Ames Albedo Reconstruction Software

A part of the NASA NeoGeography Toolkit

Version 0.2

Ara Nefian
Oleg Alexandrov
Zachary Moratto

Intelligent Robotics Group
NASA Ames Research Center

June 21, 2012

Contents

1 The albedo software inputs and outputs

The albedo reconstruction software takes the following inputs:

1. A set of DRG (Digital Raster Graphic) images of the Moon in the GeoTIFF format
2. A set of DEM (Digital Elevation Model) images in the GeoTIFF format
3. The Sun and spacecraft position at the moment each DRG image was taken.

The algorithm uses the Sun/spacecraft position and DEM data to correct for the illumination conditions in the DRG images, thus computing Moon's true albedo. The output will be a set of non-overlapping DRG tiles containing the albedo.

In addition to creating albedo images, the software can also be used for generating image mosaics (see the parameter REFLECTANCE_TYPE in section 5). In that case, the DEM images and Sun/spacecraft positions are not required.

2 Using the software

The albedo reconstruction software can be called by using the script *reconstruct.sh* in the directory PhotometryTK/src/tools. It is used as follows:

reconstruct.sh setings.txt labelStr

Here, setings.txt is the settings file (section 5), and labelStr is a string label. All outputs of running this command will be stored in the directory albedo_labelStr.

Using this script requires that Vision Workbench and PhotometryTK be complied (see the installation notes of these two packages for instructions). The script reconstruct.sh needs to be edited to specify the path to these packages.

The output albedo will go to the directory albedo_labelStr/albedo. The output will be a set of image tiles, with each pixel being uint8 (the same as the input DRG images).

The exposure time for each image computed during albedo reconstruction is stored in the directory albedo_labelStr/exposure. Other subdirectories in albedo_labelStr contain data used for intermediate computations, such as the weights assigned to each pixel in the DRG images, the averaged DEM in each tile, etc.

3 Example runs

Two *example runs* (settings files and other necessary input data) are available in the directory PhotometryTK/src/photk/tests. They are executed automatically if the user runs *make check* in PhotometryTK/build.

4 Running the software on multiple machines

The software can be distributed over a very large number of computing nodes (using GNU parallel), as long as those nodes share storage. The computing nodes need to be specified in a file, with one machine name per line, and the enviromental variable PBS_NODEFILE needs to be set to the path to this file. If this variable is not defined, the jobs are only run on the current machine. On NASA'S Pleiades Supercomputer the variable PBS_NODEFILE is set automatically.

The parameter NUM_PROCESSES in the settings file (section 5) can be used to control how many processes to start on a given computing node. This number depends on the number of cores available on the node, the amount of memory, and the resolution of the input images (higher resoluton requires more memory and therefore fewer processes can be run simultaneously on a current machine).

5 The settings file

The parameters controlling the albedo reconstruction software are specified in a settings file (a sample file is shown in the appendix). It has the following entries:

DRG_DIR	The path to the directory of DRG images.
DEM_DIR	The path to the directory of DEM images.
SUN_POSITION_FILE	The Sun position file.
SPACECRAFT_POSITION_FILE	The spacecraft position file.
NUM_PROCESSES	How many simultaneous processes to start on a given computing node. This number is best set to the number of available cores on the node.
TILE_SIZE	Output albedo tile size, in degrees (usually 4 by 4 degrees).
SIMULATION_BOX	The region in which to compute the albedo, in the format lon_min : lon_max : lat_min : lat_max. If this parameter is not provided, the region will be assumed to be the entire Moon.
REFLECTANCE_TYPE	The reflectance type. Values: 0 – the reflectance is not be used, the result of albedo reconstruction is just an image mosaic, 1 – the Lambertian reflectance model is used, 2 – the Lunar-Lambertian reflectance model (Equation 1.2) is used.
SHADOW_THRESH	Shadow threshold, with values between 0 to 255 (the input DRG images have values between 0 to 255, see section 6). Its use depends on the variable SHADOW_TYPE.

SHADOW_TYPE	Determines how to classify which input DRG image pixels are in the shadow (section 7). Values: 0 – no pixels are in the shadow, 1 – pixels in the shadow are those with values below SHADOW_THRESH, this is the default. Advanced values: 2 – pixels in the shadow satisfy $I/(R \cdot T) < t$ [1], 3 – analogous to option 2, but use the Lunar-Lambertian reflectance model (Equation 1.2) instead of the Lambertian model.
TR_CONST	A constant which determines how bright the albeo will be, a larger value will make the albedo appear darker.
PHASE_COEFF_C1	The coefficient c_1 in reflectance formula (Equation 1.2).
PHASE_COEFF_C2	The coefficient c_2 in the same expression.
MAX_NUM_ITER	Maximum number of albedo update iterations. If it equals 0, the image exposure, phase coefficients, and the albedo itself will only be initialized, and no subsequent iterations will take place to improve the initial values (see the algorithm in section 7).
NO_DEM_DATA_VAL	The value used in the DEM images to denote that no data is available at the current pixel. This number is normally stored within the DEM images; only if it absent there will the value specified here be used.
USE_WEIGHTS	If to use weights to seamlessly blend the albedo values obtained for individual images (1 means true, and 0 means false). See section 7.
USE_NORMALIZED_COST_FUN	If to enforce that the sum of weights over all images add up to 1 at each pixel. If true, all pixels will contribute in equal amount to the cost function (Equation 1.1). This flag makes a difference only if multiple albedo iterations are performed (MAX_NUM_ITER > 0). Turning this on incurs a significant performance hit.

[1] I is the image value, R is the Lambertian reflectance, T is the image exposure, and t is the shadow threshold.

COMPUTE_ERRORS	If at the end of the algorithm to compute the albedo error map (Equation 1.7), showing at each pixel what the error was in computing the albedo at that pixel.

6 Requirements for the input

As mentioned in section 1, the inputs to the albedo reconstruction software are a set of DRG images, a set of DEM images, and the Sun and spacecraft position for each DRG image. Here we specify the exact requirements for the data as expected by the software.

1. The DRG images should be in a directory and stored in the GeoTIFF format. It is assumed that the intensity values for each image pixel are between 0 and 255 (uint8). The first 11 characters of each image in that directory should be unique, and will be the means by which we will later look up the Sun and spacecraft position for that image. For example, given the image:

   DRG_input_sub64/AS15-M-0074_0075-DRG.tif

   the lookup key is the 11-character string: AS15-M-0074.

2. The DEM images should be in a directory and stored in the GeoTIFF format. They may or may not overlap. If the DEM images overlap, the values from all images at a given pixel will be averaged. It is expected that the DEM images are either in int16 or float format. If the images lack the No Data Value, it should be explicitly set in the settings file using the variable NO_DEM_DATA_VAL.

3. The Sun positions for the DRG images should be provided as a list in a file, with one line in the file for each DRG image. The same should hold for the spacecraft position. Each line in these lists needs to be in the format:

   lookup_key x y z

   where lookup_key is the 11-character string that identifies the DRG image, and x, y, and z are the coordinates of the Sun (spacecraft) in the Cartesian coordinate system whose origin is the center of the Moon. The values should be in kilometers. An example entry for the Sun position is

   AS15-M-0074 -1345328.6130903 151849061.51791 689530.60102555

4. The directories containing the DRG and DEM images should be user-writeable, as the software will create in those directories a list of the GeoTIFF images contained within them, together with the coordinates of the corners of each image. These lists are created only once and speed up subsequent computations.

7 Modeling and the algorithm

In section we describe the mathematical modeling underlying the albedo reconstruction software.

Let I_{ij}^k, A_{ij}, X_{ij}, R_{ij}^k, T^k be the observed image value, albedo, DTM, reflectance, and exposure time at pixel (i, j) and k-th image.

We would like to determine the set $\tilde{A}_{ij}, \tilde{T}^k = \arg\min_{A_{ij},T^k} \mathbf{Q}$ where

$$\mathbf{Q} = \sum_k \sum_{ij} \left[(I_{ij}^k - A_{ij} T^k R_{ij}^k)^2 S_{ij}^k w_{ij}^k \right] \tag{1.1}$$

Here, S_{ij}^k is a shadow binary variable, $S_{ij}^k = 0$ when the pixel (i, j) is in the shadow and 1 otherwise. The weights w_{ij}^k are chosen such that they have linearly decreasing values from the center of the image ($w_{ij}^k = 1$) to the image boundaries ($w_{ij}^k = 0$). The choice of these weighs insures that the reconstructed albedo mosaic is seamless.

In the above equation the reflectance is computed using the Lunar-Lambertian model and is given by

$$R_{ij}^k = (e^{-c_1 \alpha} + c_2) \left[(1 - L(\alpha)) \cos(\mathbf{i}_{ij}^k) + 2L(\alpha) \frac{\cos(\mathbf{i}_{ij}^k)}{\cos(\mathbf{i}_{ij}^k) + \cos(\mathbf{e}_{ij}^k)} \right] \tag{1.2}$$

where $L(\alpha) = 1 + A\alpha + B\alpha^2 + C\alpha^3$ is a weighting factor between the Lunar and Lambertian reflectance models that depends on the phase angle (α), $\mathbf{i}_{ij}^k$ and $\mathbf{e}_{ij}^k$ are the incident and emission angles at image k and pixel (i, j). The numbers c_1 and c_2 are called the *phase coefficients*.

Determining the best albedo reconstruction from a set of images and the corresponding DTM is formulated as a cost function minimization problem for the albedo A_{ij}, image exposures T_k, and phase coefficients c_1 and c_2. (Equation 1.1). An iterative solution to the above least square problem is given by the Gauss-Newton updates described below.

- **Step 1 (initialization)**: Compute the average DEM and the weights. Normalize the weights so that the sum of weights at each pixel is 1 if the variable USE_NORMALIZED_COST_FUN is set to 1 in the configuration file. Initialize the exposure time as inversely proportional to the average image reflectance. Initialize the phase coefficients c_1 and c_2 to some reasonable values. Initialize the albedo as the arg min of the cost function $\mathbf{Q}$ for fixed exposure time and phase coefficients,

$$A_{ij} = \frac{\sum_k I_{ij}^k T^k R_{ij}^k S_{ij}^k w_{ij}^k}{\sum_k (T^k R_{ij}^k)^2 S_{ij}^k w_{ij}^k} \tag{1.3}$$

- **Step 2**: Re-estimate the exposure time using

$$\tilde{T}^k = T^k + \frac{\sum_{ij} (I_{ij}^k - T^k A_{ij} R_{ij}^k) A_{ij} R_{ij}^k S_{ij}^k w_{ij}^k}{\sum_{ij} (A_{ij} R_{ij}^k)^2 S_{ij}^k w_{ij}^k} \tag{1.4}$$

- **Step 3**: Re-estimate the phase coefficients using

$$\tilde{c_1} = c_1 + \frac{\sum_{ijk} (I_{ij}^k - T^k A_{ij} R_{ij}^k) T^k A_{ij} \frac{\partial R_{ij}^k}{\partial c_1} S_{ij}^k w_{ij}^k}{\sum_{ijk} \left(T^k A_{ij} \frac{\partial R_{ij}^k}{\partial c_1} \right)^2 S_{ij}^k w_{ij}^k} \tag{1.5}$$

  and analogously for c_2. The partial derivatives $\partial R_{ij}^k / \partial c_1$ and $\partial R_{ij}^k / \partial c_2$ are computed from Equation 1.2.

- **Step 4**: Re-estimate the albedo using

$$\tilde{A}_{ij} = A_{ij} + \frac{\sum_k (I_{ij}^k - T^k A_{ij} R_{ij}^k) T^k R_{ij}^k S_{ij}^k w_{ij}^k}{\sum_k (T^k R_{ij}^k)^2 S_{ij}^k w_{ij}^k} \tag{1.6}$$

- **Step 5**: Compute the error cost function $\mathbf{Q}$ for the re-estimated values of the albedo and exposure time.

- **Convergence**: If the convergence error between the consecutive iterations falls below a fixed threshold then stop. Otherwise return to step 2.
- **Step 6**: Estimate the accuracy of the albedo reconstruction at each pixel using the formula

$$E_{ij} = \frac{\sum_k (I^k_{ij}/(T^k R^k_{ij}) - A_{ij})^2 S^k_{ij} w^k_{ij}}{\sum_k w^k_{ij}} \tag{1.7}$$

8 Advanced usage

The script *reconstruct.sh* computes the albedo by performing a series of calls to the executable *PhotometryTK/build/src/tools/reconstruct*. Advanced users may call this program directly to execute a certain stage of the algorithm (section 7).

The *reconstruct* executable takes the following options:

```
-s [ --settings-file ] arg     Settings file
-r [ --results-directory ] arg Results directory
-f [ --images-list ] arg       The list of images
-t [ --tiles-list ] arg        The list of albedo tiles
-i [ --image-file ] arg        Current image or tile
--initial-setup                Initial setup
--save-weights                 Save the weights
--compute-weights-sum          Compute the sum of weights at each pixel
--init-dem                     Initialize the DEM
--init-exposure                Initialize the exposure times
--init-albedo                  Initialize the albedo
--update-exposure              Update the exposure times
--update-tile-phase-coeffs     Update the phase coefficients per tile
--update-phase-coeffs          Update the phase coefficients by combining the
                               results over all tiles
--update-albedo                Update the albedo
--compute-errors               Compute the errors in albedo
--is-last-iter                 Is this the last iteration
-h [ --help ]                  Display this help message
```

For example, to initialize the exposure times for a given image, one may issue the command:

reconstruct -s settings.txt -r albedo_run -f albedo_run/imagesList.txt -t albedo_run/albedoTilesList.txt –init-exposure -i DRG_DIR/AS16-M-2961.tif

Each time the *reconstruct* executable is run, it echoes the precise command and options which were used to call it, so these calls can be inferred by looking at the output of *reconstruct.sh*.

9 Retrieving the input data from ISIS cubes

The albedo reconstruction software assumes that the input DRG images are in GeoTIFF format, and that the sun and spacecraft positions are in text files (section 1). This section describes how to get this data from ISIS cubes.

The DRG images can be obtained by orthoprojecting the ISIS cubes onto the terrain data (a set of DEM tiles) using the script named *orthoproject_cube.sh* in the directory PhotometryTK/src/tools. The script assumes that the following packages are installed: Ames Stereo Pipeline, ISIS libraries, and the GDAL tools. The paths to them can be set in the script. It can be called as:

orthoproject_cube.sh input.cub input_isis_adjust DEM_dir mpp num_proc output.tif

The arguments passed to it are, respectively, the cube file to orthoproject, the ISIS adjust file storing the adjusted camera position, the directory containing the DEM tiles to orthoproject onto, the desired output resolution in meters per pixel, the number of processors to use, and the name of the output image.

To get the sun and spacecraft position from a list of ISIS cubes the script *get_sun_or_spacecraft_position.pl* is used, also located in PhotometryTK/src/tools. The usage is:

get_sun_or_spacecraft_position.pl sun-or-spacecraft inputCubesList.txt outputPositionList.txt

A A sample settings file

```
# Files/directories
DRG_DIR                    DIM_input_2560mpp
DEM_DIR                    DEM_tiles_sub64
SUN_POSITION_FILE          meta/sunpos.txt
SPACECRAFT_POSITION_FILE   meta/spacecraftpos.txt

# Constants
NUM_PROCESSES              8
TILE_SIZE                  4.0
SIMULATION_BOX             -180:180:-180:180
REFLECTANCE_TYPE           2
SHADOW_THRESH              40
SHADOW_TYPE                1
TR_CONST                   1.6
PHASE_COEFF_C1             1.4
PHASE_COEFF_C2             0.5
MAX_NUM_ITER               0
NO_DEM_DATA_VAL            -32767

# Actions
USE_WEIGHTS                1
USE_NORMALIZED_COST_FUN    0
COMPUTE_ERRORS             0
```

Bibliography

[1] M. J. Broxton, Z. M. Moratto, A. Nefian, M. Bunte, and M. S. Robinson. Preliminary Stereo Reconstruction from Apollo 15 Metric Camera Imagery. *40th Lunar and Planetary Science Conference*, 2009.

[2] Ara V. Nefian, Kyle Husmann, Michael J. Broxton, Vinh To, Michael Lundy, and Matthew Hancher. A Bayesian Formulation for Sub-pixel Refinement in Stereo Orbital Imagery. *IEEE International Conference on Image Processing*, 85, November 2009.

[3] Michael J. Broxton, Ara V. Nefian, Zachary Moratto, Taemin Kim, Michael Lundy, and Aleksandr V. Segal. 3D Lunar Terrain Reconstruction from Apollo Images. *International Symposium on Visual Computing*, 2009.