

A beginner's guide to stereo-derived DEM production and analysis using ISIS, ASP, and ArcMap

v. 1.1

Teemu Öhman

LPI, June 2013

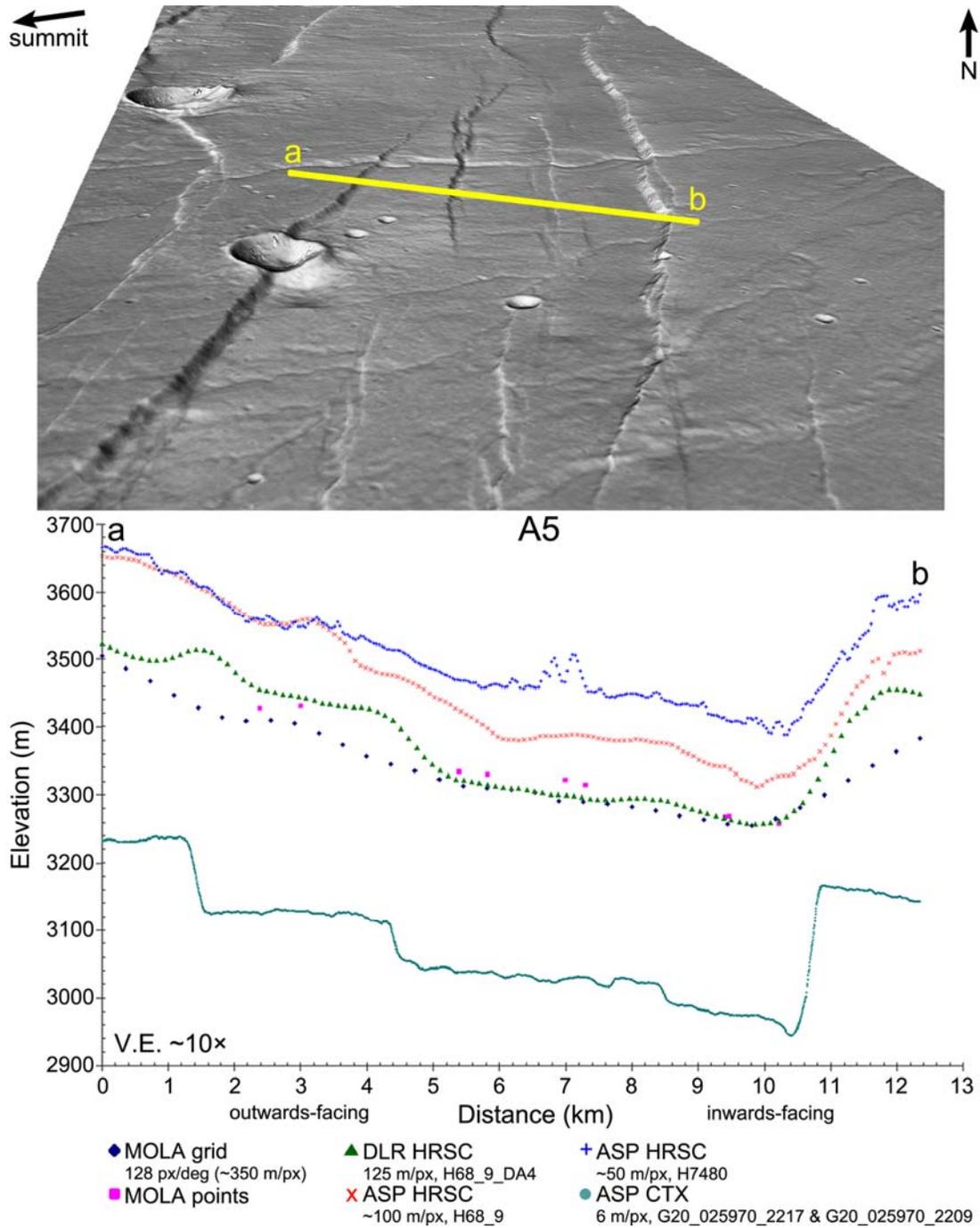


Table of contents

Preface, or things to note before you start.....	3
1. Finding the right images.....	4
1.1. CTX.....	4
1.2. HRSC	4
1.3. LRO NAC	5
2. Processing the images with ISIS	5
2.1. Cubing the raw	5
2.2. Adding some spice	6
2.3. Radiometric calibration.....	6
2.4. Creating a map template.....	6
2.5. Projecting the cubes	7
2.6. Giving the final touches to LRO NACs for ArcGIS/ArcMap.....	7
3. Creating DEMs using ASP.....	8
3.1. Crop before you start.....	8
3.2. Running stereo.....	9
3.3. Changing the correlation parameters.....	10
3.4. Creating the DEMs.....	12
3.4.1. The Moon	12
3.4.2. Exporting the DEM to ArcGIS.....	12
3.4.3. Visualising the DEM in ASP	12
3.4.4. Martian DEMs and the areoid	14
4. Processing and analysing the DEMs in ArcMap.....	16
4.1. Importing and re-projecting the data.....	16
4.2. Subtracting the areoid.....	17
4.3. Displaying the DEM.....	17
4.4. Georeferencing.....	18
4.5. Analysing the DEM and exporting the data	19
Acknowledgements	19
Appendix 1. Some additional links and references	20
Appendix 2. Some very basic UNIX/LINUX commands.....	22
Appendix 3. An example of a stereo parameter file for processing LRO NAC stereo pairs	23
Appendix 4. An example of a stereo parameter file for processing CTX stereo pairs.....	26
Appendix 5. An example of a stereo parameter file for processing HRSC stereo pairs	29

Preface, or things to note before you start

The following is a brief beginner's guide to producing and analysing planetary (the Moon and Mars, but the basics are the same for the other heavenly bodies as well) stereo image-based digital elevation models (DEMs) with USGS Integrated Software for Imagers and Spectrometers (ISIS), NASA Ames Stereo Pipeline (ASP), and ESRI ArcGIS (Geographic Information System). The procedures have been tested with ISIS 3.3.0.3646 and ASP 1.0.5 (both running on Ipi24), and ArcGIS 10.0. More recent versions of all three exist, but this set-up has the benefit that I have proven it works. I am mostly describing Martian DEMs, but will mention lunar-specific commands whenever necessary. The example files mentioned here are files that I have successfully used in creating the DEMs, so this is thoroughly tested in practice.

An important thing to note is that these DEMs described here do not use ground control points and bundle adjustment, and therefore do not provide accurate absolute elevations. Hence, these elevations should not be compared between different DEMs. Within a single DEM, however, the elevations have on all tests that I have done been consistent. Ground control points can at least theoretically be used with substantially more ISISing and ASPing, and it is quite time-consuming and difficult. Even the ASP developers admit that. With the current set-up, I have never managed to do it. Nevertheless, peer-reviewed research has been published using this simplified approach (Lefort et al. 2012).

Another very significant thing to realise is that ASP does *not* allow you to create DEMs from mosaics, only from single stereo pairs. For mosaic DEMs, you have to use Socet Set or other similar commercial software. In the future, it would also be useful to look into ASP 2.x, which is already available, and is claimed to be substantially faster.

The approach here is from a dummy, to a dummy. Even if you are not going to be using DEMs, the first and second parts of this short set of instructions can be used as a guide for extreme beginners to find the correct set of images and to process them with ISIS. Some of the ArcMap analysis part might be handy as well.

Although there are bound to be inconsistencies, I have made some attempts for a consistent style to make this easier to follow by just glancing. Commands that are supposed to be written exactly like that in the command line are indented and written in `arial`. In the case of ASP, these are usually followed by things like output prefixes, which are written in *times new roman italics*. Options in the ISIS GUI that are mentioned in the text are also given in *times new roman italics*, and it is used for emphasis in a few other places as well. In case you don't want to use the GUI, it probably helps that the actual ISIS options you would use in the command line are written with *CAPITALS* in *times new roman italics*. When you need to type something in ISIS GUI, it is given in "quotation marks in italics". ASP stereo parameter file text, mostly given in the appendices, is written in `courier new`.

1. Finding the right images

There is a confusing number of places to download planetary images and other data. Some of the places provide the same data as other places, some don't, even if they are supposed to. For the purposes of creating DEMs, I have found the following places to be the best for me.

1.1. CTX

The best Martian DEMs that I have encountered are done using MRO CTX (Mars Reconnaissance Orbiter Context Camera) images, with a resolution of about 6 m/px. The stereo coverage is not extensive, and some apparent stereo pairs have in practise too large or small differences in illumination or viewing geometries to work well. To get an idea if a stereo pair has the potential of producing a good DEM, it helps and makes things faster if you are able to see the browse versions of the stereo pairs on screen in 3D without stereo glasses. If the 3D effect looks good, the chances are that the DEM will too. Some people cannot do it, but most people can learn it – either cross-vision or parallel-vision – with little practice.

For CTX, the best place to obtain the images is the *MRO CTX image map*:

<http://global-data.mars.asu.edu/bin/ctx.pl>

Select the image, and download the *PDS source EDR* raw image. The correct image format should have a name like G20_025970_2217_XN_41N102W.IMG or G20_025904_2209_XN_40N102W.IMG from the eastern flank of Alba Mons.

1.2. HRSC

MEX HRSC (Mars Express High Resolution Stereo Camera) stereo channel images are obviously very good for DEMs as well, although they usually require more work to produce nice DEMs with ASP. Finding them is also more difficult. The place to go is the *HRSCview data explorer*:

<http://hrscview.fu-berlin.de/cgi-bin/ion-p?page=entry2.ion>

Select the *HRSC Archive DTM resolution map*. Once there, select the image you want using the very confusing map interface. When there are two or more images available, the way to choose the image further down the list for downloading is to select it on the drop-down menu and then hit the *Update view* button. This updates the almost hidden link to the actual download page at the bottom of the page.

At the download page, the images to download are in the *Download (level 2) PSA (PDS format)* column. The stereo channel images have names like H0068_0009_S12.IMG and *S22.IMG, also from the eastern flank of Alba Mons.

If DLR (German aerospace center, or Deutsches Zentrum für Luft- und Raumfahrt) has produced a DEM of this stereo pair, it is available from this page as well. The areoid-referenced DEMs are the ones that give elevation data that is referenced to MOLA, so you should directly get the approximately same elevation numbers from the DLR areoid-referenced DEM, MOLA points, and MOLA DEM without any additional fiddling. In this case, the level-4 areoid-referenced DLR DEM made from the images mentioned above is H0068_0009_DA4.IMG.

1.3. LRO NAC

For the Moon and Lunar Reconnaissance Orbiter Narrow Angle Camera images, or LRO NACs (and Wide Angle Camera Images, or WACs), I would start with the *LROC QuickMap*:

<http://target.lroc.asu.edu/q3/>

You can select the NAC from there, or if you know the coordinates for your images, you can go directly to the LROC Image Search page:

<http://wms.lroc.asu.edu/lroc/search>

And from there, after setting your search parameters, you will get to the download page. What you want to download is the EDR, not the CDR! The correct image file should have a name like M152451994LE.IMG or M152445210LE.IMG, as is the case here for crater Eimmart A. One thing that might be good to know is that those numbers in LROC names are seconds (mission elapsed time, I suppose). You most likely would want to get both the left (*LE.IMG) and the right (*RE.IMG) NACs of both of your stereo pairs, but that does of course depend on the purpose and the stereo coverage. And no, the tiny little zone of overlap between the two NACs (left and right) does not work as a stereo pair. You'll just use that overlap to mosaic them accurately in Photoshop.

2. Processing the images with ISIS

First of all you need to have a user account on lpi24, and you need to be able to establish an SSH Secure Shell connection to lpi24. These have to be done by the IT-guys. Once that is done and you have logged in, hit the *New File Transfer Window* button, then create *New Folder*, and copy the files to this folder (just drag and drop). Then you can start ISISing.

ISIS can be run either from the graphic user interface (GUI) or directly from the command line. To start with, GUI is easier, and that is what's being used here. There is no "ISIS GUI", but each program in ISIS has its own GUI, which you start by typing the name of the program on the command line. After becoming more familiar with ISIS, some (if not all) of the following stages are a bit faster to do directly from the command line. One thing that command line usage is good for is batch processing, which is something to look into particularly if you are about to process large amounts of images with the same set of parameters.

2.1. Cubing the raw

The first step is to convert the raw images to cubes. These are done with mission-specific commands. For CTX, HRSC, and LRO NAC the commands are

mroctx2isis

hrsc2isis

and

Ironac2isis

respectively. In the GUI, simply select your raw (*.IMG) image in the *FROM*-field, and potentially rename it to something more understandable in the *TO*-field. Adding .cub to the file name is not necessary, as ISIS does it automatically. Hit the green button to run it.

For other useful datasets, for example Kaguya TC, you would use `raw2isis`, but that is somewhat more complicated and not dealt with here (although I emphasize that Kaguya TC is a fantastic but overlooked dataset with a resolution of about 7.4 m/px and, therefore, it is extremely useful between the 60–100 m/px LRO WACs and 0.4–1.2 m/px LRO NACs).

2.2. Adding some spice

Adding SPICE data (spacecraft location and attitude, camera pointing data, etc. See the link in Appendix 1.) to the images can be done either now or after radiometric calibration. For CTX and LRO NAC (and LRO WAC as well) you simply run

```
spiceinit
```

and keep all the settings at defaults. However, for HRSC you have to select the *CKPREDICTED* (*Use predicted pointing kernels if available*) option. Note that for very recent images, the spice kernels on `lpi24` may not always be up-to-date. In that case, you need the IT-guys to update them.

2.3. Radiometric calibration

Radiometric calibration does very little visible things to the images, but it should be run anyway. It's perhaps good to mention here that for CTX and LRO NAC images, photometric calibration is totally pointless. For making sensible-looking LRO WAC mosaics it is vital, but that is another story and not dealt with here. For LRO NAC and CTX the radiometric calibration is simply run with commands

```
Ironaccal
```

and

```
ctxcal
```

and keeping all the settings at defaults. For HRSC, ISIS does not have a radiometric calibration routine, and things work just fine without it.

2.4. Creating a map template

This step is only required when you start a new project, because you will mostly be using the same template in all of your images (unless you are studying, for example, both poles and equatorial regions). Start creating the template by typing:

```
maptemplate
```

Give your map a name (again, sensible and descriptive names are advisable, because you are likely going to be using several map templates during different stages of different projects and without good names you will end up using a wrong template at some point). ISIS will add .map to the template name. Skip the *FROMLIST*-field. Choose the projection and projection parameters

according to your liking. *Central longitude (CLON)* is always required, *Central latitude (CLAT)* in many cases.

In *TARGOPT* under *Target parameters*, choose *User input (USER)*. For *TARGETNAME*, type either “moon” or “mars”. In the most straightforward case you can keep all the remaining settings at defaults. Alternatively, you can define the latitude and longitude extent and the resolution of your map, but just for doing a simple DEM it is not necessary. If you plan to do an image mosaic (not DEMs), then it is a good idea to set the resolution already here in the map template so you don’t have to worry about it while projecting the individual images.

2.5. Projecting the cubes

Doing the map projection is the time-consuming part of the basic ISISing. Start it by typing:

```
cam2map
```

In the *FROM*-field, select your calibrated cube, and in the *MAP*-field choose the map template you just made. In the *TO*-field I would give the file a new, again sensible and descriptive name so you know what you have done to the image just by looking at its name.

Under *Output Map Resolution* you have several useful options, depending on the case. You can just use the default setting *Compute resolution from input cube (CAMERA)*, but that’s usually not the best way. If your final goal is an image mosaic with a consistent resolution, then the easiest thing to do is to define the resolution in the map template, and then choose *Read resolution from input map file (MAP)*. If you just want to keep the best possible resolution but make a mosaic of LRO NAC left and right images, then choose *Get resolution from user in meters per pixel (MPP)*. As an aside, if you only need an LRO NAC left and right mosaic without any geographic information, you’ll get better results and you’ll get them much faster if you just do it in Photoshop rather than trying any mosaicking options of ISIS.

The next options (*Output Map Ground Range* and *Longitude Seam Options*) are usually best kept at defaults, but the choice for the last one, *Options*, depends on the images you use. Nearest neighbor is the fastest, and the one that I have almost always used for LRO NACs and CTXs. However, for low-resolution things like the areoid (see below), a bi-linear interpolation can often give the best result.

If you haven’t already looked at the intermediate versions of your cube, you probably want to look at this final product:

```
qview
```

In *qview* it’s useful to verify that no mistakes were made during the map projection. Clicking the icon with footprints opens the *Advanced Tracking* window. Move the mouse over the image and check that the coordinates change the way they ought to.

2.6. Giving the final touches to LRO NACs for ArcGIS/ArcMap

There are a number of ways to export cubes so they can be used in ArcGIS. You can try *isis2std* or various *gdal* scripts to create geotiffs or jpeg2000s, but the best way, however, is not to export them in different file formats at all, because ArcGIS 10 can handle most cubes without problems. LRO

NACs are problematic though, because with the processing done so far, NACs in Arc end up having large black boundaries. For example a jp2-file exported with `gdal` will be a rectangle with horizontal and vertical boundaries in ArcMap, so if the original image is taken/projected so that it runs “obliquely” in your ArcMap project, you will end up with much of your image being black. To avoid that, you need to delete some of the groups in the image header. Run

```
editlab
```

and choose *Delete group (DELG)*. Under the *Input Values*, type “*kernels*” in *GRPNAME*. Do the same for “*radiometry*”. After this, your cubes are ready to be used in ArcMap. Other cubes (including LRO WACs, although they require additional processing steps not described here) work just fine without this step.

If you are not going to be using DEMs, you can jump straight into section 4 where basic ArcMapping is discussed.

3. Creating DEMs using ASP

3.1. Crop before you start

Finishing all of the above steps (except for deleting the groups of LRO NACs described in section 2.6) are a pre-requisite for creating a good DEM with ASP. So what you need is a stereo pair with both images having the same resolution and same projection. What you also need is to know which image is the “left eye” image and which is the “right eye” image. This is easy if you can see stereo without using stereo glasses, but if not, then you just have to try to figure it out by looking at the different perspectives of the images. If everything else fails, just try to create a DEM, and if you don’t get a sensible result despite everything being alright, you most likely have left and right reversed.

Creating DEMs is very time-consuming (ASP 2 is, at least they say so, significantly faster), so for the initial tests you need to have your images heavily cropped. Don’t use cubes larger than 10 Mb for testing, and smaller ones will do. It is worthwhile to make the cropped image as representative as possible, so if your image has different terrains (e.g., flat, fairly featureless areas, as well as craters) try to include them both, because your initial set of parameters might work fine for the rougher topography, but utterly fail in the smoother parts of the image.

Cropping in ISIS is simply done by:

```
crop
```

FROM and *TO*-fields are by now obvious, so they’re just the file names. *Sample* and *Line Extraction* are a bit more complicated. Samples increase from left to right (so can be thought of as columns), and lines from top to bottom (so the spreadsheet analogue is rows). In ISIS, the top left corner of the top left pixel has coordinates 0.5 0.5! That is actually an important piece of information.

When doing the cropping of a stereo pair in practice, I have found it best to have both of the images open in *qview* at the same time, locating a good place for cropping, and then using the *Advanced Tracking* to get the upper left coordinates (sample and line) of the to-be-the-cropped-image, obviously for both images of the stereo pair. These upper left corner coordinates are what you are going to type into the *SAMPLE* and *LINE* fields in the *crop* GUI.

While still in *qview*, you should also record the sample and line of the lower right corner of your to-be-the-cropped-images. In *crop*, the *NSAMPLES* and *NLINES* fields are the number of samples and lines of the cropped image. So to get the numbers of lines and samples, subtract the upper left values of your to-be-the-cropped-image from the lower right ones. The better your left and right images match, the faster your processing times are, and the better your DEM (if the match is poor, you need a larger search window to find the matching pixels, which makes processing slow, and you may end up in a situation where ASP does not find a match, resulting in a hole or a mismatch). It is normally easy to find matching pixels in the cropping stage within a couple of pixels. Note also that you have to do this cropping at least twice for each image: first to create the small image for testing, and after you have found the correct parameters, you need to crop the full (or as large as you need) images to be as good matches as possible.

SINC and *LINC* in *crop* are the sample and line increment. The default values are 1, so the resolution of the cropped image will be the same as the original resolution. If you set *SINC* to 2, only every second sample will be used, and so forth. In the case of high-resolution data (like CTX and LRO NAC), you will often need to fiddle with these, because full resolution DEM files become too large to handle, or ASP just doesn't produce good DEMs with full resolution. The best sampling rate varies in each case.

3.2. Running stereo

Unlike ISIS, ASP has no GUI, but luckily the number of commands needed for the basic use is very limited. First of all, you obviously need to have the stereo parameter file in the same folder where you have your cubes and where you are running ASP. If you don't specify which stereo parameter file you will use, the program will look for a file called *stereo.default*. You can always keep this name for your stereo parameter file, or you may have a large number of slightly modified versions, so if the new version results in a worse DEM than in the previous run, you can easily go back to the previous set of parameters and modify them a bit differently. Changing the parameters is discussed in section 3.3.

The basic command to run ASP is:

```
stereo --stereo-file stereo.NACdefault leftimage.cub rightimage.cub
outputfolder/outputprefix
```

This uses your specified stereo parameter file (in this example called *stereo.NACdefault*) to correlate your “leftimage” cube with your “rightimage” cube, creates an output folder named *outputfolder* where it creates a large number of files, all of which have a file name starting with *outputprefix*. It is useful to read about all of these files in the ASP guide book and learn to understand at least the basics of them. The most straightforward of them is the *outputprefix-GoodPixelMap.tif* file. Grey pixels are the good correlations, the red ones are the ones where the correlation failed. This, and the number of holes you see given on the screen during the running of stereo are the easiest ways to see if your DEM is going to be any good.

The most important file created at this stage is the *outputprefix-PC.tif*. Don't bother trying to open it. It's a point cloud file, and normal image processors can't handle it. But that is what you need in creating the actual DEM. If the *GoodPixelMap.tif* looks good, you can skip the next section and move straight into creating the DEMs instead, but if there are plenty of holes and mismatches, you will need to change the correlation parameters in the *stereo.default* file.

3.3. Changing the correlation parameters

Although the default stereo parameter files that come with ASP and can also be found in the ASP guidebook often work reasonably well for LRO NAC and CTX, there is no such file available for HRSC, and in practice the parameters also for LRO NAC and CTX need to be changed to get the best results. To do so, the *stereo.default* file has to be opened in a text editor. The easiest thing to do is to use

```
gedit
```

The most important things to modify are the template window size, taken from the left image, and the search range in the right image, from which ASP is trying to find the match. The idea should become clearer from Figure 1, taken from the ASP guidebook (Broxton et al. 2012).

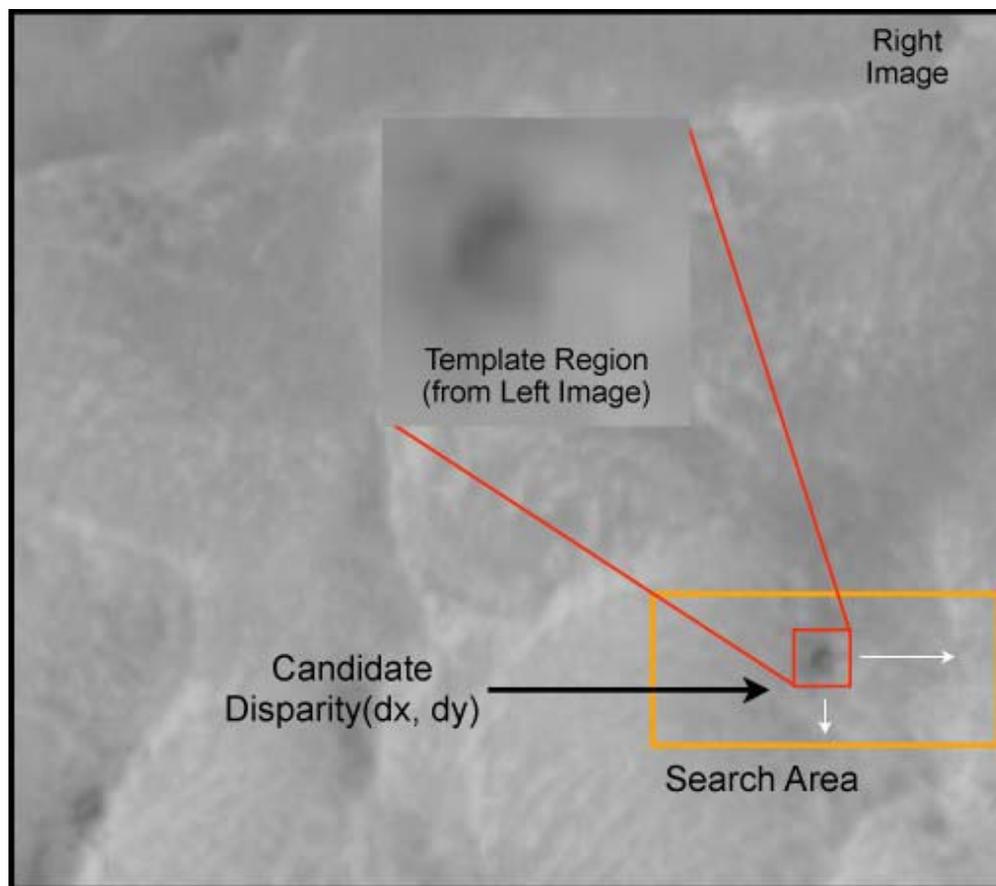


Figure 1. Correlation kernel (called Template Region in the figure) size is defined by parameters `H_KERNEL` and `V_KERNEL` in the stereo file. Parameters `H_CORR_MIN`, `H_CORR_MAX`, `V_CORR_MIN` and `V_CORR_MAX` define the correlation window (the search area). From Broxton et al. (2012).

When you open the stereo parameter file, you'll find something like this in it about two thirds down the file:

```
# Initialization step: correlation kernel size
H_KERNEL 7
V_KERNEL 7

# Initialization step: correlation window size
H_CORR_MIN -500
H_CORR_MAX 275
V_CORR_MIN -175
V_CORR_MAX 175

# Subpixel step: subpixel modes
#
# 0 - disable subpixel correlation (fastest)
# 1 - parabola fitting (draft mode - not as accurate)
# 2 - affine adaptive window, bayes EM weighting (slower, but much
more accurate)

SUBPIXEL_MODE 2

# Subpixel step: correlation kernel size

SUBPIXEL_H_KERNEL 50
SUBPIXEL_V_KERNEL 50
```

Correlation kernel (called template region in Figure 1) is the part of the left image for which ASP is trying to find a counterpart in the right image. Its size is defined by parameters `H_KERNEL` and `V_KERNEL` in the stereo file. Parameters `H_CORR_MIN`, `H_CORR_MAX`, `V_CORR_MIN` and `V_CORR_MAX` define the correlation window, or the search area in the right image. In this example the correlation kernel is unusually small, the more common values being around 29 or so, but this is just an example where difficult areas failed with more standard settings. In contrast, the correlation window size is here somewhat larger than in the default settings.

For refining the results, you may want to change the subpixel correlation size. Here it is larger than usually, the default setting value being 25. These examples should give some idea what sort of values may be useful. It is good to keep some track of what settings have worked in the past, so you can easily use the same settings if you think the problem may be of similar type. If you don't want to make a huge number of different stereo parameter files, it is useful to know that one of the files created by ASP in the *outputfolder* is called *outputprefix-stereo.default*, which is the parameter file used to create the point cloud.

Full example stereo parameter files used in creating DEMs from LRO NAC, CTX, and HRSC images are given in Appendices 3–5, respectively.

3.4. Creating the DEMs

3.4.1. The Moon

The point cloud file is not yet the actual DEM. That is created with `point2dem` command. There are several different ways to use it for different purposes. Martian DEMs, if you want to compare them with other datasets, are somewhat more complicated to produce and are discussed separately in section 3.4.4. below. The Moon is an easier case, so to create a lunar DEM, the following works:

```
point2dem -r moon outputprefix-PC.tif -n
```

This creates a DEM (*outputprefix-DEM.tif*) assuming a spherical Moon with a radius of 1737.4 km, which is what you would normally use. The switch `-n` is not necessary, but it does create a smaller normalized image (*outputprefix-DEM-normalized.tif*) of the DEM for a quick check. This is handy, because for example Photoshop does not easily understand the actual DEM.

For georeferencing in ArcMap (see section 4.4.) and for a number of other purposes, you do want to have your image to match the DEM. This is done with `point2dem` as well:

```
point2dem -r moon --orthoimage outputprefix-L.tif outputprefix-PC.tif
```

This overlays the orthorectified left image (*outputprefix-L.tif*) of your stereo pair to exactly match the dimensions of the DEM. The resulting image (*outputprefix-DRG.tif*) will have all the holes that the DEM does, so I only use that in ArcMap for georeferencing the DEM. For the actual image analysis (e.g., mapping) after the georeferencing is done I use the original image cube, not the *DRG.tif produced here.

If you have reached this point by working on the cropped test image, skip the next section and go straight to the visualisation section 3.4.3.

3.4.2. Exporting the DEM to ArcGIS

To be able to use the final DEM and the orthorectified image (DRG) in ArcGIS, you need to use `gdal` for exporting them:

```
gdal_translate -ot float32 outputprefix-DEM.tif outputprefix-DEM_your_gdal_file.tif
```

```
gdal_translate -ot float32 outputprefix-DRG.tif outputprefix-DRG_your_gdal_file.tif
```

These resulting files are now ready to be used in ArcGIS, discussed in chapter 4. However, a couple of visualisation steps might be useful, so those are described next.

3.4.3. Visualising the DEM in ASP

Nothing described in this section is necessary for the analysis of DEMs, so this is for visualisation purposes only. However, for seeing if a cropped test DEM is any good and, therefore, if it makes sense to use the same parameters for the full stereo pair, these visualisation tools can give additional easily understandable information about the DEM quality. The good pixel map may not always tell the whole truth, and other debugging images provided by ASP and described in the guidebook are not that simple to understand at first.

To see how the DEM really looks like without having to go through all the trouble with ArcGIS, ASP has some handy visualisation tools. You can easily create hillshades, colour maps, coloured hillshades, and perhaps most importantly, perspective views. Note that these are only for visualisation purposes, the actual research on the DEM has to be done in ArcGIS. To create a simple colour-coded DEM (blue=low, red=high) instead of the greyscale normalized DEM created above, run this:

```
colormap outputprefix-DEM.tif -o outputprefix-DEM_color.tif
```

Making a hillshade is equally easy:

```
hillshade outputprefix-DEM.tif -o outputprefix-DEM_shade.tif -e 30
```

The `-e` switch defines the elevation of the light source, in this case 30° (0° being logically the horizon). You can also change the azimuth of the light source, but that works in a less logical manner. The default setting produces illumination from the left, and the azimuth increases from there anti-clockwise. Therefore, using switches `-e 5 -a 90` produces illumination from the bottom of the image at a shallow angle of 5°, and `-e 60 -a 225` more higher angle illumination from the upper right, and so on.

You can also colourise your hillshade:

```
colormap outputprefix-DEM.tif -s outputprefix-DEM_shade.tif -o outputprefix-DEM_colorshade.tif
```

The fanciest visualisation tool, however, uses the point cloud and the left image to create perspective views that you can rotate freely, and even make your own flyovers (although that is rather clumsy to do). Run the following:

```
point2mesh -l -s 5 outputprefix-PC.tif outputprefix-L.tif
```

Neither of the switches `-l` and `-s` is necessary, but both are recommendable. The first one gives you the option of toggling additional illumination in the perspective view on or off. The second one, `-s`, is a crucial one, as it defines the sampling rate. The default value is 10, which means it samples only every 10th pixel. Depending on the size of your DEM, you probably want a higher sampling rate, like every 5th pixel in this example. The limiting factor is the resulting file size. To see and play with the perspective view, use the `osgviewer`:

```
osgviewer outputprefix.ive
```

In the `osgviewer` you can fly around using mouse, but the following keys are useful:

- c: takes a screenshot (the same thing as hitting print screen)
- h: allegedly displays help, but in practice they are tiny hieroglyphs on the screen
- l: lighting on/off (if `-l` switch was used)
- t: texture on/off (i.e. only the DEM-based surface model (mesh), or with the left image)

w: wire model on/off

z: starts recording the camera path

Z: stops recording the camera path and saves it to *saved_animation.path*

To playback the recorded camera path:

```
osgviewer outputprefix.ive -p saved_animation.path
```

While in the playback, hitting space bar sets you back to the beginning of the path, and hitting c produces a short pause. Note that much of the information in the ASP guidebook about this part seems not to work real life, at least using the settings at LPI.

ASP does not include the full OpenSceneGraph package. I have never used it, so I don't know if it has some smart way of exporting the perspective views, but the *osgviewer* included in ASP does not. Therefore, the perspective views will be low resolution ones, but they can still be used in papers as long as you don't tell them they have a low resolution, and add some air in them in Photoshop. Examples of these perspective views can be seen in the cover page and in Figures 2–4 in the Appendices.

If the DEM looks good in whatever visualisation you prefer (to me the perspective views are by far the most useful ones), you can export it to ArcGIS (chapter 3.4.2), or if you are still testing with a cropped stereo pair, go back to cropping (chapter 3.1) and make the full-sized versions of the images if you don't have them already, and then go through the whole ASP process again and repeat it as many times as necessary.

3.4.4. Martian DEMs and the areoid

To create Martian DEMs and to compare them with MOLA or DLR DEM (which are referenced to MOLA) data, you need several additional steps. Run the following commands, which assume a spherical Mars with a radius of 3396 km (that's the MOLA spherical Mars, different from the IAU Mars spheroid given by the option *-r mars*, and also different from the IAU Mars 2000 ellipsoid):

```
point2dem --semi-major-axis 3396000 --semi-minor-axis 3396000 outputprefix-PC.tif
-n
```

```
point2dem --semi-major-axis 3396000 --semi-minor-axis 3396000 --orthoimage
outputprefix-L.tif outputprefix-PC.tif
```

Then the complicated part begins. First you have to download the 16 pixels per degree (3.705 km/px) areoid file *mega90n000eb.img* and the accompanying label file *mega90n000eb.lbl* from:

http://pds-geosciences.wustl.edu/mgs/mgs-m-mola-5-megdr-l3-v1/mgsl_300x/meg016/

I would make a separate directory just for the various areoid files that will be created.

The areoid must of course first be ISISed to a cube using:

```
raw2isis
```

Open the label file in a text editor. In the label file, *SAMPLES* are called “LINE_SAMPLES”, and there are 5760 of them. *LINES* are 2880. Although the label doesn’t directly say it, the *BITTYPE* is *16 bit signed data (SIGNEDWORD)*. *BYTEORDER* is found in the label, and it is *Most Significant Byte (MSB)*. Run it, and open the resulting cube in *qview* to make sure it looks alright (dark close to the poles, brighter near the equator, Olympus Mons visible as a bright spot).

Create a *maptemplate* for the areoid:

maptemplate

In *MAP* type the name of this new map template you are about to create. In *PROJECTION*, select *Simple Cylindrical Projection (SIMPLECYLINDRICAL)*. Under *Projection Parameters*, type “180” for the central longitude *CLON*. In *TARGOPT*, choose *User Input (USER)* and type “mars”. Keep the other *Target parameters* at default values, but you can change both *EQRADIUS* and *POLRADIUS* to 3396000 if you want to. However, these can be left to *IAU Default* settings as well, because when no ground control points and bundle adjustment are used, in my test case on Alba Mons this resulted in ~5 m elevation difference between the very final DEMs in ArcMap, while the difference to the MOLA point elevation was ~110 m. So as long as you are not making comparisons between different DEMs, whether you use the MOLA sphere (3396000 m) or the IAU defaults for the radii, in the end it makes no practical difference one way or the other.

Under *Ground Range Parameters*, select *User Entered Values (USER)* for *RNGOPT*. The areoid is global, so *MINLAT* is “-90”, *MAXLAT* “90”, *MINLON* “0”, and *MAXLON* “360”. Under *Resolution Parameters*, choose *RESOPT* to be *User Entered Scale (PPD)*, type “16” for *RESOLUTION*, and run it.

Next the areoid and the mapping information in the map template have to be combined. This is done by adding mapping labels to the areoid:

maplab

FROM is your areoid, and *MAP* is the newly created map template. Both *SAMPLE* and *LINE* are 0.5, which means the top left corner of the top left pixel. For *COORDINATES*, select *Allows LAT, LON as Input Values (LATLON)*. For *LAT*, type “90”, and for *LON*, type “0”. After creating the mapping labels, it’s useful to check with tracking in *qview* that the coordinates really are correct in the areoid. The bright dot of Olympus Mons has latitude of about 18°N and longitude 226°E.

The previous two steps, *maptemplate* and *maplab*, you only have to do once for the global areoid. Once it’s done, you can keep on using it as a basis for all of your DEMs.

Now you need to crop the global areoid to approximately match the DEM. What I do, is I open the left image of the stereo pair in *qview* and record the latitude and longitude of the top left corner of the top left pixel and the bottom right corner of the bottom right pixel. Because of undesirable edge effects in the next stage, it is best to add some margin. I usually use 0.05° margins. So do the math for the coordinates recorded for the left image of the stereo pair, for example like this:

Top left longitude -0.05°

Top left latitude +0.05°

Bottom right longitude +0.05°

Bottom right latitude -0.05°

Then open the global areoid in *qview*, and use tracking to find the coordinates (including the margins) of the DEM (i.e. the coordinates of the left image of the stereo pair recorded above), and record (write down, really) the Sample and Line for the top left and bottom right corners. For top left coordinates, round both Sample and Line down to the nearest integer, and for the bottom right coordinates round them up (again just to play it safe with the margins, and in the end you can't use decimals in Sample and Line anyway).

Based on the lines and samples, calculate the size (subtract the top left samples and lines from the bottom right ones) of the areoid to be cropped:

crop

Cropping is done same as before: *FROM* is the global areoid, *TO* would be smart to name it to be specific for this DEM, *SAMPLE* and *LINE* are the top left sample and line recorded above, *NSAMPLES* and *NLINES* are the size of the areoid calculated above, and *SINC* and *LINC* are 1.

Because of the enormous resolution difference between the 3705 m/px areoid and the ~6 m/px CTX (and whatever HRSC resolution happens to be), the cropped areoid has to be scaled up to be about the same fake resolution as the DEM. First calculate the scaling factor by dividing the areoid resolution with the DEM resolution, for example $3705 / 6 = \sim 618$. The scaling up is done with:

enlarge

FROM is naturally the cropped areoid, *TO* would benefit from a new name. Under *Scaling Parameters*, choose *INTERP* to be *Bi-linear interpolation (BILINEAR)*, as in my experience it gives the best results in these types of situations. Cubic convolution would be another good option, but nearest neighbor doesn't work in this case. *MODE* should be *Enlarge by a scaling factor (SCALE)*, and both *SSCALE* and *LSCALE* should be, in this example, "618". After running it, it's good to once again check it in *qview*. The left and right edge of the upscaled areoid will look coarser than the rest of it, but that's not a problem, as it fits into the margins. And there is also likely going to be some "washboarding" in the areoid, but that's just how it is. Record the latitude and longitude of the top left and bottom right coordinates of the enlarged areoid.

This final areoid.cub (along with the gdaled *-DEM.tif and the accompanying gdaled *-DRG.tif) can now be copied (just drag and drop from the SSH File Transfer Window, that's the easiest way) to the folder where your ArcGIS project is (or is going to be).

4. Processing and analysing the DEMs in ArcMap

4.1. Importing and re-projecting the data

Whether or not you already have a project in ArcMap, it doesn't really matter. As always in ArcMap, you should add the new datasets using the *Add Data* button. Start by adding the areoid (assuming you are working with Martian data, which is the default assumption here – in the lunar case everything is otherwise the same, just skip the areoid parts), DEM, and DRG. If and when Arc wants to build pyramids, OK that. It may also give a Geographic Coordinate Systems Warning (if you have an existing project, it will), but you can just ignore it for now, because it will be fixed

later. You should have three grey boxes now, one slightly larger (the areoid) and other two identical ones inside it (not necessarily in the center of the areoid).

ArcGIS, of which ArcMap is the most commonly used part, is your worst enemy, so some routines are useful, just to make sure things get always done the right way, and simply to make life a little easier. Even if Arc didn't complain now about the projection, it will eventually (or you just won't be able to georeference anything, which is more likely), which is why I always reproject every single dataset that I import to my Arc projects. And I do it before I really do anything else.

To start with, try to locate a *Search* window. Type “*project raster*” in it. It should find *Project Raster (Data Management)* tool. The chances are Arc will crash when you click it. It happens. It just did to me right now. It's not a bug, it's just a fancy feature of Arc. So saving the project immediately after importing the datasets would have been a smart thing to do... Once *Project Raster* chooses to work, the steps for each file are the same: *Input Raster* is your input file (areoid, DEM, or DRG). *Output Raster Dataset* only accepts file names up to 13 characters. You can give them longer apparent names in the table of contents later. Arc is not always (ever?) too logical about the default folder locations, so be sure you know where your output raster will be.

In *Output Coordinate System*, you should follow this path: *Spatial Reference Properties* → *Select* → *Geographic Coordinate Systems* → *Solar System* → *Mars* → *Mars 2000.prj*. For lunar work, *Moon 2000.prj* can be found under *Earth*. Then just *Add* and *OK*. Back in the *Project Raster* window, you should now see the *Output Coordinate System* to be *GCS_Mars_2000*. The rest of the settings can be defaults. Repeat this to all of the datasets.

While projecting, ArcMap may have already stretched the rasters so you can actually see them. If not, right-clicking on the file in the *Table of Contents* gives you the option of fiddling with *Properties*, although it is not that important yet if you are working with Martian data. If you're working with lunar data, jump right to the section 4.3. about Displaying the DEM.

4.2. Subtracting the areoid

Using the *Search* window, find the tool *Minus (Spatial Analyst)*. ArcMap may crash, as it again just did for me. When it eventually works (it may crash several times; mine just crashed for the third time during the writing of this chapter 4), *Input raster or constant value 1* is your DEM, *Input raster or constant value 2* is the areoid, and *Output raster* is the final DEM that has been the goal of this whole lengthy exercise. Note that because no ground control points or bundle adjustment were used in the creation of the DEM, the elevations will still be different from the MOLA point values, which are the closest approximation of “the truth”. However, the elevations of the DEM are still comparable to MOLA points, they are merely offset by some fairly random value. Usually the DEM and MOLA point elevations are within a couple of hundred meters. Based on all the tests that I have run, the research I've carried out and what I have read, DEMs created this way are a reliable source of detailed topographic information as long as one does not try to compare the elevations between different DEMs. For sanity checks, always use MOLA points (obviously LOLA in the lunar case), photogeology, and common geologic sense.

4.3. Displaying the DEM

In the *Table of Contents*, the rasters can be moved up and down by simply grabbing them with the mouse. You'll want the DEM to be on top of the DRG. Right-clicking the name of the file in the

Table of Contents gives you the option of changing the *Properties*. In the *Symbology* tab, *Color Ramp* is something you very likely want to change. The traditional blue-to-red ramp is available in the lower part of the pull-down menu. If you use it, remember to check the *Invert* box at lower right. The type of stretch varies between each individual case, but a good starting point is *Standard Deviations*, with n being “2”. In the *Display* tab, you will likely want to change the *Transparency* to somewhere between 40 and 60%. During this process, ArcMap will likely ask for your permission to calculate statistics, which is what you want it to do. Now you should have your partially transparent DEM overlaid exactly on top of your DRG, and basically it is ready for analysis. However, if your project includes some background data, for example a WAC image or mosaic, THEMIS mosaic, or MOLA DEM, it is likely that your DEM and DRG are slightly offset, and need to be georeferenced.

4.4. Georeferencing

Normally you want to have some sort of a background layer that you presume to represent the immovable reference framework. Some of them are closer to “the truth” than others, but for example for Mars, THEMIS 100 m/px infrared daytime mosaic is a pretty good option (see Appendix 1 for the URL). Assuming you have such a background, and your DEM, DRG, and/or your original image file (cube, tif, jpeg2000...) are in the wrong place, you need to use the *Georeferencing* toolbar. If it is not visible, choose it from *Customize* → *Toolbars*.

In the *Georeferencing* toolbar you should have all your rasters available in the *Layer* menu. If they are not, you need to run *Project raster* on the missing ones again. ArcGIS 10 is particularly stupid in this sense, hence my routine of re-projecting every raster I add immediately after I import it even though it is supposedly already in the correct datum and projection.

I have so far met Martian and lunar ASP DEMs that have been only offset, and none have required warping. Thus, no control points are required, and georeferencing is easy. Choose the raster you want to move to the correct place from the *Layer* pull-down menu. The icon to the right of it usually shows *Rotate* as a default, but the pull-down menu allows you to choose *Shift*, which is what you want. This changes the cursor to a hand, which is very confusing, because it looks exactly the same as the hand of the *Pan* tool that you would normally use to move around. With this *Shift* hand, move the raster you want to the correct location. If everything is offset with respect to everything else (which is quite likely), for example in the case of the Moon I would first move the NAC cube to match the WAC background and georeference it (see below), then move the DRG (which, as a reminder, is the left NAC cube projected to match exactly the DEM) to match the NAC, and finally move the DEM to match the DRG. This last part can be done very precisely on sub-pixel level, because the spatial extent of the DEM and the DRG are the same, so if you zoom in close enough, you can exactly match the corners of the DEM and the DRG.

Once you have moved your raster-to-be-georeferenced to the right location, choose *Rectify* from the *Georeferencing* pull-down menu. This opens up a *Save as* window. As usual, double-check the *Output Location*, because it is likely going to be inherited from the previous project where georeferencing was used. Give the raster a new *Name*, and save it.

Although the raster you moved appears to be in the place where you moved it to, it is not permanently there, so you have to use the *Add Data* button to include the newly georeferenced raster in your project. And just for the fun of it, you also have to stretch it again if you want it to look the same as the original. Repeat this for all of the rasters you need to georeference. Depending

on the type of data, *Rectify* may not always work. In such a case you need to use *Update Georeferencing* instead.

4.5. Analysing the DEM and exporting the data

Once all the rasters are in their correct locations, you can actually start analysing the DEM. There are countless ways of doing that of course and I know virtually nothing about most of them, but taking profiles is extremely useful and something that is almost always required. For that you need the *3D Analyst* toolbar. Again, if it is not visible, choose it from *Customize* → *Toolbars*. Choose the *Layer* you want analyse, and draw your profile using the *Interpolate Line* button (the one with the zigzag icon). First click starts the profile, subsequent clicks create points where you might want to make a turn in the profile, and double-clicking ends the profile. Note that you cannot use the same profile for more than one layer. So if you need to take the same profile on several different rasters, you need to draw the profile separately for every raster. For precise work, zoom in a lot so you are at least in the same pixel for each of the profiles.

Create Profile Graph button on the right, surprisingly enough, usually creates the profile graph. When it does, right-click on the graph. Choose *Export* from the menu that appears, select *Data* tab, choose the format you like, and then just *Save*. In the resulting Excel-file (or whatever you want to use), the Y column displays the elevation from your DEM in meters. The X column is normally a bit more problematic, because it tends to be given in degrees. There is probably some way to solve this in ArcMap, but I don't know what it is. Therefore, I use the *Measure* tool (button with a ruler icon) to measure the length of the profile, and then in Excel I use the degrees to calculate the X-values in kilometers for the points between the starting and ending points of the profile.

It is worth noting that right-clicking the profile gives the option of seeing its *Properties*. In the *Length* tab, coordinates of the starting and ending points are correct and useful information, whereas the length of the profile there is practically always wrong (There are sort of sensible reasons why this is so. For the same reasons, inserting a scale bar when creating a map is not advisable, because it will be wrong.). So to get the true length of the profile, you have to use the *Measure* button.

And that is pretty much all there is to it from a processing point of view. The rest is just doing the analysis and trying to make some sense of it all.

Acknowledgements

This project was supported by NASA Mars Data Analysis Program (MDAP) grant NNX09AI42G (PI Patrick J. McGovern), and earlier kinks in the learning curve were funded by NASA Lunar Science Institute contract NNA09DB33A (PI David A. Kring), and by NASA under the LPI Cooperative Agreement NNX08AC28A. A big thank you to Brian Fessler for all his help with ISIS and ArcGIS, and setting up ASP.

Appendix 1. Some additional links and references

Planetary datasets, and particularly any useful information about using them, are notoriously difficult to find. Here just some of the sites that I have found useful.

- ◆ The reference to the ASP guidebook is:

Broxton M. J., Beyer R. A., Moratto Z., Lundy M., Husman K., 2012. The Ames Stereo Pipeline: NASA's Open Source Automated Stereogrammetry Software. A part of the NASA NeoGeography Toolkit. Version 2.0.0. Intelligent Robotics Group, NASA Ames Research Center, 88 p.

- ◆ A couple of LPSC abstracts that the ASP developers want to be referenced are:

Broxton M. J., Edwards L. J., 2008. The Ames Stereo Pipeline: automated 3D surface reconstruction from orbital imagery, Abstract #2419, 39th Lunar and Planetary Science Conference.

<http://www.lpi.usra.edu/meetings/lpsc2008/pdf/2419.pdf>

Moratto Z. M., Broxton M. J., Beyer R. A., Lundy M., Husmann K., 2010. Ames Stereo Pipeline, NASA's open source automated stereogrammetry software. Abstract #2364, 41st Lunar and Planetary Science Conference.

<http://www.lpi.usra.edu/meetings/lpsc2010/pdf/2364.pdf>

- ◆ One paper where I know a similar approach as presented here (no ground control points, no bundle adjustment) was used, with the author list even including one of the ASP developers, is:

Lefort A., Burr D. M., Beyer R. A., Howard A. D., 2012. Inverted fluvial features in the Aeolis-Zephyria Plana, western Medusae Fossae Formation, Mars: Evidence for post formation modification. Journal of Geophysical Research, 117, E03007, doi: 10.1029/2011JE004008.

- ◆ ASP home page, with links to the ASP guidebook and other useful things, including the invaluable email list:

<http://ti.arc.nasa.gov/tech/asr/intelligent-robotics/ngt/stereo/>

- ◆ ASP-developer Zach Moratto's blog with many useful tips:

<http://lunokhod.org/>

- ◆ ISIS guidebook:

<http://isis.astrogeology.usgs.gov/Application/index.html>

- ◆ ISIS discussion forum:

<https://isis.astrogeology.usgs.gov/IsisSupport/>

- ◆ USGS guide on Working with Mars Reconnaissance Orbiter CTX Data:

http://isis.astrogeology.usgs.gov/IsisWorkshop/index.php/Working_with_Mars_Reconnaissance_Orbiter_CTX_Data

- ◆ GDAL conversion tips from Trent Hare:

<https://isis.astrogeology.usgs.gov/IsisSupport/index.php/topic,2172.0.html>

- ◆ What is SPICE:

<http://naif.jpl.nasa.gov/naif/spiceconcept.html>

- ◆ Global THEMIS IR daytime 512 ppd (100 m/px) mosaic:

ftp://pdsimage2.wr.usgs.gov/pub/pigpen/mars/themis/themis_ir_global_mosaic/Global_IRday_512ppd/

- ◆ LOLA points, to compare with lunar stereo DEMs:

<http://ode.rsl.wustl.edu/moon/indextools.aspx>

- ◆ LOLA DEM in simple cylindrical projection and jpeg2000 format:

http://imbrium.mit.edu/DATA/LOLA_GDR/CYLINDRICAL/JP2/

- ◆ LRO WAC 128 px/degree stereo DEM:

http://wms.lroc.asu.edu/lroc/global_product/color_shaded_relief_grid

- ◆ MOLA points, to compare with Martian stereo DEMs:

<http://ode.rsl.wustl.edu/mars/indextools.aspx?displaypage=molapedr>

- ◆ MOLA DEMs and areoid, as well as radius data:

<http://pds-geosciences.wustl.edu/missions/mgs/megdr.html>

- ◆ CTX home page:

http://www.msss.com/all_projects/mro-ctx.php

- ◆ LROC home page:

<http://www.lroc.asu.edu/>

Appendix 2. Some very basic UNIX/LINUX commands

<code>cd <i>directoryname</i></code>	Goes to a directory.
<code>cd ..</code>	Goes up one level in directories.
<code>cp <i>oldfilename newfilename</i></code>	Copies a file.
<code>exit</code>	A sophisticated way of quitting. Note that you will lose your terminal history (available with the up-arrow) when you exit.
<code>gedit</code>	Starts a text editor.
<code>ls</code>	Lists the files in the current directory.
<code>mkdir <i>directoryname</i></code>	Makes a new directory.
<code>more <i>filename</i></code>	Shows the contents of the file on the screen, stopping when the screen is full. Hitting Ctrl-z stops it.
<code>rm <i>filename</i></code>	Removes (deletes) the file.
<code>rmdir <i>directoryname</i></code>	Removes (deletes) an <i>empty</i> directory.

Appendix 3. An example of a stereo parameter file for processing LRO NAC stereo pairs

These parameters were used to produce nice DEMs of the central part of crater Eimmart A from LRO NAC images M152451994LE.IMG and M152445210LE.IMG.

```
#####
##      PREPROCESSING      ##
#####

# Pre-alignment options

DO_INTERESTPOINT_ALIGNMENT 0
INTERESTPOINT_ALIGNMENT_SUBSAMPLING 0
DO_EPIPOLAR_ALIGNMENT 0

# Normalization options

FORCE_USE_ENTIRE_RANGE 1
DO_INDIVIDUAL_NORMALIZATION 0

# Preprocessing filter mode:
#
# 0 - None
# 1 - Gaussian Blur
# 2 - Laplacian of Gaussian Blur (recommended)
# 3 - Sign of the LoG Filter

PREPROCESSING_FILTER_MODE 2

# Gaussian convolution kernel size (1-sigma) for pre-processing
# modes 1, 2 and 3.

SLOG_KERNEL_WIDTH 1.5

#####
#####      CORRELATION      #####
#####

# Select a cost function to use for initialization:
#
# 0 - absolute difference (fast -- use in conjunction with SLoG)
# 1 - squared difference
# 2 - normalized cross correlation (recommended)

COST_MODE 2

# Turn this up to improve the results of the discrete correlation
# step. This will reduce the number of missing pixels, but reduce the
# overall accuracy of the disparity estimates.
#
# WARNING: Cost blurring must be used in conjunction with affine
# adaptive window subpixel below, which are capable of achieving
# highly accurate results even when seeded by slightly inaccurate
# results from the initialization step.

COST_BLUR 12
```

```

# Initialization step: correlation kernel size

H_KERNEL 7
V_KERNEL 7

# Initialization step: correlation window size
H_CORR_MIN -500
H_CORR_MAX 275
V_CORR_MIN -175
V_CORR_MAX 175

# Subpixel step: subpixel modes
#
# 0 - disable subpixel correlation (fastest)
# 1 - parabola fitting (draft mode - not as accurate)
# 2 - affine adaptive window, bayes EM weighting (slower, but much more
accurate)

SUBPIXEL_MODE 2

# Subpixel step: correlation kernel size

SUBPIXEL_H_KERNEL 25
SUBPIXEL_V_KERNEL 25

#####
#####          FILTERING          #####
#####

# Fill in holes up to 100,000 pixels in size with an inpainting method

FILL_HOLES 1

# Automatic "erode" low confidence pixels

RM_H_HALF_KERN 5          # Note: a 5*5 halfkernel = 11*11 kernel -> 121 pixels
RM_V_HALF_KERN 5
RM_MIN_MATCHES 60        # Units = percent
RM_THRESHOLD 3
RM_CLEANUP_PASSES 1

#####
#####          POST_PROCESSING          #####
#####

# Size max of the universe in meters and altitude off the ground.
# Setting both values to zero turns this post-processing step off.

NEAR_UNIVERSE_RADIUS 0.0
FAR_UNIVERSE_RADIUS 0.0

```

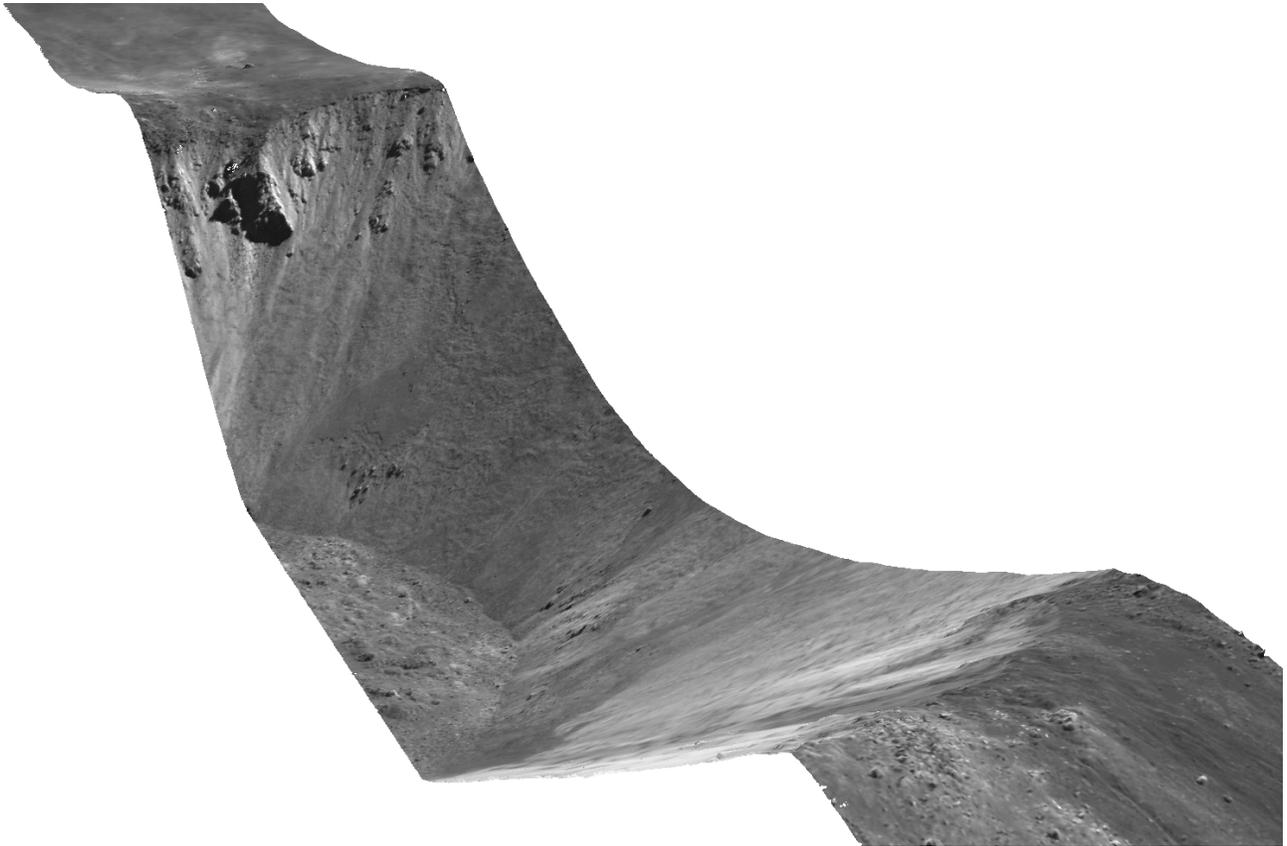


Figure 2. A perspective view of Eimmart A created with ASP using NAC images and stereo parameters described above. Diameter of the crater is ~7 km.

Appendix 4. An example of a stereo parameter file for processing CTX stereo pairs

These parameters were used to produce nice DEMs of the graben on the eastern flank of Alba Mons from CTX images G20_025970_2217_XN_41N102W.IMG and G20_025904_2209_XN_40N102W.IMG. See the cover page for an example and comparisons with other datasets.

```
#####
##      PREPROCESSING      ##
#####

# Pre-alignment options

DO_INTERESTPOINT_ALIGNMENT 1
INTERESTPOINT_ALIGNMENT_SUBSAMPLING 0
DO_EPIPOLAR_ALIGNMENT 0

# Normalization options

FORCE_USE_ENTIRE_RANGE 1
DO_INDIVIDUAL_NORMALIZATION 0

# Preprocessing filter mode:
#
# 0 - None
# 1 - Gaussian Blur
# 2 - Laplacian of Gaussian Blur (recommended)
# 3 - Sign of the LoG Filter

PREPROCESSING_FILTER_MODE 3

# Gaussian convolution kernel size (1-sigma) for pre-processing
# modes 1, 2 and 3.

SLOG_KERNEL_WIDTH 1.5

#####
#####      CORRELATION      #####
#####

# Select a cost function to use for initialization:
#
# 0 - absolute difference (fast)
# 1 - squared difference (faster .. but usually bad)
# 2 - normalized cross correlation (recommended)

COST_MODE 0

# Turn this up to improve the results of the discrete correlation
# step. This will reduce the number of missing pixels, but reduce the
# overall accuracy of the disparity estimates.
#
# WARNING: Cost blurring must be used in conjunction with affine
# adaptive window subpixel below, which are capable of achieving
# highly accurate results even when seeded by slightly inaccurate
# results from the initialization step.
```

```

COST_BLUR 0

# Initialization step: correlation kernel size

H_KERNEL 25
V_KERNEL 25

# Initializaion step: correlation window size
#H_CORR_MIN -100
#H_CORR_MAX 100
#V_CORR_MIN -100
#V_CORR_MAX 100

# Subpixel step: subpixel modes
#
# 0 - disable subpixel correlation (fastest)
# 1 - parabola fitting (draft mode - not as accurate)
# 2 - affine adaptive window, bayes EM weighting (slower, but much more
accurate)

SUBPIXEL_MODE 2

# Subpixel step: correlation kernel size

SUBPIXEL_H_KERNEL 25
SUBPIXEL_V_KERNEL 25

#####
#####          FILTERING          #####
#####

# Fill in holes up to 100,000 pixels in size with an inpainting method

FILL_HOLES 1

# Automatic "erode" low confidence pixels

RM_H_HALF_KERN 5          # Note: a 5*5 halfkernel = 11*11 kernel -> 121 pixels
RM_V_HALF_KERN 5
RM_MIN_MATCHES 60        # Units = percent
RM_THRESHOLD 3
RM_CLEANUP_PASSES 1

#####
#####          POST_PROCESSING          #####
#####

# Size max of the universe in meters and altitude off the ground.
# Setting both values to zero turns this post-processing step off.

NEAR_UNIVERSE_RADIUS 0.0
FAR_UNIVERSE_RADIUS 0.0

```

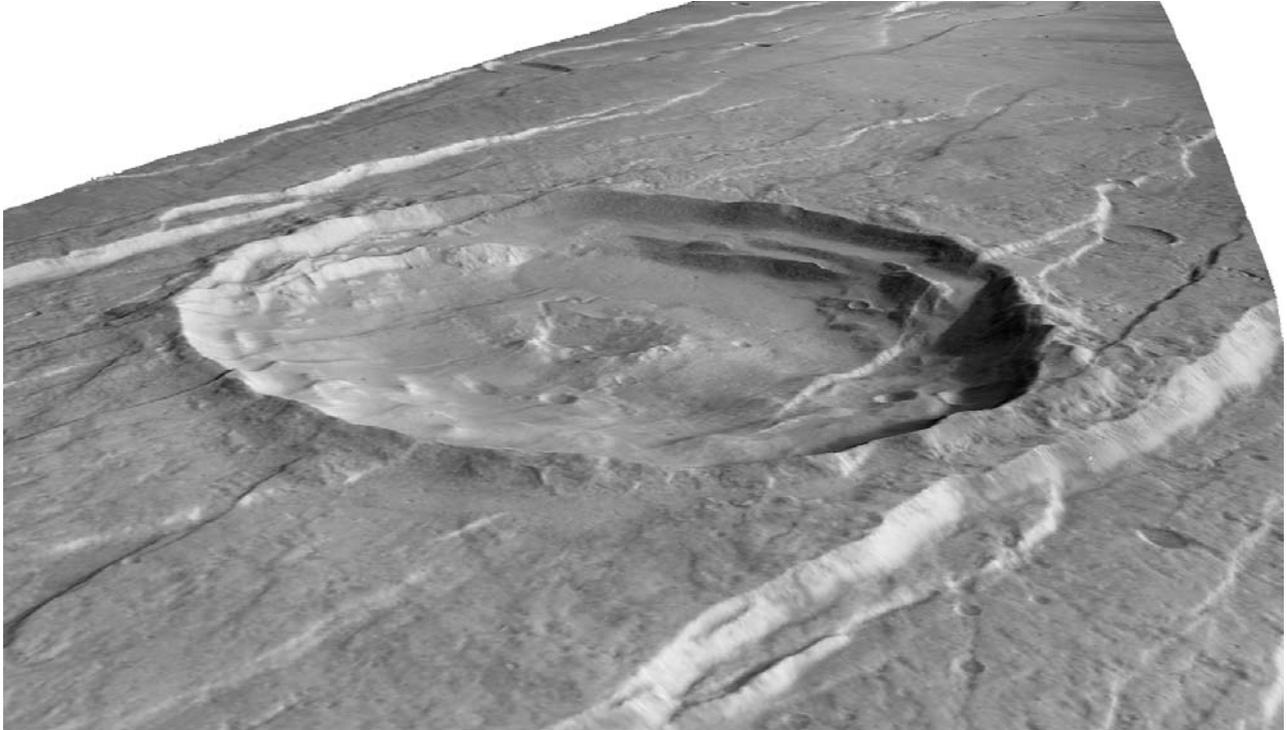


Figure 3. A perspective view of a central pit crater and graben in eastern Alba Mons created with ASP using CTX images and stereo parameters described above. Diameter of the crater is ~19 km. Another example can be seen on the cover page.

Appendix 5. An example of a stereo parameter file for processing HRSC stereo pairs

These parameters were used to produce reasonably nice DEMs of the graben on the eastern flank of Alba Mons using HRSC images H0068_0009_S12.IMG and H0068_0009_S22.IMG.

```
DO_INTERESTPOINT_ALIGNMENT 0
INTERESTPOINT_ALIGNMENT_SUBSAMPLING 0
DO_EPIPOLAR_ALIGNMENT 0

FORCE_USE_ENTIRE_RANGE 1
DO_INDIVIDUAL_NORMALIZATION 0

PREPROCESSING_FILTER_MODE 2
SLOG_KERNEL_WIDTH 1.5

COST_MODE 0
COST_BLUR 0

H_KERNEL 22
V_KERNEL 22

# These numbers need to be customized for every HRSC stereo session
H_CORR_MIN -30
H_CORR_MAX 30
V_CORR_MIN -130
V_CORR_MAX 20

SUBPIXEL_MODE 2

SUBPIXEL_H_KERNEL 35
SUBPIXEL_V_KERNEL 35

FILL_HOLES 1

RM_H_HALF_KERN 5          # Note: a 5*5 halfkernel = 11*11 kernel -> 121 pixels
RM_V_HALF_KERN 5
RM_MIN_MATCHES 60        # Units = percent
RM_THRESHOLD 3
RM_CLEANUP_PASSES 1

NEAR_UNIVERSE_RADIUS 0.0
FAR_UNIVERSE_RADIUS 0.0
```

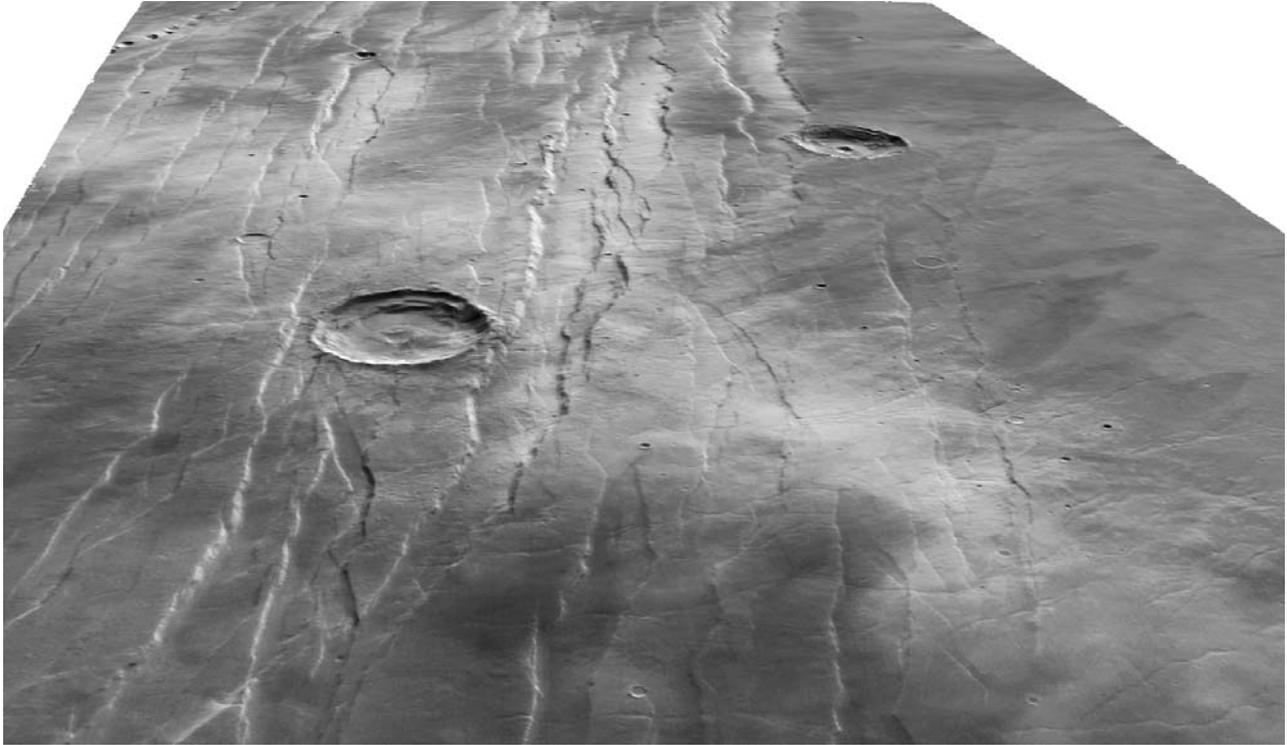


Figure 4. A perspective view of graben and central pit craters in eastern Alba Mons created with ASP using HRSC images and stereo parameters described above. The width of the view is ~155 km, and the crater in the center left is the same as in Figure 3.