**JavaScript is not enabled. Please enable JavaScript for this site or** [**view Printer Friendly version of this document**](../../PrinterFriendly/jigsaw/jigsaw.html)**.**

[Home](../../../../index.html)

## User Documentation

[Getting Started](../../../../UserStart/index.html)

[Learn More](../../../../UserLearn/index.html)

[Explore in Detail](../../../../UserExplore/index.html)

[Get Inspired](../../../../UserInspire/index.html)

## Contributor Documentation

[Getting Started](../../../../DevStart/index.html)

[Learn More](../../../../DevLearn/index.html)

[Explore in Detail](../../../../DevExplore/index.html)

[Get Inspired](../../../../DevInspire/index.html)

## Quick Links

[Software Manual](../../../../Application/index.html)

[AstroDiscuss](https://astrodiscuss.usgs.gov/)

[GitHub](https://github.com/USGS-Astrogeology/ISIS3)

[API Reference](../../../../Object/Developer/index.html)

## Documentation Versions

Public Release6.0.03.9.03.5.0

## ISIS 2

[Documentation](../../../../documents/Isis2UserDocs/index.html)

[Tutorials](../../../../documents/Isis2Tutorials/index.html)

[Technical Documents](../../../../documents/Isis2TechnicalDocs/index.html)



ISIS Application Documentation

|  |  |
| --- | --- |
| jigsaw | [Printer Friendly View](../../PrinterFriendly/jigsaw/jigsaw.html) | [TOC](../../../index.html) | [Home](../../../../index.html) |

*Improves image alignment*

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| Overview  | Parameters  | Example 1 | Example 2 | Example 3 | Example 4 | Example 5 | Example 6 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Description OverviewThe **jigsaw** application performs a bundle adjustment on a group of overlapping Isis, level 1, [cube](../../../../documents/Glossary/Glossary.html#Cube)s from framing and/or line-scan cameras. The adjustment simultaneously defines the selected image geometry information (camera pointing, spacecraft position) and [control point](../../../../documents/Glossary/Glossary.html#ControlPoint) coordinates (x,y,z or lat,lon,radius) to reduce boundary mismatches in mosaics of the images. This functionality is demonstrated below in a zoomed-in area of a mosaic of a pair of overlapping Messenger images. In the before jigsaw mosaic on the left (uncontrolled), the features on the edges of the images do not match. In the after jigsaw mosaic on the right (controlled), the crater edges meet correctly and the seam between the two images is no longer visible. Before jigsaw (uncontrolled)After jigsaw (controlled)The **jigsaw** application assumes **spiceinit** or **csminit** has been run on the input [cube](../../../../documents/Glossary/Glossary.html#Cube)s so that camera information is included in the Isis [cube](../../../../documents/Glossary/Glossary.html#Cube) labels. In order to run the program, the user must provide a list of input [cube](../../../../documents/Glossary/Glossary.html#Cube)s, an input [control net](../../../../documents/Glossary/Glossary.html#ControlNet), the name of an output [control net](../../../../documents/Glossary/Glossary.html#ControlNet), and the solve settings. **jigsaw** outputs a new [control net](../../../../documents/Glossary/Glossary.html#ControlNet) that includes the initial state of the points in the network and their final state after the adjustment. The initial states of the points are tagged as [a priori](../../../../documents/Glossary/Glossary.html#APriori) in the [control net](../../../../documents/Glossary/Glossary.html#ControlNet), and their final states are tagged as [adjusted](../../../../documents/Glossary/Glossary.html#Adjusted). The measured [sample](../../../../documents/Glossary/Glossary.html#Sample)/[line](../../../../documents/Glossary/Glossary.html#Line) positions associated with the [control point](../../../../documents/Glossary/Glossary.html#ControlPoint)s in the net are not changed. camera information in the [cube](../../../../documents/Glossary/Glossary.html#Cube) labels is updated at the end of the adjustment only if the bundle converges and the **UPDATE** parameter is selected. Optional output files can be selected to provide more information for analyzing the results. **BUNDLEOUT\_TXT** provides an overall summary of the bundle adjustment. It lists the user input parameters selected and tables of statistics for both the images and the points. The image statistics can also be written to a separate **CSV** file and likewise for the point statistics with the **OUTPUT\_CSV** option selected. **RESIDUALS\_CSV** provides a table of the measured image coordinates and the final [sample](../../../../documents/Glossary/Glossary.html#Sample), [line](../../../../documents/Glossary/Glossary.html#Line), and overall residuals in both millimeters and pixels. Observation EquationsThe **jigsaw** application attempts to minimize the reprojective error within the control network. For each control measure and its control point in the control network, the reprojective error is the difference between the control measures image coordinate and the image coordinate that the control point projects to through the camera model. We also refer to the reprojective error as the measure residual. So, for each control measure we can write an observation equation:  reprojective\_error = control\_measure - camera(control\_point) The camera models use additional parameters such as the instrument position and pointing to compute what image coordinate a control point projects to. So, the camera function in our observation equation actually takes a variety of parameters:  reprojective\_error = control\_measure - camera(control\_point, camera\_parameters) The primary decision when using the jigsaw application is which parameters to include as variables in the observation equations. Later sections in this documentation will cover choosing parameters. By combining all of the observation equations for all of the control measures in the control network we can create a system of equations. Then, we can find the camera parameters and control point coordinates that minimize the sum of the squared errors using that system. Below is an example system of obseration equations for a control network with three images (A, B, and C) and three control points (1, 2, and 3). Control point 1 has control measures on images A and B, Control point 2 has control measures on images B and C, and control point 3 has control measures on images A, B, and C. control\_point\_1 is the coordinate of control point 1, camera\_parameters\_A are the camera parameters for image A's camera model, camera\_A is the projection function for image A's camera model, and control\_measure\_1\_A is the image coordinate of control point 1's control measure on image A. Simple control network with 3 images and 3 points | control\_measure\_1\_A - camera\_A(control\_point\_1, camera\_parameters\_A) | | control\_measure\_1\_B - camera\_B(control\_point\_1, camera\_parameters\_B) | | control\_measure\_2\_B - camera\_B(control\_point\_2, camera\_parameters\_B) | reprojective\_error = | control\_measure\_2\_C - camera\_C(control\_point\_3, camera\_parameters\_C) | | control\_measure\_3\_A - camera\_A(control\_point\_3, camera\_parameters\_A) | | control\_measure\_3\_B - camera\_B(control\_point\_3, camera\_parameters\_B) | | control\_measure\_3\_C - camera\_C(control\_point\_3, camera\_parameters\_C) | If the camera projection functions were linear, we could easily minimize this system. Unfortunately, the camera projection functions are quite non-linear. So, instead we have to perform a linearization and iteratively compute updates that will gradually approach a minimum. The actual structure of the observation equations is complex and vary significantly based on the dataset and which parameters are being solved for. If the parameters are sufficiently far away from a minimum, the updates can even diverge. So, it is important to check your input data set and solve options. Tools like the **cnetcheck** application and extra options like outlier rejection can help with this. As with all iterative methods convergence criteria is important. The default for the **jigsaw** application is to converge based on the sigma0 value. This is a complex value that includes many technical details outside the scope of this documentation. In a brief, it is a measurement of the remaining error in a typical observation. The sigma0 value is also a good first estimate of the quality of the solution. If the only errors in the control network are purely random noise, then the sigma0 should be 1. So, if the sigma0 value for your solution is significantly larger than 1, then it is an indication that there is an unusual source of error in your control network such as poorly registered control measures or poor a-priori camera parameters. Solve OptionsChoosing solve options is difficult and often requires experimentation. As a general rule of thumb, the more variation in your dataset, the more you can solve for. Here are some suggestions about what to start with. Control NetworkControl networks can contain a wide variety of information, but the primary concern for choosing solve options is the presence or lack of ground control. Most control networks are generated by finding matching points between groups of images. This defines a relative relationship between the images, but it does not relate those images to the actual surface. So, additional information is needed to absolutely control the images. In terrestrial work, known points on the ground are identified in the imagery via special target markers, but this is usually not possible with planetary data. So, instead, we relate to an additional dataset such as a previous basemap or surface model. This additional relationship is called ground control. Sometimes, there is not sufficient previous data to do this or it will be done as a secondary step after running jigsaw. In these cases, we say that ground control is not present. If your control network does not have ground control, then we recommended that you hold the instrument position fixed and only solve for the instrument pointing. This is because without ground control, the instrument position and pointing can become correlated and there will be multiple minimums for the observation equations. We recommend holing the instrument position fixed becaue a-priori data for it generally is more accurate than the instrument pointing. Sensor TypesFraming sensors are the simplest type or sensor to work with. The entire image is exposed at the same time, so there is only a single position and pointing value for each image. With these datasets you will want to use the either NONE or POSITIONS for the **SPSOLVE** parameter and ANLGES for the **CAMSOLVE** parameter. Pushbroom sensors require additional consideration because the image is exposed over a period of time and the position and pointing are time dependent. The **jigsaw** application handles this by modeling the position and pointing as polynomial functions of time. First polynomial functions are fit to the existing position and pointing values on each image. Then, each iteration updates the polynomial coefficients. For example, setting **CAMSOLVE** to velocities will use first degree, linear, polynomials and setting **CAMSOLVE** to accelerations will use second degree, quadratic, polynomials for the pointing angles. For most pushbroom observations, we recommend using second degree polynomials. There are also a variety of parameters that allow for more precise control over the polynomails used and the polynomial fitting process. The **OVERHERMITE** parameter changes the initial polynomial setup so that it does not replace the original position values. Instead, the polynomial starts out as a zero polynomial and is added to the original position values. The **OVEREXISTING** parameter does the same thing for pointing values. These two options allow you to preserve complex variation like jitter. We recommend enabling these options for jittery images and very long exposure pushbroom images. The **SPSOLVE** and **CAMSOLVE** parameters only allow up to a second degree polynomial. To use a higher degree polynomial for pointing values, set these to ALL and then set the **CKDEGREE** parameter to the degree of the polynomial you want to fit over the original pointing values and set the **CKSOLVEDEGREE** parameter to the degree of the polynomial you want to solve for. The **SPKDEGREE** and **SPKSOLVEDEGREE** parameters do the same thing for position values. For example, if you want to fit a 3rd degree polynomial for the position values and then solve for a 5th degree polynomial set **SPKDEGREE=3** and **SPKSOLVEDEGREE=5**. These parameters should only be used when you know the a-priori position and/or pointing values are well modeled by high degree polynomails. Using very high degree polynomials can result in extreme over-fitting where the solution looks good in areas near control measures and control points but is poor in areas far from control measures and control points. The end behavior, the solution at the start and end of images, becomes particularly unstable as the degree of the polynomial used goes up. Running the **jigsaw** application with synthetic aperture radar (SAR) images is different because SAR sensors do not have a pointing component. Thus, you can only solve for the sensor position when working with SAR imagery. SAR imagery also tends to have very long duration images which can necessitate using the **OVERHERMITE**, **SPKDEGREE**, and **SPKSOLVEDEGREE** parameters. Parameter SigmasThe initial, or a-priori, camera parameters and control points all contain error, but it is not equally distributed. The **jigsaw** application allows users to input a-priori sigmas for different parameters that describe how the initial error is distributed. The parameter sigmas are also sometimes called uncertainties. Parameters with higher a-priori sigma values are allowed to vary more freely than those with lower a-priori sigma values. The a-priori sigma values are not hard contraints that limit how much parameters can change, but rather relative accuracy estimates that adjust how much parameters change relative to each other. The **POINT\_LATITUDE\_SIGMA**, **POINT\_LONGITUDE\_SIGMA**, and **POINT\_RADIUS\_SIGMA** parameters set the a-priori sigmas for control points *that do not already have a ground source*. If a control point already has a ground source, such as a previous basemap or DEM, then it already has internal sigma values associated with it or is held comletely fixed. The exact behavior depends on how the ground source was set. All of these sigmas are in meters on the ground, so the latitude and longitude sigmas are uncertainty in the latitude and longitude directions. Generally, you will choose values for these parameters based on the resolution of the images and a visual comparison with other datasets. The radius sigma is also usually higher than the latitude and longitude sigmas because there is not a very large separation angle between different observations of the ground point. The **CAMERA\_ANGLES\_SIGMA**, **CAMERA\_ANGULAR\_VELOCITY\_SIGMA**, and **CAMERA\_ANGULAR\_ACCELERATION\_SIGMA** parameters set the a-priori sigmas for the instrument pointing. Similarly, the **SPACECRAFT\_POSITION\_SIGMA**, **SPACECRAFT\_VELOCITY\_SIGMA**, and **SPACECRAFT\_ACCELERATION\_SIGMA** parameters set the a-priori sigmas for the instrument position. The velocity and acceleration sigmas are only used if you are solving for them. Finding initial values for these sigmas is challenging as they are not a part of the SPICE kernels used to generate the initial position and pointin values. Some teams will publish uncertainties with their SPK and CK solutions which can be used here, but it is not a part of the SPK or CK specification. Also keep in mind that the velocity and acceleration sigmas are in meters per second or meters per second squared (degrees per second and degrees per second squared for angles). So, the velocity sigma will generally be smaller than the position sigma and similarly, the acceleration sigma will be smaller than the velocity sigma. Coming up with a-priori sigma values can be quite challenging, but the **jigsaw** application has the ability to produce updated sigma values for your solution. By enabling the **ERRORPROPAGATION** parameter, you will get a-posteriori (updated) sigmas for each control point and camera parameter in your solution. This is helpful for analyzing and communicating the quality of your solution. Computing the updated sigmas, imposes a significant computation and memory overhead, so it is recommended that you wait until you are happy with the rest of your settings before enabling the **ERRORPROPAGATION** parameter. Community Sensor ModelThe options that control which parameters to solve for only work with ISIS camera models. When working with community sensor model (CSM) camera models, you will need to manually specify the exact camera parameters that you want to solve for. This is because the CSM API only exposes a generic camera parameter interface that may or may not be related to the physical state of the sensor when an image was exposed. See the **CSMSOLVELIST** parameter documentation for how to do this. Each CSM plugin library defines its own set of adjustable parameters. See the documentation for the CSM plugin library you are using for what different adjustable parameters are and when to solve for them. The sigma values for parameters are also baked into the CSM plugin library and image support data (ISD). So, you do not need to specify camera parameter sigmas when working with CSM camera models; you still need to specify control point coordinate sigmas. For convenience, the **csminit** application adds information about all of the adjustable parameters to the image label. The CSM API natively uses a rectangular coordinate system. So, we also recommend that you have the **jigsaw** application perform calculations in a rectangular coordinate system when working with CSM camera models. To do this, set the **CONTROL\_POINT\_COORDINATE\_TYPE\_BUNDLE** parameter to RECTANGULAR. You can also set the **CONTROL\_POINT\_COORDINATE\_TYPE\_REPORTS** parameter to RECTANGULAR to see the results in a rectangular coordiante system, but this is more a matter of personal preference. RobustnessMuch of the mathematical underpinning of the **jigsaw** application expects that the error in the input data is purely random noise. If there are issues with control measures not being poorly registered or blatently wrong, it can produce errors in the output solution. Generally, users will need to manually identify and correct this errors prior to running the **jigsaw** application, but there are a handful of parameters than can be used to make your solutions more robust to them. The **OUTLIER\_REJECTION** parameter adds a simple calculation to each iteration that looks for potential outliers. The median average deviation (MAD) of the measure residuals is computed and then any control measure whose residual is larger than a multiple of the MAD of the measure residuals is flagged as rejected. Control measures that are flagged as rejected are removed from future iterations. The multiplier for the rejection threshold is set by the **REJECTION\_MULTIPLIER** parameter. The residuals for rejected measures are re-computed after each iteration and if they fall back below the threshold they are un-flagged and included again. Enabling this parameter can help eliminate egregious errors in your control measures, but it is not very sensitive. It is also not selective about what is rejected. So, it is possible to introduce islands into your control network when the **OUTLIER\_REJECTION** parameter is enabled. For this reason, always check your output control network for islands with the **cnetcheck** application when using the **OUTLIER\_REJECTION** parameter. Choosing an appropriate value for the **REJECTION\_MULTIPLIER** is about deciding how aggresively you want to eliminate control measures. The default value will only remove extreme errors, but lower values will remove many control measures that are not atually errors. It is recommended that you start with the default value and then if you want to remove more potential outliers reduce it gradually. If you are seeing too many rejections with the default value, then your input data is extremely poor and you should investigate other methods of improving it prior to running the **jigsaw** application. You can use the **qnet** application to examine and improve your control network. You can use the **qtie** application to manually adjust your a-priori camera parameters. The **jigsaw** application also has the ability to set sigmas for individual control measures. Similar to the sigmas for control points and parameters, these are a measurement of the relative quality of each control measure. Control measures with a higher sigma have less impact on the solution than control measures with a lower sigma. Most control networks have a very large number of control measures and it is not realistic to manually set the sigma for each one. So, the **jigsaw** application can instead use maximum likelihood estimation (MLE) to estimate the sigma for each control measure each iteration. There are a variety of models to choose from, but all of them start each control measure's sigmas at 1. Then, if the control measure's reprojective error falls beyond a certain quantile the sigma will be increased. This reduces the impact of poor control measures in a more targeted fashion than simple outlier rejection. In order to produce good results, though, a high density of control measures is required to approximate the true reprojective error probability distributions. Some of the modules used with MLE can set a measure's sigma to infinty, effectively removing it from the solution. Similar to when using the **OUTLIER\_REJECTION** parameter, this can create islands in your control network. If you are working with a large amount of data and want a robustness option that is more targeted and customizable than outlier rejection, then using the MLE parameters is a good option. See the **MODEL1**, **MODEL2**, **MODEL3**, **MAX\_MODEL1\_C\_QUANTILE**, **MAX\_MODEL2\_C\_QUANTILE**, and **MAX\_MODEL3\_C\_QUANTILE** parameter documentation for more information. Observation ModeSometimes multiple images are exposed at the same time. They could be exposed by different sensors on the same spacecraft or they could be read out from different parts of the same sensor. When this happens, many of the camera parameters for the two images are the same. By default, the **jigsaw** application has a different set of camera parameters for each image. To allow different images that are captured simultaneously to share camera parameters enable the **OBSERVATIONS** parameter. This will combine all of the images with the same observation number into a single set of parameters. By default the observation number for an image is the same as the serial number. For some instruments, there is an extra keyword in their serial number translation file called ObservationKeys. This specifies the number of keys from the serial number to use in the observation number. For example JunoCam serial numbers consist of SpacecraftName/InstrumentId/StartTime/FrameNumber/FilterName and it has ObservationKeys = 4. So, the observaiton number for a JunoCam image is the first four keys of its serial number, SpacecraftName/InstrumentId/StartTime/FrameNumber. Thus, if two JunoCam images have the same StartTime and FrameNumber but different FilterNames and the **OBSERVATIONS** parameter is enabled, then they will share camera parameters. To check if observation mode is setup for use with your data set, check the serial number translation file located in $ISISROOT/appdata/translations. ReferencesFor information on what the original **jigsaw** code was based on checkout [Rand Notebook](https://www.rand.org/pubs/notes/N3330.html) A more technical look at the **jigsaw** application can be found in *Jigsaw: The ISIS3 bundle adjustment for extraterrestrial photogrammetry* by K. L. Edmundson et al. A rigorous development of how the **jigsaw** application works can be found in these [lecture notes](https://earthsciences.osu.edu/sites/default/files/2019-12/OSU_Adjustment_Notes_Part_1.pdf) from Ohio Sate University. Known Issues**Running jigsaw with a control net containing *JigsawRejected* flags may result in bundle failure** When running **jigsaw** with *Outlier Rejection* turned on, control points and/or control measures may be flagged as *JigsawRejected* in the output control net file. If this output net file is then used in a subsequent **jigsaw** run, these points and measures will be erroneously ignored, potentially causing the bundle adjustment to fail. **--Workarounds** 1. Run jigsaw with *Outlier Rejection* **off**.
2. Do not use the output control net file in subsequent jigsaw runs.
3. Convert the output control net file from binary to PVL and back using ***cnetbin2pvl*** and ***cnetpvl2bin***. This will clear the *JigsawRejected* flags.

**Solving for the target body radii (triaxial or mean) is NOT possible and likely increases error in the solve.** With the current jigsaw implementation, it is NOT possible to individually solve for either the mean or triaxial radii as seperate calculations in the bundle adjustment. More specifically, the target body radii has no effect when solving for individual points and thus cannot be solved for in the bundle. A local radii solve is already part of the sequence of equations that **jigsaw** uses to compute various partial derivatives to populate the solve matrix. A seperate mean or triaxial radii solve can be applied to the target body and the partials from this separate application are added to the solve matrix. This option adds additional computation time to **jigsaw** and creates additional uncertainty/error in the bundle adjust. We advise the mean and triaxial radii solve be avoided. If you are trying to generate a spheroid from a control network there are other programs that can do this for you. An easier but more naive method is ingesting the *OUTPUT\_CSV* from your network, gather local radii information from those points, then generate a sphereiod from those local radii. Categories * [Control Networks](../../../index.html#Control_Networks)

Related Objects and Documents Applications* [deltack](../deltack/deltack.html)
* [qnet](../qnet/qnet.html)

History

|  |  |  |
| --- | --- | --- |
| Jeff Anderson | 2007-04-27 | Original version  |
| Steven Lambright | 2007-07-23 | Changed category to Control Networks and corrected XML bugs  |
| Debbie A. Cook | 2007-10-05 | Revised iteration report to list the errors and sigmas from the same iteration. Previous version reported errors from previous iteration and sigmas from current iteration.  |
| Christopher Austin | 2008-07-03 | Cleaned the Bundle Adjust memory leak and fixed the app tests.  |
| Tracie Sucharski | 2009-04-08 | Added date to the Jigged comment in the spice tables.  |
| Tracie Sucharski | 2009-04-22 | If updating pointing, delete the CameraStatistics table from labels.  |
| Mackenzie Boyd | 2009-07-23 | Modified program to write history to input cubes.  |
| Debbie A. Cook | 2010-08-12 | Commented out Heldlist until mechanism in place to enter individual image parameter constraints.  |
| Debbie A. Cook | 2010-08-12 | Merged Ken Edmundson version with system and binary control net.  |
| Debbie A. Cook | 2011-06-14 | Modified code to prevent updates to cube files in held list.  |
| Debbie A. Cook | 2011-09-28 | Removed SC\_SIGMAS from user parameter list because it is not fully implemented; changed method name SPARSE to OLDSPARSE and CHOLMOD to SPARSE; and improved the documentation for the Isis3.3.0 release.  |
| Debbie A. Cook, Ken Edmundson, and Orrin Thomas | 2011-10-03 | Added images showing before and after to demonstrate the program. Added Krause's collinearity diagram and a brief explanation on the output options. Also added a lien for example(s) to be added later.  |
| Debbie A. Cook | 2011-10-06 | Corrected previous history entry and added references to glossary. Also changed application names to bold type.  |
| Debbie A. Cook and Ken Edmundson | 2011-10-07 | Removed glossary references from briefs. Also changed the definition of angles to state right ascension and declination to be consistent with the output.  |
| Ken Edmundson | 2011-10-14 | Added internal default and minimum inclusive tags to global apriori uncertainties.  |
| Ken Edmundson | 2011-10-18 | Added Known Issues section and JigsawRejected flag issue.  |
| Debbie A. Cook | 2011-11-04 | Added minimums to parameters, corrected SOLVEDEGREE description, and added to the camsolve option descriptions in response to Mantis issue #514.  |
| Ken Edmundson | 2011-12-20 | Added REJECTION\_MULTIPLIER to interface, part of Mantis issue #637.  |
| Ken Edmundson | 2012-01-19 | Added SPKDEGREE and SPKSOLVEDEGREE; changed name of SOLVEDEGREE to CKSOLVEDEGREE.  |
| Ken Edmundson | 2014-02-13 | Added separate group for Error Propagation with option to write inverse matrix to binary file. For extremely large networks where memory/time for error propagation is limited.  |
| Ken Edmundson | 2014-07-09 | Added USEPVL and SC\_PARAMETERS parameters.  |
| Jeannie Backer | 2014-07-14 | Modified appTests to use SPARSE method only. Commented out bundleout\_images.csv references. Created observationSolveSettings() method to create an observation settings object from the user entered values.  |
| Ken Edmundson | 2015-09-05 | Added preliminary target body functionality. Added SOLVETARGETBODY and TB\_PARAMETERS.  |
| Jesse Mapel | 2016-08-16 | Added a connection to allow jigsaw to surface exceptions from BundleAdjust. Fixes #2302  |
| Jeannie Backer | 2016-08-18 | Removed the user parameter called METHOD (i.e. the method used for solving the bundle matrix). This solve method is no longer user-selected. The program will now use what was called the SPARSE option for the METHOD parameter (i.e. solve with CholMod sparse decomposition). This method should give the same results as the other options and should run faster. So the other options were no longer needed. References #4162.  |
| Ian Humphrey | 2016-08-22 | Reviewed documentation and updated small spelling and grammar errors. References #4226.  |
| Adam Paquette | 2016-08-31 | Updated how jigsaw handles its prefix parameter along with a small documentation change. Fixes #4309.  |
| Jesse Mapel | 2016-09-02 | Updated how input parameters are output when using multiple sensor solve settings. Fixes #4316.  |
| Ian Humphrey | 2016-09-22 | Output from jigsaw will again provide "Validating network" and "Validation complete" messages to inform user that their control network has been validated. Fixes #4313.  |
| Ian Humphrey | 2016-10-05 | When running jigsaw with error propagation turned on, the correlation matrix file, inverseMatrix.dat, is no longer generated. Fixes #4315.  |
| Tyler Wilson | 2016-10-06 | Added the IMAGES\_CSV parameter to the "Output Options" group so that the user can now request the bundleout\_images.csv file in addition to the other output files such as bundleout.txt. Fixes #4314.  |
| Ian Humphrey | 2016-10-13 | Implemented HELDLIST functionality for non-overlapping held images. Any control points that intersect the held images are fixed, and a priori surface points for these control points are set to the held images' measures' surface points. Disabled USEPVL/SC\_PARAMETERS. Fixes #4293.  |
| Ian Humphrey | 2016-10-25 | Added the "Generating report files" and Rejected\_Measures keyword back to jigsaw's standard output. Fixes #4461. Fixed spacing in standard output. Fixes #4462, #4463."  |
| Ian Humphrey | 2016-10-26 | The bundleout.txt output file will record default values for unsolved parameters. The default position will be the instrument position's center coordinate, and the default pointing will be the pointing's (rotation's) center angles. The bundleout\_images.csv file will also have these defaults provided. Fixes #4464.  |
| Makayla Shepherd | 2016-10-26 | Removed the underscores from the new parameters IMAGESCSV and TBPARAMETERS.  |
| Ian Humphrey | 2016-11-16 | Exceptions that occur during the solving of the bundle adjustment will now pop up as message boxes when running jigsaw in GUI mode. Fixes #4483.  |
| Ken Edmundson | 2016-11-17 | Output control net will be now be written regardless of whether bundle converges. Fixes #4533.  |
| Ken Edmundson | 2017-01-17 | Updated description and brief for SOLVETARGETBODY and TBPARAMETERS.  |
| Summer Stapleton | 2017-08-09 | Fixed bug where an invalid control net was not throwing exception. Fixes #5068.  |
| Ken Edmundson | 2018-05-23 | Modifed call to bundleAdjustment->solveCholeskyBR() to return a raw pointer to a BundleSolutionInfo object. Am also deleting this pointer because jigsaw.cpp takes ownership from BundleAdjust.  |
| Debbie A. Cook | 2018-06-04 | (BundleXYZ modified on 2017-09-11) Added options for outputting and/or solving for body-fixed x/y/z instead of lat/lon/radius. References #501.  |
| Debbie A. Cook | 2018-06-04 | (BundleXYZ modified on 2017-09-17) Fixed a problem in the xml that was causing the input parameters to be omitted from the history. References #501.  |
| Debbie A. Cook | 2018-06-04 | (BundleXYZ modified on 2018-03-18) Fixed a problem in the xml that excluded entry of values for latitudinal point sigmas when the coordinate type for reports was set to Rectangular and vice versa. References #501.  |
| Tyler Wilson | 2019-05-17 | Cleaned up the bundleout.txt file and added new information in the header. Fixes #3267.  |
| Ken Edmundson and Debbie A. Cook | 2019-05-20 | Added initial support for simultaneous LIDAR data. Added LIDARDATA, OLIDARDATA, and OLIDARFORMAT arguments.  |
| Aaron Giroux | 2019-12-19 | Added SCCONFIG parameter which allows users to pass in a pvl file with different settings for different instrumentIDs. Added logic into the observationSolveSettings function to construct BundleObservationSolveSettings objects based off of the settings in the pvl file.  |
| Adam Paquette | 2020-12-23 | Added a warning when solving for target body radii/radius that is output to the application log. Updated the documentation to include the original rand notebook that jigsaw was based on. Also added a section in the documentation describing the target body radii solve issue.  |
| Jesse Mapel and Kristin Berry | 2021-06-29 | Added the ability to bundle adjust images that use a CSM based model. New parameters CSMSOLVESET, CSMSOLVELIST, and CSMSOLVEYPE were added to specify which parameters to solve for. These parameters can also be used as keys in the SCCONFIG file. Modified images CSV file to generate a separate CSV for each sensor being adjusted.  |
| Jesse Mapel | 2021-11-09 | Fixed measure residual reporting in bundleout.txt file to match the residuals reported in the residuals CSV file.  |

 |

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Parameter Groups Files

|  |  |
| --- | --- |
| **Name**  | **Description**  |
| FROMLIST | List of cubes in the input control network  |
| HELDLIST | List of (non-overlapping) cubes to hold in the adjustment  |
| CNET | Input control network  |
| ONET | Output control network  |
| LIDARDATA | Input Lidar Point File - requires SPSOLVE to be turned on  |
| OLIDARDATA | Output lidar data file - requires SPSOLVE to be turned on  |
| OLIDARFORMAT | Output lidar data file format |
| SCCONFIG | File containing Camera/Spacecraft parameters  |

Solve Options

|  |  |
| --- | --- |
| **Name**  | **Description**  |
| OBSERVATIONS | Keep instances of the same observation in different cube files the same |
| RADIUS | Solve for local radii of points |
| UPDATE | Update cube label |
| OUTLIER\_REJECTION | Auto-rejection of outliers |
| REJECTION\_MULTIPLIER | Rejection multiplier |
| ERRORPROPAGATION | Compute variance-covariance matrix |

Maximum Likelihood Estimation

|  |  |
| --- | --- |
| **Name**  | **Description**  |
| MODEL1 | A maximum likelihood estimation model selection |
| MAX\_MODEL1\_C\_QUANTILE | Quantile of the |residual| distribution used to set the tweaking constant of the maximum likelihood estimation model |
| MODEL2 | A maximum likelihood estimation model selection |
| MAX\_MODEL2\_C\_QUANTILE | Quantile of the |residual| distribution used to set the tweaking constant of the maximum likelihood estimation model |
| MODEL3 | A maximum likelihood estimation model selection |
| MAX\_MODEL3\_C\_QUANTILE | Quantile of the |residual| distribution used to set the tweaking constant of the maximum likelihood estimation model |

Convergence Criteria

|  |  |
| --- | --- |
| **Name**  | **Description**  |
| SIGMA0 | Standard deviation of unit weight  |
| MAXITS | Maximum number of iterations  |

Camera Pointing Options

|  |  |
| --- | --- |
| **Name**  | **Description**  |
| CKDEGREE | Degree of polynomial fit to original camera angles  |
| CKSOLVEDEGREE | The degree of the polynomial being fit to in the bundle adjustment  |
| CAMSOLVE | Camera pointing parameters to include in the bundle adjustment |
| TWIST | Solve for twist |
| OVEREXISTING | Fit polynomial over the existing pointing |

Spacecraft Options

|  |  |
| --- | --- |
| **Name**  | **Description**  |
| SPKDEGREE | Degree of polynomial fit to original camera position  |
| SPKSOLVEDEGREE | The degree of the camera position polynomial being fit to in the bundle adjustment.  |
| SPSOLVE | Spacecraft position parameters to include in the adjustment |
| OVERHERMITE | Fit polynomial over the existing Hermite spline |

Community Sensor Model Options

|  |  |
| --- | --- |
| **Name**  | **Description**  |
| CSMSOLVESET | Specify a set of a CSM parameters to solve for. |
| CSMSOLVETYPE | Specify a type of a CSM parameters to solve for. |
| CSMSOLVELIST | Specify an explicit list of CSM parameters to solve for. |

Target Body

|  |  |
| --- | --- |
| **Name**  | **Description**  |
| SOLVETARGETBODY | Solve for target body parameters. The parameters, their a priori values, and uncertainties are input using a PVL file specified by TBPARAMETERS below. |
| TBPARAMETERS | File containing target body parameters to solve for, their a priori values and uncertainties.  |

Control Point Parameters

|  |  |
| --- | --- |
| **Name**  | **Description**  |
| CONTROL\_POINT\_COORDINATE\_TYPE\_BUNDLE | Coordinate type to use for bundling and outputting control points |
| CONTROL\_POINT\_COORDINATE\_TYPE\_REPORTS | Coordinate type to use for bundling and outputting control points |

Parameter Uncertainties

|  |  |
| --- | --- |
| **Name**  | **Description**  |
| POINT\_LATITUDE\_SIGMA | Global latitude uncertainty for all points (meters)  |
| POINT\_LONGITUDE\_SIGMA | Global longitude uncertainty for all points (meters)  |
| POINT\_RADIUS\_SIGMA | Global radius uncertainty for all points (meters)  |
| POINT\_X\_SIGMA | Global body-fixed X uncertainty for all points (meters)  |
| POINT\_Y\_SIGMA | Global body-fixed X uncertainty for all points (meters)  |
| POINT\_Z\_SIGMA | Global body-fixed X uncertainty for all points (meters)  |
| SPACECRAFT\_POSITION\_SIGMA | Global uncertainty for spacecraft coordinates (meters)  |
| SPACECRAFT\_VELOCITY\_SIGMA | Global uncertainty for spacecraft velocity (meters/second)  |
| SPACECRAFT\_ACCELERATION\_SIGMA | Global uncertainty for spacecraft acceleration (meters/second/second) |
| CAMERA\_ANGLES\_SIGMA | global uncertainty for camera angles (decimal degrees) |
| CAMERA\_ANGULAR\_VELOCITY\_SIGMA | Global uncertainty for camera angular velocity (decimal degrees/second) |
| CAMERA\_ANGULAR\_ACCELERATION\_SIGMA | Global uncertainty for camera angular acceleration (decimal degrees/second/second) |

Output Options

|  |  |
| --- | --- |
| **Name**  | **Description**  |
| FILE\_PREFIX | Output file prefix |
| BUNDLEOUT\_TXT | Standard bundle output file - bundleout.txt  |
| IMAGESCSV | Outputs image data (body-fixed) to csv file - bundleout\_images.csv  |
| OUTPUT\_CSV | Outputs point and image data (body-fixed) to csv file - bundleout\_points.csv  |
| RESIDUALS\_CSV | Outputs image coordinate residuals to csv file - residuals.csv |
| LIDAR\_CSV | Outputs lidar data to csv file - lidar.csv |

 |

**X**

### *Files*: FROMLIST

#### Description

This file contains a list of all [cube](../../../../documents/Glossary/Glossary.html#Cube)s in the [control network](../../../../documents/Glossary/Glossary.html#ControlNetwork)

|  |  |
| --- | --- |
| **Type**  | filename |
| **File Mode**  | input |
| **Filter**  | \*.txt \*.lis  |

Close Window

**X**

### *Files*: HELDLIST

#### Description

This file contains a list of all [cube](../../../../documents/Glossary/Glossary.html#Cube)s whose orientation and position will be held in the adjustment. These images will still be included in the solution, but their camera orientation and spacecraft position will be constrained to keep the values from changing. This is an optional parameter and the default is to not hold any of the images. *Note that held images must not overlap each other to work properly.*

|  |  |
| --- | --- |
| **Type**  | filename |
| **File Mode**  | input |
| **Internal Default**  | none |
| **Filter**  | \*.txt \*.lis  |

Close Window

**X**

### *Files*: CNET

#### Description

This file is a [control network](../../../../documents/Glossary/Glossary.html#ControlNetwork) generated from programs such as **autoseed** or **qnet**. It contains the [control point](../../../../documents/Glossary/Glossary.html#ControlPoint)s and associated measures.

|  |  |
| --- | --- |
| **Type**  | filename |
| **File Mode**  | input |
| **Filter**  | \*.net  |

Close Window

**X**

### *Files*: ONET

#### Description

This output file contains the updated [control network](../../../../documents/Glossary/Glossary.html#ControlNetwork) with the final coordinates of the [control point](../../../../documents/Glossary/Glossary.html#ControlPoint)s and residuals for each measurement.

|  |  |
| --- | --- |
| **Type**  | filename |
| **File Mode**  | output |
| **Filter**  | \*.net  |

Close Window

**X**

### *Files*: LIDARDATA

#### Description

This file is a [lidar point data](../../../../documents/Glossary/Glossary.html#LidarPointData) generated from **lrolola2isis.cpp**. It contains [lidar control point](../../../../documents/Glossary/Glossary.html#LidarControlPoint)s and associated measures for simultaneous images. If [lidar point data](../../../../documents/Glossary/Glossary.html#LidarPointData) are used, SPSOLVE must be turned on (anything besides NONE)

|  |  |
| --- | --- |
| **Type**  | filename |
| **File Mode**  | input |
| **Internal Default**  | none |
| **Exclusions**  |  |
| **Filter**  | \*.dat \*.json  |

Close Window

**X**

### *Files*: OLIDARDATA

#### Description

This output file contains the adjusted [lidar data](../../../../documents/Glossary/Glossary.html#LidarData) with the final coordinates of the [lidar point](../../../../documents/Glossary/Glossary.html#LidarPoint)s and residuals for each measurement.

|  |  |
| --- | --- |
| **Type**  | filename |
| **File Mode**  | output |
| **Internal Default**  | none |
| **Filter**  | \*.dat \*.json  |

Close Window

**X**

### *Files*: OLIDARFORMAT

#### Description

Output lidar data file format.

|  |  |
| --- | --- |
| **Type**  | string |
| **Default**  | BINARY  |
| **Option List:**  |

|  |  |  |
| --- | --- | --- |
| **Option** | **Brief** | **Description** |
| BINARY | Output lidar data in binary format | Output lidar data in binary format.  |
| JSON | Output lidar data in json format  | Output lidar data in json format.  |

 |

Close Window

**X**

### *Files*: SCCONFIG

#### Description

This file contains the Camera/Spacecraft parameters to use when processing images from different sensors. This file should be in PVL format. It should contain an object called SensorParameters with one group per spacecraft/instrument combination. The SpacecraftName and InstrumentId keywords in the Instrument group of an image file are used to create the name of each group in the PVL file. The group pertaining to each spacecraft/instrument should contain the keyword/value pairs needed to process images taken with that sensor: CKDEGREE, CKSOLVEDEGREE, CAMSOLVE, TWIST, OVEREXISTING, SPKDEGREE, SPKSOLVEDEGREE, SPSOLVE, OVERHERMITE, SPACECRAFT\_POSITION\_SIGMA, SPACECRAFT\_VELOCITY\_SIGMA, SPACECRAFT\_ACCELERATION\_SIGMA, CAMERA\_ANGLES\_SIGMA, CAMERA\_ANGULAR\_VELOCITY\_SIGMA, CAMERA\_ANGULAR\_ACCELERATION\_SIGMA. If any of these keywords are missing, then their defaults will be used. There is an example template at $ISISROOT/appdata/templates/jigsaw/SensorParameters.pvl that can be used as a guide.

|  |  |
| --- | --- |
| **Type**  | filename |
| **File Mode**  | input |
| **Exclusions**  | * CKDEGREE
* CKSOLVEDEGREE
* CAMSOLVE
* OVEREXISTING
* SPKDEGREE
* SPKSOLVEDEGREE
* SPSOLVE
* OVERHERMITE
* CAMERA\_ANGLES\_SIGMA
* CAMERA\_ANGULAR\_VELOCITY\_SIGMA
* CAMERA\_ANGULAR\_ACCELERATION\_SIGMA
* SPACECRAFT\_POSITION\_SIGMA
* SPACECRAFT\_VELOCITY\_SIGMA
* SPACECRAFT\_ACCELERATION\_SIGMA
 |
| **Filter**  | \*.pvl  |

Close Window

**X**

### *Solve Options*: OBSERVATIONS

#### Description

This option will solve for [SPICE](../../../../documents/Glossary/Glossary.html#SPICE) on all cubes with a matching observation number as though they were a single observation. For most missions, the default observation number is equivalent to the [serial number](../../../../documents/Glossary/Glossary.html#SerialNumber) of the [cube](../../../../documents/Glossary/Glossary.html#Cube), and a single [cube](../../../../documents/Glossary/Glossary.html#Cube) is an observation. However, for the Lunar Orbiter mission, an image has a defined observation number that is a substring of its [serial number](../../../../documents/Glossary/Glossary.html#SerialNumber). This feature allows the three subframes of a Lunar Orbiter High Resolution frame to be treated as a single observation when this option is used; otherwise, each subframe is adjusted independently.

|  |  |
| --- | --- |
| **Type**  | boolean |
| **Default**  | No  |

Close Window

**X**

### *Solve Options*: RADIUS

#### Description

Select this option to solve for the local radius of each [control point](../../../../documents/Glossary/Glossary.html#ControlPoint). If this button is not turned on, the radii of the points will not change from the [cube](../../../../documents/Glossary/Glossary.html#Cube)'s shape model.

|  |  |
| --- | --- |
| **Type**  | boolean |
| **Default**  | No  |
| **Inclusions**  | * POINT\_RADIUS\_SIGMA
 |

Close Window

**X**

### *Solve Options*: UPDATE

#### Description

When this option is selected, the application will update the labels of the individual [cube](../../../../documents/Glossary/Glossary.html#Cube)s in the FROMLIST with the final values from the solution if the adjustment converges. The results are written to the [SPICE](../../../../documents/Glossary/Glossary.html#SPICE) blobs attached to the cube, overwriting the previous values. If this option is not selected, the [cube](../../../../documents/Glossary/Glossary.html#Cube) files are not changed. All other output files are still created.

|  |  |
| --- | --- |
| **Type**  | boolean |
| **Default**  | No  |

Close Window

**X**

### *Solve Options*: OUTLIER\_REJECTION

#### Description

Select this option to perform automatic outlier detection and rejection.

|  |  |
| --- | --- |
| **Type**  | boolean |
| **Default**  | No  |
| **Exclusions**  | * MODEL1
* MODEL2
* MODEL3
* MAX\_MODEL1\_C\_QUANTILE
* MAX\_MODEL2\_C\_QUANTILE
* MAX\_MODEL3\_C\_QUANTILE
 |
| **Inclusions**  | * REJECTION\_MULTIPLIER
 |

Close Window

**X**

### *Solve Options*: REJECTION\_MULTIPLIER

#### Description

Rejection multiplier

|  |  |
| --- | --- |
| **Type**  | double |
| **Default**  | 3.0  |
| **Inclusions**  | * OUTLIER\_REJECTION
 |

Close Window

**X**

### *Solve Options*: ERRORPROPAGATION

#### Description

Select this option to compute the variance-covariance matrix of the parameters. The parameter uncertainties can be computed from this matrix.

|  |  |
| --- | --- |
| **Type**  | boolean |
| **Default**  | No  |

Close Window

**X**

### *Maximum Likelihood Estimation*: MODEL1

#### Description

A maximum likelihood estimation model selection.

|  |  |
| --- | --- |
| **Type**  | string |
| **Default**  | NONE |
| **Option List:**  |

|  |  |  |
| --- | --- | --- |
| **Option** | **Brief** | **Description** |
| NONE | None: no tier one maximum likelihood estimation | None: no tier one maximum likelihood estimation Exclusions* MODEL2
* MODEL3
* MAX\_MODEL1\_C\_QUANTILE
* MAX\_MODEL2\_C\_QUANTILE
* MAX\_MODEL3\_C\_QUANTILE
 |
| HUBER | Huber: approximates the L2 norm near 0, and the L1 norm thereafter. Has one continuous derivative. | A highly recommended model that works well in many situations. Exclusions* REJECTION\_MULTIPLIER
* OUTLIER\_REJECTION
* ERRORPROPAGATION
 |
| HUBER\_MODIFIED | Huber Modified: approximates the L2 norm near 0 and the L1 norm thereafter. Has two continuous derivatives. | An adaptation of the highly recommended Huber model that has two continuous derivatives. Exclusions* REJECTION\_MULTIPLIER
* OUTLIER\_REJECTION
* ERRORPROPAGATION
 |

 |

Close Window

**X**

### *Maximum Likelihood Estimation*: MAX\_MODEL1\_C\_QUANTILE

#### Description

The tweaking constant has different meanings depending on the model being used: Huber models: The point at which the transformation motion from L2 to L1 norms takes place. Recommended quantile: 0.5 Welsch model: Residuals whose absolute value is twice the tweaking constant are approaching negligible significance. Recommended quantile: 0.7 Chen model: Residuals whose absolute value is greater than the tweaking constant are totally ignored. Recommended quantile: > 0.9

|  |  |
| --- | --- |
| **Type**  | double |
| **Default**  | 0.5  |
| **Minimum**  | 0 (exclusive)  |
| **Maximum**  | 1 (exclusive)  |

Close Window

**X**

### *Maximum Likelihood Estimation*: MODEL2

#### Description

A maximum likelihood estimation model selection.

|  |  |
| --- | --- |
| **Type**  | string |
| **Default**  | NONE |
| **Option List:**  |

|  |  |  |
| --- | --- | --- |
| **Option** | **Brief** | **Description** |
| NONE | None: no tier two maximum likelihood estimation | None: no tier two maximum likelihood estimation Exclusions* MODEL3
* MAX\_MODEL2\_C\_QUANTILE
* MAX\_MODEL3\_C\_QUANTILE
 |
| HUBER | Huber: approximates the L2 norm near 0, and the L1 norm thereafter. Has one continuous derivative. | A highly recommended model that works well in many situations.  |
| HUBER\_MODIFIED | Huber Modified: approximates the L2 norm near 0 and the L1 norm thereafter. Has two continuous derivatives. | An adaptation of the highly recommended Huber model that has two continuous derivatives.  |
| WELSCH | Welsch: approximates the L2 norm near 0, but then decays exponentially to zero. | This model reduces the significance of large residuals more aggressively than Huber. Large residuals will have less influence than small residuals, and they approach negligibility as they approach infinity. Measures can be effectively 'removed' by this method, which may cause singularities and/or islands.  |
| CHEN | Chen: a highly aggressive method that intentionally removes the largest few percent of residuals. | This method dramatically increases the influence of smaller residuals (beyond the L2 norm) while simultaneously totally ignoring the largest few percent of the residuals.  |

 |

Close Window

**X**

### *Maximum Likelihood Estimation*: MAX\_MODEL2\_C\_QUANTILE

#### Description

The tweaking constant has different meanings depending on the model being used: Huber models: The point at which the transformation motion from L2 to L1 norms takes place. Recommended quantile: 0.5 Welsch model: Residuals whose absolute value is twice the tweaking constant are approaching negligible significance. Recommended quantile: 0.7 Chen model: Residuals whose absolute value is greater than the tweaking constant are totally ignored. Recommended quantile: > 0.9

|  |  |
| --- | --- |
| **Type**  | double |
| **Default**  | 0.5  |
| **Minimum**  | 0 (exclusive)  |
| **Maximum**  | 1 (exclusive)  |

Close Window

**X**

### *Maximum Likelihood Estimation*: MODEL3

#### Description

A maximum likelihood estimation model selection.

|  |  |
| --- | --- |
| **Type**  | string |
| **Default**  | NONE |
| **Option List:**  |

|  |  |  |
| --- | --- | --- |
| **Option** | **Brief** | **Description** |
| NONE | None: no tier three maximum likelihood estimation | None: no tier three maximum likelihood estimation Exclusions* MAX\_MODEL3\_C\_QUANTILE
 |
| HUBER | Huber: approximates the L2 norm near 0, and the L1 norm thereafter. Has one continuous derivative. | A highly recommended model that works well in many situations.  |
| HUBER\_MODIFIED | Huber Modified: approximates the L2 norm near 0 and the L1 norm thereafter. Has two continuous derivatives. | An adaptation of the highly recommended Huber model that has two continuous derivatives.  |
| WELSCH | Welsch: approximates the L2 norm near 0, but then decays exponentially to zero. | This model reduces the significance of large residuals more aggressively than Huber. Large residuals will have less influence than small residuals, and they approach negligibility as they approach infinity. Measures can be effectively 'removed' by this method, which may cause singularities and/or islands.  |
| CHEN | Chen: a highly aggressive method that intentionally removes the largest few percent of residuals. | This method dramatically increases the influence of smaller residuals (beyond the L2 norm) while simultaneously totally ignoring the largest residuals.  |

 |

Close Window

**X**

### *Maximum Likelihood Estimation*: MAX\_MODEL3\_C\_QUANTILE

#### Description

The tweaking constant has different meanings depending on the model being used: Huber models: The point at which the transformation motion from L2 to L1 norms takes place. Recommended quantile: 0.5 Welsch model: Residuals whose absolute value is twice the tweaking constant are approaching negligible significance. Recommended quantile: 0.7 Chen model: Residuals whose absolute value is greater than the tweaking constant are totally ignored. Recommended quantile: > 0.9

|  |  |
| --- | --- |
| **Type**  | double |
| **Default**  | 0.5  |
| **Minimum**  | 0 (exclusive)  |
| **Maximum**  | 1 (exclusive)  |

Close Window

**X**

### *Convergence Criteria*: SIGMA0

#### Description

Converges on stabilization of Sigma0. Convergence occurs when the change in sigma0 between iterations is less than or equal to Sigma0.

|  |  |
| --- | --- |
| **Type**  | double |
| **Default**  | 1.0e-10  |
| **Minimum**  | 0 (exclusive)  |

Close Window

**X**

### *Convergence Criteria*: MAXITS

#### Description

Maximum number of times to iterate. The application stops iterating at MAXIT, or when convergence is reached.

|  |  |
| --- | --- |
| **Type**  | integer |
| **Default**  | 50  |
| **Minimum**  | 1 (inclusive)  |

Close Window

**X**

### *Camera Pointing Options*: CKDEGREE

#### Description

The degree of the polynomial fit to the original camera angles and used to generate [a priori](../../../../documents/Glossary/Glossary.html#APriori) camera angles for the first iteration.

|  |  |
| --- | --- |
| **Type**  | integer |
| **Default**  | 2  |
| **Minimum**  | 0 (inclusive)  |

Close Window

**X**

### *Camera Pointing Options*: CKSOLVEDEGREE

#### Description

The degree of the polynomial being fit to in the bundle adjust solution. This polynomial can be different from the one used to generate the [a priori](../../../../documents/Glossary/Glossary.html#APriori) camera angles used in the first iteration. In the case of an instrument with a jitter problem, a higher degree polynomial fit to each of the camera angles might provide a better solution (smaller errors). For framing cameras, the application automatically sets degree to 0.

|  |  |
| --- | --- |
| **Type**  | integer |
| **Default**  | 2  |
| **Minimum**  | 0 (inclusive)  |

Close Window

**X**

### *Camera Pointing Options*: CAMSOLVE

#### Description

This parameter is used to specify which, if any, camera pointing parameters to include in the adjustment.

|  |  |
| --- | --- |
| **Type**  | string |
| **Default**  | ANGLES  |
| **Option List:**  |

|  |  |  |
| --- | --- | --- |
| **Option** | **Brief** | **Description** |
| NONE | Don't solve for any camera pointing factors | If this option is selected, no camera pointing parameters will be adjusted. Exclusions* CKDEGREE
* CKSOLVEDEGREE
* TWIST
* OVEREXISTING
 |
| ANGLES | Solve for camera angles: right ascension, declination and optionally twist  | Camera angles in each [cube](../../../../documents/Glossary/Glossary.html#Cube) will be adjusted in the solution, but not angular velocities or accelerations. Solving for the first two camera angles translates images in sample and line. Adding the third angle to the solution (TWIST option) allows for rotation corrections. Adjustments are not applied unless the solution converges and UPDATE is selected. Solving for angles only is equivalent to using CKSOLVEDEGREE=0. Exclusions* CKDEGREE
* CKSOLVEDEGREE
 |
| VELOCITIES | Solve for camera angles AND their angular velocities | Camera angles and their angular velocities will be adjusted in the solution. Solving for angles and velocities is equivalent to using CKSOLVEDEGREE=1. Exclusions* CKDEGREE
* CKSOLVEDEGREE
 |
| ACCELERATIONS | Solve for camera angles, their angular velocities and accelerations | Camera angles, their angular velocities, and accelerations will be adjusted in the solution. Solving for angles, angular velocities, and accelerations is equivalent to using CKSOLVEDEGREE=2. Exclusions* CKDEGREE
* CKSOLVEDEGREE
 |
| ALL | Solve for all coefficients in the polynomials fit to the camera angles. | If this option is selected, all coefficients of the solve equation will be adjusted in the solution (CKSOLVEDEGREE+1 coefficients)  |

 |

Close Window

**X**

### *Camera Pointing Options*: TWIST

#### Description

If this option is selected, the twist angle will be adjusted in the bundle adjustment solution.

|  |  |
| --- | --- |
| **Type**  | boolean |
| **Default**  | Yes  |

Close Window

**X**

### *Camera Pointing Options*: OVEREXISTING

#### Description

This option will fit a polynomial over the existing pointing data. This data is held constant in the adjustment, and the initial value for the each of the coefficients in the polynomials is 0. When this option is used, the current pointing is used as a priori in the adjustment.

|  |  |
| --- | --- |
| **Type**  | boolean |
| **Default**  | No  |

Close Window

**X**

### *Spacecraft Options*: SPKDEGREE

#### Description

The degree of the polynomial fit to the original camera position and used to generate [a priori](../../../../documents/Glossary/Glossary.html#APriori) camera positions for the first iteration.

|  |  |
| --- | --- |
| **Type**  | integer |
| **Default**  | 2  |
| **Minimum**  | 0 (inclusive)  |

Close Window

**X**

### *Spacecraft Options*: SPKSOLVEDEGREE

#### Description

The degree of the polynomial being fit to in the bundle adjust solution. This polynomial can be different from the one used to generate the [a priori](../../../../documents/Glossary/Glossary.html#APriori) camera positions used in the first iteration. In the case of an instrument with a jitter problem, a higher degree polynomial fit for the camera position might provide a better solution (smaller errors). For framing cameras, the application automatically sets degree to 0.

|  |  |
| --- | --- |
| **Type**  | integer |
| **Default**  | 2  |
| **Minimum**  | 0 (inclusive)  |

Close Window

**X**

### *Spacecraft Options*: SPSOLVE

#### Description

This parameter is used to specify which, if any, spacecraft position parameters to include in the adjustment.

|  |  |
| --- | --- |
| **Type**  | string |
| **Default**  | NONE  |
| **Option List:**  |

|  |  |  |
| --- | --- | --- |
| **Option** | **Brief** | **Description** |
| NONE | Don't solve for any spacecraft position parameters | No spacecraft position parameters will be adjusted. Exclusions* SPKDEGREE
* SPKSOLVEDEGREE
* OVERHERMITE
 |
| POSITIONS | Solve for the spacecraft positions | Spacecraft positions will be adjusted in the solution, but not the velocity or the acceleration. Exclusions* SPKDEGREE
* SPKSOLVEDEGREE
 |
| VELOCITIES | Solve for the spacecraft positions and velocities | Spacecraft positions will be adjusted in the solution, as well as the velocities of the spacecraft at each instance. Exclusions* SPKDEGREE
* SPKSOLVEDEGREE
 |
| ACCELERATIONS | Solve for the spacecraft positions, velocities, and accelerations | Spacecraft positions will be adjusted in the solution, as well as the velocities and accelerations of the spacecraft at each instance. Exclusions* SPKDEGREE
* SPKSOLVEDEGREE
 |
| ALL | Solve for all coefficients in the polynomials fit to the camera position. | If this option is selected, all coefficients of the solve equation will be adjusted in the solution (SPKSOLVEDEGREE+1 coefficients)  |

 |

Close Window

**X**

### *Spacecraft Options*: OVERHERMITE

#### Description

This option will fit a polynomial over the existing Hermite cubic spline used to interpolate the coordinates of the spacecraft position. The spline is held constant in the adjustment, and the initial value for the each of the coefficients in the polynomials is 0. When this option is used, the current positions are used as a priori in the adjustment.

|  |  |
| --- | --- |
| **Type**  | boolean |
| **Default**  | No  |

Close Window

**X**

### *Community Sensor Model Options*: CSMSOLVESET

#### Description

Specify one of the parameter sets from the CSM GeometricModel API to solve for. All parameters belonging to the specified set will be solved for.

|  |  |
| --- | --- |
| **Type**  | string |
| **Internal Default**  | none |
| **Option List:**  |

|  |  |  |
| --- | --- | --- |
| **Option** | **Brief** | **Description** |
| VALID | Solves for CSM parameters that are not NONE. |  |
| ADJUSTABLE | Solves for real or fictitous CSM parameters. |  |
| NONADJUSTABLE | Solves for fixed CSM parameters. | Solve for fixed CSM parameters. These parameters are generally not adjusted but do have uncertainty which can help constrain the solutions and improve a posteriori uncertainties for other parameters.  |

 |
| **Exclusions**  | * CSMSOLVETYPE
* CSMSOLVELIST
 |

Close Window

**X**

### *Community Sensor Model Options*: CSMSOLVETYPE

#### Description

Specify a parameter type from the CSM GeometricModel API to solve for. All parameters of the specified type will be solved for.

|  |  |
| --- | --- |
| **Type**  | string |
| **Option List:**  |

|  |  |  |
| --- | --- | --- |
| **Option** | **Brief** | **Description** |
| NONE | Solve for unitialized CSM parameters |  |
| FICTITIOUS | Solve for CSM parameters calculted by resection |  |
| REAL | Solve for measured CSM parameters |  |
| FIXED | Solve for fixed CSM parameters |  |

 |
| **Exclusions**  | * CSMSOLVESET
* CSMSOLVELIST
 |

Close Window

**X**

### *Community Sensor Model Options*: CSMSOLVELIST

#### Description

All CSM parameters in this list will be solved for. Trailing and leading whitespace will be stripped off. Use standard ISIS parameter array notation to specify multiple parameters.

|  |  |
| --- | --- |
| **Type**  | string |
| **Exclusions**  | * CSMSOLVESET
* CSMSOLVETYPE
 |

Close Window

**X**

### *Target Body*: SOLVETARGETBODY

#### Description

Solve for target body parameters. The parameters, their a priori values, and uncertainties are input using a PVL file specified by TBPARAMETERS below. An example template PVL file is located at $ISISROOT/appdata/templates/jigsaw/TargetBodyParameters.pvl.

|  |  |
| --- | --- |
| **Type**  | boolean |
| **Default**  | false |
| **Inclusions**  | * TBPARAMETERS
 |

Close Window

**X**

### *Target Body*: TBPARAMETERS

#### Description

This file contains target body parameters to solve for in the bundle adjustment, their a priori values, and uncertainties. The file must be in PVL format. An example template PVL file is located at $ISISROOT/appdata/templates/jigsaw/TargetBodyParameters.pvl. Instructions for the PVL structure are given in the template.

|  |  |
| --- | --- |
| **Type**  | filename |
| **File Mode**  | input |
| **Filter**  | \*.pvl  |

Close Window

**X**

### *Control Point Parameters*: CONTROL\_POINT\_COORDINATE\_TYPE\_BUNDLE

#### Description

This parameter indicates which coordinate type will be used to present the control points in the bundle adjustment and bundle output.

|  |  |
| --- | --- |
| **Type**  | string |
| **Default**  | LATITUDINAL  |
| **Option List:**  |

|  |  |  |
| --- | --- | --- |
| **Option** | **Brief** | **Description** |
| LATITUDINAL | Coordinates will be planetocentric latitudinal | If this option is selected all control points will be adjusted, corrected, and reported in planetocentric latitudinal coordinates (latitude, longitude, and radius). Exclusions* POINT\_X\_SIGMA
* POINT\_Y\_SIGMA
* POINT\_Z\_SIGMA
 |
| RECTANGULAR | Coordinates will be body-fixed rectangular | If this option is selected all control points will be adjusted, corrected, and reported in body-fixed rectangular coordinates (X, Y, and Z). Exclusions* POINT\_LATITUDE\_SIGMA
* POINT\_LONGITUDE\_SIGMA
* POINT\_RADIUS\_SIGMA
 |

 |

Close Window

**X**

### *Control Point Parameters*: CONTROL\_POINT\_COORDINATE\_TYPE\_REPORTS

#### Description

This parameter indicates which coordinate type will be used to present the control points in the bundle adjustment and bundle output.

|  |  |
| --- | --- |
| **Type**  | string |
| **Default**  | LATITUDINAL  |
| **Option List:**  |

|  |  |  |
| --- | --- | --- |
| **Option** | **Brief** | **Description** |
| LATITUDINAL | Coordinates will be planetocentric latitudinal | If this option is selected all control points will be adjusted, corrected, and reported in planetocentric latitudinal coordinates (latitude, longitude, and radius).  |
| RECTANGULAR | Coordinates will be body-fixed rectangular | If this option is selected all control points will be adjusted, corrected, and reported in body-fixed rectangular coordinates (X, Y, and Z).  |

 |

Close Window

**X**

### *Parameter Uncertainties*: POINT\_LATITUDE\_SIGMA

#### Description

This optional value will be used as the global latitude uncertainty for all points. Units are meters.

|  |  |
| --- | --- |
| **Type**  | double |
| **Internal Default**  | none |
| **Minimum**  | 0 (inclusive)  |

Close Window

**X**

### *Parameter Uncertainties*: POINT\_LONGITUDE\_SIGMA

#### Description

This optional value will be used as the global longitude uncertainty for all points. Units are meters.

|  |  |
| --- | --- |
| **Type**  | double |
| **Internal Default**  | none |
| **Minimum**  | 0 (inclusive)  |

Close Window

**X**

### *Parameter Uncertainties*: POINT\_RADIUS\_SIGMA

#### Description

This value will be used as the global radius uncertainty for all points. Units are meters.

|  |  |
| --- | --- |
| **Type**  | double |
| **Internal Default**  | none |
| **Minimum**  | 0 (inclusive)  |
| **Inclusions**  | * RADIUS
 |

Close Window

**X**

### *Parameter Uncertainties*: POINT\_X\_SIGMA

#### Description

This optional value will be used as the global uncertainty for all points. Units are meters.

|  |  |
| --- | --- |
| **Type**  | double |
| **Internal Default**  | none |
| **Minimum**  | 0 (inclusive)  |

Close Window

**X**

### *Parameter Uncertainties*: POINT\_Y\_SIGMA

#### Description

This optional value will be used as the global uncertainty for all points. Units are meters.

|  |  |
| --- | --- |
| **Type**  | double |
| **Internal Default**  | none |
| **Minimum**  | 0 (inclusive)  |

Close Window

**X**

### *Parameter Uncertainties*: POINT\_Z\_SIGMA

#### Description

This optional value will be used as the global uncertainty for all points. Units are meters.

|  |  |
| --- | --- |
| **Type**  | double |
| **Internal Default**  | none |
| **Minimum**  | 0 (inclusive)  |

Close Window

**X**

### *Parameter Uncertainties*: SPACECRAFT\_POSITION\_SIGMA

#### Description

This value will be used as the global uncertainty for spacecraft coordinates. Units are meters.

|  |  |
| --- | --- |
| **Type**  | double |
| **Internal Default**  | none |
| **Minimum**  | 0 (inclusive)  |

Close Window

**X**

### *Parameter Uncertainties*: SPACECRAFT\_VELOCITY\_SIGMA

#### Description

This value will be used as the global uncertainty for spacecraft velocity. Units are meters/second.

|  |  |
| --- | --- |
| **Type**  | double |
| **Internal Default**  | none |
| **Minimum**  | 0 (inclusive)  |

Close Window

**X**

### *Parameter Uncertainties*: SPACECRAFT\_ACCELERATION\_SIGMA

#### Description

This value will be used as the global uncertainty for spacecraft acceleration. Units are meters/second/second.

|  |  |
| --- | --- |
| **Type**  | double |
| **Internal Default**  | none |
| **Minimum**  | 0 (inclusive)  |

Close Window

**X**

### *Parameter Uncertainties*: CAMERA\_ANGLES\_SIGMA

#### Description

This value will be used as the global uncertainty for camera angles. Units are decimal degrees.

|  |  |
| --- | --- |
| **Type**  | double |
| **Internal Default**  | none |
| **Minimum**  | 0 (inclusive)  |

Close Window

**X**

### *Parameter Uncertainties*: CAMERA\_ANGULAR\_VELOCITY\_SIGMA

#### Description

This value will be used as the global uncertainty for camera angular velocity. Units are decimal degrees/second.

|  |  |
| --- | --- |
| **Type**  | double |
| **Internal Default**  | none |
| **Minimum**  | 0 (inclusive)  |

Close Window

**X**

### *Parameter Uncertainties*: CAMERA\_ANGULAR\_ACCELERATION\_SIGMA

#### Description

This value will be used as the global uncertainty for camera angular acceleration. Units are decimal degrees/second/second.

|  |  |
| --- | --- |
| **Type**  | double |
| **Internal Default**  | none |
| **Minimum**  | 0 (inclusive)  |

Close Window

**X**

### *Output Options*: FILE\_PREFIX

#### Description

File prefix to prepend for the generated output files. Any prefix that is not a file path will have an underscore placed between the prefix and file name.

|  |  |
| --- | --- |
| **Type**  | string |
| **Internal Default**  | none |

Close Window

**X**

### *Output Options*: BUNDLEOUT\_TXT

#### Description

Selection of this parameter flags generation of the standard bundle output file

|  |  |
| --- | --- |
| **Type**  | boolean |
| **Default**  | yes  |

Close Window

**X**

### *Output Options*: IMAGESCSV

#### Description

Selection of this parameter flags output of image data (in body-fixed coordinates) to a csv file.

|  |  |
| --- | --- |
| **Type**  | boolean |
| **Default**  | yes  |

Close Window

**X**

### *Output Options*: OUTPUT\_CSV

#### Description

Selection of this parameter flags output of point and image data (in body-fixed coordinates) to csv file.

|  |  |
| --- | --- |
| **Type**  | boolean |
| **Default**  | yes  |

Close Window

**X**

### *Output Options*: RESIDUALS\_CSV

#### Description

Selection of this parameter flags output of image coordinate residuals to a csv file

|  |  |
| --- | --- |
| **Type**  | boolean |
| **Default**  | yes  |

Close Window

**X**

### *Output Options*: LIDAR\_CSV

#### Description

Selection of this parameter flags output of lidar data points to a csv file

|  |  |
| --- | --- |
| **Type**  | boolean |
| **Default**  | no  |

Close Window

|  |
| --- |
| Example 1Solve with CSM parametersDescription This is an example where the cubes in the network have been run through csminit. When specified in CSMSOLVELIST, parameters described in the CsmInfo group can be adjusted during the bundle. Command Line **jigsaw jigsaw fromlis=cubes.lis cnet=input.net onet=output.net radius=yes csmsolvelist="(Omega Bias, Phi Bias, Kappa Bias)" control\_point\_coordinate\_type\_bundle=rectangular control\_point\_coordinate\_type\_reports=rectangular point\_x\_sigma=50 point\_y\_sigma=50 point\_z\_sigma=50** *jigsaw CSM example*  |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Example 2Simple run of jigsaw with images from a linescanner Description This example runs jigsaw in a very simple way using four MRO CTX images. Only the required parameters are entered, with all other parameters left at their default settings. This bundle solution only solves for the camera orientation (see the CAMSOLVE and TWIST parameters). This command does not update the SPICE information attached to the four cubes. A possible use for this simple run would be to test a network to identify control points that are not well placed. Command Line **jigsaw fromlist=ctx.lis cnet=hand\_dense.net onet=hand\_dense-jig.net** *This command line only sets the three required parameters.* GUI Screenshot

|  |  |
| --- | --- |
| jigsaw Linescan image examplejigsaw Linescan image example | **Example GUI** The top of the GUI shows the parameters filled in for input cube list, the input control network and the output control network. All other parameters were left at their default values.  |

Data File Links open in a new window.

|  |  |
| --- | --- |
| [**Cube list file**](assets/linescan/ctx.lis)  | Input file defining the file names of the four cubes used in the control network. The file names can include a path if needed. Each file name is on a separate line and the last file name can have new line at the end, but it is not required.  |

 |

|  |
| --- |
| Example 3Run of jigsaw parameterized for Kaguya Terrain Camera (TC) images and a relative control network covering the Apollo 15 landing site. Description A relative network is a network that connects overlapping images with tie points but has no tie points connected to a ground source. Since there is no connection to ground in the network, this bundle will only solve for camera specific parameters. The bundle could still solve for other parameters and be correct relative to the camera position, but it would increase the complexity of the bundle. The proceeding two examples will include grounded networks, so we will wait to increase the complexity of the bundle until ground points are included. Additionally, in this example we are evaluating the solution and do not want to apply it to the images yet, therefore, update is set to 'no'. This relative bundle turns on and parameterizes the camera twist and camera acceleration solve parameters. Camera twist is a flag that allows the bundle to solve for the camera's rotation around the bore sight axis and uses the same uncertainty estimation as the other two rotations. Setting the camera solve parameter to acceleration, however, does require uncertainties to be set for the angles (deg), angular velocity (deg/s), and angular accelerations (deg/s\*\*2). These values were set with increasing constraint because of the increasing affect alterations of higher order parameters have on the bundle solution. The 'overexisting' flag tells the bundle solution to approximate the camera rotation with a zero polynomial function added to the existing rotation data (adding the polynomial **over** the **existing** data). Without the 'overexisting' flag, the bundle fits a polynomial to the existing rotations, throws out the existing data points, and uses the polynomial to calculate the approximate ephemerides when needed. This bundle also lowers the max iterations to 10 (from default 50). Lowering the max iterations does not affect the bundle solution. However, setting a lower iteration limit can serve as a flag if you expect your network to bundle quickly. Finally, the sigma0 convergence criteria was not changed from its default value, it was only explicitly stated in this call. Command Line **jigsaw fromlist=cubes.lis cnet=relative\_reg\_jig4\_edit.net onet=relative\_reg\_jig5.net update=no file\_prefix=jig5 sigma0=1.0e-10 maxits=10 twist=yes camsolve=accelerations overexisting=yes camera\_angles\_sigma=.25 camera\_angular\_velocity\_sigma=.1 camera\_angular\_acceleration\_sigma=.01**  |

|  |  |  |
| --- | --- | --- |
| Example 4Run of jigsaw parameterized for Kaguya Terrain Camera (TC) images and an intermediate ground control network covering the Apollo 15 landing site. Description This example jigsaw bundle is run directly after adding ground control points to the previous relative network. With ground control points inserted into the bundle solution, we will expand the bundle solve parameters to attempt solving for point radius values and the position of the spacecraft. Ground points are weighted more heavily in the bundle than relative control points and adding parameters adds more complexity to the bundle solution. Therefore, it is common to create and refine a network with only relative points and add ground point in **after** the relative network (and its bundle solution) is of sufficient quality. The purpose of this bundle is to ensure the network bundle converges with the added ground points and solve parameters, before committing to updating the camera pointing on the images, so update is set to 'no'. In addition to the previous solve parameters, this bundle turns on the point radius and spacecraft position solve parameters. Applying the point radius solve parameter requires the point\_radius\_sigma to be set. This value is a representation the uncertainty (in meters) of the cameras apriori pointing corresponding to the correct elevation on the shape model; this value is not a hard constraint. Often this value can be set, and the appropriateness of the set value can be checked using the 'POINTS DETAIL' section of the bundleout.txt file output by jigsaw. If more than half of the radius total corrections exceed the provided sigma, the uncertainty may need to be increased. Applying the space craft position solve parameter requires an uncertainty estimation through spacecraft\_position\_sigma (again this value is not a hard constraint). The appropriateness of the provided sigma can be evaluated through the bundleout\_images.csv X Correction, Y Correction, and Z Correction columns. The 'overhermite' flag allows an estimation the spacecraft position like the 'overexisting' flag estimates the camera pointing, with a zero-polynomial added over the existing data (for spacecraft position this is a cubic Hermite spline). These options require more memory but provide a solution more representative of small variations in the original ephemeris data. Command Line **jigsaw fromlist=cubes\_for\_ground\_update.lis cnet=grounded\_relative\_reg.net onet=grounded\_rr\_jig1.net update=no file\_prefix=jig1rr sigma0=1.0e-10 maxits=10 twist=yes camsolve=accelerations overexisting=yes camera\_angles\_sigma=.25 camera\_angular\_velocity\_sigma=.1 camera\_angular\_acceleration\_sigma=.01 radius=yes point\_radius\_sigma=500 spsolve=positions overhermite=yes spacecraft\_position\_sigma=1000** Data File Links open in a new window.

|  |  |
| --- | --- |
| [**jig1rr\_bundleout.txt**](assets/ap15/jig1rrEP_bundleout.txt)  | A shorten example of a typical bundleout.txt file produced by a jigsaw run with error propegation. This file was trimmed to hold 10 images, 50 relative points, and all ground points along with their associated detail sections. Bundleout files typically contain all image and point from a network put through jigsaw.  |

 |

|  |
| --- |
| Example 5Solve with lidar dataDescription This is an example of how to use simultaneous lidar measurments to constrain the instrument position. The constraints on the lidar data are already contained in the lidardata input file. Command Line **jigsaw jigsaw lidar\_csv=yes fromlis=cubes.lis cnet=input.net onet=output.net lidardata=lidar\_data.json olidardata=lidar\_data\_jigged.json radius=yes twist=yes camsolve=accelerations overexisting=yes camera\_angles\_sigma=.25 camera\_angular\_velocity\_sigma=.1** *lidar example. Note the lidar\_csv, lidardata, and olidardata arguments.*  |

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Example 6Run of jigsaw parameterized for Kaguya Terrain Camera (TC) images and a final ground control network covering Apollo 15 landing site. Description This last example is of a final jigsaw run of a grounded network. In this example all network adjustments are done, the bundle is converging, the resulting residuals are acceptable, and therefore we are ready to update the camera pointing on the cubes. During the final run we turn on the error propagation flag, this provides the variance-covariance matrix of the parameters, from which uncertainties can be computed. This is valuable if you plan to compute certainties for your update cubes camera pointing (or any kernels resulting from these updated camera pointings). If you want to double check the update was completed, see cathist or catlab (search for 'Jigged'). Command Line **jigsaw fromlist=cubes\_for\_ground\_update.lis cnet=grounded\_relative\_reg.net onet=grounded\_rr\_jig1\_ErrProp.net update=yes file\_prefix=jig1rrEP sigma0=1.0e-10 maxits=10 twist=yes camsolve=accelerations overexisting=yes camera\_angles\_sigma=.25 camera\_angular\_velocity\_sigma=.1 camera\_angular\_acceleration\_sigma=.01 radius=yes point\_radius\_sigma=500 spsolve=positions overhermite=yes spacecraft\_position\_sigma=1000 errorpropagation=yes** Data Files Links open in a new window.

|  |  |
| --- | --- |
| [**jig1rrEP\_bundleout.txt**](assets/ap15/jig1rrEP_bundleout.txt)  | A shorten example of a typical bundleout.txt file produced by a jigsaw run with error propegation. This file was trimmed to hold 10 images, 50 relative points, and all ground points along with their associated detail sections. Bundleout files typically contain all image and point from a network put through jigsaw.  |
| [**Instrument pointing and position tables before jigsaw update**](assets/ap15/tables.txt)  | Table summaries for the InstrumentPointing and InstrumentPosition tables extracted from a cube label.  |
| [**Instrument point and position tables after jigsaw update**](assets/ap15/jigged_tables.txt)  | Table summaries for the InstrumentPointing and InstrumentPosition tables extracted from a jigsaw updated cube label. The InstrumentPointing table was updated due to the camera updates solved for in the bundle (camsolve=accelerations). The InstrumentPosition table was updated due to the spacecraft updates solved for in the bundle (spsolve=positions).  |

 |