VISUAL MOTION ESTIMATION AND TERRAIN MODELING FOR PLANETARY ROVERS

Stephen Se, Timothy Barfoot, Piotr Jasiobedzki
MDA Space Missions
Brampton, Ontario, Canada, L6S 4J3
{Stephen.Se, Tim.Barfoot, Piotr.Jasiobedzki}@mdacorporation.com

ABSTRACT
The next round of planetary missions will require increased autonomy to enable exploration rovers to travel great distances with limited aid from a human operator. For autonomous operations at this scale, localization and terrain modeling become key aspects of onboard rover functionality. Previous Mars rover missions have relied on odometric sensors such as wheel encoders and inertial measurement units/gyros for on-board motion estimation. While these offer a simple solution, they are prone to wheel-slip in loose soil and drift of biases, respectively. Alternatively, the use of visual landmarks observed by stereo cameras to localize a rover offers a more robust solution but at the cost of increased complexity. Additionally rovers will need to create photo-realistic three-dimensional models of visited sites for autonomous operations on-site and mission planning on Earth.

1. INTRODUCTION
The ExoMars Rover is a key element of the ExoMars mission, the first flagship mission of the Aurora Programme initiated by the European Space Agency (ESA). The aim of this programme is to characterize in detail the Mars biological environment in preparation for future missions, including human exploration. Carrying a large suite of exobiology instruments, the ExoMars Rover will be capable of operating autonomously, traveling several kilometers over rocky Martian terrain, and drilling to collect samples for analysis by the instruments. Planned for launch in 2011, the main purpose of the ExoMars mission is to search for signs of past and present life on Mars. In a Phase A study performed for ESA, MDA led an international industrial team to develop an optimized conceptual design of the Rover (see Figure 1), incorporating specialized electrical power generation, thermal control, navigation, telecommunications and vehicle control subsystems.

Absolute localization techniques such as radiolocation and horizon feature matching to elevation data provide updates too infrequently to be used throughout a sol [1]. These techniques are more appropriate to making corrections at the end of a sol or every few sols.

To localize throughout a sol we typically require a relative localization system that tries to estimate the pose of a rover relative to a reference frame attached to the initial pose of the robot. No attempt is made to find the correspondence between the initial reference frame and a global reference frame. The frequency of updates from a relative localization system is typically much higher than an absolute localization system.

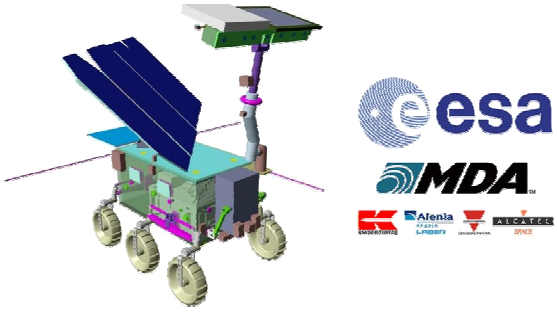

Figure 1: Proposed rover design for the European Space Agency's 2011 ExoMars mission

Wheel odometry estimates the velocities of each wheel and since they are part of the motion control components, utilizing these sensor data is relatively simple and inexpensive. The velocities are integrated over time to produce a position estimate. However, wheel odometry is extremely prone to slip on natural terrains. Many studies have proven that localization relying on odometry alone can produce 20%-25% error of distance travel in position estimate [2]. The 2004 Mars Exploration Rovers experienced considerable slip on occasion, corrupting odometry measurements.

Inertial measurement units (IMU) can provide translation and attitude of a rover by using 3-axis accelerometers and 3-axis gyroscope rate sensors. Both accelerometers and gyros can however be influenced by bias errors which can lead to unbounded growth in error over time. Bias fluctuations over an entire sol preclude using an IMU alone.

Improved orientation estimates can be obtained by employing a Sun sensor, which compares a detected sun vector with internal knowledge of Sun's expected location based on ephemeris data. The inclusion of a Sun sensor was one of the main recommendations after the 1997 Mars Pathfinder mission [3]. On hard terrain, a Sun sensor in combination with odometry can provide a cheap relative positioning device [4]. For baseline operations, 2004 Mars Exploration Rovers employed Sun sensors in combination with other sensors, but estimates of translation relied heavily on odometry for which slip was a problem on loose terrain.

Proc. 'ISAIRAS 2005 Conference', Munich, Germany,
5-8 September 2005 (ESA SP-603, September 2005)

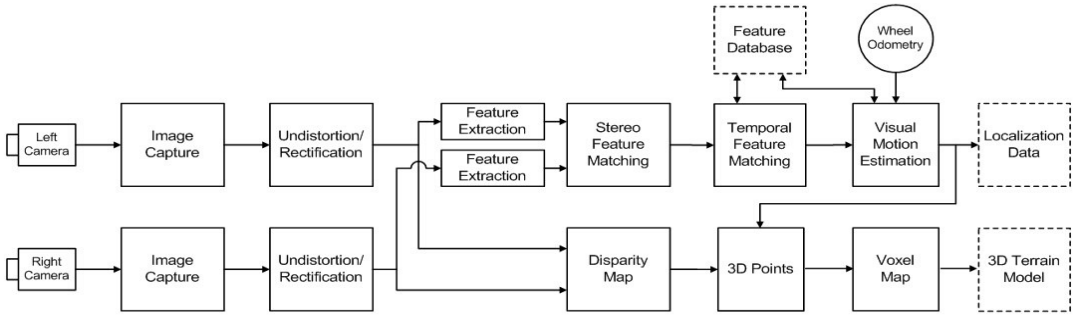

Figure 2: System architecture for vision system

Stereo cameras have been present on planetary rovers for other purposes, namely obstacle detection and avoidance, and modeling. However, only the recent Mars Exploration Rovers have used visual odometry as a technology demonstration (discussed below). The slow acceptance of vision-based localization may be due to limited computational resources and power.

Based on knowledge of past rover missions and the anticipated requirements for future rovers to travel longer distances per sol and to generally perform more autonomously, an improved relative localization system will be needed. There are several possible avenues that might be pursued to provide such a system. However, given that baseline rover operations already rely on stereo cameras for obstacle avoidance and modeling, it is logical to attempt to use these same sensors to help improve localization.

In this paper, we will describe our development of a vision-based localization system to allow a planetary rover to position itself with errors limited to a few percent of distance traveled over a several kilometer traverse across unknown terrain. A consequence of solving this problem is that it facilitates the creation of a high-resolution three-dimensional terrain model of the environment for visualization and planning.

2. SYSTEM OVERVIEW

In this section we present our approach to vision-based localization and terrain modeling, as shown in Figure 2. Stereo imagery is used for two purposes: localization and terrain modeling. We can see that terrain modeling relies on the output of visual motion estimation. This is because we seek to create terrain models from a moving platform and so data from images taken in different locations must be merged. The 3D terrain map can be used for situational awareness and it can be converted to a cost map for autonomous motion planning.

The Undistortion/Rectification block transforms the raw images into a rectified form that simplifies stereo processing. This requires that the stereo camera undergo a calibration procedure in advance of use.

As our intended application is for a planetary rover, we will use images obtained by the Mars Exploration Rover, Spirit, as a running example. They were obtained from the MER Analyst's Notebook web repository [5]. They were taken by Spirit's Front HazCam on Sol 15 of the primary mission as it approached a rock feature called "Adirondack".

2.1 Feature Extraction

In our vision-based localization, we automatically identify and track a large number of visual landmarks, or features, as the rover moves. We have chosen to use a high level set of natural visual features called Scale Invariant Feature Transform (SIFT) as the visual landmarks to compute the camera motion. SIFT was developed for image feature generation in object recognition applications [6]. The features are invariant to image translation, scaling, rotation, and partially invariant to illumination changes and affine or 3D projection. These characteristics make them suitable as landmarks for robust matching when the cameras are moving around in an environment. Such natural landmarks are observed from different angles, distances or under different illumination.

Previous approaches to feature detection, such as the widely used Harris corner detector [7], are sensitive to the scale of an image and therefore are less suitable for building feature databases that can be matched from a range of camera positions.

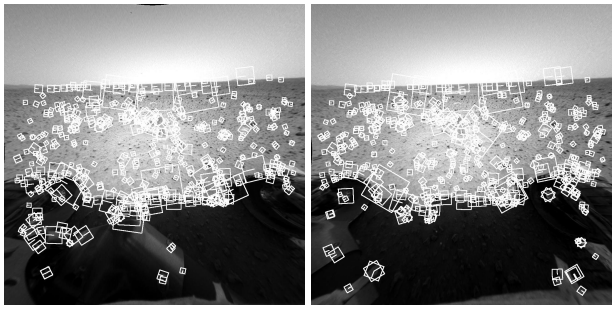

Figure 3: SIFT features extracted from a stereo pair from the Mars Exploration Rover Spirit

Figure 3 shows the hundreds of SIFT features that were identified in the left and right images in our example. Each feature is marked with a white box. The size of the box represents the scale of the feature while

the rotation of the box represents the orientation of the feature. It is worth noting that features were found at many scales and orientations both near the rover and out to the horizon.

Although SIFT features are reasonably unique in their description as compared to Harris corners, there is an added computational burden associated with their use. For this reason, we have implemented the Feature Extraction block on a Field Programmable Gate Array (FPGA), to be described in Section 3.2.

2.2 Stereo Feature Matching

With known stereo camera geometry, the SIFT features in the left and right images are matched using the following criteria: epipolar constraint, disparity constraint, orientation constraint, scale constraint, local feature vector constraint and unique match constraint [8]. All of these constraints are essentially inequality-type constraints with tunable thresholds. By varying the thresholds we may trade off the number of features against the quality of matched features.

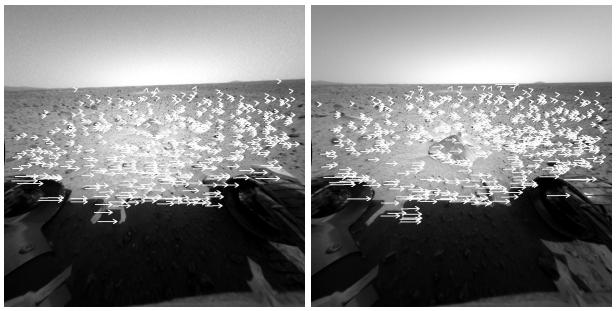

Figure 4: Stereo matching of SIFT features for two consecutive stereo pairs

The quality of the matched features is typically very high coming out of the Stereo Feature Matching block. Figure 4 shows the hundreds of stereo-matched SIFT features for two consecutive stereo pairs in our example. The images shown are the right images for each stereo pair and each match is marked with a white arrow. The tail of the arrow is at the location of the feature in the right image and the head of the arrow is at the location of the feature in the left image. Thus, the length of the arrow represents the disparity. We can see that the lines are all horizontal as expected due to the epipolar constraint and disparity is larger close up and smaller far away.

2.3 Temporal Feature Matching

Temporal feature matching is typically done in one of two ways: single frame or multi-frame. In single frame matching, a stereo pair is compared only with the previous frame. In multi-frame matching a database is built up and the current frame is compared to the database. A 28% reduction in rover navigation error has been reported [9] when multi-frame matching is used, rather than considering each pair of frames separately. Our preferred approach is to maintain a database, but for the planetary application a trade study would be necessary to balance performance against processing limitations. This is because there is a cost in terms of the additional memory and computational cycles needed to maintain and search the database.

The Temporal Feature Matching block takes each of the stereo-matched features from the current frame and finds the best match in our growing database of features. If a feature cannot be found in the database, we add it and assign it an id number. To maintain fast access in our implementation, a kd-tree is built online and matching of observed features to the database (i.e., data association) is carried out by a best-bin-first search [10]. We have experimented with databases sizes up to 200,000 features.

Figure 5 shows some temporally-matched features between the two stereo frames. The line connecting the previous position (tail) to the current position (head) is analogous to optical flow. Roughly 100 temporal matches were found here and we can see the movement of these features is qualitatively correct as the rover moved forward.

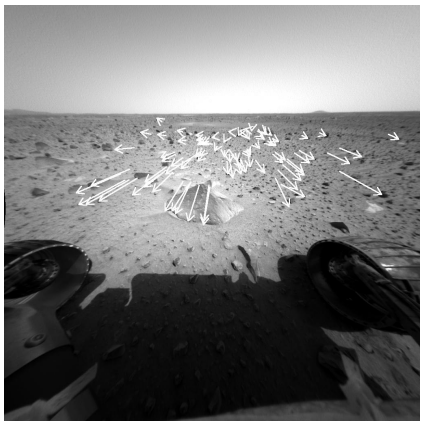

Figure 5: Temporal matching of SIFT features between frames

2.4 Visual Motion Estimation

Once we have completed all of the feature extraction and feature matching steps, we estimate the full three-dimensional motion of the camera from the temporally-matched features. To do this we employ a Simultaneous Localization And Mapping (SLAM) approach that has a few essential steps:

1. Predict camera motion using odometric sensors.
2. Correct this camera motion using the observations of SIFT features that have been temporally-matched to the database. This is done using a weighted least squares technique that accounts for the feature uncertainty.
3. Update the features in the database (our map) using the final camera motion.

In more detail, our estimation algorithm is derived from the FastSLAM 2.0 algorithm [11][12]. Some modifications were necessary to make the algorithm compatible with our scenario [13]. The biggest change to

the original algorithm is that we observe a large number, K, of SIFT landmarks simultaneously (e.g., K = 500). We also needed to incorporate outlier detection as some of the visual landmarks are inevitably mismatched.

We seek to simultaneously estimate the trajectory of a vehicle as well as the states of L landmarks. Mathematically this is expressed as the joint probability density for the vehicle trajectory and landmarks positions, given all the observations:

$$p(\mathbf{s}^t, \mathbf{x}_1, \ldots, \mathbf{x}_L \mid \mathbf{z}^t, \mathbf{u}^t, \alpha^t) = p(\mathbf{s}^t \mid \mathbf{z}^t, \mathbf{u}^t, \alpha^t) \prod_{l=1}^{L} p(\mathbf{x}_l \mid \mathbf{s}^t, \mathbf{z}^t, \mathbf{u}^t, \alpha^t)$$

which can be factored into L landmark state-estimators and one vehicle trajectory estimator. The vehicle states, up to time t (a.k.a., its trajectory up to time t), is denoted $\mathbf{s}^t$. The l^{th} landmark state is denoted $\mathbf{x}_l$ (which is assumed to be stationary). The sensor observations, up to time t, are denoted $\mathbf{z}_t$. The control inputs (or odometry measurements), up to time t, are denoted $\mathbf{u}^t$. The data associations, which assign particular observations to particular landmarks, up to time t, are denoted α^t.

As described in [11][12], a Rao-Blackwellized particle filter will be used to update the posterior as new observations are gathered. This type of particle filter uses samples to represent uncertainty in the vehicle trajectory. Within each particle (a.k.a., sample), an independent Kalman filter [14] is implemented for each landmark in the map. Thus for each landmark (in each particle) we are estimating a mean and covariance:

$$p(\mathbf{x}_l \mid \mathbf{s}^{(m),t}, \mathbf{z}^t, \mathbf{u}^t, \alpha^t) \sim \mathcal{N}\left(\bar{\mathbf{x}}_{l,t}^{(m)}, \mathbf{C}_{l,t}^{(m)}\right)$$

where (*m*) is the particle index. This has the advantage of not requiring a monolithic filter to represent the joint density for the vehicle and all the landmarks. In our real-time implementation to date we have only been able to use a single particle to represent the vehicle trajectory. However, having formulated the problem in this way allows more particles to be added later if computational resources permit.

2.5 Disparity Map and 3D Points

To compute disparity maps offline, we use either Point Grey Research's optimized Triclops library based on the Sum of Absolute Differences (SAD) or MDA normalised correlation-based dense stereo algorithm.

To compute disparity maps online for real-time applications, we use the 3DAware PCI card from Tyzx for dense stereo computation. It consists of a DeepSea2 chip, which is an optimized hardware implementation of the Census stereo algorithm [15]. As with other stereo algorithms, texture is required for stereo matching, and hence there is no match for uniform regions. The Tyzx system can compute dense stereo at 30Hz but is limited to 512x512 resolution.

Whether online or offline, a simple pinhole stereo camera model is used to reconstruct the dense 3D points from the disparity map. As the rover moves around, dense 3D points are obtained relative to the camera position at each frame. All data sets must be transformed to the initial camera coordinate system using the camera pose estimated before they can be combined together.

2.6 Voxel Map and 3D Terrain Map

Using all 3D points obtained from the stereo processing is not efficient as there are a lot of redundant measurements, and the data may contain noise and missing regions (due to incorrect matches or lack of texture). Representing 3D data as a triangular mesh reduces the amount of data when multiple sets of 3D points are combined and thus also reduces the amount of bandwidth needed to send the resulting models offboard (e.g., to Earth). Furthermore, creating surface meshes fills up small holes and eliminates outliers, resulting in smoother and more realistic reconstructions.

To generate triangular meshes as 3D models, we employ a voxel-based method [16], which accumulates 3D points with their associated normals. It creates a mesh using all the 3D points, fills up holes and works well for data with significant overlap. The 3D data is accumulated into voxels at each frame. Outliers are filtered out using their local orientation and by selecting the threshold of range measurements required per voxel for a valid vertex.

Photo-realistic appearance of the reconstructed scene is created by mapping camera images as texture. Such surfaces are more visually appealing and easier to interpret as they provide additional surface details. Colour images from the stereo camera are used for texture mapping.

As each triangle may be observed in multiple images, the algorithm selects the best texture image for each triangle. A texture image is considered to be better if it is captured when the camera is facing the triangle directly. If the camera is looking at the triangle at an angle, then its quality is lower due to the lower and non-uniform resolution caused by perspective distortion. To find the best texture, the algorithm analyses all the images and selects the one that gives the largest area upon 2D projection according to the camera pose.

Figure 6 shows a textured terrain map for the image pair in our running example. The "Adirondack" rock feature is clearly visible in 3D when viewed from this perspective. A three-dimensional model of the Spirit rover was inserted for visualization.

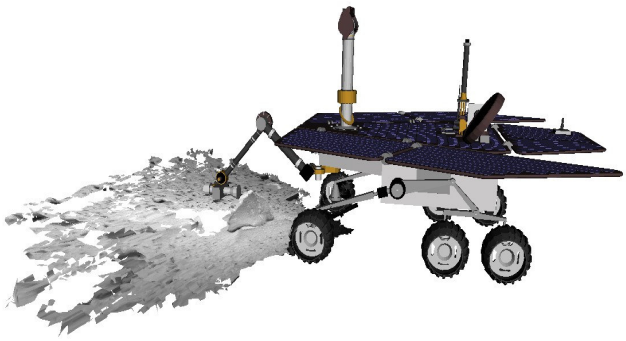

Figure 6: Terrain map with Spirit rover model

3. IMPLEMENTATION

3.1 Rover Testbeds

To date we have used two different rover testbeds for hardware testing. Initial testing of our methodology was conducted using the rover shown in Figure 7 (left). This rover consists of a 4-wheel chassis developed by the University of Toronto Institute for Aerospace Studies (UTIAS). The limited computational power of this platform required that all images be uploaded to a ground station for processing [17].

Figure 7: (Left) Custom rover with Bumblebee stereo camera. (Right) ATRVJr rover with custom stereo camera.

Our second testbed is depicted in Figure 7 (right). The chassis of this rover is an iRobot ATRVJr with a custom vision system. The stereo camera was constructed using a pair of Sony DFW-X700 cameras, mounted on a rigid aluminum bar, affixed to a pan-tilt unit. The images we use are 8-bit 1024x768 resolution. The field of view of the cameras is approximately 45 degrees horizontal and 35 degrees vertical. There are currently two computers on board, a dual Pentium III 1 GHz with 1 Gb of RAM (inside the red box) and a dedicated vision computer consisting of a Pentium M 1.8 GHz with 1 Gb of RAM. The vision computer also houses our hardware accelerated vision processing boards, a Tyzx DeepSea2 for dense stereo calculations and an AlphaData ADM-XRC board with a Virtex II Xilinx FPGA running our implementation of SIFT feature extraction. There are various other sensors onboard as well: sonar rangefinders, SICK laser rangefinder, DGPS, compass, inertial measurement unit, and inclinometer.

3.2 FPGA Implementation

The high computational requirements of vision algorithms often limit the distance and science investigation that can be safely achieved by rovers equipped with radiation hardened processors. In order to speed up performance, we used dedicated hardware such as Field Programmable Gate Arrays (FPGA) for some intensive image processing to offload the processor.

For this work, we have implemented the SIFT extraction on a Virtex II Xilinx FPGA (computationally intensive). The fixed point hardware implementation of SIFT was developed based on the floating point software version. To implement the complex SIFT algorithm directly using Very High-Level Design Language (VHDL) would have been a lengthy and time consuming task. An alternative high level environment was needed.

We chose System Generator, which is a software tool for modeling and designing FPGA-based signal processing systems in the Matlab-Simulink environment. Simulink provides a graphical environment for creating and modeling dynamical systems. System Generator consists of a Simulink library called a Xilinx Blockset, and software to translate a Simulink model into an equivalent faithful hardware realization of the model.

Even though the majority of the design was created with System Generator, there was coding in VHDL for low level processes that were not efficient to do with the Xilinx Block sets (such as DMA transfers, memory access routines and wrapper files). The System Generator design, low level VHDL coding and wrapper files were all brought into the Xilinx Integrated Synthesis Environment (ISE) software tool. The final bit file was generated within the ISE environment which then could be uploaded to the FPGA for execution.

To extract SIFT features from a 640x480 image, it takes 600 ms for a Pentium III 700MHz processor, while the FPGA can do so within 60 ms and leaving the processor available for other tasks.

4. EXPERIMENTAL RESULTS

A set of field trials was conducted on a sandy surface (10 traverses total). Figure 8 shows the results of an approximately 40 m traverse at maximum speed of 5 cm/s on loose sand. During this run, the motion planning software chose to go left around an obstacle (8 m into the traverse). While executing this turn, a considerable amount of slip occurred, causing a significant error in the odometric-based orientation estimate. Our visual technique did a much better job of estimating the robot path, as can be seen by comparison to GPS.

Using a tape measure, the final position of the rover was 37.8 m from the start. Visual motion estimation found 39.4 m and GPS found 38.8 m. The tape measure was taken as ground truth, indicating the visual motion estimation over-predicted the position by 4% of distance travelled. However, it should be noted that most of this positioning error was in the longitudinal direction (along the line joining the start and final positions). Visually, the lateral error was extremely small, indicating that orientation was likely estimated very well throughout the traverse. Similar results were found for all the traverses. The repeatability of the system was found to be quite high across trials.

A second set of field trials was conducted on a large gravel area (5 traverses total). Figure 9 shows the results of an approximately 120 m traverse at maximum speed of 10 cm/s on gravel. Here we found that odometry did not experience isolated positioning errors, as was the case on

sand, but did experience a systematic error (gradual curve to the left). All of the trials at this site had similar errors, likely due to a slightly higher tire pressure on one side of the rover than the other. This systematic error was not observed prior to the field test; it was attributed to changes in tire pressure on the test day and hence miscalibration of odometry.

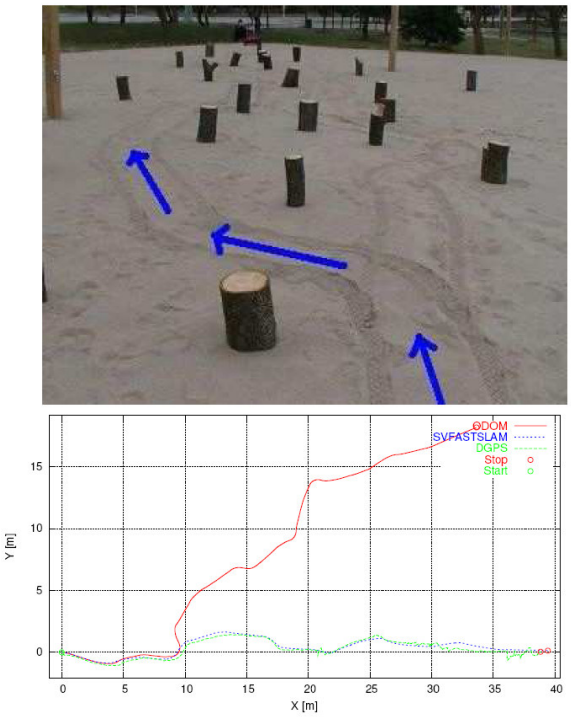

Figure 8: (Above) Sand test site with arrows indicating path taken by rover. (Below) Path estimated by FastSLAM (blue), odometry (red), and GPS (green) on loose sand.

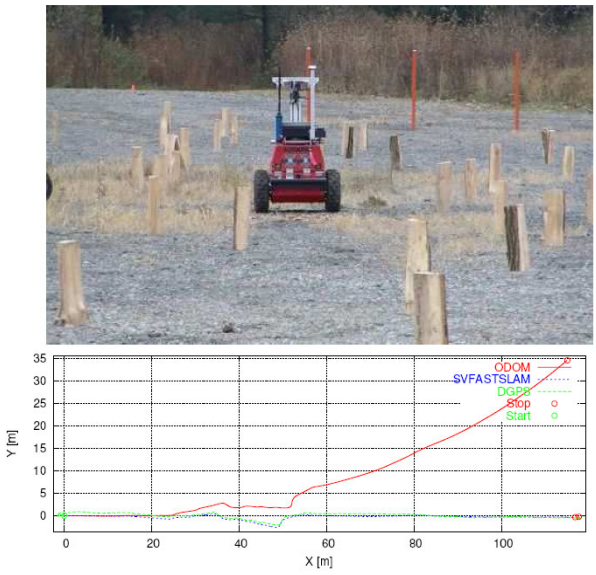

Figure 9: (Above) Gravel test site. (Below) Path estimated by FastSLAM (blue), odometry (red) and GPS (green) on gravel.

Using a tape measure, the final position of the rover was 117.4 m from the start. Visual motion estimation found 116.8 m and DGPS found 117.6 m. The tape measure was taken as ground truth, indicated the visual motion estimation under-predicted the position by 0.5% of distance travelled. Again, most of this positioning error was in the longitudinal direction (along the line joining the start and final positions). Visually, the lateral error was extremely small, indicating that orientation was estimated very well throughout the traverse.

The results of other trials at this site were mixed as we tried to push the system to move more quickly and use fewer images. Increasing the vehicle speed to 20 cm/s, or decreasing the frame-rate to 1.5 Hz, resulted in decreased performance.

Figure 10 shows a model we created from a moving rover that captured a sequence of 101 stereo pairs. A three dimensional model of the rover has also been inserted into the resulting terrain model for visualization.

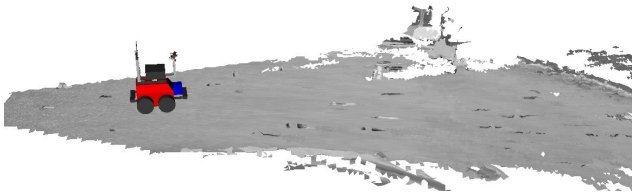

Figure 10 Terrain model with rover model inserted

5. DISCUSSIONS

5.1 Related Work

There have been various works on visual motion estimation for planetary rovers with promising results. Semi-sparse terrain maps were constructed and matched successively to obtain a vision-based state estimate in [18]. An extended Kalman Filter was then applied to fuse with wheel odometry. Experiments at JPL's rover pit showed that the results had more than double the accuracy of the dead reckoning estimate.

A maximum likelihood estimation technique for rover localization in natural terrain was presented in [19] by matching range maps. Stereo vision generated local terrain range map which was matched to a previously generated 3D occupancy map to estimate rover pose. Good qualitative results were obtained when tested with Sojourner data, running on-board Rocky 7 Mars rover prototype.

Pixel tracking in stereo image sequences was proposed in [20] to estimate visual odometry in outdoor unstructured terrain, with around 4% error over 25 metres. [21] evaluated a similar algorithm on the Marsokhod robot on many runs totaling several hundreds of metres and achieved about 2% translation error. [22] proposed that a set of concurrent and complementary algorithms are required for rover localization, as no

single localization algorithm is robust enough to fulfill various localization needs during long range navigation.

In addition to stereo vision, [23] discussed the use of inertial sensors to estimate camera ego-motion and to augment stereo tracking on rough terrain. [9][24] showed that even with a robust stereo ego-motion method, the system accumulated super-linear error due to increasing orientation error. Therefore, they proposed incorporating an absolute orientation sensor to reduce the error growth to linear. They achieved 1.2% error in experiments carried out with a prototype Mars rover.

This same technique was also used to perform rover path tracking [25]. The Mars Exploration Rovers, Spirit and Opportunity, have been using a derivative of this visual odometry technology on Mars. Initial demonstrations were conducted on Spirit during Sols 175-178 with positive results. The technique was also used to improve odometry when Spirit was forced to drive using only five of its six wheels, although some problems were reported during Sols 416-421. An official report of the results is pending.

Real-time visual odometry results for terrestrial applications have been reported by [26], which uses Harris corners for features. Our technique differs in that we are actually building a database of landmarks. Real-time SLAM results using SIFT features have recently been demonstrated indoors by [27] with a monocular camera and much smaller images than ours.

Most of the previous work used vision systems for localization only, whereas we also use the vision system for 3D modeling. Recently, [28] proposed using stereo images for recalibration and also for reconstructing 3D terrain models which were texture mapped with the original images. They have carried out preliminary experiments to create digital elevation maps at the ESA planetary terrain testbed. The model was then used to plan a trajectory for the Nanokhod rover. However, their vision system was part of the lander, not on-board of the rover. Therefore, the terrain map generated will be limited to the surroundings of the landing site only.

5.2 Advantages/Disadvantages of Our Approach

The key advantages of our localization approach are:

- No new sensors needed as stereo cameras are already baselined on most future rovers.
- Does not rely on any artificial infrastructure to localize and hence can be used far away from a lander for long-range science missions.
- Highly distinctive SIFT features are used as visual landmarks, enabling the repeated identification of landmarks to be quite robust.
- A SLAM approach is used for motion estimation (as opposed to single-frame odometry). This allows landmarks to be tracked over frames and thereby helps reduce accumulation of error.
- A probabilistic algorithm is used to estimate the rover's pose based on a large number of landmarks taking into account their respective quality.

The current disadvantages of our localization approach are:

- SIFT extraction requires considerable computational effort. We have addressed this through implementation of vision components on FPGAs in preparation for flight.
- Although our vision-based localization was based on the FastSLAM 2.0 algorithm, we found the use of more than a single particle to represent rover trajectory to be computational too expensive. Our experimental results have shown with a single particle we may still achieve reasonable results.

The key advantages of our terrain mapping approach are:

- It can be seamlessly integrated with our vision-based localization technique and hence terrain models can be created while the rover is in motion.
- Through the use of a voxel representation of the models, terrain maps for both visualization by ground operators and cost maps for autonomous operations can be generated.
- The resulting visualization models (with texture mapping) can be transmitted over a communication channel at a greatly reduced bandwidth than all of the raw images.

The key disadvantages of our terrain mapping approach are:

- The computational burden of generating disparity maps, voxel maps, and texturing is reasonably high and may require implemention in hardware for flight operations.

6. CONCLUSIONS

We have demonstrated the ability for a rover to use a stereo camera and SIFT features as the landmarks for efficient localization and terrain mapping. The resulting online visual motion estimation was used for autonomous outdoor rover traverses up to 120 m long on loose terrain. The final positioning errors were 0.5% to 4% of distance travelled, a major improvement over using odometry alone. We have reconstructed terrain models in many environments, both artificial and natural including underground mines and buildings. We have also begun to address the computational requirements of our approach by implementing one of the core vision blocks on FPGA. All of these preliminary findings have shown promise and hence we continue to develop our vision technologies for planetary rovers.

In terms of our estimation algorithm, a future step in our work is to incorporate loop-closure detection and possibly backwards correction [29]. This could be incorporated in our outlier detection scheme as the number of outliers tends to spike when loops are closed. This is because a large number of SIFT matches are made

but not in the expected locations. If loop closure can be robustly detected, we could switch to a `kidnapped robot' scenario to reset the localization and make corrections backwards in time. Work must also be done to prune and rebalance the kd-tree to allow significantly longer operation of the algorithm. We also seek to make our approach more robust to the translation and rotation that can occur between consecutive images. Currently, we can tolerate only small translations and rotations. For practical applications we would like to be able to move 1 m in translation and 20 degrees in rotation. This would make the algorithm efficient enough for planetary exploration, where computational resources are scarce.

We are also currently building a prototype of the ExoMars rover design shown in Figure 1 that will be used to further test our vision-based localization and terrain modeling. We plan to conduct long-range (i.e., traverses of kilometers) field trials with this new rover during the summer of 2006 in a Mars-like environment (possibly Haughton Crater in the Canadian High Arctic).

ACKNOWLEDGEMENTS

We would like to thank Ho-Kong Ng, Kirk Harasym, Lucas Szajek, Tariq Rafique, Robert Nguyen, and Raymond Oung for their contributions and useful discussions. MDA would also like to thank David Lowe at the University of British Columbia for providing most of the code related to SIFT features under license.

REFERENCES

[1] T. Parker, M. Malin, M. Golombek, T. Duxbury, A. Johnson, J. Guinn, T. McElrath, R. Kirk, B. Archinal, L. Soderblom, R. Li, and the MER Navigation Team and Athena Science Team, Localization, Localization, Localization. Lunar and Planetary Science XXXV (2004).

[2] P. Goel, S.I. Roumeliotis, and G.S. Sukhatme, Robust Localization Using Relative and Absolute Position Estimates, Proceedings of the 1999 IEEE/RSJ.

[3] B. Wilcox and T. Nguyen, Sojourner on Mars and Lessons Learned for Future Planetary Rovers, In Proc. of the 28th International Conference on Environmental Systems (ICES'98), SAE publication 981695, July, 1998.

[4] E. T. Baumgartner, H. Aghazarian, and A. Trebi-Ollennu, Rover Localization Results for the FIDO Rover, Sensor Fusion and Decentralized Control in Robotics System III, Proceedings of SPIE Vol. 4196, 2000.

[5] http://anserver1.eprsl.wustl.edu/

[6] D. G. Lowe, Distinctive image features from scale-invariant keypoints, Int. Journal of Computer Vision, vol. 60, no. 2, pp. 91-110, 2004.

[7] C.J. Harris and M. Stephens. A combined corner and edge detector. In Proceedings of 4th Alvey Vision Conference, pages 147–151, Manchester, 1988.

[8] S. Se, D. Lowe, and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. Int. J. of Rob. Res., 21(8):735–758, Aug. 2002.

[9] C.F. Olson, L.H. Matthies, M. Schoppers, and M.W. Maimone. Robust stereo ego-motion for long distance navigation. In Proc. of IEEE Conf. on Comp. Vis. and Patt. Recog. (CVPR) Volume 2, pages 453–458, South Carolina, June 2000.

[10] J. S. Beis and D. G. Lowe, Indexing without invariants in 3d object recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 21, no. 10, pp. 1000–1015, 1999.

[11] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit, Fastslam 2.0: An improved particle filtering algorithm for simultaneous localization and mapping that provably converges. Acapulco, Mexico: International Joint Conference on Artificial Intelligence (IJCAI), August 9-15 2003.

[12] M. Montemerlo, Fastslam: A factored solution to the simultaneous localization and mapping problem, Ph.D. dissertation, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, July 2003.

[13] T. Barfoot, Online Visual Motion Estimation using FastSLAM with SIFT Features. In Proc. of the Int, Conf, on Intell. Rob. and Systems (IROS), Edmonton, Alberta, August 2-6, 2005.

[14] R. E. Kalman, A new approach to linear filtering and prediction problems, Trans. ASME, J. of Basic Eng., vol. 82, pp. 35-45, 1960.

[15] R. Zabih and J. Woodfill. Non-parametric local transforms for computing visual correspondence. In Proceedings of European Conference on Computer Vision (ECCV) Volume 2, pages 151–158, Stockholm, Sweden, 1994.

[16] G. Roth and E. Wibowo. An efficient volumetric method for building closed triangular meshes from 3-d image and point data. In Proceedings of Graphics Interface (GI), pages 173–180, Kelowna, B.C., Canada, 1997.

[17] S. Se, H. Ng, P. Jasiobedzki and T. Moyung. Vision based modeling and localization for planetary exploration rovers. In Proc. of Int. Astronautical Congress (IAC), Vancouver, Canada, Oct., 2004.

[18] B. Hoffman, E.T. Baumgartner, T.L. Huntsberger, and P.S. Schenker. Improved rover state estimation in challenging terrain. Autonomous Robots, 6:113–130, 1999.

[19] C.F. Olson and L.H. Matthies. Maximum likelihood rover localization by matching range maps. In Proc. of Int. Conf. on Rob. and Aut., pages 272–277, Leuven, Belgium, May 1998.

[20] A. Mallet, S. Lacroix, and L. Gallo. Position estimation in outdoor environment using pixel tracking and stereovision. In Proceedings of IEEE International Conference on Robotics and Automation (ICRA), pages 3519–3524, April 2000.

[21] S. Lacroix, A. Mallet, D. Bonnafous, G. Bauzil, S. Fleury, M. Herrb, and R. Chatila. Autonomous rover navigation on unknown terrains: functions and integration. Int. J. of Robotics Research, 21(10–11):917–942, Oct.-Nov. 2002.

[22] S. Lacroix and A. Mallet. Integration of concurrent localization algorithms for a planetary rover. In Proc. of Int. Symp. on Art. Intell. Rob. and Aut. in Space: i-SAIRAS, St-Hubert, Quebec, Canada, June 2001.

[23] K. Nickels and E. Huber. Inertially assisted stereo tracking for an outdoor rover. In Proceedings of IEEE Int.Conf. on Rob. and Aut. (ICRA), pages 3078–3083, Seoul, Korea, May 2001.

[24] C. F. Olson, L. H. Matthies, M. Schoppers, and M. W. Maimone, Rover navigation using stereo ego-motion, Rob. and Autonomous Systems, vol. 43, no. 4, pp. 215-229, 2003.

[25] D. M. Helmick, Yang Cheng, D. S. Clouse, L. H. Matthies, S.I. Roumeliotis, Path Following using Visual Odometry for a Mars Rover in High-Slip Environments, In Proc. IEEE AERO 2004.

[26] D. Nister, O. Naroditsky, and J. Bergen, Visual odometry, Washington, DC: Proceedings of the IEEE Computer Society Conf. on Comp. Vision and Patt. Recog. (CVPR), June 2004.

[27] N. Karlsson and E. Di Bernardo, The vSLAM Algorithm for Robust Localization and Mapping Barcelona, Spain: IEEE Int. Conf. on Rob. and Aut. (ICRA), April 2005.

[28] M. Vergauwen, M. Pollefeys, and L. Van Gool. A stereo-vision system for support of planetary surface exploration. Machine Vision and Applications, 14:5–14, 2003.

[29] S. Se, D. Lowe and J. Little, Vision-based Global Localization and Mapping for Mobile Robots, IEEE Transactions on Robotics, Volume 21, Issue 3, pages 364-375, June 2005.