



# Earth and Space Science

## RESEARCH ARTICLE

10.1029/2018EA000409

### Special Section:

Planetary Mapping: Methods, Tools for Scientific Analysis and Exploration

### Key Points:

- The Ames Stereo Pipeline is an open source user-friendly suite of stereogrammetry tools
- The software can be applied to data from the Earth and other planets in the solar system
- The software suite contains tools to help prepare data for stereo correlation and processes output terrain data

### Correspondence to:

R. A. Beyer,  
Ross.A.Beyer@nasa.gov

### Citation:

Beyer, R. A., Alexandrov, O., & McMichael, S. (2018). The Ames Stereo Pipeline: NASA's open source software for deriving and processing terrain data. *Earth and Space Science*, 5, 537–548.  
<https://doi.org/10.1029/2018EA000409>

Received 1 MAY 2018

Accepted 27 AUG 2018

Accepted article online 31 AUG 2018

Published online 21 SEP 2018

## The Ames Stereo Pipeline: NASA's Open Source Software for Deriving and Processing Terrain Data

Ross A. Beyer<sup>1,2</sup> , Oleg Alexandrov<sup>2</sup>, and Scott McMichael<sup>2</sup>

<sup>1</sup>Sagan Center at the SETI Institute, USA, <sup>2</sup>NASA Ames Research Center, USA

**Abstract** The NASA Ames Stereo Pipeline is a suite of free and open source automated geodesy and stereogrammetry tools designed for processing stereo images captured from satellites (around Earth and other planets), robotic rovers, aerial cameras, and historical images, with and without accurate camera pose information. It produces cartographic products, including digital terrain models, ortho-projected images, 3-D models, and bundle-adjusted networks of cameras. Ames Stereo Pipeline's data products are suitable for science analysis, mission planning, and public outreach.

### 1. Introduction

The NASA Ames Stereo Pipeline (ASP) software (latest version 2.6.1, Beyer et al., 2018) is a suite of software for deriving high-quality terrain information from visual images via stereogrammetry and, optionally, photocalinometry. It also contains a collection of tools meant to help manipulate and process terrain information and perform cartography tasks, including map-projection, point cloud and digital terrain model (DTM) registration, automatic registration of cameras, data format conversion, and data visualization.

Although initially developed for ground control and scientific visualization applications, the Stereo Pipeline has evolved to address orbital stereogrammetry and cartographic applications. In particular, long-range spacecraft mission planning requires detailed knowledge of planetary topography, and high-resolution topography is often derived from stereo pairs captured from orbit. Orbital mapping satellites are sent as precursors to planetary bodies in advance of landers and rovers. They return a wealth of images and other data that help mission planners and scientists identify areas worthy of more detailed study. Topographic information often plays a central role in this planning and analysis process.

ASP has been worked on in one form or another since the mid-1990s at NASA Ames Research Center. It was initially developed locally to support stereo cameras on test rovers. Its first use on a planetary lander was with the Imager for Mars Pathfinder camera (Nguyen et al., 2001). It was further enhanced by Edwards et al. (2005) to support the Mars Exploration Rover (MER) cameras. As part of an industry-led project to create terrain generation for the Mars Global Surveyor (MGS) mission, ASP was adapted to work with orbital cameras (Edwards & Broxton, 2006).

In 2006 the code was forked with private versions still in use by previous developers carrying the ASP name, but at this time a version intended for public release was developed, this effort was led by Michael Broxton and Ross Beyer. By 2008, the code had been adapted to integrate with the Integrated Software for Imagers and Spectrometers version 3 (ISIS3, Anderson, 2008; Anderson et al., 2004; U. S. Geological Survey, 2018), which allowed ASP to more easily operate on all of the instruments for which there was an ISIS3 camera model (Broxton & Edwards, 2008).

In October 2009, the software was publicly released as version 1.0.0 (Moratto et al., 2010), under a NASA Open Source Agreement license. In 2010, we migrated to develop and distribute the software from GitHub:

<https://github.com/NeoGeographyToolkit/StereoPipeline>

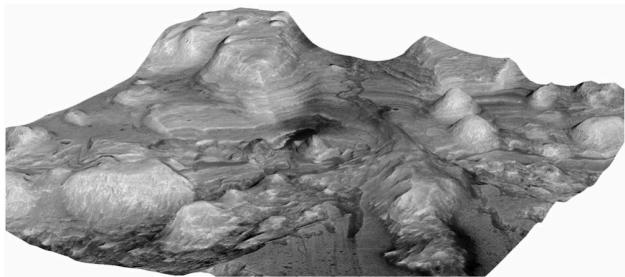
Since version 2.0, released in June 2012, ASP has been under the Apache 2 license.

In the last several years, more ASP development effort has been focused toward Earth images, and the software now has the ability to process commercial Earth-orbiting data (Shean et al., 2016), hence allowing ASP to be applied to almost every solid body in the Solar System.

©2018. The Authors.

This is an open access article under the terms of the Creative Commons Attribution-NonCommercial-NoDerivs License, which permits use and distribution in any medium, provided the original work is properly cited, the use is non-commercial and no modifications or adaptations are made.





**Figure 1.** Perspective view of the mound in Gale Crater on Mars that the Curiosity Rover will explore. This terrain was created from HiRISE images PSP\_009149\_1750 and PSP\_009294\_1750, and the scene is approximately 2.8 km wide.

ASP operates on any data that were produced by an instrument that has an ISIS3 sensor model (e.g., HiRISE in Figure 1; CTX in Figure 2; MOC, MER, High Resolution Stereo Camera, LROC in Figure 3; Apollo Metric, GalileoSSI, New Horizons LORRI, etc.), any data that have a rational polynomial camera model (e.g., WorldView 1/2/3, Pleiades, ASTER, GeoEye, SPOT5, Cartosat, and Perusat), and data from sensors that can be approximated as pinhole cameras (e.g., Operation IceBridge in Figure 4).

In addition to source code downloads, ASP is also available for download as a binary release for Linux and macOS and comes with a comprehensive 200-page manual that describes how to install it, provides several tutorials and practical data processing examples, and dwells on the inner workings of algorithms and approaches in more detail than this manuscript (Beyer et al., 2018).

In the following sections we will give an overview of ASP's stereo processing and results, as well as its other tools and features.

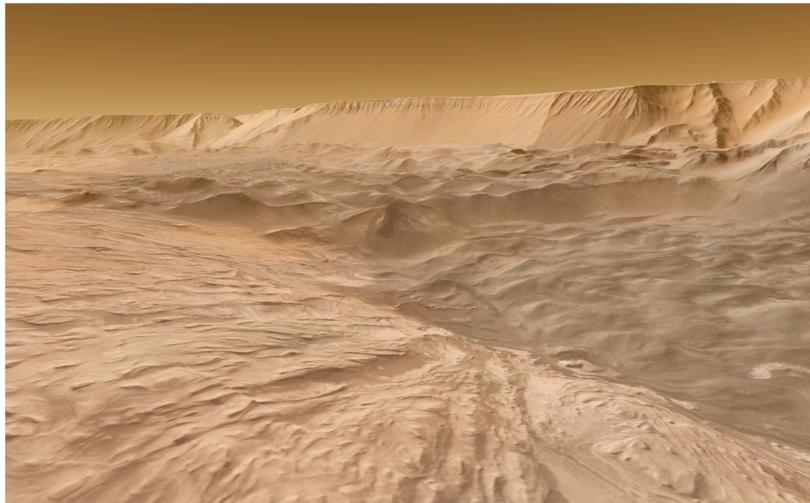
## 2. From Images to a 3-D Point Cloud

The inputs to ASP are two images, having different perspectives on the same scene, and their associated camera information. ASP takes several steps to go from these to a 3-D point cloud which can then be visualized (Figure 5). These steps are implemented as separate tools, that are called sequentially, which we describe as follows.

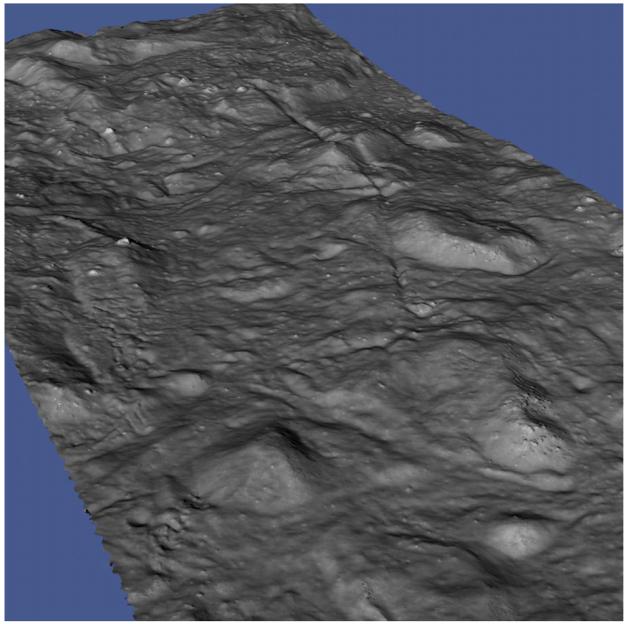
### 2.1. Preprocessing

Often, data as they are delivered from instruments are not amenable to be immediately ingested into a stereo algorithm. Real data are complicated, not only are there potential issues with instrument-related factors like read-out noise and flat fielding, but also issues related to the known positions of the camera stations and complicating factors from the landscape of the scene itself.

Fortunately, most instruments have detailed camera models that describe the optics of the camera, referred to as camera intrinsic parameters, and for most scientific instruments, they are well known. They also typically have custom photometric normalization algorithms that provide photometric processing to clean and condition the images to remove instrument-related artifacts. It is imperative to apply these corrections before stereo correlation.



**Figure 2.** Perspective view across the floor of West Candor Chasma on Mars, the north rim is visible in the distance. This terrain was created from several CTX image pairs that span the almost 90 km from the foreground terrain to the rim in the distance, and then draped with a mosaic of those CTX images.

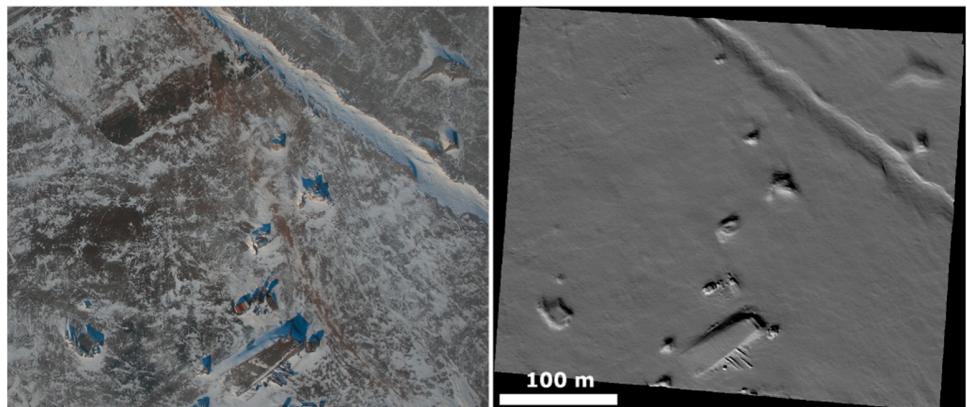


**Figure 3.** Perspective view of the floor of Jackson Crater on the Moon. This terrain was created from LROC images M105585995R and M105593167R, and the scene is approximately 7.5 km wide.

loration is mostly horizontal with only a small vertical component. We found out that this works well even for linescan cameras when, strictly speaking, the epipolar geometry assumption does not hold and epipolar lines become curves.

A second approach ASP employs to make the left and right images more similar, and hence correlation more successful, is to map-project them onto a preexisting lower-resolution terrain model (Figure 6). In effect, this generalizes affine epipolar alignment by replacing it with a nonlinear transform. It is important to note that after we find pixel-to-pixel correspondences among the two map-projected images, we convert these pixels back to the original images for the purpose of doing triangulation.

The disadvantage of this approach is that it requires an existing terrain model of some kind on which to map-project the images. If such a data set is not available, a solution that we employ is to use a two-pass approach. First we have ASP create a rough terrain model, then this terrain is blurred and its holes are filled, and then it is used in map-projection. Following this, a second pass of stereo produces a higher quality terrain.



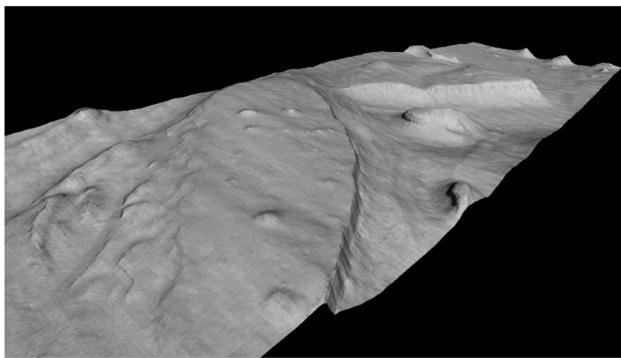
**Figure 4.** Left is Operation IceBridge aerial photo 2015\_04\_01\_14658, and on the right a hillshaded terrain model created from that image and its neighbor along this flightline over Arctic ice near Greenland.

The stereo triangulation process later on also requires that the position and pose of the camera stations are well known. Typically, some camera information is available for the data, but this often needs to be refined via a technique known as bundle adjustment (Triggs et al., 2000). ASP provides such functionality, which we describe in section 4.1, but bundle adjustment can also be performed via other tools.

The camera position and pose are often referred to as camera extrinsic parameters, and the best case is when they are available via metadata delivered with the images. However, there can certainly be cases where such extrinsic information does not exist, either from historic data sets which do not capture it, or images from modern systems that simply do not capture that kind of detailed information (like drone camera images, etc.). In this case, ASP has a **camera\_solve** tool based on the Theia Vision Library (Sweeney, 2013) that can help attempt to derive those extrinsics for the images, but it has not been extensively tested.

## 2.2. Image Alignment

Before performing stereo correlation, typically some kind of alignment process is needed to narrow down the amount of image pixels that must be searched through to find correspondences among the two images used in stereo. The default alignment in ASP is affine epipolar, using the algorithm in Hartley and Zisserman (2004). In this approach, an affine transformation is applied to both the left and right images to make the epipolar lines approximately horizontal, hence the search range for corre-



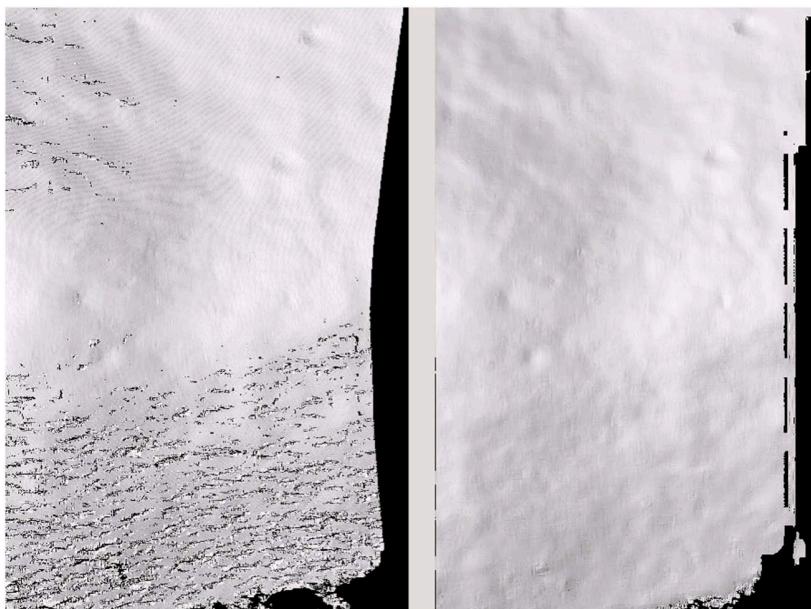
**Figure 5.** This 3-D model was created from a Mars Orbiter Camera image pair M01/00115 and E02/01461 with an early version of Ames Stereo Pipeline. The model, shown here with the original image draped over the terrain, is shown without vertical exaggeration, and is roughly 2 km wide in the cross-track dimension.

the computational work among multiple CPU cores and/or separate computers. Iterating over increasing resolutions limits the search range and thus reduces computation time.

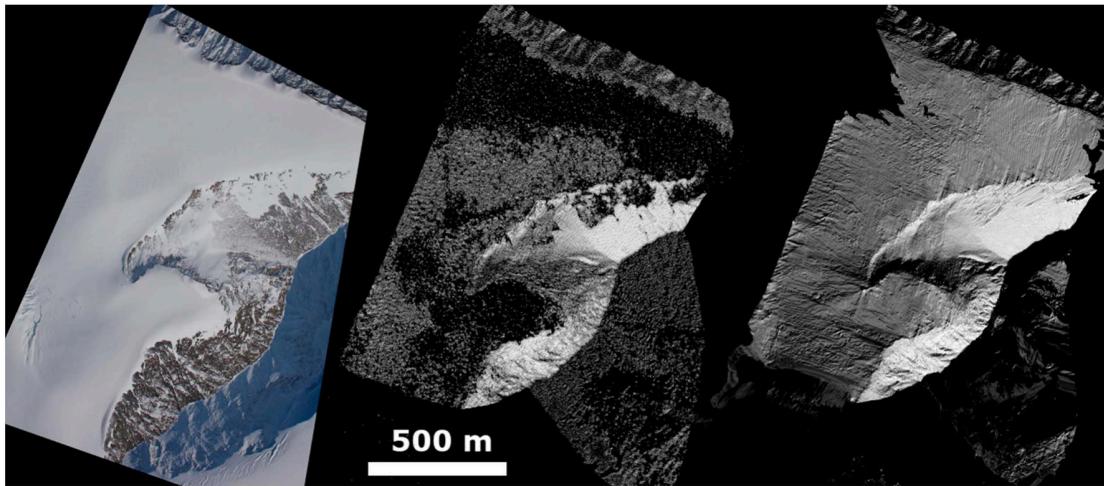
### 2.3.1. Block Matching

Block matching correlation is carried out by moving a small, rectangular template window from the left image over the specified search region of the right image. The best match is determined by applying a cost function that compares the two windows. The location at which the window evaluates to the lowest cost compared to all the other search locations is reported as the disparity value. ASP has a selection of cost functions that the user can select from, but normalized cross-correlation (Menard, 1997) is the most robust to lighting and contrast variations between a pair of images. Other options sacrifice quality for speed.

In order to accelerate the block-matching method, we use a box filter-like accumulator that reduces duplicate operations during correlation (Sun, 2002) and we partition the disparity search space into rectangular subregions with similar values of disparity determined in the previous lower-resolution level of the pyramid



**Figure 6.** A digital terrain model obtained using plain stereo (left) and stereo with map-projected images (right). Their quality will be comparable for relatively flat terrain and the second will be much better for rugged terrain. The right image has some artifacts, but those are limited to areas close to the boundary.



**Figure 7.** On the left is Operation IceBridge image 2011\_11\_14\_01456 of mountains on the Antarctic Peninsula. Traditional block matching of this difficult scene with low Sun angle and large areas of bland snow results in a terrain model that is hillshaded in the center (lots of gaps and data dropouts). The SGM algorithm applied to the same scene results in the hillshade to the right.

(Sun, 2002). The end result of the block-matching process is an integer-valued disparity estimate which is then passed to the subpixel refinement step for improvement.

### 2.3.2. Semiglobal Matching

A new option for integer stereo correlation available in ASP is the popular SGM algorithm introduced in Hirschmüller (2008). Among its other uses this algorithm is used to process High Resolution Stereo Camera images (Hirschmüller et al., 2006). The version of the algorithm implemented by ASP has a few modifications relative to the original implementation. The most significant difference is that ASP's implementation performs a 2-D disparity search, similar to what is done in the Neighbor-Guided Semi Global Matching algorithm (Xiang et al., 2016). Since ASP processes a wide variety of cameras with varying degrees of metadata quality, the standard assumption with SGM that the disparity search can be performed only along a one-dimensional epipolar line does not hold. The other major change is that ASP uses a multiresolution hierarchical search combined with a compressed memory scheme similar to what is used in the Rothermel et al. (2012) SGM algorithm. With these two features, the ASP SGM algorithm can be used for larger and imperfectly rectified images. ASP also supports a mode using the MGM algorithm (Facciolo et al., 2015), which is a modification of SGM. This algorithm reduces the amount of high frequency artifacts in textureless regions at the cost of a longer run time. ASP also offers the option of a hybrid SGM/MGM mode where MGM is used only for the final resolution level which obtains results somewhere between the pure SGM and MGM options.

The greatest advantage of the SGM algorithm over the normal ASP block-correlation algorithm is an improved ability to find disparity matches in areas of repetitive or low texture (Figure 7). SGM can also discern finer resolution features than the standard correlation algorithm since it tends to use much smaller matching kernels. Along with these advantages come several disadvantages. First, SGM is computationally expensive and can require gigabytes of memory. Second, additional blending between processed overlapping tiles is required to avoid producing artifacts at tile boundaries. Third, it can sometimes produce spurious results in textureless regions (Figure 7). With careful parameter selection and usage these disadvantages can be mitigated. Mayer (2018) outlines a workflow that combines SGM, block-matching, and disparity map smoothing in a multistage processing approach to achieve better results.

### 2.4. Subpixel Refinement

Once the disparity map initialization is complete, we have a correlation matching pixel to pixel between the left and right images. If we were to go on to the triangulation step at this point, the ray that we would trace out of the optics would come from the pixel's center. However, there is additional information that we can use to refine the correlation position within that pixel.

Again, there are several subpixel refinement approaches that could be used, and ASP allows the user to select between them. The first mode is parabola-fitting subpixel refinement. This technique fits a 2-D parabola

to points on the correlation cost surface in an 8-connected neighborhood around the cost value that was the best as measured during disparity map initialization. The parabola's minimum can then be computed analytically and taken as the new subpixel disparity value.

This method is easy to implement and extremely fast to compute, but it exhibits a problem known as pixel-locking: the subpixel (floating point) disparities tend toward their integer estimates and can create noticeable *stair steps* on surfaces that should be smooth (Stein et al., 2006; Szeliski & Scharstein, 2004). Furthermore, the parabola subpixel mode is not capable of refining a disparity estimate by more than one pixel, so although it produces smooth disparity maps, these results are not much more accurate than the results that come out of the disparity map initialization in the first place. However, the speed of this method makes it very useful as a *draft* mode for quickly generating a DTM for visualization (i.e., nonscientific) purposes.

For high-quality results, ASP implements a Bayes expectation-maximization (EM) weighted affine-adaptive window correlator (Broxton et al., 2009; Nefian et al., 2009). This advanced method produces extremely high-quality stereo matches that exhibit a high degree of immunity to image noise. The Bayes EM subpixel correlator also features a deformable template window from the left image that can be rotated, scaled, and translated as it zeros in on the correct match in the right image. This adaptive window is essential for computing accurate matches on crater or canyon walls, and on other areas with significant perspective distortion due to foreshortening.

This affine-adaptive behavior is based on the Lucas-Kanade template tracking algorithm, a classic algorithm in the field of computer vision (Baker & Matthews, 2004). ASP has extended this technique; developing a Bayesian model that treats the Lucas-Kanade parameters as random variables in an EM framework. This statistical model also includes a Gaussian mixture component to model image noise that is the basis for the robustness of the algorithm.

ASP also has a simple affine correlator. This is essentially the Bayes EM mode with the noise correction features removed in order to decrease the required run time. In data sets with little noise this mode can yield good results similar to Bayes EM mode in approximately one fifth the time.

However, we do note that like the computations in the disparity map initialization stage, we adopt a multiscale approach for subpixel refinement. At each level of the pyramid, the algorithm is initialized with the disparity determined in the previous lower-resolution level of the pyramid, thereby allowing the subpixel algorithm to shift the results of the disparity initialization stage by many pixels if a better match can be found using the affine, noise-adapted window. Hence, this subpixel algorithm is able to significantly improve upon the results to yield a high-quality, high-resolution result.

## 2.5. Triangulation

Once the correlation step has found accurate matches on each image, we can move on to the next part of the stereo process: triangulation. During this step, correlation results are converted to 3-D coordinates in the world.

Camera models supported by ASP account for all aspects of camera geometry, including both intrinsic (i.e., focal length, optical center, pixel size, and lens distortion) and extrinsic (e.g., camera position and orientation) camera parameters. Points in the real world are imaged onto the focal planes of the two camera stations, and if we know the location of the same feature in both images (via the matching and correlation processes described above), we can use the camera intrinsics and extrinsics to *back project* a ray from each camera station starting at the matched pixels out through the camera optics that converge on a point in 3-D space. That intersection location is the position of the feature in the real world.

In practice, the two rays rarely intersect perfectly because any slight error in the camera position or pointing information will affect the rays' positions as well. Instead, we take the closest point of intersection of the two rays as the location of the point.

Additionally, the actual distance between the rays at this point is an interesting and important error metric that measures how self-consistent our two camera models are for this point.

This error in the triangulation—the distance between two rays—is not the true accuracy of the DTM. It is only another indirect measure of quality. A DTM with high triangulation error is always bad and should probably have its images bundle-adjusted. A DTM with low triangulation error is at least self-consistent but could still be bad. A map of the triangulation error should only be interpreted as a relative measurement. Small areas

with high triangulation error are probably the result of correlation mistakes and large areas of triangulation error are probably the result of camera model inadequacies.

### 3. Results of Stereo

The result of running the ASP **stereo** program is a point cloud file. Since the correlation step creates a disparity map that has information for every pixel (with some exceptions where the correlation algorithm could not find a match), the triangulation step marches through that disparity map and computes a 3-D point for every input pixel. One can think of this point cloud as a fuzzy collection of 3-D points, not all of which represent truly independent measures of a terrain surface, due to the nature of the correlation windowing and even the way that our subpixel refinement process uses information from a window of pixels.

However, data in this point cloud format are rarely useful, and so we provide command line tools that take that point cloud and either rasterize it to create a gridded terrain model (**point2dem**) or create a mesh file (**point2mesh**).

#### 3.1. Errors in the Computed Terrain

Ideally, each feature on the ground would be perfectly identified in the two images, and when rays are traced through those image locations, they would intersect at the precise ground data point the image information arose from. In practice, this is not possible, and there are several sources of error that result in terrain inaccuracies.

The most obvious source of error is that the image resolution is finite, and hence the ground is not represented perfectly in the images. The terrain accuracy is directly proportional to image resolution. Stereo correlation usually involves some sort of matching window, with a larger window increasing the likelihood of successfully finding a pixel match. The drawback is that a larger window acts like a low-pass filter, reducing the effective resolution of the generated terrain. The size of this window also has implications for what the *ideal* ground scale (or post spacing) of the output terrain model should be. Terrain models are generally made with post spacings of three to five times the ground scale distance of the stereo image pair (even though windows are sometimes larger than this). It would be nice if the ideal terrain model ground sample distance could be independently derived, or deterministically yielded from the selection of the correlation window size, but it is not that straightforward. In fact, some studies indicate that the *true resolution* of terrain data may be even coarser than these three to five pixel values that are typically used (Kirk et al., 2017, 2016).

The baseline distance between the cameras in the stereo pair is also very important. A small baseline results in rays from the cameras to the ground that have a small convergence or parallax angle, and hence a high uncertainty about where they intersect. A large baseline has the consequence that the images can have very different perspectives and even occlusions, hence features in them may be hard to match, also resulting in errors. These factors are described by Cook et al. (1996) in their equation which predicts stereo height accuracy.

Stereo matching on its own can result in errors, such as when an image block is inaccurately matched among the images. We usually filter these out using various heuristics, such as ray intersection error or a median filter. The SGM and MGM algorithms, being semiglobal in nature, are more robust to these than the purely local block matching. And, of course, featureless terrains such as over fresh snow, or water, dusty areas, or those in shadows can confuse the matching process. It may not perform well on images of surfaces with complicated reflection, such as translucent or transparent terrains, trees and vegetation in general, or occlusions, as seen often in urban environments.

Inaccurate camera positions and orientations are another source of error. Those can be alleviated, as mentioned earlier, by using bundle adjustment to reduce intrinsic errors among the cameras. To reduce errors to the ground, including in translation, rotation, and possibly even scale, one can use ground control points or register the computed terrain to a known ground truth using ASP's **pc\_align** alignment tool.

For frame imagers, such as consumer digital cameras, a source of errors—also discussed earlier—is imperfectly known camera intrinsics, which result in nonlinear deformations in the final terrain, and which we are able to solve for in certain circumstances. For linescan (pushbroom) cameras, in addition to lens errors, there is also the fact that the line sensor can oscillate somewhat as it moves with the satellite or aircraft, resulting in jitter, and this is not always perfectly measured and compensated for. ASP provides a partial research solution to this problem, but it is still work in progress. Lastly, there can be a situation where the linescan sensor

is made up of multiple individual charge coupled device segments (Digital Globe's World View 1 and 2, the Mars Reconnaissance Orbiter's HiRISE camera, etc.) and result in brightness differences when combining their pixels into a single image. This results in clearly visible fault-like artifacts in computed terrains. ASP provides the **wv\_correct** tool to handle these for World View images. Note that this effect is practically absent in the newer WV-3 satellite, at least for the higher-resolution pan-chromatic band.

A detailed analysis of results produced by ASP from WorldView data can be found in Shean et al. (2016). After aligning ASP-produced DTMs to lidar data and computing the standard deviation of the elevations at each pixel, they found a mean value of 0.19 m. Based on inspection of their results, they theorized that the major sources of elevation difference were spacecraft jitter artifacts and optical distortion near the outer sides of the images. Watters et al. (2015) performed a comparison of ASP-produced DTMs of craters to DTMs produced by the HiRISE data team using the commercial SOCET Set software. They found that for small craters which could be accurately compared the typical average elevation discrepancy was around 0.5 m or less.

One approach to reduce the effect of these errors is to take advantage of multiple images by averaging together terrains computed from multiple image pairs using our mosaicking tool (section 4.3).

## 4. Other Functionality Included with ASP

### 4.1. Bundle Adjustment

ASP provides a bundle adjustment tool named **bundle\_adjust**, built around the Google Ceres least squares solver (Agarwal et al., 2012). The purpose of bundle adjustment is to correct for errors in camera positions and orientations. This is accomplished by selecting matching tie points among images, computing corresponding triangulated points in 3-D space, and minimizing the pixel reprojection error for these points. This makes the cameras more self-consistent and reduces discontinuities at the borders between DTMs (Figure 8) but does not necessarily result in improved positioning of the final 3-D terrain to what is known on the ground. To accomplish the latter, our tool can take as input so-called *ground control points*, which are sets of matching tie points among the images together with known 3-D triangulated position on the ground.

For high-quality and well-calibrated satellite stereo images, such bundle adjustment (coupled with alignment done in postprocessing as detailed in section 4.2) is usually sufficient to result in a high-quality terrain. When it is not, we have found that optimizing not only the camera positions and orientations, but also the camera intrinsics (e.g., focal length, optical center, and lens distortion) can provide improvements to the solutions. For this to be successful it is important that there be good overlap among images (at least three views whenever possible), an external ground truth terrain (such as provided by lidar or laser altimeters), and a reasonably dense set of tie points among the images.

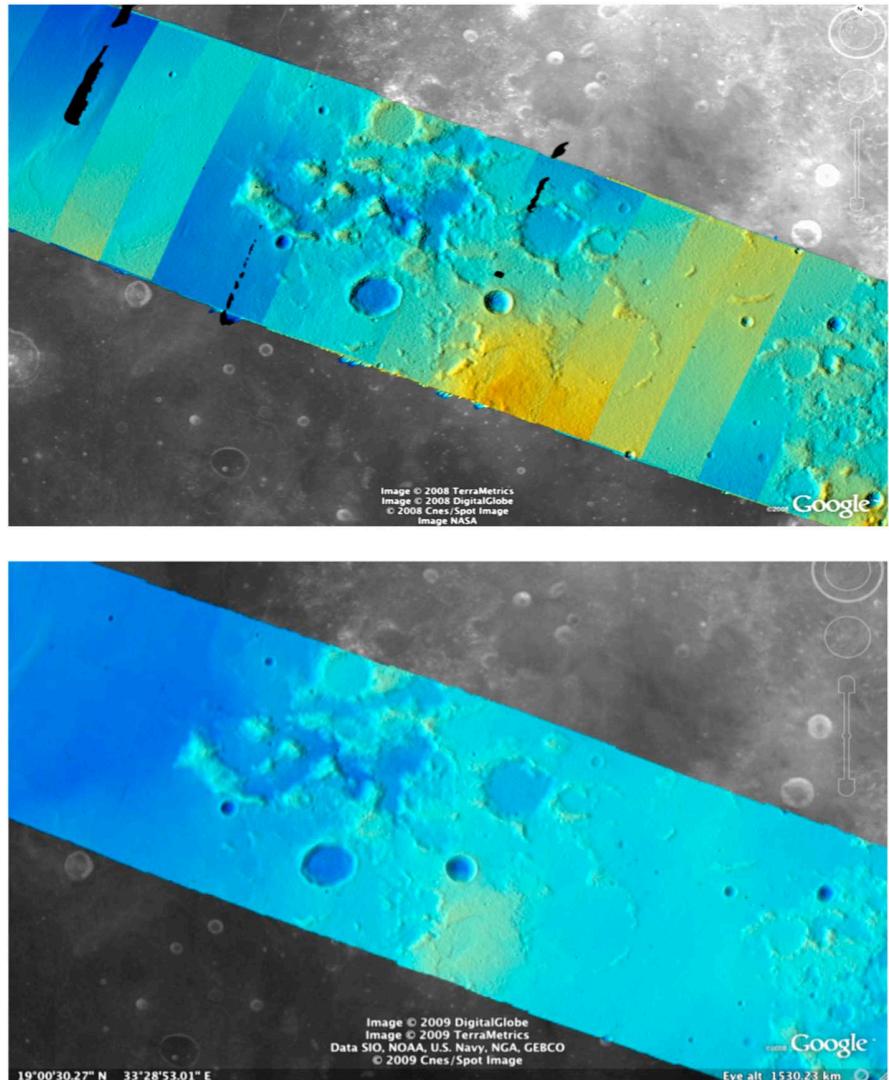
Our bundle adjustment tool can work with any number of images. It is important to note that it is well known that any bundle adjustment with a long sequence of images and without any external constraints, such as ground control points, will result in accumulation of error in the positions, orientations, and scale of the final DTMs. In the absence of ground control points, but in the presence of some kind of external ground truth, such as lidar or a previously known DTM, an effective solution is to apply bundle adjustment on short sequences and align each obtained DTM to the ground truth using the provided alignment tool (section 4.2).

We implemented the ability to keep fixed the positions and orientations of any number of the cameras. It is possible, in principle, for bundle adjustment to work with different sensors, as long as their format is the same. For example, one can use one WorldView-1 image together with a WorldView-2 image, or a HiRISE with a CTX image (both being .cub files).

### 4.2. Alignment of Point Clouds

Often, the 3-D terrain models output by **stereo** (point clouds and DTMs) can be intrinsically quite accurate, yet their actual position on the planet may be off by several meters or several kilometers, depending on the image source. This can result from small errors in the position and orientation of the cameras taking the pictures.

Such errors can be corrected in advance using bundle adjustment, as described above, requiring ground control points that may not be easy to collect. Alternatively, the images and cameras can be used as they are (or with bundle adjustment but without ground control points), and the absolute position of the output point clouds can be corrected in postprocessing. For that, ASP provides a tool named **pc\_align** (Beyer et al., 2014). It aligns a 3-D terrain to a much more accurately positioned (if potentially sparser) data set using an iterative



**Figure 8.** The effects of bundle adjustment illustrated on color-mapped, hillshaded digital terrain model mosaics from Apollo 15 metric camera images. Without bundle adjustment (left) discontinuities between digital terrain models are clearly visible, but bundle adjustment is able to almost completely eliminate them (right).

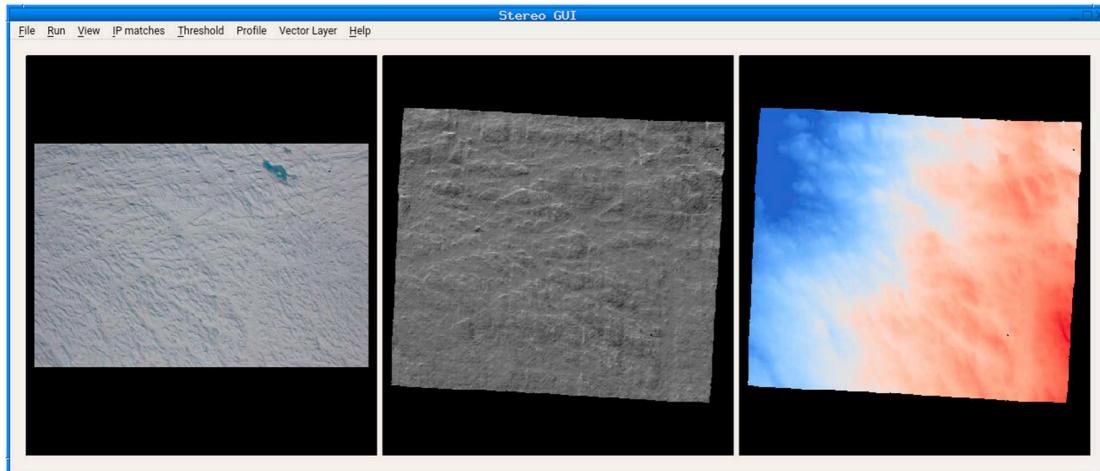
Closest Point algorithm (we use the **libpointmatcher** implementation of Pomerleau et al., 2013). Such data sets can be made up of GPS measurements (in the case of Earth), or from laser altimetry instruments. It takes a generic input (i.e., in addition to ASP-created point clouds it can use DTMs and plain text files), and it can be used for a variety of point cloud alignment tasks, with the input data not necessarily produced by ASP.

#### 4.3. Mosaicking of Terrain Models

There are situations when a desired area of interest cannot be covered just by one stereo pair. Then, multiple stereo pairs are acquired, with each resulting in a point cloud, those clouds are aligned to a known ground truth, and individual terrain models are generated. Those can be combined into a single seamless terrain model using ASP's **dem\_mosaic** tool. This program has various options of how to merge the terrain models and can also compute various statistics, and so forth.

#### 4.4. The ASP GUI

ASP comes with its own graphical user interface (GUI) tool, named **stereo\_gui**. It can visualize all images supported by ASP and Geospatial Data Abstraction Library. It deals with very large images gracefully: for images that are too large to fit in memory, it builds a pyramid of increasingly lower-resolution versions of the images, and loads into memory only the portion that is in view at the moment, at the requested resolution (Figure 9).



**Figure 9.** From left to right, an image (part of a stereo pair), the hillshaded digital terrain model obtained from that stereo pair, and its colorized version, as displayed by `stereo_gui`.

The user can invoke `stereo_gui` with exactly the same options as the ASP `stereo` command, that is, with two images in a stereo pair and corresponding cameras, and so forth, then manually select a subregion in the GUI upon which the user wishes to perform stereo correlation, and then start the process from the GUI. This can be very helpful when the input images are very large (10s of thousands of pixels in each dimension), and only a small region needs to be processed for evaluation purposes.

In addition, `stereo_gui` can display DTMs as hillshaded images, show colorized DTMs (generated with ASP's `colormap` tool or with Geospatial Data Abstraction Library tools), it can show interest points on top of images (interest points are vital in bundle adjustment and for seeding the stereo correlation), display the stereo disparity (after being broken into channels by ASP's `disparitydebug`), it can overlay georeferenced images, show multiple images side-by-side or in a grid pattern, with the option of locking the zoom window (hence if one zooms in one pane, the images in the other panes zoom to the same pixel region or georeferenced area), and so forth.

#### 4.5. Other Tools

In addition to the `stereo` tool for creating a point cloud from input images, ASP has a version named `parallel_stereo` that can scale this functionality to multiple processes, potentially over multiple machines reachable via `ssh` and sharing a file system. ASP also provides a shape-from-shading tool named `sfs` for refining a stereo DTM based on one or more images with various illumination conditions (Alexandrov & Beyer, 2017); the tool `dem_geoid` for converting a DTM from being relative to an ellipsoid to being relative to a geoid, and vice-versa (for Earth and Mars); the program `mapproject` for projecting a camera image onto DTM; `image_calc` for performing per-pixel arithmetic operations on a set of images; tools for hillshading and colorizing DTMs, taking their (absolute) difference; converting a DTM to a laser (LAS) file, and many others.

## 5. Conclusions

The NASA ASP is a practical set of software that allows users to derive terrain from orbital stereo cameras, aerial cameras, and even surface cameras. It has a tremendous amount of flexibility and power. ASP can handle all of the steps of terrain creation, from bundle adjustment to terrain model mosaicking. Beyond the stereo terrain generation abilities at its core, ASP also distributes a variety of tools which can be used to help prepare data for stereo correlation and for working with output terrain and orthoimages. ASP can operate on a single laptop but also has parallel computing capabilities so that it can be run on clusters and super-computers. More importantly, since ASP is built with open source concepts from the ground up, most of the software can be used atomically or broken down into component parts and used to build entirely new workflows to accomplish new analysis.

## Acknowledgments

The authors would like to thank the large number of people that have developed and contributed to ASP over the years, especially Michael Broxton, Matt Hancher, Ara Nefian, and Zachary Moratto. The authors would also like to thank the two anonymous reviewers whose comments improved this manuscript, and Carter Emmart who provided the visualization in Figure 2. All the data used are listed in the references or archived in the Planetary Data System or National Snow and Ice Data Center repositories.

## References

- Agarwal, S., Mierle, K., et al. (2012). Ceres solver. Retrieved from <http://ceres-solver.org>
- Alexandrov, O., & Beyer, R. A. (2017). Multi-view shape-from-shading for planetary images with challenging illumination. In *Lunar and Planetary Science Conference, Lunar and Planetary Science Conference* (p. 3024).
- Anderson, J. A. (2008). ISIS camera model design. In *Lunar and Planetary Science Conference, Lunar and Planetary Inst. Technical Report* (p. 2159).
- Anderson, J. A., Sides, S. C., Soltesz, D. L., Sucharski, T. L., & Becker, K. J. (2004). Modernization of the integrated software for imagers and spectrometers. In S. Mackwell & E. Stansbery (Eds.), *Lunar and Planetary Science XXXV, Abstract2039*. Houston (CD-ROM): Lunar and Planetary Institute.
- Baker, S., & Matthews, I. (2004). Lucas-Kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3), 221–255. <https://doi.org/10.1023/B:VISI.0000011205.11775.fcd>
- Beyer, R. A., Alexandrov, O., & McMichael, S. (2018). NeoGeographyToolkit/StereoPipeline: Ames Stereo Pipeline version 2.6.1. <http://doi.org/10.5281/zenodo.1345235>
- Beyer, R. A., Alexandrov, O., & Moratto, Z. M. (2014). Aligning terrain model and laser altimeter point clouds with the Ames Stereo Pipeline. In *Lunar and Planetary Science Conference 45* (p. 2902).
- Broxton, M. J., & Edwards, L. J. (2008). The Ames Stereo Pipeline: Automated 3D surface reconstruction from orbital imagery. In *Lunar and Planetary Science Conference, Lunar and Planetary Inst. Technical Report* (p. 2419).
- Broxton, M. J., Nefian, A. V., Moratto, Z., Kim, T., Lundy, M., & Segal, A. V. (2009). 3D Lunar Terrain Reconstruction from Apollo Images. In G. Bebis, et al. (Eds.), *Advances in Visual Computing. ISVC 2009, Lecture Notes in Computer Science* (Vol. 5875). Berlin, Heidelberg: Springer Berlin Heidelberg.
- Cook, A. C., Oberst, J., Roatsch, T., Jaumann, R., & Acton, C. (1996). Clementine imagery: Selenographic coverage for cartographic and scientific use. *Planetary and Space Science*, 44, 1135–1148.
- Edwards, L., & Broxton, M. (2006). Automated 3D surface reconstruction from orbital imagery. In M. Broxton (Ed.), *Space 2006, American Institute of Aeronautics and Astronautics*. San Jose, CA: NASA Ames Research Center Space 2006. <https://doi.org/10.2514/6.2006-7435>
- Edwards, L., Sims, M., Kunz, C., Lees, D., & Bowman, J. (2005). Photo-realistic terrain modeling and visualization for Mars exploration rover science operations. In *2005 IEEE International Conference on Systems, Man and Cybernetics* (pp. 1389–1395). <https://doi.org/10.1109/ICSMC.2005.1571341>
- Faccioli, G., de Franchis, C., & Meinhardt, E. (2015). MGM: A significantly more global matching for stereovision. In X. Xie, M. W. Jones, & G. K. L. Tam (Eds.), *Proceedings of the British Machine Vision Conference (BMVC)* (pp. 90.1–90.12). Swansea, UK: BMVA Press. <https://doi.org/10.5244/C.29.90>
- Hartley, R., & Zisserman, A. (2004). *Multiple view geometry in computer vision* (2nd ed.). Cambridge: Cambridge University Press.
- Hirschmüller, H. (2008). Stereo processing by semiglobal matching and mutual information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(2), 328–341. <https://doi.org/10.1109/TPAMI.2007.1166>
- Hirschmüller, H., Mayer, H., & Neukum, G. (2006). The HRSC Co-Investigator Team Stereo processing of HRSC Mars Express images by semi-global matching. In *Symposium of ISPRS Commission IV in Goa, India* (pp. 305–310).
- Kirk, R. L., Howington-Kraus, E., Edmundson, K., Redding, B., Galuszka, D., Hare, T., et al. (2017). Community tools for cartographic and photogrammetric processing of Mars Express HRSC Images, ISPRS - International Archives of the Photogrammetry. *Remote Sensing and Spatial Information Sciences*, XLII-3/W1, 69–76. <https://doi.org/10.5194/isprs-archives-XLII-3-W1-69-2017>
- Kirk, R. L., Howington-Kraus, E., Hare, T. M., & Jordá, L. (2016). The effect of illumination on stereo DTM quality: Simulations in support of Europa exploration, ISPRS annals of photogrammetry. *Remote Sensing and Spatial Information Sciences*, III-4, 103–110. <https://doi.org/10.5194/isprs-annals-III-4-103-2016>
- Mayer, D. P. (2018). An improved workflow for producing digital terrain models of Mars from CTX stereo data using the NASA Ames Stereo Pipeline. In *Lunar and Planetary Science Conference* (p. 1604).
- Menard, C. (1997). Robust stereo and adaptive matching in correlation scale-space (PhD thesis). Institute of Automation, Vienna Institute of Technology (PRIP-TR-45).
- Moratto, Z. M., Broxton, M. J., Beyer, R. A., Lundy, M., & Husmann, K. (2010). Ames Stereo Pipeline, NASA's open source automated stereogrammetry software. In S. Mackwell & E. Stansbery (Eds.), *Lunar and Planetary Science Conference 41, Abstract 2364*. Houston: Lunar and Planetary Institute.
- Nefian, A. V., Husmann, K., Broxton, M., To, V., Lundy, M., & Hancher, M. D. (2009). A Bayesian formulation for sub-pixel refinement in stereo orbital imagery. In *2009 16th IEEE International Conference on Image Processing (ICIP)* (pp. 2361–2364). <https://doi.org/10.1109/ICIP.2009.5413749>
- Nguyen, L. A., Bualat, M., Edwards, L. J., Flueckiger, L., Neveu, C., Schwehr, K., et al. (2001). Virtual reality interfaces for visualization and control of remote vehicles. *Autonomous Robots*, 11(1), 59–68. <https://doi.org/10.1023/A:1011208212722>
- Pomerleau, F., Colas, F., Siegwart, R., & Magnenat, S. (2013). Comparing ICP variants on real-world data sets. *Autonomous Robots*, 34(3), 133–148. <https://doi.org/10.1007/s10514-013-9327-2>
- Rothermel, M., Wenzel, K., Fritsch, D., & Haala, N. (2012). *SURE: Photogrammetric surface reconstruction from imagery* (Vol. 8). Berlin: Proceedings LC3D Workshop.
- Shean, D. E., Alexandrov, O., Moratto, Z. M., Smith, B. E., Joughin, I. R., Porter, C., et al. (2016). An automated, open-source pipeline for mass production of digital elevation models (DEMs) from very-high-resolution commercial stereo satellite imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 116, 101–117. <https://doi.org/10.1016/j.isprsjprs.2016.03.012>
- Stein, A. N., Huertas, A., & Matthies, L. (2006). Attenuating stereo pixel-locking via affine window adaptation. In *Proceedings 2006 IEEE International Conference on Robotics and Automation, 2006. ICRA 2006* (pp. 914–921). <https://doi.org/10.1109/ROBOT.2006.1641826>
- Sun, C. (2002). Fast stereo matching using rectangular subregioning and 3D maximum-surface techniques. *International Journal of Computer Vision*, 47(1), 99–117. <https://doi.org/10.1023/A:1014585622703>
- Sweeney, C. (2013). *Theia multiview geometry library*. Tutorial & reference. Retrieved from <http://theia-sfm.org>
- Szeliski, R., & Scharstein, D. (2004). Sampling the disparity space image. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(3), 419–425. <https://doi.org/10.1109/TPAMI.2004.1262341>
- Triggs, B., McLauchlan, P. F., Hartley, R. I., & Fitzgibbon, A. W. (2000). Bundle adjustment—A modern synthesis. In B. Triggs, A. Zisserman, & R. Szeliski (Eds.), *Vision algorithms: Theory and practice* (pp. 298–372). Berlin, Heidelberg: Springer.
- U. S. Geological Survey (2018). Integrated software for imagers and spectrometers (ISIS), U.S. Geological Survey, Flagstaff, AZ.

- Watters, W. A., Geiger, L. M., Fendrock, M., & Gibson, R. (2015). Morphometry of small recent impact craters on Mars: Size and terrain dependence, short-term modification. *Journal of Geophysical Research: Planets*, 120, 226–254. <https://doi.org/10.1002/2014JE004630>
- Xiang, J., Li, Z., Blaauw, D., Kim, H. S., & Chakrabarti, C. (2016). Low complexity optical flow using neighbor-guided semi-global matching. In *2016 IEEE International Conference on Image Processing (ICIP), IEEE* (pp. 4483–4487). <https://doi.org/10.1109/ICIP.2016.7533208>