### Terrain Occlusion Mask Generation Algorithm

#### Background/Introduction

The heritage Landsat and ALI/EO-1 image resampling procedures ignored the possibility of multiple terrain intersections due to off-nadir viewing toward the edges of the imaging swath. This was a reasonable simplification for Landsat with its fixed nadir viewing geometry. Although the ALI was capable of off-nadir pointing, this capability was mostly used to acquire different portions of the nominal Landsat swath, given that the ALI’s focal plane was only 20 percent populated. Furthermore, EO-1 was a technology demonstration project with a minimal budget for ground processing algorithm development, so Landsat capabilities were reused as-is wherever possible.

Ignoring the multiple terrain intersection effect is less defensible for Landsat 8/9, which will routinely acquire off-nadir scenes from adjacent WRS paths for product generation. The approach to this problem adopted here is to compute the ground locations where the OLI line-of-sight is obstructed by terrain, and provide this information in a mask. The image resampling logic will be permitted to populate all output image pixels with apparent values according to the heritage algorithm. Some of these will be erroneous data that actually represent terrain intersection points closer to the imaging sensor. These can be subsequently identified, and if appropriate, replaced with fill by the user, based on the contents of the terrain occlusion mask generated by this algorithm. This approach was felt to be preferable to inserting fill in the product image, as some image exploitation algorithms (e.g., control point mensuration) are sensitive to the presence of fill.

Generating the terrain occlusion mask can also be performed without reference to the output image itself, requiring only the DEM and the LOS projection grid as inputs. For each pixel in the output image, the algorithm uses the grid file to locate the corresponding pixel in the input (L1R) space. It then uses the grid to compute the output space line/sample location corresponding to the same input line/sample at the maximum elevation plane. The line connecting the original output pixel location with the maximum elevation location corresponds to the projection of that pixel’s line-of-sight into output space. By interpolating elevation model heights for points along this line and comparing them to the computed LOS height, terrain intersection points that are closer to the imager can be detected. Each point in the output terrain occlusion mask will be flagged as either visible or occluded by terrain.

#### Dependencies

The terrain occlusion algorithm assumes that the LOS Projection and Gridding algorithm (ADD 6.2.2) has created the output product LOS projection grid. The elevation planes in the LOS projection grid must span the range of elevations in the elevation model.

#### Inputs

The terrain occlusion algorithm and its component sub-algorithms use 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).

|  |
| --- |
| **Algorithm Inputs** |
| ODL file (implementation) |
| OLI Grid file |
| DEM Grid file |
| Original Unresampled DEM file |
| Terrain Occlusion Mask file name |
| Terrain Occlusion band |

#### Outputs

|  |
| --- |
| TO (terrain occlusion) mask file |
| TO mask data descriptor record (DDR) (see note 4) |
| TO mask image |

#### Options

None.

#### Procedure

Read the unresampled DEM to determine the maximum elevation within the file (maximum\_elevation).

Initialize the terrain mask to 0.

For each SCA:

For each output pixel:

1. Retrieve the elevation for the current output pixel location (current elevation) from the DEM.
   1. Using the DEM resampling grid, map the L1TP output pixel location to geographic unresampled DEM line/sample location.
      1. Calculate grid cell row and column index.

grid row = output line / number grid cell lines

grid col = output sample / number grid cell samples

* + 1. Determine grid cell number.

grid cell number = grid row \* number grid cell samples + grid col

* + 1. Look up grid mapping coefficients based on grid cell.

coeff = grid cell coefficient reverse[grid cell number].

* + 1. Calculate DEM line/sample location.

DEM line = coeff.line[0] +

output sample \* coeff.line[1] +

output line \* coeff.line[2] +

output sample \* output line \* coeff[3].line

DEM sample = coeff.sample[0] +

output sample \* coeff.sample[1] +

output line \* coeff.sample[2] +

output sample \* output line \* coeff[3].sample

* 1. Perform bilinear interpolation at location in DEM from step 1a) to determine elevation of current L1TP output location.
     1. Determine subpixel location

Integer line = (int)DEM line

Integer sample = (int) DEM sample

ds = DEM sample – Integer sample

dl = DEM line – Integer line

* + 1. Determine location in DEM image buffer.

dem\_ns = number samples in DEM

dem\_nl = number lines in DEM

loc = Integer line \* dem\_ns + Integer Sample

* + 1. Interpolate elevation for floating point location.

elevation =

(1.0 - ds) \* (1.0 - dl) \* dem.data[loc] +

ds \* (1.0 - dl) \*dem.data[loc+1] +

(1.0 - ds) \*dl \* dem.data[loc+dem\_ns] +

ds \* dl \* dem->data[loc+dem\_ns + 1]

Note:

For off-nadir images, pixel line-of-sight ground projections can extend outside of the product image area. Using the unresampled DEM as the source of elevation data should prevent elevations from being needed outside of the available data range as the terrain occlusion calculation performs its “stepping process.” However, a check to ensure that the elevation being retrieved is greater than 0 in line and sample while less than dem\_nl-1 and dem\_ns-1 should be implemented. The process should issue a warning that the data to be retrieved is outside of the DEM, and return the DEM elevation value for the closest edge line/sample position (i.e., clip the DEM line/sample values at the DEM edges).

1. Run ols2ils (OLI Resampling Algorithm ADD 6.2.4) to find the input location for the corresponding output location. This will be based on the elevation for current output pixels (lc,sc).
2. For the input location calculated in 2) calculate the corresponding output location for the maximum elevation (lm,sm) (OLI Resampling Algorithm ADD 6.2.4).
3. Define the parametric equation for a line that connects (lc,sc) to (lm,sm).

sp = s0 + t \* f

lp = l0 + t \* g

where: 

At t=0: lp=lc and sp=sc.

At t=1: lp=lm and sp=sm

Therefore

l0 = lc,

s0=sc,

g=(lm-lc),

f=(sm-sc)

1. Compute the length of the line in output space:



1. Compute the increment of t to use to walk along the line:



1. Walk along the line in increments of t, testing each point for terrain occlusion:

For j = 0 to (int)ceil(1/Δt)

t = j \* t

1. Calculate the point of intersection:

lp = l0 + t \* g

sp = s0 + t \* f

1. Round (lp,sp) to get (lp',sp'). Find the elevation for (lp',sp') (pixel elevation) using the DEM resampling grid as described in steps 1a) and 1b) above.
2. The value of t represents the ratio used to measure whether the elevation of (lp',sp') is large enough to obscure the current pixel of interest (lc,sc).

if( (t \* maximum elevation + (1.0-t) \* current elevation) < pixel elevation )

Current pixel location (lc,sc) is occluded. Set the terrain mask to 1 and exit loop.

else

Current pixel location (lc,sc) is not occluded. Continue to loop.

*Determining Elevation (change from using co-registered DEM)*

Due to the “walk-a-line” process of step 7) of the previous procedure, the location of an elevation requested could reside outside of the co-registered DEM used in creating the L1TP. To account for this, the unresampled DEM and DEM geomgrid (used to resample the DEM) are used for terrain occlusion calculations instead of the co-registered DEM. Since the unresampled DEM should extend outside the boundary of the L1TP, this will allow the retrieval of elevations outside the product image extent.

#### Notes

Some additional background assumptions and notes include the following:

1. The new logic required to calculate the terrain occlusion mask (particularly for off-nadir scenes) is documented here, but may be implemented as part of the resampling software for processing efficiency. The Terrain Occlusion (TO) mask output by this algorithm is also included as a possible (to be resolved) output in the resampling algorithm.
2. The current concept is to allow the user to specify the band(s) to use in testing for occlusion. However, for the terrain mask that is to accompany the L1TP L8/9 product, generation of the mask for the SWIR1 band should be sufficient.
3. The problem of multiple terrain intersections needs to be addressed, particularly for off-nadir acquisitions. A terrain occlusion mask will be generated to identify these obstructed pixels, but the current thinking is that it would not alter the behavior of the resampler, as sprinkling fill pixels throughout a product image can wreak havoc with some applications. Generating a separate terrain occlusion mask will allow users to evaluate the extent of the problem and apply the mask if appropriate to a particular application.