CAMERA RESPONSE SIMULATION FOR PLANETARY EXPLORATION

Dr. Richard Madison[(1)], Marc Pomerantz[(2)], Dr. Abhinandan Jain[(3)]

[(1)] *Jet Propulsion Laboratory, M/S 125-209, 4800 Oak Grove Drive, Pasadena CA 91109, USA, Madison@robotics.jpl.nasa.gov*
[(2)] *Jet Propulsion Laboratory, M/S 301-270, 4800 Oak Grove Drive, Pasadena CA 91109, USA, Marc.I.Pomerantz@jpl.nasa.gov*
[(3)] *Jet Propulsion Laboratory, M/S 198-235, 4800 Oak Grove Drive, Pasadena CA 91109, USA, Abhi.Jain@jpl.nasa.gov*

ABSTRACT

Current and future Mars rover missions have spurred the development of the ROAMS rover simulator to provide realistic, closed-loop simulation with onboard rover control software. A key component of the simulator is the ability to model camera response and synthesize images for onboard vision-based control algorithms. ROAMS image synthesis incorporates models to support user specifiable terrain geometry and texture, CAHV and CAHVOR camera models, and realistic shadowing. This paper describes these models as well as results from verification tests to measure the accuracy of the models.

1. INTRODUCTION

NASA's planetary surface exploration missions, such as the Mars Exploration Rover (MER) mission with the Spirit and Opportunity rovers and the upcoming Mars Science Laboratory (MSL), use ground testbeds for software development, test and operations. NASA's Mars Program is developing and validating the Rover Analysis, Modelling and Simulation (ROAMS) rover simulator [1,2] to complement the hardware testbeds for these surface missions.

ROAMS provides interfaces to close rover control loops ranging from low level motor control to locomotion estimation/control to vision-based autonomous navigation. A number of rover control algorithms are based on computer vision, including path planning, hazard avoidance, instrument pointing and placement, target tracking, and visual odometry. Such algorithms interpret an input image stream and generate commands that, in turn, affect the contents of the image stream. To support closed-loop development and testing of such vision-based algorithms, ROAMS incorporates facilities that synthesize images from the perspective of virtual cameras attached to simulated rovers.

This paper describes the implementation and verification of image synthesis algorithms in ROAMS. We describe the imaging features that the synthetic images model and how ROAMS implements these features. We also describe the method used to verify the geometric accuracy of the synthetic images and the verification results.

2. IMAGING FEATURES

The primary requirement on the fidelity of the ROAMS-synthesized imagery is that it be sufficient to test the functionality of computer vision algorithms in the rover control loop. The next important requirement is that ROAMS be able to close the loop in real-time between the simulation and the control algorithms. Photo-realism beyond what the vision algorithm needs adds little value and can instead reduce the simulation speed. Hence, careful attention has been paid to prioritizing and selecting the imaging features that need to be simulated.

Table 1 shows the imaging features considered for inclusion into ROAMS image synthesis. They are classified according to their difficulty of implementation and their benefit to algorithm testing. We identified surface texture, camera model with radial distortion, and shadowing as the most important features to implement. This reflects the fact that the majority of rover-based, visual navigation algorithms tested at JPL rely on stereo processing, which is very sensitive to texture, shadows, and camera model. The three camera models, CAHV, CAHVOR, and CAHVORE [3] are the models used within the NASA machine vision community and on the MER rovers. Other effects from the table will be implemented when needed by future algorithms.

3. IMAGE SYNTHESIS

The implementation of ROAMS image synthesis consists of separate components to model textured terrain, real world cameras, and realistic shadowing.

Table 1. Imaging features sorted by cost and benefit

		Benefit		
		High	Moderate	Low
Complexity	Low	Pinhole camera (CAHV) Lighting angle and shadows Pixel noise (shot, readout, …) Vignetting	Global gain Pixel response Dark current Fine lens fog/dust, Pixel dropout Saturation, blooming	Pixel response Permanent hot pixels Large detector dust Fast flush & frame transfer noise Single event hot pixels
	Moderate	Radial lens distortion (CAHVOR) Moving entrance pupil (CAHVORE)	Motion blur Ray tracing specularity, reflectivity, turbulence, transmittance IR camera Large lens dust Atmosphere fog/dust	Depth of field, focal distance Lens point spread function Color CCD: single chip or multi-chip Wavelength sensitivity Spherical aberration, Chromatic aberration Unsynchronized stereo cameras
	High	Textures with realistic granularity 3D textures (bump maps) @ 5mm cell size Opposition effect	Fixed/repeating pattern noise from EM interference Volatiles on lens Lens flare, Aliasing, radiosity Cable reflection ghosts, analog antialiasing, pixel jitter	

3.1 Terrain Model and Texture

The first critical component of a synthesized image is a suitable scene. Computer vision algorithms for navigation require realistic terrain geometry with targets to track and obstacle features to detect. Furthermore, many computer vision algorithms rely on image correlation and thus require sufficient scene texture.

ROAMS uses terrain modelling software called SimScape [4] to manage terrain models. SimScape represents terrains as 2.5D height fields, which can be fully specified in a data file or specified with an analytical function. SimScape also allows the user to specify texture maps for both. The user can select terrain geometry and texture detail and realism as needed to test an algorithm. Textures may be user provided or generated from terrain surface properties.

Dspace [1], the ROAMS graphics rendering engine, uses OpenInventor [5] and OpenGL [6] to render imagery of SimScape's terrains. Dspace converts a terrain into an OpenInventor *scenegraph* with 3D vertices defined by the terrain's height field. OpenInventor converts a scenegraph into an image utilizing the OpenGL pinhole camera model and, polygon rendering. Polygon rendering is significantly faster that typical ray-tracing.

3.2 CAHV Camera Model

The CAHV camera model is equivalent to thestandard linear photogrammetric model for a pinhole camera. It is useful for very small field of view cameras and as a building block for more complex camera models. The CAHV model consists of four 3-vectors: *C*, *A*, *H*, and *V*. Vector *C* gives the location of the pinhole. Vector *A* gives the camera axis, defined as the normal to the image plane. Vector *H* encodes the horizontal axis of the image plane (*H*'), the coordinate (*Hc*) of the image column at the optical centre of the image plane, and the horizontal focal length (*Hs*) of the camera, in pixels. Vector *V* encodes corresponding information (*V*', *Vc*, *Vs*) in the vertical direction. The angle (theta) between horizontal and vertical vectors *H'* and *V'* is about 90°. Non-orthogonal *H'* and *V'* generally represent an attempt to compensate for distortion that CAHV vectors cannot directly model. Image dimensions are supplied along with the CAHV model.

ROAMS can synthesize images for a CAHV-model camera. First, Dspace renders an image using information in the CAHV vectors to define the position, orientation, and field of view of the OpenGL camera, as well as the dimensions of the resulting image. OpenGL's perfect pinhole camera has its optical centre at the centre of the image and has perpendicular image plane axes. To accommodate an optical centre at arbitrary (*Hc,Vc*), ROAMS renders an oversized image and then crops it to the desired, off-centred, CAHV sub-image.

OpenGL does not support non-perpendicular image plane axes. This is reasonable, because in real cameras, image plane axes are nearly perpendicular. If CAHV-specified axes *H'* and *V'* are not exactly perpendicular, ROAMS replaces *V'* with a vector perpendicular to vectors *A* and *H'*. These orthogonalized axes are then suitable for specifying OpenGL camera pointing.

3.3 CAHVOR Camera Model

The CAHVOR model describes a camera with radial lens distortion about the lens axis. In addition to the CAHV parameters, it includes 3-vectors *O* and *R*. Vector *O* is the optical axis of the lens, slightly different from vector *A* due to imperfect lens mounting. A ray from a point in space, passing through the pinhole, bends at the pinhole according to a function of the angle between the ray and optical axis *O*. The function is a polynomial whose coefficients are stored in vector *R*.

ROAMS synthesizes images from CAHVOR-model cameras in two steps. First, it generates a CAHV image as described above, complete with approximated vertical image plane vector and OpenGL's accelerated, pinhole camera rendering engine. Second, it warps that image to introduce the distortion specified by the *O* and *R* vectors as well as the effects of non-perpendicular axes.

A ray of light passing through the camera pinhole projects to one image pixel under the CAHV model and to a different pixel after bending under the CAHVOR model. ROAMS creates a map of these pairs, identifying the "equivalent" CAHV and CAHVOR pixels representing the same, incoming ray of light. After rendering a CAHV image, ROAMS fills each CAHVOR image pixel by copying the equivalent pixel from the CAHV image, using bilinear interpolation when equivalent coordinates fall between CAHV pixels. The CAHV model uses the modified vertical image plane axis but the CAHVOR model does not, so the mapping undoes any error introduced in the CAHV imaging by the orthogonalized axes.

CAHV imaging could be made to support non-perpendicular image plane axes by performing a warping step just as CAHVOR imaging does.

3.4 CAHVORE Camera Model

A third model, CAHVORE, describes more general cameras including those with fisheye or otherwise wide field of view lenses. An additional vector, *E*, describes the apparent motion of the camera entrance pupil. ROAMS does not yet support this camera model.

CAHV and CAHVOR models obtained via calibration usually are accurate at the image centre but not at high angles, such as in image corners. Because ROAMS faithfully follows the CAHVOR model in image synthesis, the images are often physically unrealistic at the corners. The planned implementation of the CAHVORE model should eliminate this behaviour.

3.5 Shadowing

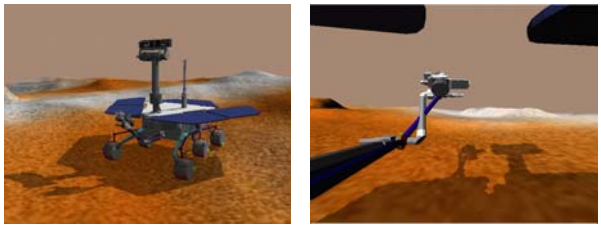

Fig. 1. A ROAMS rover casting a shadow on the ground and on its own arm, as seen from a chase camera (left) and a forward hazcam (right).

Shadows can impact computer vision algorithms by obscuring parts of the image, changing the scene appearance over time, indicating the presence of large obstacles, and adding false edges to the scene. Dspace, the ROAMS visualization engine, uses two methods to generate realistic scene shadows – one for terrain shadows (which change slowly) and the other for the rover itself.

The first method utilizes the Povray renderer [7] to render a shadow map of the terrain. ROAMS creates a Povray Scene Description Language file that describes the sun position, terrain height field, camera position and material information. Povray uses this input to render a shadow map, which is then combined with the terrain's original colour texture image. Dspace uses this augmented texture to render the terrain, now with terrain shadows. To save time in the simulation, terrain shadows need not be rendered at every simulation time step. This component generates very realistic shadows for the terrain, but it does not include shadows from the rover.

The second method, the Z-buffer shadowing algorithm [8], generates near real-time rover shadows, and terrain shadows in close proximity to the rover, at every simulation time step. This algorithm was implemented inside the Dspace rendering engine and was chosen because it is relatively simple to code yet general enough to allow objects to "self-shadow" if the object geometry and light position so dictate.

The Z-Buffer algorithm is a two-pass rendering method that renders a scene first from the light's point of view and then from the camera's point of view, saving the

associated Z-buffers (depth at each pixel) from each rendering pass. Using the Z-buffer from the camera view, each camera-view pixel is transformed into pixel coordinates and a depth value (*Ez*) as seen from the light's viewpoint. If the light-view Z-buffer value (*Lz*) at those pixel coordinates is less than depth *Ez*, then that camera-view pixel is shadowed by something closer to the light. Shadowed pixels are darkened according to a pre-determined shadowing factor.

4. IMAGING VERIFICATION

We tested our synthetic images to evaluate the geometric accuracy of their representation of the rover's 3D world. These tests evaluated whether the images synthesized by ROAMS are geometrically consistent with the CAHV and CAHVOR models used to generate them. This consistency is critical, because computer vision algorithms that use stereo ranging expect to use these camera models (modulo noise) to recover the 3D structure of the world from the images.

4.1 Verification Method

We used a two-step process to evaluate the accuracy of our camera modelling. We created a set of synthetic images from a variety of camera models for different rover cameras. These images were put through a "camera calibration" process to estimate the camera models represented by the images. The estimated camera models were then compared with the original models. The closeness of the match is a measure of the geometric accuracy of the ROAMS camera image synthesis algorithms.

4.1.1 Cameras

We selected 22 cameras (Table 2) as a representative set of camera models for this verification study. This set includes hazcams (belly-mounted, fisheye cameras used for hazard detection), navcams, and pancams (mast-mounted cameras used for navigation and science) on testbed and MER rovers. We extracted CAHVOR parameters for these cameras from existing CAHVOR and CAHVORE data for the testbed rovers and from the headers of MER image files.

Table 2. Camera IDs (left,right) for the 22 cameras used in testing

	Rear (bogey) hazcams	Front (rocker) hazcams	navcams	pancams
Fido	1,2	4,3		
Rocky8	17,18	20,19	21,22	
MER A	5,6	8,7		
MER B	9,10	12,11	15,16	13,14

For each camera, we made a CAHV model and a CAHVOR model. Most of the original model files used the CAHVORE model. We omitted the *E* vector since ROAMS' support for fisheye effects is still under development.

4.1.2 Camera Calibration

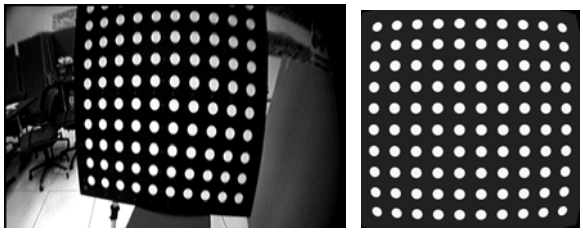

Fig. 2. The calibration target is a board with a 10x10 grid of dots. The 3D location of the real target, left, is measured using a surveying station and reflectors on the target corners. ROAMS knows the 3D locations of the corners of the simulated target board, right.

For each camera model (2 models each from 22 cameras), we used the JPL camera calibration procedure to estimate model parameters. The calibration procedure works as follows.

- Place a calibration target (fig. 2) in front of the camera, take an image, and use a surveying station to record the 3D coordinates of the corners of the target. Repeat this for six target positions. For our test, synthesize the images and report the true corner coordinates.

- Identify the 2D image coordinates of the centres of the dots on the target, and match them to 3D world coordinates based on the known geometry of the target and the measured 3D corner coordinates.

- Solve for the CAHV or CAHVOR parameters that best explain the matched 2D and 3D coordinates.

- If using the CAHVOR model, repeat the previous two steps, this time considering the radial distortion caused by the estimated camera model when determining the centres of the dots on the target. Solving again for the CAHVOR model using these refined dot centres refines the model.

- Project the 3D dot coordinates using the estimated camera model, and compare them against the observed 2D coordinates. The distances ("residuals") between observed and projected 2D dot positions will be small if the data matches the model well.

4.2 Residuals

The final step of calibration calculates "residuals". These are the distances between where calibration target dots appear in the synthesized images and where they should appear, according to the estimated model and the known target geometry. Fig. 3 shows that RMS residual hovers around 1/30 of a pixel for both CAHV and CAHVOR. This generally is regarded as a good match. The worst error is 0.14 pixels, suggesting that the synthetic images are geometrically consistent with the recovered models.

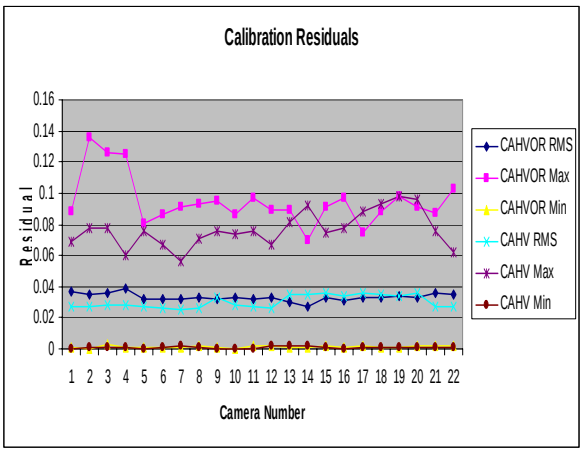

Fig.3. Minimum, maximum, and RMS calibration residuals for each test camera. Sub-pixel residuals indicate that the images are consistent with CAHV and CAHVOR models.

4.3 Other comparisons

This section examines in more detail the difference between the camera models used to synthesize images and those estimated by calibration from the synthesized images. Table 2 (above) identifies the camera numbers that appear in the plots.

4.3.1 Optical Center of Image Plane (*Hc,Vc*)

Figs. 4 and 5 suggest that ROAMS rounds (*Hc,Vc*) to half-pixel accuracy and then synthesizes images that very accurately model the new values. Indeed, ROAMS does fail to adjust (*Hc,Vc*) after rounding the dimensions of the CAHV image that it will render.

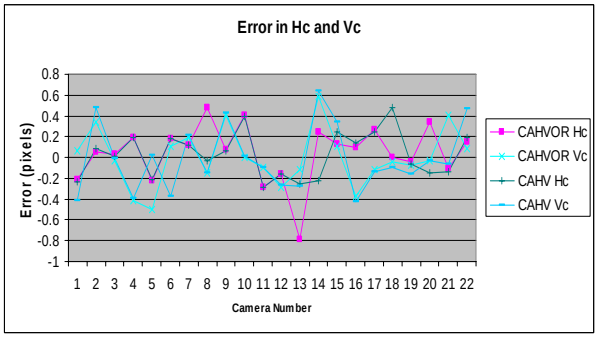

Fig. 4. Error between actual and recovered (Hc,Vc) is distributed rather randomly in the range of ±0.5 pixels.

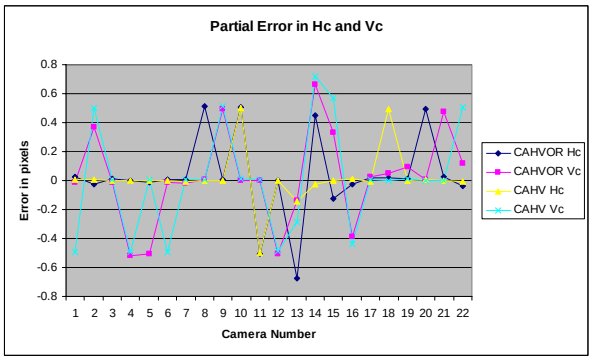

Fig. 5. Error between recovered (*Hc,Vc*) and actual values that have been rounded to the nearest half pixel.

4.3.2 Focal Lengths (*Hs,Vs*)

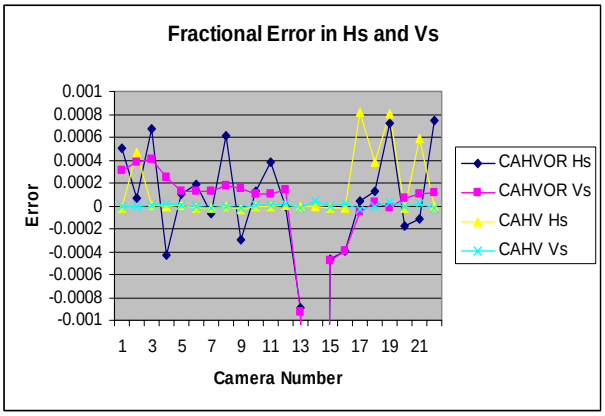

Fig. 6. Errors in focal lengths (*Hs,Vs)*, shown as fractions of the actual *Hs* and *Vs*. Actual values and their absolute errors vary by an additional order of magnitude across cameras.

The most notable feature of error in recovered focal length (fig. 6), is the apparently poor recovery of *Vs* for CAHVOR cameras 13-16 (MER B mast cameras). This is actually not significant, as discussed below under Radial Distortion.

For the remaining CAHVOR cameras, *Vs* error is a consistent underestimate of 0.04% or less. This causes offsets along the top and bottom image borders of about 0.1 pixels in the worst case (cameras 1-4) and at most half that in all other cameras. CAHV *Vs* error is much smaller – a 0.03 pixel offset in the worst case. As discussed below, error in recovered *R* accounts for most of the error under CAHVOR.

The large error in *Hs* is most likely due to a second instance of rounding in the code, when calculating the image aspect ratio. OpenGL uses the rounded aspect ratio instead of *Hs* to calculate horizontal field of view. This speculation is supported by the fact that the MER cameras (5-16) under the CAHV model, being well centred and having aspect ratio of 1.0, have little to no rounding and very low error in *Hs*.

4.3.1 Theta

Theta is the angle between the camera's horizontal and vertical image-plane-axes. It is almost exactly 90° for well manufactured cameras, but calibration may perturb it to allow a model to better fit noisy data. ROAMS CAHV imaging forces theta to 90°, and calibration recovers theta of 90° to float precision, suggesting that ROAMS very accurately models the incorrect theta. In the set of test cameras, the maximum error in theta of 0.17° offsets the top and bottom rows by 0.7 pixels on a 640x480 image or 1.49 pixels on a 1024x1024 image. ROAMS CAHVOR imaging models arbitrary theta, and calibration recovers the original theta to within 0.002°, offsetting the top or bottom rows of the image by only 0.02 pixels in a 1024x1024 image.

4.3.1 Camera Position

Vector *C* gives the 3D location of the camera pinhole. Fig. 7 shows the error in recovered *C*. For CAHV models, error is less than 0.06mm in all cases, the worst offender being at the limit of float precision.

Error in CAHVOR pinhole locations is a fraction of a millimetre in all cases. The error in *C* is roughly in the direction of camera axis *A* for all but the MER mast cameras, which had virtually no error. The error in C results from *Vs* being consistently underestimated, corresponding to an overestimated field of view. The calibration process overestimates the angle occupied by objects in the overestimated field of view. The calibration target has known dimensions, so its overestimated angular size can only be explained by underestimating distance to the target.

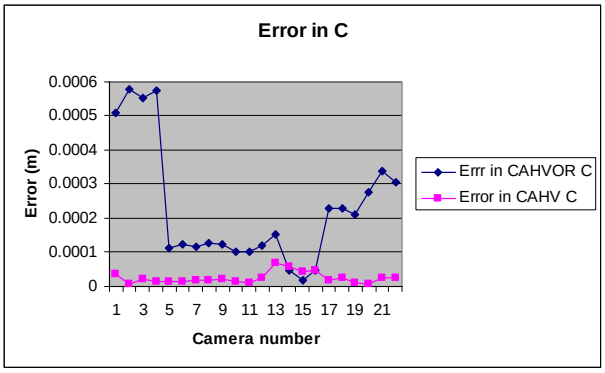

Fig. 7. Error in recovered camera position, vector C.

4.3.1 Image Plane Axis

Vector *A* is the normal vector to the camera's image plane. Much of the angular error in *A*, shown in fig. 8, results from error in *Hc* and *Vc* observed in fig. 5. CAHVOR camera 13's error is a right pan of 0.012°, giving an offset of 0.75 pixels, comparable to the observed *Hc* error. CAHV camera 13's error is largely a 0.004° tilt, corresponding to a 0.25 pixel vertical offset, compensating for the error observed in *Vc*. CAHVOR Camera 2 recovers about 0.022° of extra downward tilt, equivalent to a vertical offset of 0.13 pixels in the image. Observed *Vc* is off by 0.13 pixels from the nearest half-pixel mark. CAHVOR Camera 19 recovered an extra tilt up of 0.0135° or 0.09 pixels, comparable to observed error in *Vc*. However, there is some additional error source. CAHVOR camera 15 recovers an excess left pan of 0.011°, equivalent to 0.23 pixels of left shift, but we observe *Hc* of half that.

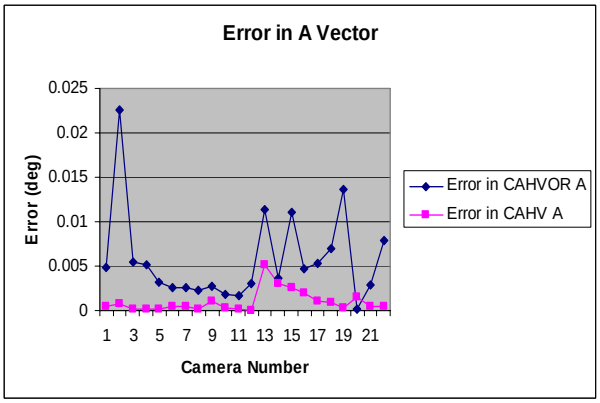

Fig. 8. For CAHV imaging, the maximum angular error in *A* is 0.005°, corresponding to a 0.3 pixel shift in a 1024x1024 image. Error under CAHVOR imaging is much worse.

4.3.1 Optical Axis

Vector *O* is the axis of the camera lens, which is not quite equal to the forward axis of the image plane. Fig. 9 shows that recovery of *O* was particularly poor for the MER mast cameras. This is an artefact of the fact that those cameras have very little distortion, so vector *O* can float considerably without having much effect on image appearance. For camera 15, the error in *O* produces only a 0.03 pixel offset at the edge of the image, based on calculations in [9]. However, camera 2 suffers a 0.3 pixel shift at image edge, because its high radial distortion is affected more by the lens orientation.

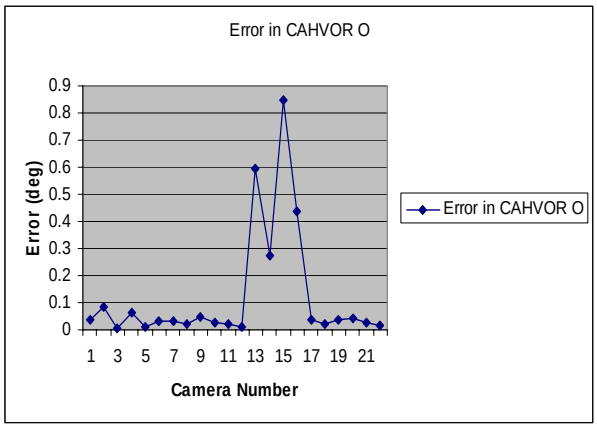

Fig. 9. The *O* vector used to synthesize images and the one estimated by calibration agree to within 0.1° for all cameras except the MER mast cameras (13-16).

4.3.1 Radial Distortion

Vector *R* gives the coefficients of the polynomial that describes radial distortion in a CAHVOR camera. Figs. 10 and 11 show the error in the three terms of *R* as absolute and fractional error values.

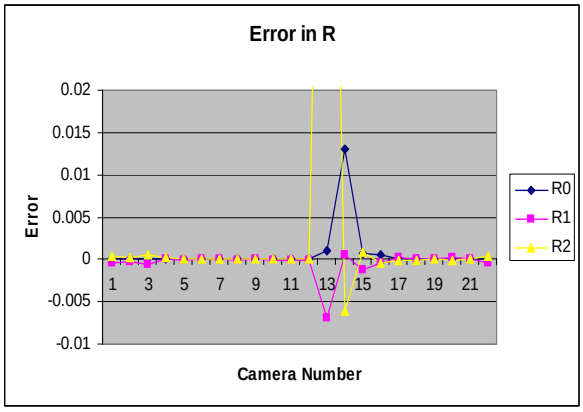

Fig. 10. Absolute error in the recovery of the camera radial distortion parameter *R*.

Absolute and fractional errors in *R1* and *R2* are well below 1% for all but the MER mast cameras (13-16). Errors for the mast cameras have little relevance, because the mast cameras have little distortion to begin with. Based on [9], we find that the *R1* and *R2* terms provide at most .02, .00, .09, and .05 pixel errors, respectively, for the four mast cameras, well below the errors introduced by other elements of the model.

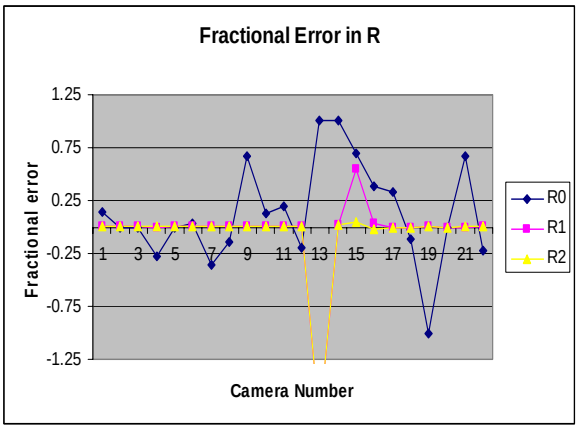

Fig. 11. Fractional error in the recovery of the camera radial distortion parameter *R*.

Element *R0* is more interesting. Correct values are on the order of 10E-5 or 10E-6 for all but the MER mast cameras. Calibration solves for the term (1+*R0*), so errors of 100% in *R0* make very little difference. The worst recovered *R0* (excluding MER mast cameras) produces a pixel offset of at most .005 pixels.

The MER mast cameras have larger absolute errors in *R0*, but these have little impact. In the CAHVOR model, (1+*R0*) couples with *Hs* and *Vs* when there is little distortion, as in the mast cameras, so a rise in one can be offset by a drop in the other. The calibration code recovers small *R0* values, so for the MER mast cameras, the only cameras whose correct models had larger *R0*, the calibration code chose a small *R0* and adjusted *Hs* and *Vs* accordingly. The error in the products, *Hs*(1+R0)* and *Vs*(1+R0)*, for the actual and recovered models is 0.02% at most.

5. CONCLUSIONS

We implemented CAHV and CAHVOR image synthesis models in ROAMS and tested their accuracy over a range of camera models. We used standard camera calibration procedures to estimate the camera models that produced the images, and we compared these with the original models. Excellent calibration residuals showed that the images were geometrically consistent with CAHV and CAHVOR models. Errors between the original and estimated models were small.

ROAMS imaging is currently being used to evaluate vision-based algorithms. Figure 12 shows the range map extracted by stereo correlation algorithms from ROAMS stereo pair images.

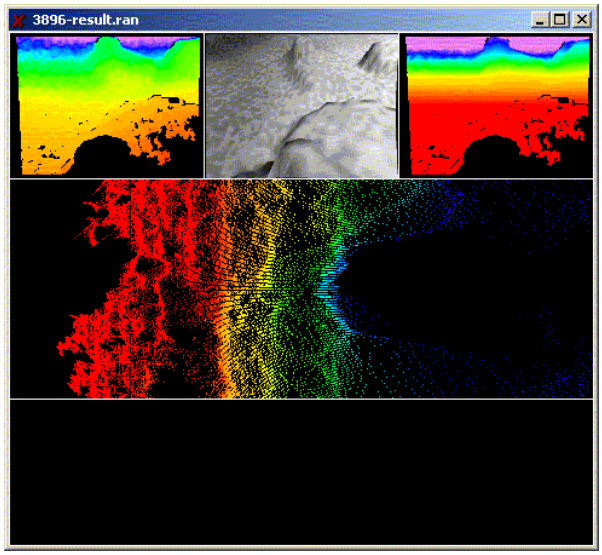

Fig 12. Output of JPL Stereo operating on ROAMS imagery. The top corners show range data coloured by distance and by elevation. Dense, undistorted range output suggests that the imagery is geometrically consistent with the camera model used to generate the stereo.

6. FUTURE DIRECTIONS

Verification testing showed that imagery synthesized by ROAMS is consistent with CAHV and CAHVOR camera models. However some additional steps may eliminate a number of small error sources and further improve the accuracy of the simulated models.

- Error in theta for CAHV models introduced noticeable shifts to some image pixels. Adding image warping to ROAMS image synthesis of CAHV images should remove this error.

- Parameters *Hc* and *Vc* were recovered to the nearest integer or half integer pixel, probably due to rounding in the image synthesis algorithm. This rounding can be compensated for during the image warping step. This should eliminate error in *Hc* and *Vc* and the resulting error in *A*.

- Error in parameter *Hs* may be due to a second instance of rounding in the image synthesis algorithm. Revising ROAMS' use of aspect ratio to eliminate the need for rounding may eliminate the error in *Hs* and the compensating error in *C*.

- Several CAHV and CAHVOR parameters were recovered to float precision. Careful scrubbing of the ROAMS camera model to convert float precision math to double precision may improve the accuracy of the imaging.

The near-term plans are to implement and verify CAHVORE image synthesis. That upgrade and verification will complete the incorporation of standard JPL camera models into ROAMS image synthesis.

7. ACKNOWLEDGEMENT

This work was supported by the Jet Propulsion Laboratory under contract to NASA.

8. REFERENCES

1. Jain, A., J.Guineau, C.Lim, W.Lincoln, M.Pomerantz, G.Sohl, R.Steele, "Roams: Planetary Surface Rover Simulation Environment", *i-SAIRAS 2003*, Nara, Japan, May 19-23, 2003.
2. Jain, A., J. Balaram, J. Cameron, J. Guineau, C. Lim, M. Pomerantz, G. Sohl, "Recent Developments in the ROAMS Planetary Rover Simulation Environment", *IEEE 2002 Aerospace Conf.*, Big Sky, Montana, March 6-13, 2004.
3. Litwin, T, *Camera Calibration References*, http://eis.jpl.nasa.gov/~telitwin/public-jpl/src/ccal/ccal-references.html
4. Jain, A, *Simscape*, http://dartslab.jpl.nasa.gov/references/index.php
5. http://oss.sgi.com/projects/inventor
6. http://www.opengl.org
7. http://www.povray.org
8. Skinner, M., http://www.gamedev.net/reference/articles/article1300.asp
9. Madison, R., "CAHV Image Synthesis in ROAMS", http://msl-lib/msl-lib/dscgi/ds.py/View/Collection-1332 , Feb 2005.