in two different contexts in the L8/9 Image Assessment System. It will function as part of the primary Level 1TP (L1TP) product generation process, where it provides control point measurements for use by the OLI LOS Model Correction algorithm in registering the L1TP products to the GLS control base. Note on terminology: although geometrically readjusted and augmented with OLI-derived GCPs, the Landsat global control point library is still referred to as the GLS control base since it was originally extracted from the GLS2000 data set. The GCP correlation algorithm will also be used for geometric assessment as a tool for measuring the locations of validation GCPs in processed L1TP imagery. These measurements will be analyzed by the Geometric Accuracy Assessment algorithm for data product characterization purposes. High precision control derived from Digital Orthophoto Quadrangle (DOQ) data will also be used at geometric supersites for instrument and platform characterization and calibration.

The L8/9 GCP correlation algorithm was originally derived from the corresponding Advanced Land Imagery (ALI) algorithm used in the ALI Image Assessment System (ALIAS). Its implementation is very similar to the gcpcorrelate application. This is a utility algorithm that is not dependent on sensor architecture.

#### Dependencies

The GCP correlation algorithm assumes that GCPs are available for the ground site and that the Model Creation, LOS Projection and Gridding, and Image Resampling algorithms have been executed to create a systematic terrain-corrected L1GT image for GCP mensuration (for the LOS model correction application). This GCP mensuration image may be either SCA-separated or SCA-combined. In either case, the image must match the GCP image chips with respect to ground sample distance, map projection, and image orientation. As such, the band selection and resolution of the input image will depend on the flow being executed/control source being used. For standard L1TP product generation processing, the GLS control points (SWIR1 band, 30m resolution) will be used, whereas for characterization and calibration flows, the DOQ control points (panchromatic band, 15m resolution) will be used. For L1TP product geometric assessment, GLS GCPs flagged as validation points will be extracted and used. A limited set of thermal (ETM+ band 6, 60m resolution) GCPs will also be available to support contingency TIRS-only accuracy characterization operations. These thermal GCPs will use a source identifier of "TM6.”

##### GCP Retrieval

The GCP mensuration process relies upon a control point storage, management, and retrieval infrastructure. Though not formally part of the GCP correlation algorithm, the availability of logic to retrieve the GCPs corresponding to a particular L1GT image is a dependency of the algorithm. The Landsat 7 production system was the model for this capability and the source of the original GLS-derived operational GCPs. The GCP retrieval logic queries the GCP repository and requests GCP records based upon the following:

1. Geography - GCPs that fall within the latitude/longitude bounds of the L1GT image being correlated. Note that GCPs meeting this criterion could come from more than one WRS path/row, particularly at high latitudes.
2. GCP Source - As noted above, operational L1TP product generation uses GLS control, while the characterization and calibration operations would use DOQ control. Though not shown as an input to this algorithm in the table below, which takes the pre-assembled control package created by the GCP retrieval process as an input, this GLS, DOQ or TM6 control source selection would be a work order input. Note that there is no requirement to combine GLS, DOQ, and/or TM6 GCPs in a single control set.
3. GCP Type - GCPs will be flagged as "control" or "validation" points so that a subset of the available GCPs can be withheld from the image correction process to provide an independent basis for accuracy assessment (see note #4). Valid query options are as follows: CONTROL, VALIDATION, and BOTH. The CONTROL GCP type will be requested for all cases except the geometric accuracy assessment operation, which will use VALIDATION points. If a correlation failure occurs, a high-priority scene may be reprocessed using the BOTH option. This could help in cases where cloud cover has limited the set of usable GCPs in a particular scene. Under this scenario, if a scene was deemed as necessary for processing (high priority), for characterization, calibration, or other reasons, this scene would be processed through the IAS using the BOTH option.

The GCP retrieval process would extract the GCPs meeting the specified criteria from the GCP repository and construct a GCP data package/library for input to this algorithm. This data package would include the information presented in the Inputs section and in Table 6‑1. This ADD does not address the implementation details of how the GCP data fields and image chips are stored, managed, and packaged for delivery.

It is probably worth mentioning that the GCP retrieval query may return no valid GCPs. This will happen if, for example, DOQ or TM6 control is requested for an area where it does not exist, or GLS control is requested for a sea ice area or island/reef where control is not available. In this case, the processing system will have to be able to gracefully handle the lack of control (e.g., treat it as a correlation failure and proceed with systematic processing). This is outside the scope of this ADD.

For the prototype code, the database retrieval is mimicked in a two-step process. The first is with a Perl script that searches for all GCPs available online that fit within the image’s geographic extent. This script produces an ASCII file listing each valid chip (chips within image geographic extent) and relevant projection information, such as UTM zone. The second step is a C executable file that reads the ASCII file created from the Perl script and places all valid chips within a local directory. These chips are then resampled if they are not projected to the same UTM zone as the image file.

#### Inputs

The GCP correlation algorithm uses the inputs listed in the following table. Note that some of these “inputs” are implementation conveniences (e.g., using an ODL parameter file to convey the values of and pointers to the input data; including data set IDs to provide unique identifiers for data trending).

|  |
| --- |
| **Algorithm Inputs** |
| ODL File (implementation) |
|  Input GCP library/package name/link |
|  Level 1G Image file name |
|  Band to process |
|  Output GCP measurement file name |
|  Calibration Parameter File name (CPF contains default values for processing parameters) |
|  Options and Parameters |
|  Correlation result fit method (see note 2) |
|  Search window size (line, sample) in pixels |
|  Maximum allowable GCP displacement in pixels |
|  Minimum correlation strength (dimensionless) |
|  Image fill value to ignore in correlation  |
|  Predicted GCP location offset (line, sample) in pixels (optional) (see note #5) |
| GCP library/package contents: (see Table 6‑1 for details) |
|  Number of GCPs |
|  For each GCP: |
|  GCP ID |
|  GCP type (GLS, DOQ or TM6) (new) |
|  GCP ground position (lat/lon/proj X/proj Y/height) for each GCP |
|  Location of control point within image chip |
|  Chip parameters (e.g., size, ground sample distance (GSD)) |
|  Image chip (see note 1)  |
| Level 1G image contents |
|  Image data |
|  Image metadata (DDR), including the following: |
|  Image dimensions (number of lines and samples) |
|  Map projection |
|  GSD/pixel size |
|  Scene corner coordinates |

##### CPF Parameters

|  |  |  |
| --- | --- | --- |
| Parameter | Type | Description |
| GCP Correlation Window Size | Int | Correlation window size |
| GCP Minimum Correlation Peak | double | Minimum allowable correlation peak strength |
| GCP Maximum Displacement | double | Maximum allowable measured offsets |
| GCP Correlation Fit Method | Int | Correlation subpixel peak fit methodology |
| GCP Correlation Threshold | double | Threshold of allowable fill values in the correlation window |
| GCP Correlation Fill Value | double | Fill value to check for in the correlation window |

#### Outputs

|  |
| --- |
| GCP Measurements (see Table 6‑2 for details) |
|  GCP ID |
|  Nominal GCP chip line/sample |
|  GCP ground position (lat/lon/height) |
|  Predicted GCP image line/sample |
|  Measured offset from predicted line/sample |
|  Correlation success flag |
|  Correlation coefficient (new) |

#### Options

Correlation Fit Method (only one choice in baseline algorithm).

Note that the control source (GLS, DOQ, or TM6) will be selected by the infrastructure software that queries the control database and constructs the GCP library data package input to this algorithm. As such, it is not strictly an option within this algorithm, but it is an option that this processing step will select.

#### Procedure

This function correlates GCPs located in reference image chips to a terrain-corrected Level 1G image. Windows are extracted from the L1GT image to do the image correlation. The correlated points are written to the GCP data file for subsequent use in precision correction or product evaluation.

The heritage Landsat 7 implementation used L1G mensuration images that were not terrain corrected. The use of terrain-corrected images reduces the size of the L1G image region that must be searched for a control point match. It also requires that the measured GCP locations are associated with elevations interpolated from the Digital Elevation Model (DEM) used to perform the terrain correction. The LOS Model Correction ADD describes this (see 6.2.3).

##### Ground Control Point Correlation Algorithm Details

GCPcorrelate performs correlations on GCPs with a Level 1GT image.

###### Ground Control Points

GCPs and reference imagery are generated from USGS Digital Orthophoto Quadrangles (DOQs). DOQs are designed to meet national map accuracy standards at 1:24,000 scale, which corresponds to a Root Mean Squared Error (RMSE) of approximately 6 meters. A mosaic of DOQs is created by subsampling the 1-meter DOQ imagery to match the PAN band at a 15-meter resolution. Multiple DOQs are combined so that the mosaic covers a Landsat Worldwide Reference System (WRS) scene extent. The ground control chip library is generated by extracting 64x64 windows from the DOQ mosaic. Since the DOQ data are only available for the United States, these GCPs cover only U.S. test sites.

Ground control chips are chosen by stepping through the DOQ imagery at evenly spaced line and sample locations. Elevation for the chips are found from DEM and stored in the GCP library. If the DOQs that comprise the mosaic have large radiometric differences, histogram equalization operations may be performed. These histogram operations include histogram matching to a reference data set or histogram balancing within the mosaic.

GCPs have also been extracted from the Global Land Survey 2000 (GLS2000) data set. These GCPs serve as the geospatial reference for standard Landsat product generation and will be used for L8/9 standard product generation. The GLS2000 GCPs provide a globally distributed, internally consistent control set.

The global set of GLS chips were extracted from Band 5 (SWIR1) at 30m resolution. A limited set of thermal GCPs will also be extracted from the GLS2000 ETM+ band 6 images using a selected subset of GLS scenes. These thermal GCPs will be used if TIRS geometric characterization must be performed without reference to OLI data.

###### GCP Mensuration

Throughout the L8/9 data processing and characterization algorithms, normalized cross correlation is used to measure spatial differences between two image sources. These image sources could be OLI and DOQ, OLI and ETM+, TIRS and ETM+, or OLI and OLI. Image windows are extracted and correlation is performed over the windowed area. The correlation process will only measure 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 correlation will produce a discrete correlation surface (i.e., one correlation value per integer pixel offset location). A sub pixel location associated with the offset is found by fitting a polynomial around a 3x3 area centered on the correlation peak. The polynomial coefficients can be used to solve for the sub pixel correlation peak, taken to be the location of the polynomial function maximum. The normalized cross correlation process helps to reduce any correlation artifacts that may arise from radiometric differences between the two image sources.

If the two image windows of size NxM are defined by f and g, the mensuration

steps are as follows:

1) Perform normalized grey scale correlation.



where:





*R(x,y)* is the grey scale discrete correlation surface.

2) Find the peak of the discrete correlation surface by searching for the integer offset with the largest correlation coefficient.

3) Fit a 2nd order polynomial to a 3x3 area centered on the correlation peak. The polynomial has the form:



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

3a) Set up matrices



where:

X = correlation locations centered around peak

Y = correlation values corresponding to X locations

a = polynomial coefficients

Note that these matrices take the form:

$$\left[\begin{matrix}\begin{matrix}R(-1,-1)\\R(-1, 0)\\R(-1, 1)\end{matrix}\\\begin{matrix}R( 0,-1\\R( 0, 0)\\R( 0, 1)\end{matrix}\\\begin{matrix}R( 1, -1)\\R( 1, 0)\\R( 1, 1)\end{matrix}\end{matrix}\right]\_{9x1}=\left[\begin{matrix}\begin{matrix}1&-1\\1& 0\\1& 1\end{matrix}&\begin{matrix}-1& 1\\-1& 0\\-1&-1\end{matrix}&\begin{matrix}1&1\\0&1\\1&1\end{matrix}\\\begin{matrix}1&-1\\1& 0\\1& 1\end{matrix}& \begin{matrix} 0& 0\\ 0& 0\\ 0& 0\end{matrix}&\begin{matrix}1&0\\0&0\\1&0\end{matrix}\\\begin{matrix}1&-1\\1& 0\\1& 1\end{matrix}&\begin{matrix} 1&-1\\ 1& 0\\ 1& 1\end{matrix}&\begin{matrix}1&1\\0&1\\1&1\end{matrix}\end{matrix}\right]\_{9x6}\left[\begin{matrix}\begin{matrix}a\_{0}\\a\_{1}\\a\_{2}\end{matrix}\\\begin{matrix}a\_{3}\\a\_{4}\\a\_{5}\end{matrix}\end{matrix}\right]\_{6x1}$$

 3b) Solve for polynomial coefficients:

 

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





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





where:

x offset = subpixel offset in x direction

y offset = subpixel offset in y direction

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

The GCP positional information and the measured sub-pixel offset is recorded for each GCP, along with a flag indicating whether the final correlation value passed simple correlation strength and maximum offset thresholds. No statistics-based (e.g., t-distribution) outlier detection is performed by this algorithm.

##### Processing Steps

The basic GCP Correlation processing flow consists of the following steps:

1. Read the GCPs and L1GT image.
2. For each GCP:
	1. Compute the predicted location of the GCP in the L1GT image using the GCP map projection coordinates and any specified predicted offset.
	2. Extract an image window from the L1GT image at the predicted location.
		1. Make sure the image window contains sufficient non-fill image data.
		2. Make sure the L1GT image and GCP image chip are in the same map projection (UTM zone). Reproject (see below) the GCP chip if necessary.
	3. Correlate the GCP image chip with the L1GT window to find the optimum match point.
	4. Test the measured correlation and offset against predefined thresholds.
	5. Write out the GCP mensuration results.

The reprojection of the GCPs in the prototype code is done as a separate step through the process called gcpretrieve. This process is a precursor to the actual correlation process. It is also worth noting that the resampling methodology is slightly different between the Landsat heritage code and the prototype. The Landsat methodology used a table of weights that were applied to each chip in order to perform the reprojection.

The GCP correlation procedure was implemented in the heritage ALIAS prototype. Though the correlation process is conceptually simple, it is computationally intensive so the ALIAS implementation was designed to be efficient. This included taking advantage of parallel processing. These processing efficiency measures make the heritage implementation somewhat more complicated than it might otherwise be. The remainder of this processing discussion follows the L8/9 prototype, which was based on the heritage ALIAS implementation, to illustrate how the conceptually simple flow outlined above was mapped to a computationally efficient implementation.

##### Prototype Code

Input to the executable is an ODL file; output is an ASCII file containing measured offsets between the input image file and GCP library. Under the prototype output/input file directory, there is a directory called chips that contains the GCP data structures and files. The prototype output/input directory also contains the ODL files needed, the HDF5 input image file, a Perl script that is needed, and the CPF.

The following sections briefly describe the functions of the primary modules of the prototype implementation.

**Get GCP Correlate Parameters (get\_gcpcorrelate\_parameters)**

This function gets parameters from the ODL parameter file.

**Get GCP Information (get\_gcp\_information)**

This function reads the GCPs from the input GCP library.

**Process GCP (process\_gcp)**

This function processes all of the GCPs by extracting the GCP image chip, extracting the image window, performing the correlation for each point, and then writing the results to the GCP data file.

Notes:

1. The correlation routines want things in sample, line order, so the fit\_offset pairs returned are horizontal (sample) and then vertical (line) offsets. In contrast, the GCP data file contains fit\_offset in line, sample order.
2. To calculate the correlated location, know the following:
	1. The correlate routines return the offset from the reference window (chip) to the search window (L1GT), which is also the offset from the nominal reference point to the actual point in the search window.
	2. The integer location of the predicted location roughly corresponds to the integer location of the reference location. We need to report the predicted search line, sample of the reference point, and the offset from the predicted point to the correlated point. To get the correlated location, we start with the integer location of the predicted point because this corresponds to the integer part of the ref point (this is why gcp[num\_used\_gcp].fit\_offset subtracts the fractional part of the predicted location). Then, we add the fractional part of the reference coordinate because this is really the point we are going after. After that, we add the correlation fit\_offset because this tells how the reference point relates to the location in the search window.
3. The calculation for fit\_offset only works correctly because we are assuming the reference and search points are at the center of the window (plus some fractional distance), and the difference in window size is accounted for by nom\_off; if the reference point was not at the middle of the chip, this would have to be adjusted.

*Initialize Parallel Correlator (xxx\_init\_parallel\_correlator)*

This function initializes an instance of the parallel correlator. All of the multiprocessing resources are created, and the memory for the chip buffers and queue structures is allocated.

*Get Correlation Chip Buffers (xxx\_get\_corr\_chip\_buffers)*

This function returns buffers for the search and reference chips that will be submitted to the parallel correlator. Getting buffers from this routine and not submitting them to the parallel correlator will quickly exhaust all of the buffers available. The buffers will be freed when the parallel correlator is closed. When compiled in single-threaded mode, the same set of buffers is used for every pair of chips.

*Close Parallel Correlator (xxx\_close\_parallel\_correlator)*

This function is the routine that needs to be called after all of the chips have been submitted to the correlator. This routine will wait until all of the threads have completed, then destroy this instance of the parallel correlator. The results of the correlation are not valid until this routine returns.

*Get Search Line/Sample (get\_search\_line\_samp)*

This function finds the line and sample that corresponds to the given projection y and x. Since the L1GT image is positioned (map projection) north up, to find the line (sample), subtract the upper-left projection y(x) value from the GCP projection y(x) value and divide the result by the line(sample) pixel size.

Notes:

The line and sample numbers are 0-relative.

This will not work for a path-oriented image.

*Extract Window (ias\_misc\_extract\_window)*

This function extracts an image window around a specific GCP. From the input image, a window of the specified size will be extracted around the GCP line and sample. If the window is an odd size, the extra line and/or sample will be at the beginning of the imagery. The data in the window representing portions outside of the imagery will be filled with zeros.

The two steps to the extraction are as follows:

1. Data type conversion of whatever the L1GT image is to float
2. Setting the calculated window correctly into the buffer (even if the calculated window falls partially outside of the image)

*Check Fill (oli\_check\_fill)*

This function checks the input window for fill data over the specified percentage. This routine is useful to determine if there is too much fill data in a window. If too much fill data exists, then the window might not be good for correlating. Fill data nominally has a value of 0.0.

*Extract Chip ( xxx\_extract\_chip)*

This function reads the specified image chip. The image chip is always assumed to be a flat binary file containing chip\_size[0] x chip\_size[1] BYTE pixels (see note #1).

*Resample chip if necessary (build\_gcp\_lib)*

New logic, derived from the Landsat Product Generation System (LPGS) heritage, will be required here to check the image chip map projection, and if necessary, resample the chip to match the L1GT image. This is necessary when working with a global GCP repository containing GCPs extracted from multiple source scenes in multiple UTM zones. The GCPs falling inside of a particular L1GT image will frequently (particularly at high northern latitudes) be drawn from source images in neighboring UTM zones. It is also worth noting that the resampling techniques between the LPGS heritage code and the L8/9 prototype are not the same.

Image chip reprojection proceeds as follows:

1. Compute the image chip UL corner coordinates from the GCP UTM coordinates, the GCP image line/sample coordinates, and the image chip pixel size:
	1. UL Corner X = GCP X - GCP sample coordinate \* chip pixel size
	2. UL Corner Y = GCP Y + GCP line coordinate \* chip pixel size

Note that the GCP line/sample coordinates are relative to a zero-origin at the center of the upper-left chip pixel.

1. Project the GCP latitude and longitude to the L1GT map projection (UTM zone) using the projection transformation package (see OLI/TIRS LOS Projection ADD) to compute GCP X' and GCP Y' projected coordinates.
2. Compute the desired "new" chip UL corner in the L1GT projection using the new GCP X' and GCP Y' coordinates, rounding off to a whole multiple of the pixel size:
	1. UL Corner X' = GCP X' - GCP sample coordinate \* chip pixel size
	2. UL Corner Y' = GCP Y' + GCP line coordinate \* chip pixel size
	3. UL Corner X' = round(UL Corner X'/chip pixel size)\*chip pixel size
	4. UL Corner Y' = round(UL Corner Y'/chip pixel size)\*chip pixel size
3. Compute the "new" GCP line/sample coordinates in the reprojected chip:
	1. GCP sample coordinate' = (GCP X' - UL Corner X')/chip pixel size
	2. GCP line coordinate' = (UL Corner Y' - GCP Y')/chip pixel size
4. For each point in the new chip:

For line = 0 to nlines-1

Compute: Y' = UL Corner Y' - line\*chip pixel size

For samp = 0 to nsamps-1

1. Compute: X' = UL Corner X' + samp\*chip pixel size.
2. Convert (X',Y') to old chip projection (X,Y) using the projection transformation package.
3. Compute: oline = (UL Corner Y - Y)/chip pixel size.
4. Compute: osamp = (X - UL Corner X)/chip pixel size.
5. If the point (oline, osamp) falls inside of the old chip boundary, then interpolate a DN value at that location using bilinear interpolation:
	1. lindex = (int)floor(oline)
	2. sindex = (int)floor(osamp)
	3. fline = oline - lindex
	4. fsamp = osamp - sindex
	5. DN(oline,osamp) =

DN(lindex,sindex)\*(1-fline)\*(1-fsamp) +

DN(lindex+1,sindex)\*fline\*(1-fsamp) +

DN(lindex,sindex+1)\*(1-fline)\*fsamp +

DN(lindex+1,sindex+1)\*fline\*fsamp

Else DN(oline,osamp) = 0

5. Use the reprojected image chip and GCP line/sample coordinates in the GCP correlation procedure.

The build\_gcp\_lib, or gcpretreive process, is separated from the GCPcorrelate process to emulate the GCP retrieval process from the database containing the GCP image chips and their corresponding metadata. This retrieval process also contains the following C modules:

 *GCPretrieve -Main driver for GCP retrieval process.*

 *get\_gcp\_lib - Reads GCPs according to set of criteria*

 *get\_gcp\_information - Wrapper for reading GCPLib information*

 *get\_gcp\_proj\_parms - Reads projection information from the image file metadata*

 *get\_gcpretrieve\_parameters - Read input ODL parameters*

This code was based on the Landsat ETM+/TM heritage code for GCP retrieval and chip reprojection.

*Put GCP (io\_put\_gcp)*

This function writes all GCP records to the specified output file. This function writes out a set of GCP data records. If the file already exists, it will be overwritten.

***Write GCP (io\_write\_gcp)***

This function writes one record to the GCP data file. The file pointer is left at the end of the current record so sequential calls of xxx\_write\_gcp will sequentially write all of the records in the file. The GCP data file is assumed to be an ASCII file containing one line of text per GCP data record. Each record contains: point\_id, reference\_line, reference\_sample, latitude, longitude, elevation, predicted\_search\_line, predicted\_search\_sample, delta\_y (line), delta\_x (sample), accept/reject\_flag, correlation coefficient, reference band number, search band number, search SCA number (0 for SCA-combined images), chip source (DOQ, GLS).

*Submit Chip to Correlator (xxx\_submit\_chip\_to\_corr)*

The xxx\_parallel\_corr module implements a parallel correlation object. Using the Posix threading interface, up to MAX\_CORR\_THREADS (or the number of processors available - whichever is less) are created to perform correlation. The main thread that creates the parallel correlator is then responsible for "feeding" the parallel correlator chips to correlate. The xxx\_submit\_chip\_to\_corr places the chips into a queue. The correlation threads remove the chips from the queue and perform the correlation. The results of the correlation are not immediately available to the application since xxx\_submit\_chip\_to\_corr returns before the correlation is complete.

Before any of the correlation results are used, the application must call xxx\_close\_parallel\_correlator to make sure all of the chips have been correlated and to destroy the correlation threads.

***Grey Correlator ( xxx\_grey\_corr)***

This function correlates a reference subimage with a search subimage using the pixel grey levels, and evaluates the results.

**Grey Cross Product Same-size (xxx\_grey\_cross\_ss)**

This function computes the unnormalized (raw) sum of pixel-by-pixel cross products between the reference and search images for every combination of horizontal and vertical offsets of the reference relative to the search image for windows of the same size (in one dimension at least).

**Grey Normalization Same-size (xxx\_grey\_norm\_ss)**

This function converts raw cross-product sums to normalized cross-correlation coefficients, using tabulated statistics from previous step (grey\_cross\_ss). This function is much simpler than the one for unequal-sized windows, since all normalizing is done by the space domain same size correlator. All that has to be done is statistics gathering to set up the peak finder.

**Grey Cross Product (xxx\_grey\_cross)**

This function computes the unnormalized (raw) sum of pixel-by-pixel cross-products between the reference and search images for every combination of horizontal and vertical offsets of the reference, relative to the search image. This function works for windows of unequal size.

**Grey Normalization (xxx\_grey\_norm)**

This function converts raw cross-product sums to normalized cross-correlation coefficients, while tabulating statistics needed for subsequent evaluation. This function works for unequal window sizes.

**Grey Evaluation (xxx\_grey\_eval)**

This function evaluates various measures of correlation validity and extracts a subarea of the cross-correlation array centered on the peak.

**Fit Registration (xxx\_fitreg)**

This function fits a quadratic surface to the neighborhood of the correlation peak and from it determines the best-fit registration offsets and their estimated errors.

##### Input and Output File Details

The details of the fields contained in the input GCP library file (Table 6‑1) and the output measured GCP file (Table 6‑2) are as follows:

|  |  |
| --- | --- |
| **Field** | **Description** |
| Header Text | Zero or more lines of ASCII text, each line beginning with the "#" symbol to designate it as a header comment, describing GCP library contents. |
| Data Start Marker | "BEGIN" - static text to indicate beginning of data area |
| Number of GCPs | Integer number (N) of GCPs to follow (new) |
| GCP Record Fields: | One set per GCP |
|  GCP Number | Sequence number of GCP in this package (1 to N) |
|  GCP ID | Unique ID for GCP of the form: ppprrrnnnn where: ppprrr = WRS path/row GCP was taken from nnnn = chip sequence number |
|  GCP Image Chip Line Coordinate | GCP location within image chip - line coordinate (fractional pixel). |
|  GCP Image Chip Sample Coordinate | GCP location within image chip - sample coordinate (fractional pixel). |
|  GCP Latitude | GCP latitude in degrees. |
|  GCP Longitude | GCP longitude in degrees. |
|  GCP X | GCP projected X coordinate in meters. |
|  GCP Y | GCP projected Y coordinate in meters. |
|  GCP Height | GCP WGS84 ellipsoid height in meters. |
|  Image Chip GSD | Chip pixel size in meters. |
|  Image Chip Lines | Number of lines in the image chip. |
|  Image Chip Samples | Number of samples in the image chip. |
|  GCP Source | Source of GCP, either "DOQ," "GLS," or "TM6" |
|  GCP Type | Control/validation point flag, either "CONTROL" or "VALIDATION" |
|  Image Chip Projection | UTM or PS |
|  Image Chip Zone | UTM zone number (1-60). Use 0 for PS. |
|  Image Chip Date | yyyymmdd = year/month/day of the GCP source |
|  Image Chip | Link to chip image data (could be in file named with GCP ID) |

Table 6‑1. Input GCP Library Contents

|  |  |
| --- | --- |
| **Field** | **Description** |
| GCP Record Fields: | One set per GCP |
| Point ID | GCP ID (see Table 6‑1) |
| GCP chip line location | Line location of the GCP within the chip |
| GCP chip sample location | Sample location of the GCP within the chip |
| GCP latitude | GCP WGS84 latitude in degrees |
| GCP longitude | GCP WGS84 longitude in degrees |
| GCP height | GCP WGS84 ellipsoid height in meters |
| Predicted GCP image line | Predicted line location of the GCP in the L1GT image |
| Predicted GCP image sample | Predicted sample location of the GCP in the L1GT image |
| GCP image line offset | Measured line offset from the predicted location |
| GCP image sample offset | Measured sample offset from the predicted location |
| Correlation success flag | Flag 0 = correlation failure, 1 = success |
| Correlation coefficient | Measured correlation coefficient (new) |
| Search band number | L1GT band number used |
| Search SCA number | L1GT SCA where the GCP was found |
| Chip source | GCP source (DOQ or GLS or TM6) |

Table 6‑2. Output GCP Mensuration File Contents

#### Notes

The following are additional background assumptions and notes:

* The heritage GLS, TM6, DOQ image chips are stored as 8-bit (BYTE) arrays, whereas the L8/9 imagery is 16-bit. The correlation is performed on floating-point data so both the image and the chips are converted to float on input. Thus, the image and chip data types need not match.
* The correlation result fit method defines the algorithm used to estimate the correlation peak location to subpixel accuracy. Only the quadratic surface-fitting method described in this ADD is supported in the baseline algorithm.
* Though the normal baseline for measuring control points is to use an SCA-separated terrain corrected image, this algorithm should also function with a combined-SCA image so that it can be used to measure test-point GCPs in L1TP product images to support the geometric accuracy characterization algorithm.
* The GCPs in the GCP repository are flagged as either “control” points, to be used for LOS model correction, or “validation” points, to be used for geometric accuracy characterization. The utility that extracts control points from this repository must be able to extract either control set. The “control” set would contain the majority of the points. The “validation” flag is only used in areas where more than some minimum threshold number of GCPs are available. The Cal/Val team set these flags when the GCP repository was loaded and could adjust them thereafter, if necessary. Criteria for selecting validation points are based upon considerations such as the following:
	+ The total number of available GCPs in the scene must exceed some minimum (e.g., 100).
	+ Points that fall on the boundary (or, more precisely, the convex hull) of the GCP set would not be validation point candidates.
	+ Points that are within some maximum distance (e.g., 25 km) of another GCP would be validation point candidates.
	+ The goal would be to develop an automated validation point identification algorithm that would operate somewhat like an outlier rejection algorithm: identify the best validation point candidate based on a set of criteria, remove it from the control point list, and iterate until no additional validation points are identified.
* Scenes with poor geolocation accuracy can lead to the actual GCP L1GT image locations being sufficiently far from their predicted locations so as to make it impractical to expand the GCP search window to the extent necessary to find the GCPs. An optional parameter to specify an a priori predicted offset provides a more reliable way to find and correctly correlate the GCPs in this situation. This can occur early in the mission, before the first on-orbit sensor alignment calibration, or during an anomaly investigation.