Seeing in Three Dimensions: Correlation and Triangulation of Mars Exploration Rover Imagery

Robert G. Deen
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA USA
Bob.Deen@jpl.nasa.gov

Jean J. Lorre
Jet Propulsion Laboratory
California Institute of Technology
Pasadena, CA USA
Jean.Lorre@jpl.nasa.gov

Abstract -*The Mars Exploration Rovers (MER) use a man-in-the-loop system for control in most cases. While capable of some autonomous driving, all arm operations and most drives are planned on the ground. Planning these operations requires a precise knowledge of the terrain surrounding the rover: where are the rocks, the sand, the hazards. This terrain is derived from images taken by stereoscopic cameras.*

This paper describes in detail the middle parts of the ground-based terrain derivation process: correlation, which finds matching points in the stereo pair, and triangulation, which converts those points to XYZ coordinates. The algorithms and free parameters are described, followed by a discussion of the results obtained, the problems encountered, and possible avenues for future development.

Keywords: Stereo, Computer Vision, Correlation, Feature Matching, Triangulation, Terrain, Teleoperation

1 Introduction

Remote operation of a robotic vehicle is challenging under any circumstances. When that vehicle is on Mars, as the Mars Exploration Rovers (MER) Spirit and Opportunity are, it raises "challenging" to a whole new level. Controlling the vehicle requires precise knowledge of the environment in which the rovers find themselves. With no other method of discovering "ground truth", the operators must rely on data returned by the rover itself, specifically on stereo imagery from any of 4 pairs of onboard cameras (the 9th camera is a microscopic imager, not used for teleoperation) [1][2].

The rovers are capable of some autonomous operation [3]. However, the most complex or hazardous drives, and all of the robotic arm (Instrument Deployment Device, or IDD) motions, are commanded by controllers on the ground [4]. These rover planners require precise knowledge of the terrain surrounding the rover (for driving), and even more detailed knowledge of the IDD workspace for arm operations. Additionally, science planners require knowledge of the terrain in order to target observations [5].

This terrain knowledge is derived from stereo imagery acquired by the rover itself. Stereo imagery on MER consists of two images (1024x1024) taken simultaneously by two different cameras (left and right) that are separated by a constant baseline distance [1][2]. The same object seen by both cameras will be at different places in the two images. The difference in these locations is called the disparity, which varies depending on how far away from the cameras the object is.

The process for MER involves four primary steps:

1) Acquisition of the images [2].

2) Correlation of stereo pairs to determine matching features and disparity.

3) Triangulation of the disparity and camera models to determine spatial XYZ coordinates of each pixel.

4) Conversion of the XYZ coordinates to integrated terrain meshes [6].

The second and third steps are described in this paper. The entire processing sequence is managed on the ground by the MIPL (Multimission Image Processing Lab) pipeline [7].

2 Stereo Correlation

The stereo correlation process can be described simply: For each pixel in one image (the reference image), find the location of the pixel that matches it in the other image. Because individual pixels are not unique enough to match, this process happens over a small area around each pixel (7x11 for most cases). By convention, the left image is the reference image and the matching location is found in the right image, so "left" and "right" will be used thus in this paper. "Left" and "right" may be swapped with no change to the process.

Finding the matching pixel is accomplished by searching the right image for an area that best matches the template (the area around the left image's pixel). The central pixel of this area is the desired result. The

difference between the coordinates of the matching pixel in the left and right images (in line/sample terms) is called the stereo disparity, and directly relates to range from the camera (see the triangulation section). Simple enough, but the devil is (as always) in the details.

2.1 Correlation Coefficient

The measure of how well the areas match is called the correlation coefficient ρ [8]:

$$\rho = \sigma_{xy} / (\sigma_x \sigma_y) \tag{1}$$

which is the covariance over the product of the standard deviations of the areas. This reduces to the more computationally efficient:

$$\rho = \frac{n^2 \Sigma xy - \Sigma x \Sigma y}{\sqrt{(n^2 \Sigma x^2 - \Sigma^2 x)(n^2 \Sigma y^2 - \Sigma^2 y)}} \tag{2}$$

where x and y are pixel values from each image, n is the size of the area, and Σx etc. is the sum of x over the area.

The value of ρ ranges from 1 (perfect correlation) to 0 (no correlation) to -1 (perfect inverse correlation). For efficiency we actually minimize $2.0 - q$, where $q = \rho^2$ (to avoid the square root) with the sign of ρ preserved to avoid inverse correlations. This value q is called the correlation quality.

2.2 Geometric Warping of Search Area

The template (left) area is not going to exactly match its partner on the right because of the geometry of the scene. Looking out at a flat plane introduces a trapezoidal distortion; a box in the left becomes a trapezoid when projected into the right. The image may also be rotated or scaled in the general case. In order to compensate for this, the right (search) area is geometrically warped before comparison with the left. This warping is done using a bilinear transform (affine transform plus xy terms):

$$\begin{aligned} x' &= ax + by + c + gxy \\ y' &= dx + ey + f + hxy \end{aligned} \tag{3}$$

In this case, x and y represent image coordinates. This process was inspired by Gruen [9] but is implemented differently: the right area is warped in a separate step before the correlation coefficient is computed.

For efficiency, only a subset of the parameters a-h are used. For most cameras, b, c, f, g are used, which incorporates translation, shear, and trapezoid in x, and translation only in y. This models a pair of epipolar-aligned cameras looking at a flat plane. The front hazcam adds a (scale), which models epipolar-aligned cameras looking at a more general scene.

2.3 Function Minimization

In order to find the matching pixel, the right image is searched for the location that best matches the template. This involves a function minimization process, which attempts to find the maximum quality measure (minimum $2.0 - q$).

The minimization process works iteratively, trying to find values for a-h that create the best match for the area. Minimization is accomplished using the "amoeba" (downhill simplex) method [10], which has the advantage of not requiring partial derivatives at the expense of being a bit slow. It can be conveniently (but somewhat inaccurately) thought of as putting all the parameters in a box and shaking the box, perturbing the parameters, and selecting those perturbations which improve the result.

The final result consists of the c and f terms, which represent the translation, or disparity, of the area. The other parameters are discarded; they exist merely to help find a good match. The q results are thresholded by a quality parameter; this prevents areas that do not match well from being included in the output.

2.4 Seed Points

The amoeba algorithm gives very good subpixel accuracy, but it needs a starting point. It is not good at searching for the pixel throughout the entire right image. This starting point needs to be within a couple of pixels of the solution for amoeba to work reliably.

These starting points, or seed points, are created using several different methods.

The initial seed points are found by running the image through the same 1-dimensional correlator that the onboard flight code uses [3][11]. This code requires that the images be epipolar aligned, meaning that matching features are on the same line in both images, i.e. vertical disparity (in the y direction) is 0. While this would be the case for perfect calibration of the cameras, in practice the vertical disparity ranges up to +/- 3 pixels (sometimes approaching 4). For this reason, a highly downsampled image is given to the 1-D correlator (the flight code also correlates only highly downsampled images). The images are downsampled by 4x, 8x, or 16x, depending on the type.

If the images are not epipolar aligned, initial seed points can instead be generated using a program that estimates disparities based on the camera models and a surface model [12][13]. While this mode is not used in operations, it is being used to create photometry cubes for the pancam [14].

After the initial seed points are obtained, the images are correlated in multiple passes using successive image pyramids, zooming up by a factor of 2 each time until full

size is reached. Thus at each step, amoeba should only have to search +/-2 pixels, and the results are used as a starting point for the next step.

2.5 Additional Details

Before the full-resolution correlation is done, the image is filtered using a small (3x3) low-pass box filter. This reduces the effect of compression noise at the expense of some resolution.

After each pass, a gore-filling algorithm is run. This makes several passes through the entire image, looking for uncorrelated pixels. For each one, it examines all 8 of its neighbors for a good correlation, and picks the highest-quality one to use as a seed point, attempting the correlation again. This allows small gaps, or gores, to be filled in. These gores are often caused by the 1-D correlator failing to find a good match.

3 Image Triangulation

Once the correlation is complete, the next step is to convert those disparities into XYZ locations. This requires that the cameras be calibrated to obtain camera models. These camera models describe the relationship between line/sample coordinates in the image, and the corresponding ray in 3-D space. The object the pixel is seeing may lie anywhere along this ray. MER uses the CAHV family of camera models [12]. Note that the camera baseline (distance between the cameras) is implicitly encoded into the two camera models.

Triangulation is conceptually simple. For each pixel in the left image, find the corresponding one in the right image using the disparity. Take each pixel location and project it through the camera models, giving us two 3-D rays. The XYZ point is where these two rays intersect.

Of course, the rays rarely if ever exactly intersect. So the XYZ point is defined to be midway between the rays at their closest approach to each other. The distance between the rays is called the error, or miss distance.

3.1 Rejecting Invalid Points

The bulk of the XYZ-generation code consists of filtering out bad points from the result of the triangulation. There are 9 different filters which reject points for various reasons. The numeric values are all parameters; the values shown are those used during the mission.

1) Obviously, if there is no correlation match, the pixel is rejected.

2) The absolute line disparity must be < 4 pixels.

3) The line disparity must be within 0.75 pixels of the average line disparity in a 51x51 pixel area. This allows the epipolar calibration to slowly drift over the image (as it does for the hazcams).

4) The XYZ intersection must be computable (i.e. the rays are not perfectly parallel).

5) The absolute miss distance must be less than 0.05 meters. This value needs to be somewhat less than the physical camera baseline.

6) The miss distance divided by the range to the pixel must be less than 0.005 meters/meter. This tightens up the allowed error up close where the models are more accurate.

7) The computed Z value must be within certain limits (-20,+20 meters for Opportunity, -20,+40 meters for Spirit).

8) The rays must not diverge.

9) The range must be within 1000 times the camera baseline.

4 Results

Perhaps the best way to analyze the results is to note that this system has been used operationally by two rovers on Mars for well over a year now with very few problems. The problems that do exist have sufficient workarounds for operations to proceed.

4.1 Speed

Execution speed is a major issue. For the Mars Pathfinder mission (1997), a 256x256 image pair took an hour or more to correlate. MER images at 1024x1024 have 16 times as many pixels, and the runtime is around 2-4 minutes (on a dual 2.4GHz Pentium-class box running Linux). While some of the speed improvement is due to faster hardware, the bulk of it is due to algorithm improvements and parameter tuning.

Four correlator parameters have a huge effect on runtime. Almost always, we trade off speed for quality. A larger template size gives more accurate results, at the expense of both resolution and speed. It would better in most cases to use all 8 geometric warp parameters (a-h), but more degrees of freedom imposes a significant speed penalty. The gore-filling algorithm is rather inefficient, so large numbers of gore passes greatly increase the time while providing fewer and fewer results, a case of diminishing returns. Finally, a tolerance parameter to amoeba tells it how hard to work, which directly relates to the precision of the results, at a huge speed penalty.

The parameter set ultimately used was the result of heuristic tradeoff studies performed over time, before and to some extent during the mission, and represents a good compromise between speed and performance.

4.2 Accuracy

Correlators are quite difficult to test in any real-world scenarios. What exactly is a "correct" correlator output? It is difficult if not impossible to get large-scale accurate ground truth from a scene for quantitative comparison.

End-to-end tests of derived XYZ's compared to surveyed points are possible, and have been done. Pre-launch tests using survey targets indicate less than 1% error at 20m range for the navcams, and less than 10% at 20m range for the hazcams. Unfortunately, these are only spot checks. They validate the range, but survey targets are easy to corrrelate due to high contrast, and are thus relatively unaffected by noise. They don't say a lot about overall correlator performance.

Another test method which has been used is to generate a "pseudo-left" image by projecting the right image back through the disparity map. The result should match the left image. Several tests have shown good matches, but radiometric differences make quantitative comparisons difficult (especially on earth-based tests where high temperature introduces a large uncorrected dark current). It is also hard to see subpixel errors this way.

Similarly, the disparity results have been spot-checked manually, but this is very time consuming, and again subpixel errors are hard to see.

Using an artificially-generated imagery is possible, and gives you perfect ground truth, but that's not very representative of real-world performance, and was not done.

Relative comparison between multiple correlator runs was very useful in tuning the parameters. One slow, high-quality run using uncompressed images was checked as much as possible and then assumed to be correct. The effects of other runs with different compression and parameters were then compared, with tradeoffs made for speed vs. accuracy.

The totality of all these checks showed that the correlator was working within the limits needed for operations. Arm operators need about a ~1cm accuracy [15] end-to-end (including mechanical placement), and this has been achieved. It is interesting to note that there is some evidence the geometry of the cameras has drifted over time during the mission, but as that does not affect the correlator, it is outside the scope of this paper.

4.3 Effects of Compression Noise

Nevertheless, there are known errors in the output. The primary source of such errors are compression artifacts. These artifacts result in a "patchiness" to the correlator output, due partially to small errors affecting an area as large as the template (since bad pixels are included in many different templates), and partially due to an anti-integer bias discussed below. See Figure 1. This patchiness is reflected in anomalous spikes in the final terrain mesh. This places an upper limit on the amount of compression allowed for images used for arm motion (especially) and driving. Typically 4 bits per pixel is used for images intended for arm motions, with no less than 1 bpp used for driving. See Table 1.

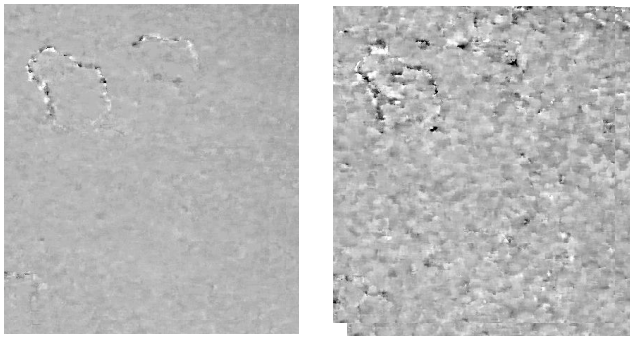

Figure 1. Line disparity of same image compressed at 9 bits per pixel (left) and 0.5 bpp (right) showing patchiness

Table 1. Compression Effects on Average Range Error

Comp (bpp)	Noise (DN)	SNR	Error (m)
9.64 (lossless)	(n/a)	(n/a)	0.018406
6.00	10.04795	193.6	0.018658
3.00	31.14266	62	0.019497
2.00	56.66446	52	0.020221
1.00	103.0099	27.1	0.021302
0.75	117.2271	22.8	0.027388
0.50	142.216	18.5	0.031038
0.25	172.4457	14.2	0.044052
0.125	206.131	11.3	0.05976
0.0624	239.2385	9.6	0.075553
0.0311	269.4861	8.4	0.15759

4.4 Anti-Integer Bias

Careful analysis of the effects of noise showed a curious phenomenon. The subpixel-level disparity results of the correlator were slightly biased away from integer values, toward half-integer values. The more compression noise, the more pronounced this effect. This is best illustrated in a histogram of disparity values over an image. Figure 2 contains histogram plots of line disparity values (actually disparity plus line number) for an image correlated against itself, with varying amounts of Gaussian noise introduced into the second image. With zero noise, the result is a single spike at each integer value (tick marks in the plot). As noise is introduced, the histogram should assume a Gaussian shape at each integer, becoming wider as the noise increases until they eventually merge. Instead, as noise is added, the values move away from the integers and cluster at the half-integers.

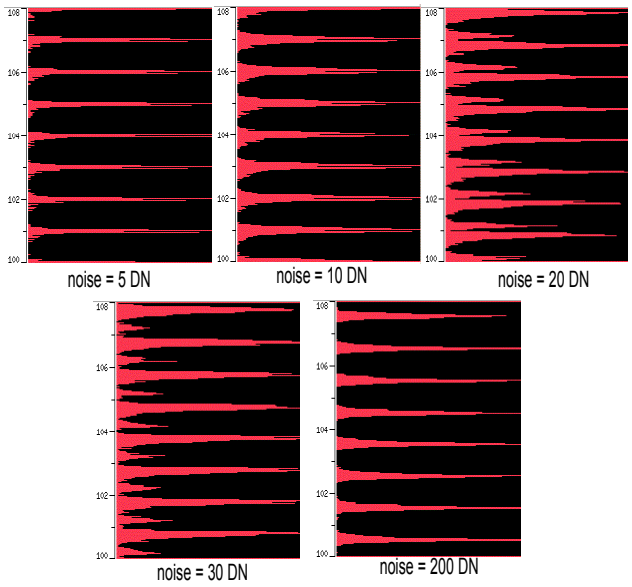

Figure 2. Anti-integer bias due to compression noise

This effect is due to the bilinear interpolation performed as part of the geometric warp. Interpolation causes a slight smoothing effect on the right-side image, reducing the noise somewhat. At integral disparities, no interpolation occurs since the raster positions of both areas exactly match. At half-integral positions, each pixel in the right area is the average of four pixels. This noise reduction "pulls" the correlator towards the half-integer points, where the noise is the least. Since the noise is not the same in both areas, the smoother interpolated data matches the spiky template better than the different spikes in the non-interpolated data.

This half-integer bias, which exists only in the presence of noise, is believed to be the ultimate source of the "patchiness" of the data. This effect is also exacerbated by "fast" correlation parameters. More degrees of freedom means more overall interpolation, reducing the relative effect of the translation terms (so there is some interpolation even at the integer disparities). Also, larger windows improve the signal-to-noise ratio, reducing the effect.

4.5 Inappropriately Correlated Areas

The last major source of error is also the most troublesome operationally. That is the inclusion of inappropriate results - pixels that should not have a match, but do. This typically manifests itself as random pixels in the sky, around edges of rocks, and in the distance where the features become indistinct. These should not be correlated but often are due to image noise appearing to be a feature. Increasing the correlation quality threshold helps somewhat to eliminate those pixels, but at the expense of eliminating many good pixels as well. These bad pixels often result in "walls" or so-called "hanging chad" in the final terrain mesh. The effect of this has been somewhat mitigated by completely masking off the area near the horizon, but more work is needed to properly solve this problem.

5 Future Directions

The correlator has worked well for MER operations, but there is room for improvement. Future missions such as MSL (Mars Science Laboratory) will have more strict requirements and require more robustness. The much longer drives planned for MSL will require better results in the distance; horizon masking is not an option. In addition, reducing the spikes and patchiness will allow the correlation results to be used for visualization and science analysis in addition to just operations. Some of the ideas are listed below.

Checking the left->right correlation by doing an inverse right->left check is an obvious method to help avoid bad pixels. This was implemented for MER but not used as the cost/benefit ratio was not high. That can probably be improved.

A method needs to be implemented to avoid correlating in the sky and other undifferentiated areas (such as featureless sand and areas compressed too much). This may be based on an interest operator (does the scene have enough detail) combined with some heuristics to help find the horizon.

Better ways of dealing with compression noise are much needed. The anti-integer bias may be reduced by using something other than bilinear interpolation. Recognizing the noise would also help to avoid bad correlations, or perhaps the correlation quality threshold could be dynamically adjusted.

Areas where there are different depths within the window, such as the edges of a rock (combined with the background behind it), are not correlated well. This imposes some operational constraints, as these edges are thus off-limits for arm operations. An adaptive window, shaping itself to avoid the multi-depth areas, could allow good results much closer to the edges of rocks.

6 Conclusion

While future work is required in certain areas, the correlation and triangulation algorithms used by MER have proven to be accurate and reliable enough for well over a year of operations in a very challenging environment. Image compression noise is the biggest factor in obtaining quality results. Careful calibration of the system is also needed for accuracy.

Finally, Figure 3 shows an image and its XYZ counterpart derived using the processes described in this paper. The lines represent areas of constant XYZ value, showing the coordinate grid lines. This kind of product has been critical to the success of MER operations.

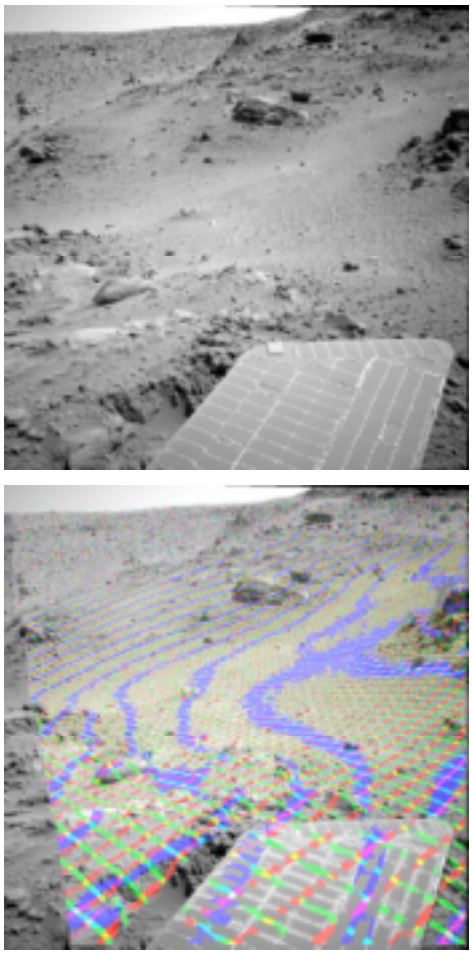

Figure 3. Spirit Sol 409 navcam image (top), XYZ results overlayed (bottom). Red/green/blue lines represent constant XYZ coordinate values (respectively) at 0.1m intervals.

7 Acknowledgements

The work described in this paper was performed at the Jet Propulsion Laboratory, California Institute of Technology, under a contract with the National Aeronautics and Space Administration. Many thanks to the entire MER team, without whom there would be no reason to do this work. Special thanks to Gary Yagi for his insights on correlation.

References

[1] J.N. Maki, *et al*, "The Mars Exploration Rover Engineering Cameras", *J. Geophysical Research - Planets*, Vol 108 No E12, 2003.

[2] J.N. Maki, "Operation and Performance of the Mars Exploration Rover Imaging System on the Martian Surface", submitted to 2005 IEEE International Conf. on Systems, Man, and Cybernetics, Waikoloa, HI.

[3] S.B. Goldberg, M.W. Maimone, L. Matthies, "Stereo Vision and Rover Navigation Software for Planetary Exploration", Proc. 2002 IEEE Aerospace Conf., Big Sky, MT, March 2002.

[4] S. Maxwell, B. Cooper, F. Hartman, J. Wright, J. Yen, C. Leger, "The Best of Both Worlds: Integrating Textual and Visual Command Interfaces for Mars Rover Operations", submitted to 2005 IEEE International Conf. on Systems, Man, and Cybernetics, Waikoloa, HI.

[5] J.S. Norris, M.W. Powell, M.A. Vona, P.G. Backes, J.V. Wick, "Mars Exploration Rover Operations with the Science Activity Planner", Proc. 2005 IEEE Conf. on Robotics and Automation, Barcelona, Spain, April 2005.

[6] J.R. Wright, A. Trebi-Ollenu, J. Morrison, "Terrain Modelling for In-Situ Activity Planning and Rehearsal for the Mars Exploration Rovers", submitted to 2005 IEEE International Conf. on Systems, Man, and Cybernetics, Waikoloa, HI.

[7] D. Alexander, P. Zamani, R. Deen, "The MIPL Pipeline: Automated Generation of Image Products for Mars Exploration Rover Mission Tactical Operations", submitted to 2005 IEEE International Conf. on Systems, Man, and Cybernetics, Waikoloa, HI.

[8] R.C. Gonzales, P. Wintz, *Digital Image Processing, Second Edition*, Addison-Wesley, Reading, MA, 1987

[9] A.W. Gruen, E.P. Baltsavias, "Geometrically Constrained Multiphoto Matching", *Photogrammetric Engineering and Remote Sensing*, Vol 54, No 5, May 1988.

[10] W.H. Press, B.P. Flannery, S.A. Teukolsky, W.T. Vetterling, *Numerical Recipes in C, The Art of Scientific Computing*, Cambridge Univ. Press, Cambridge, 1988.

[11] L. Matthies, "Stereo Vision for Planetary Rovers: Stochastic Modeling to Near Real-Time Implementation", *Int. J. Computer Vision*, Vol 8 No 1, 1992.

[12] T.E. Litwin, J.N. Maki, "Imaging Services Flight Software on the Mars Exploration Rovers", submitted to 2005 IEEE International Conf. on Systems, Man, and Cybernetics, Waikoloa, HI.

[13] R. Deen, O. Pariser, J. Lorre, "Creation of Surface-Based Image Mosaics for MER/FIDO", JPL Information Technology Symposium Poster Session, Pasadena, CA, Nov. 2002.

[14] J.M. Soderblom, J.F. Bell, R.E. Arvidson, J.R. Johnson, M.J. Johnson, F.P. Seelos, "Mars Exploration Rover Pancam Photometric Data QUBs: Definition and Example Uses", *Eos Trans. AGU*, Vol 85 No 47, 2004.

[15] E.T. Baumgartner, R.G. Bonitz, J.P. Melko, L.R. Shiraishi, P.C. Leger, "The Mars Exploration Rover Instrument Pointing System", Proc. 2005 IEEE Aerospace Conf., Big Sky, MT, March 2005.