Terrain model registration for single cycle instrument placement

Matthew C. Deans, Clay Kunz, Randy Sargent, and Liam Pedersen
QSS Group / Autonomy and Robotics Area
NASA Ames Research Center
Mailstop 269-3, Moffett Field, CA 94035, USA
{mdeans,ckunz,rsargent,pedersen}@arc.nasa.gov

Abstract

This paper presents a method for registration of terrain models created using stereo vision on a planetary rover. Most 3D model registration approaches use some variant of iterated closest point (ICP), which minimizes a norm based on the distances between corresponding points on an arbitrary 3D surface, where closest points are taken to be corresponding points. The approach taken here instead projects the two surface models into a common viewpoint, rendering the models as they would be seen from a single range sensor. Correspondence is established by determining which points on the two surfaces project to the same location on the virtual range sensor image plane. The norm of the deviations in observed depth at all pixels is used as the objective function, and the algorithm finds the rigid transformation which minimizes the norm. This recovered transformation can be used for visual odometry, rover pose estimation and feature hand off.

1 Introduction

Single cycle instrument placement (SCIP) is a critical need for the planned MSL 2009 rover mission to Mars. The goal of SCIP is to enable a planetary rover to place an instrument on a scientifically interesting point on the terrain from a distance of 10 meters with a single rover command[1].

The first step in SCIP is the navigation of the rover to a location that places the terrain point within the workspace of an arm which carries an instrument. Once positioned, the rover servos the instrument into place for taking a measurement.

Figure 1: K9 rover placing its microscopic camera on a science target in the Marscape

All of this happens within one command cycle, so that once an operator selects a science target, the next response from the rover is the data that was requested. Single cycle instrument placement will significantly increase science return per unit of operational time over the stop and move, human-in-the-loop operation of the Sojourner rover.

Two technology needs within SCIP are pose estimation and feature hand off. Pose estimation is required in order to place the rover within reach of the science target. Registration of terrain models from different rover locations can be used to recover the pose of the rover in the different views. Feature hand off may be required to provide a better 3D model of the instrument goal location for the final step of instrument placement using imagery from hazard avoidance cameras closer to the ground. Registration of terrain models from the navigation cameras and hazard cameras provides a calibration which can be used for handing off the goal location from one stereo pair to the other.

This paper focuses on the problem of terrain model registration. The method presented in this paper uses stereo vision to build 3D terrain models, then uses an algorithm similar to ICP to find the rigid transformation which aligns two terrain models. The important difference between the method presented here and the popular ICP algorithm is the use of a sensor model which projects the two views into a virtual range sensor and minimizes the difference between the measured depth to points on the two models. Using a rendering model removes the need to search for corresponding points with a distance heuristic. At the correct registration, corresponding points project onto the same range sensor pixel.

2 Previous work

The Iterated Closest Point algorithm (ICP) algorithm was proposed independently by Chen and Medioni[2] and by Besl and MacKay[3] to recover a rigid transformation between two point clouds with unknown correspondence. The method relied on a step which used a nearest neighbor heuristic to establish correspondence between points, followed by a step which computed the rigid transformation between the point clouds in closed form using techniques which had been studied earlier[4, 5]. ICP was later extended to handle multiple views[6, 7, 8, 9], where there is no longer a known closed form solution for the alignment of all views even when the correspondences are known. A good survey of efficient variants on the original ICP algorithm can be found in [10].

An important extension to the original ICP algorithm was a modification to the objective function which minimizes the distance between a vertex in one model and the nearest point on the surface of the other model, rather than the nearest vertex[11]. This extension captures the fact that changes in the rigid transformation which cause a point to move along a surface should not be penalized. The derivatives of the cost function reflect the fact that the distance between a point and a plane only changes when the point moves in a direction parallel to the surface normal.

Methods other than ICP have also been used for model registration, including the general purpose Expectation-Maximization algorithm[12], and general purpose nonlinear optimization using robust M-estimators[13]. The latter approach is attractive. Fitzgibbon showed that besides increasing the robustness of the registration solution to outliers in the data, using Levenberg-Marquardt to minimize a robust norm converges to a solution rapidly and has a significantly larger basin of attraction than least squares. Because of these properties, robust estimation with Levenberg-Marquardt is used in this work.

3 Approach

This section describes the technical approach used for terrain model registration. The approach relies on two key parts. The first is a sensor model which predicts the observations that should be seen under a hypothesized transformation for the surface models. The second is the Levenberg-Marquardt[14, 15, 16] nonlinear optimization method, along with an extension which incorporates robust estimation using iteratively reweighted least squares[17, 18, 19].

3.1 Sensor modeling

For every pixel in the left camera image for which a correspondence is found in the right camera image, the stereo algorithm estimates the depth to that point. These depth estimates are combined to produce a 3D model of the surface. If two models of a surface are made from different locations, the rigid transformation that aligns the two models can be used to determine the coordinate transformation between views.

The surface models are represented by triangulated meshes with vertices $\mathbf{v}$ and $\mathbf{v}'$. If the two 3D models contain some region of overlap, there should be a rigid transformation that aligns the two overlapping regions on the two surfaces. We represent the rigid transformation using the parameter vector $\mathbf{p} = (x, y, z, \alpha, \beta, \gamma)^T$ corresponding to 3 translation directions plus rotation in roll, pitch, and yaw. These parameters define a transformation matrix $\mathbf{T_p}$. If $\mathbf{p}$ is the parameter describing the transformation between surfaces $\mathbf{v}$ and $\mathbf{v}'$, then for every pair of corresponding points $\mathbf{v}_i$ and $\mathbf{v}'_i$ the relationship

$$\mathbf{v}'_i - \mathbf{T_p}\mathbf{v}_i = 0 \tag{1}$$

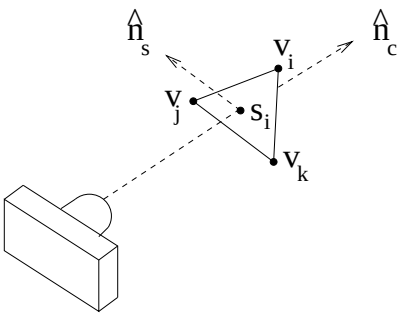

Figure 2: Each pixel in the range image is predicted by rendering the corresponding mesh facet into a virtual range sensor.

holds. With real observations this equality will not hold. The approach taken in ICP is to minimize the Euclidean distance between points. In this work, we project these two models into a virtual range sensor view and minimize the difference between the depths at each point.

Rendering takes $O(n)$ operations, where n is the number of pixels in the virtual range sensor. For each facet on the mesh $\mathbf{v}'$, the three vertices $\mathbf{v}'_i$, $\mathbf{v}'_j$, and $\mathbf{v}'_k$ are projected onto the image plane, creating a triangle. For every pixel inside that triangle, the location of the intersection of the camera ray $\hat{\mathbf{n}}_c$ and the facet of the mesh is a point $\mathbf{s}'_i$ which is a convex combination of the vertices,

$$\mathbf{s}'_i = \alpha_i \mathbf{v}'_i + \alpha_j \mathbf{v}'_j + \alpha_k \mathbf{v}'_k \tag{2}$$

with $\alpha_i + \alpha_j + \alpha_k = 1$. The depth to the intersection point is the z coordinate in the camera frame,

$$z_i = \hat{\mathbf{n}}_c \cdot \mathbf{s}'_i \tag{3}$$

The vector of all depths z_i is denoted $\mathbf{z}$. The surface model $\mathbf{v}'$ does not move during registration, so $\mathbf{z}$ is a constant.

The depth to the point $\mathbf{v}_i$ changes with $\mathbf{p}$. Similarly to (2) and (3), we write

$$\begin{aligned} \mathbf{s}_i &= \mathbf{T}_{\mathbf{p}}(\alpha_i \mathbf{v}_i + \alpha_j \mathbf{v}_j + \alpha_k \mathbf{v}_k) \\ h_i(\mathbf{p}) &= \hat{\mathbf{n}}_c \cdot \mathbf{s}_i \end{aligned} \tag{4}$$

and $\mathbf{h}(\mathbf{p})$ is the vector containing all predicted depths $h_i(\mathbf{p})$. We then define an objective function which is the sum of squared deviations between the projected depths

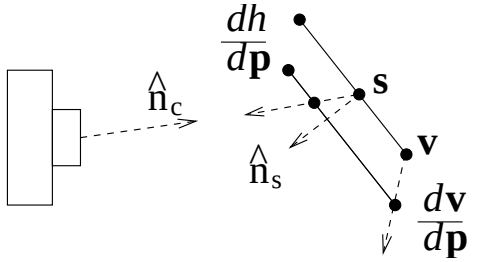

Figure 3: The Jacobian of the depth measurement is found by projecting the derivative of the vertex locations onto the surface normal.

$$J_2 = \frac{1}{2}(\mathbf{z} - \mathbf{h}(\mathbf{p}))^T \mathbf{R}^{-1} (\mathbf{z} - \mathbf{h}(\mathbf{p})) \tag{5}$$

The goal of the registration then is to minimize (5). Using Levenberg-Marquardt requires the gradient and approximate Hessian

$$\begin{aligned} \nabla_{\mathbf{p}} J_2 &= -\mathbf{H}^T \mathbf{R}^{-1} (\mathbf{z} - \mathbf{h}(\mathbf{p})) \\ \nabla^2_{\mathbf{p}} J_2 &\approx \mathbf{H}^T \mathbf{R}^{-1} \mathbf{H} \end{aligned} \tag{6}$$

to compute the updated parameter

$$\mathbf{p}^{(i+1)} = \mathbf{p}^{(i)} + (\nabla^2_{\mathbf{p}} J_2 + \lambda \mathbf{I})^{-1} \nabla_{\mathbf{p}} J_2 \tag{7}$$

The parameter λ is used to dynamically mix Newton-Raphson style updates with gradient descent style updates. If an update results in $J(\mathbf{p}^{(i+1)}) > J(\mathbf{p}^{(i)})$ then λ is increased and $\mathbf{p}^{(i+1)}$ recomputed. A more complete description of the general use of Levenberg-Marquardt optimization is beyond the scope of this paper. Several useful descriptions exist[20, 16]. However, the Jacobian $\mathbf{H} = \frac{\partial \mathbf{h}}{\partial \mathbf{p}}$ warrants some explanation.

The Jacobian $\mathbf{H}$ is the change in the rendered depth at each point given a change in the transformation parameter $\mathbf{p}$. Any motion of the polygon on which the point $\mathbf{s}_i$ lies can be decomposed into motion normal to the plane and motion parallel to the plane. Motions parallel to the plane do not change the depth to the point. The depth only changes with motion normal to the plane.

The change of the point $\mathbf{s}_i$ is described by $\partial \mathbf{s}_i / \partial \mathbf{p}$, which is a linear combination of the derivatives $\partial \mathbf{v} / \partial \mathbf{p}$ with the same coefficients used during rendering in (4). The projection of the derivative onto the surface normal is

$$\frac{\partial \mathbf{s}_i}{\partial \mathbf{p}}_{\perp} = \hat{\mathbf{n}}_c \cdot (\alpha_i \frac{\partial \mathbf{v}_i}{\partial \mathbf{p}} + \alpha_j \frac{\partial \mathbf{v}_j}{\partial \mathbf{p}} + \alpha_k \frac{\partial \mathbf{v}_k}{\partial \mathbf{p}}) \tag{8}$$

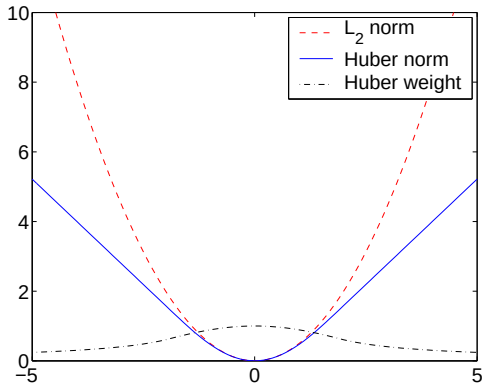

Figure 4: Comparison of the L_2 norm and the Huber robust kernel used in this work, and the weight function for weighted least squares.

The change in depth h_i lies along the camera normal. Its projection onto the surface normal is

$$\frac{\partial h_i}{\partial \mathbf{p}}_{\perp} = \hat{\mathbf{n}}_c \cdot \hat{\mathbf{n}}_s \frac{\partial h_i}{\partial \mathbf{p}} \tag{9}$$

Equating the projections (8) and (9) we find

$$\frac{\partial h_i}{\partial \mathbf{p}} = \frac{1}{\hat{\mathbf{n}}_c \cdot \hat{\mathbf{n}}_s} \hat{\mathbf{n}}_c \cdot (\alpha_i \frac{\partial \mathbf{v}_i}{\partial \mathbf{p}} + \alpha_j \frac{\partial \mathbf{v}_j}{\partial \mathbf{p}} + \alpha_k \frac{\partial \mathbf{v}_k}{\partial \mathbf{p}}) \tag{10}$$

The Jacobian $\mathbf{H}$ is the matrix containing all of the gradients $\partial h_i / \partial \mathbf{p}$.

3.2 Robust Estimation

The L_2 norm is optimal when the observation noise is Gaussian. However, the L_2 norm is known to exhibit problems when the data contains outliers. For data which may contain outliers, there are a family of norms $\rho()$ which are robust to large deviations. These are functions which have a unique minimum at zero and a bounded derivative far from zero, so that large deviations provide only a small contribution to the gradient of the objective function. The kernel used in this work is the Huber norm[19],

$$\rho(x) = \begin{cases} c^2(1 - \cos(x/c)) & \text{if } |x|/c < \pi/2 \\ c|x| + c^2(1 - \pi/2) & \text{if } |x|/c \geq \pi/2 \end{cases} \tag{11}$$

shown in Figure 4. When the deviation is close to zero, the Huber norm behaves similarly to the L_2 norm. When the deviation is large, the norm behaves similarly to L_1, i.e. the derivative of the norm does not increase as the deviation increases. Robust estimators have been used for a variety of applications[17]. In particular, this norm has been shown to perform well for ICP[13]. Using the robust norm, we rewrite (5) as

$$J_H(\mathbf{p}) = \frac{1}{2} \sum_i \rho(z_i - h_i(\mathbf{p})) \tag{12}$$

and the derivatives as

$$\begin{aligned} \nabla_{\mathbf{p}} J_H &= \mathbf{H}^T \mathbf{\Omega} \mathbf{R}^{-1} (\mathbf{z} - \mathbf{h}(\mathbf{p})) \\ \nabla^2_{\mathbf{p}} J_H &= \mathbf{H}^T \mathbf{\Omega} \mathbf{R}^{-1} \mathbf{H} \end{aligned} \tag{13}$$

where $\mathbf{\Omega}$ is a diagonal matrix of weights

$$\omega_{ii} = w(z_i - h_i(\mathbf{p})) \tag{14}$$

The weight function is derived from the robust norm, and several examples are shown in [19]. The weight function for the norm used in this work is

$$w(x) = \begin{cases} c/x \sin(x/c) & \text{if } |x|/c < \pi/2 \\ c/|x| & \text{if } |x|/c \geq \pi/2 \end{cases} \tag{15}$$

with $c = 1.2107$. The weights are recomputed during each iteration of Levenberg-Marquardt, resulting in an iteratively reweighted least squares (IRLS) optimization.

4 Experimental results

To empirically validate the performance of the registration, we tested the algorithm using a few examples of 3D surface registration problems which are part of a single cycle instrument placement system. The 3D models used in these experiments are all built using stereo algorithms which run onboard the K9 rover using images from three different stereo camera pairs. Table 1 describes the stereo camera pairs used on the rover. The terrain models derived from stereo imaging are used for a variety of purposes including visualization for

Camera pair	Location	FOV
Science cam	top of mast	narrow
Nav cam	top of mast	moderate
Haz cam	below solar panel	wide

Table 1: Stereo camera pairs on K9 rover

scientists and rover operators as well as onboard autonomous hazard avoidance. Registration of 3D models can also enable the applications outlined in Table 2. Examples of these are shown below.

Application	Camera pair	Camera pair
Visual odometry	Nav cam	Nav cam
Pose estimation	Nav cam	Science cam
Target hand off	Haz cam	Nav cam

Table 2: Applications of 3D registration and the stereo pairs used for each.

4.1 Visual odometry

Visual odometry can be achieved by registration of 3D surface models created using images from the navigation cameras before and after the rover moves. The transformation which aligns the views is directly related to the change in pose of the rover, leading to an odometry measurement between poses. For the visual odometry experiment the rover was placed in the Marscape and incrementally commanded to move forward one meter or turn 45°. The 3D models acquired before and after each move were registered using dead reckoning as an initial guess for the alignment.

Figure 5 shows the differences in depth between the rendered surfaces using the transformation from dead reckoning, before the registration algorithm is applied. Medium gray pixels denote zero error, while white and black denote errors of more than 10 cm. Figure 6 shows the depth errors found after optimizing the L_2 norm in (5) using Levenberg-Marquardt. While the ground plane is well aligned, there are large errors where a rock in the foreground is misaligned. Figure 7 shows the result of optimizing the robust norm in (12). By minimizing the robust norm the algorithm is able to correct the errors seen in Figure 6. This is consistent with Fitzgibbon's results and the claim that the robust norm has a wider basin of attraction than L_2[13].

Figure 8 shows the convergence of the robust norm through time for several runs. Typically, convergence is achieved within 5-10 iterations. The two convergence curves which show 20 and 38 iterations correspond to point turns, where the overlap in the models is poor.

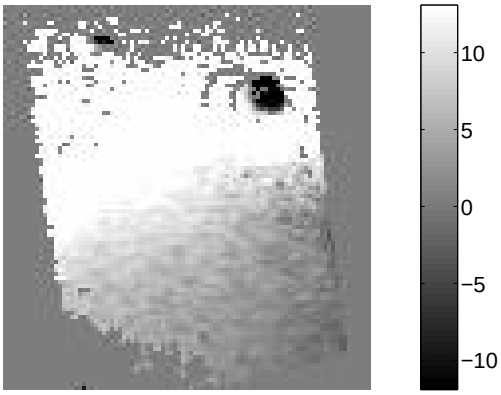

Figure 5: Initial depth errors for starting guess provided by dead reckoning for the visual odometry example. Error measured in centimeters.

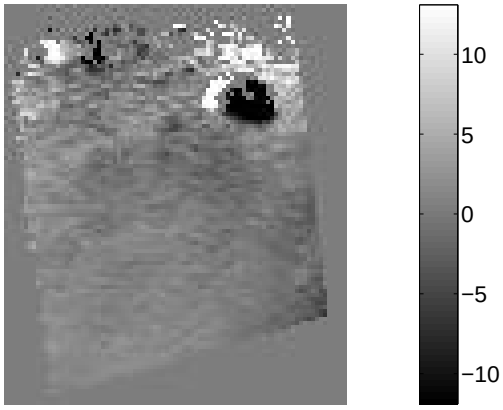

Figure 6: Final depth errors after convergence with L2 norm for the visual odometry example. Error measured in centimeters.

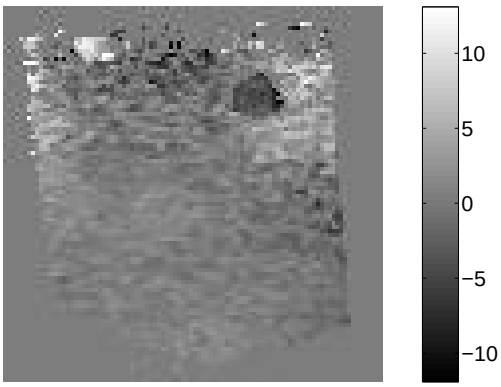

Figure 7: Final depth errors after convergence with Huber norm for the visual odometry example. Error measured in centimeters.

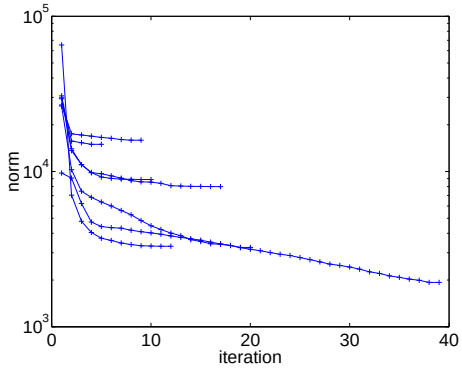

Figure 8: Convergence of the norm for several visual odometry examples

4.2 Pose estimation

Pose estimation may also be done relative to a map of the environment built earlier. This may be done, for example, by building an extensive 3D panoramic model using the high resolution, narrow field of view science cameras from a single location and using the model as a 3D map. The rover can repeatedly register terrain models captured by the navigation cameras against the larger more accurate model as it moves. The transformation which aligns the nav cam model to the previous map is directly related to the pose of the vehicle in the coordinate frame of the rover when the panoramic model was created. Figure 9 and 10 show an example of the registration of 3D models created by the navigation cameras and science cameras. The initial alignment again comes from dead reckoning, but once the rover has localized, the next pose can be initialized using the previous registration plus an incremental dead reckoning estimate. Figure 11 shows the initial and final depth errors for the models in Figure 9 and Figure 10.

4.3 Feature hand off

In the context of instrument placement it may be necessary to coordinate multiple stereo pairs. For example, for the task of placing an instrument against a rock in front of the rover, the Haz cams might have a better vantage point to image the goal location when the rover is close to a rock which was selected by rover operators using the navigation cameras. By registering the 3D models from

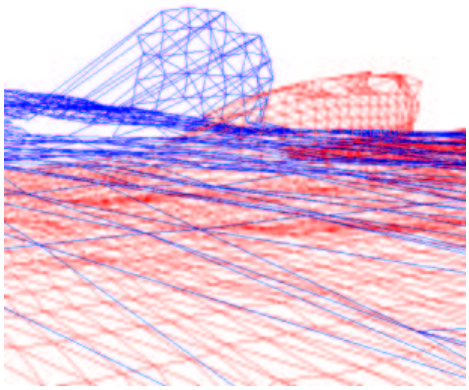

Figure 9: Alignment before registration for the pose estimation example.

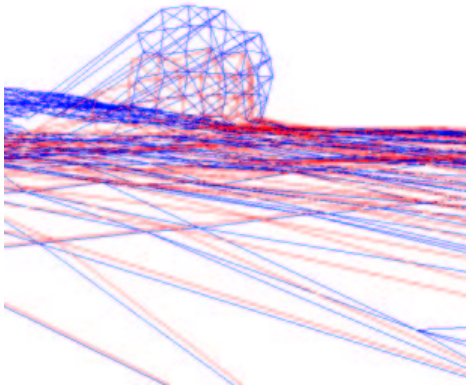

Figure 10: Alignment after registration for the pose estimation example.

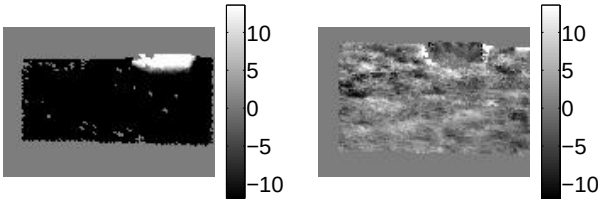

Figure 11: Depth error before and after registration for the pose estimation example. Error measured in centimeters.

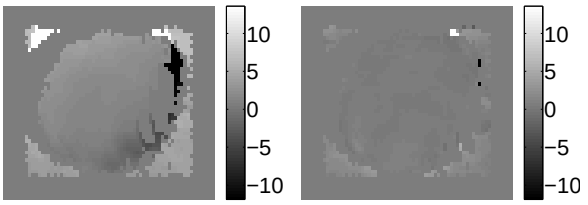

Figure 12: Depth error before and after registration for goal location hand off example. Errors measured in centimeters.

the Haz cam and Nav cam, the rover can hand off the target location from one view to the other. Figure 12 shows the depth errors before and after registration of surface models created with the Nav cam and Haz cam.

5 Discussion

Registration of 3D surface models is an attractive approach for rover localization. As long as the lighting conditions permit the acquisition of images for stereo, the resulting 3D surface models are independent of the lighting conditions. This is attractive compared to 2D approaches which might have difficulty with tracking features or recognizing places when lighting conditions change.

Furthermore, 2D visual tracking approaches require the rover to spend computational effort on something that it may only be doing for the purpose of visual odometry. However, NASA's current plans call for stereo vision to be used as a hazard avoidance technique on MER in 2003 and likely on MSL in 2009. Registering the 3D models that are already created in order to do local path planning and obstacle avoidance makes dual use of data that is being generated anyway. The marginal computation for registration is less than the computational effort for building the 3D surfaces, so most of the computational work is already done.

Accurate stereo calibration is fundamentally important for the approach in this paper. A miscalibration of stereo can cause the resulting models to be distorted by a projective transformation. Such a deformation will not be recoverable by the parameters in $\mathbf{p}$.

The robust estimation method used in this paper works quite well, as was reported in [13]. The surface models used in the examples here were not "cleaned" in any way and the results are still promising. Other reported approaches require mesh regularization and cleaning in order to ensure that there are no outliers before minimizing a norm which is sensitive to large deviations. These steps may improve the results we can acheive using robust estimation but empirically are not required for it to work.

We are currently working to further extend this work. Algorithmically we are investigating ways to optimize the implementation, perhaps making use of some very efficient existing rendering techniques. We would also like to extend this to multiview registration in order to handle more than two views at a time. It should be possible to extend the objective function to include a term which matches the albedo at each point along with the depth.

In terms of rover autonomy, this method might benefit from some sensor planning to determine when a next view is necessary for localization. We have already seen the difficulties with registering views where the rover has turned far enough that there is a small overlap between surface models.

This method is also being incorporated into a larger demonstration of single cycle instrument placement for improved efficiency of planetary rovers and increased science return for future Mars missions.

6 Acknowledgements

The authors would like to thank Larry Edwards, David Lees, Anne Wright, Susan Lee, and Maria Bualat for their help with the Viz visualization package, the K9 rover and the experiments in this paper. We would also like to thank Daniel Huber from Carnegie Mellon for many useful discussions on surface registration.

References

[1] L. Pedersen, R. Sargent, M. Bualat, M. Deans, C. Kunz, S. Lee, and A. Wright. Single cycle instrument deployment for mars rovers. *To appear in Proceedings of i-SAIRAS*, 2003.

[2] Y. Chen and G. Medioni. Object modeling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–155, 1992.

[3] P. J. Besl and N. D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[4] B. K. P. Horn. Closed-form solution of absolute orientation using unit quatemions. *Journal of the Optical Society of America*, 4:629–642, 1987.

[5] K. S. Arun, T. S. Huang, and S. D. Blostein. Least-squares fitting of two 3-d point sets. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-9(5):698–700, 1987.

[6] A. J. Stoddart and A. Hilton. Registration of multiple point sets. In *Proceedings of ICPR '96*, pages 40–44, 1996.

[7] R. Benjemaa and F. Schmitt. A solution for the registration of multiple 3d point sets using unit quaternions. In *Proceedings of the European Conference on Computer Vision*, volume 2, pages 34–50, 1998.

[8] A. Ardeshir Goshtasby. Three-dimensional model construction from multiview range images: survey with new results. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(11):1705–1714, 1998.

[9] Kari Pulli. Multiview registration for large data sets. In *Int.Conf. on 3D Digital Imaging and Modeling*, pages 160–168, 1999.

[10] S. Rusinkiewicz and M. Levoy. Efficient variants of the icp algorithm. In *Third International Conference on 3D Digital Imaging and Modeling (3DIM)*, 2001.

[11] P. Neugebauer. Geometrical cloning of 3d objects via simultaneous registration of multiple views. In *Proceedings of the International Conference on Shape Modelling and Applications*, pages 130–9, March 1997.

[12] J. Goldberger. Registration of multiple point sets using the em algorithm. In *Proceedings of the International Conference on Computer Vision*, volume 2, pages 730–736, 1999.

[13] A. Fitzgibbon. Robust registration of 2d and 3d point sets. In *British Machine Vision Conference*, pages 411–420, 2001.

[14] Kenneth Levenberg. A method for the solution of certain non-linear problems in least-squares. *Quarterly of Applied Mathematics*, 2(2):164–168, July 1944.

[15] Donald W. Marquardt. An algorithm for the least-squares estimation of nonlinear parameters. *SIAM Journal of Applied Mathematics*, 11(2):431–441, June 1963.

[16] Philip E. Gill, Walter Murray, and Margaret H. Wright. *Practical Optimization*. Academic Press, 1981.

[17] P. J. Huber. *Robust Statistics*. John Wiley & Sons, 1981.

[18] William. J. J. Rey. *Introduction to Robust and Quasi-Robust Statistical Methods*. Springer, 1983.

[19] Zhengyou Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. Technical Report No. 2676, INRIA, 1995.

[20] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C*. Cambridge University Press, 1988.