VISION BASED MODELING AND LOCALIZATION FOR PLANETARY EXPLORATION ROVERS

Stephen Se, Ho-Kong Ng, Piotr Jasiobedzki, Tai-Jing Moyung

MD Robotics
9445 Airport Road
Brampton, Ontario
Canada L6S 4J3
{sse,hng,pjasiobe,tmoyung}@mdrobotics.ca

Abstract

Exploration of large unknown planetary environments will rely on rovers that can autonomously cover distances of kilometres and maintain precise information about their location with respect to local features. During such traversals, the rovers will create photo-realistic three dimensional (3D) models of visited sites for autonomous operations on-site and mission planning on Earth. Currently rover position is estimated using wheel odometry, which is sufficient for short traversals but as error accumulates quickly, it is unsuitable for long distances. At MD Robotics, we are working on imaging technologies for future planetary rover missions. Two complementary technologies are currently investigated: a stereo based vision system and a scanning time-of-flight LIDAR system. Both imaging systems have been installed on board of two experimental rovers and tested in laboratory and outdoor environments. With stereo cameras, the rover can create photo-realistic 3D model as well as provide visual odometry that is more accurate than the rover dead reckoning. With the LIDAR, the rover can match 3D scans to estimate the relative location to improve the wheel and visual odometry.

1 Introduction

Future planetary rovers will have to operate autonomously with minimum operator supervision due to transmission delay, limited bandwidth and intermittent communication. A prerequisite for autonomous operation is the ability to sense and perceive the environment to react appropriately and to make the right decisions.

Among various perception tasks, the ability to estimate the rover position using locally sensed data is essential for navigation as errors in wheel odometry grow quickly. It prevents the autonomous rover from getting lost, and allows the rover to reach required destinations, perform structured exploration and recognize previously visited places. Localization with respect to an absolute coordinates system is possible by matching locally sensed data with remotely sensed data from satellites.

At MD Robotics, we are working on two complementary imaging technologies for future planetary rover missions. A stereo based vision system and a scanning time-of-flight LIDAR system have been integrated on board of two small rovers.

The stereo vision system creates photo-realistic 3D surface models from sequences of images obtained from cameras mounted on a moving rover. Computed camera motion is used as visual odometry to complement the rover dead reckoning for accurate localization. The prototype system can run in near real time using a combination of dedicated hardware and software.

Scanning LIDAR systems offer much longer operational range (100s of metres) and accuracy than stereo systems (10s of metres). Therefore, they allow better planning and execution of autonomous rover operations. This comes, however, with an additional mass and power requirements. We have developed a rover localization technique that uses multiple LIDAR scans to estimate the rover relative location. The scans are obtained from different locations and automatically matched. Multiple scans are registered together to provide a 3D model of visited sites.

Section 2 gives a brief literature survey on visual odometry for planetary rovers. Section 3 overviews

the MDR rover system architecture and operations. The stereo vision system and the LIDAR system are described in Sections 4 and 5 respectively. Experimental results with both systems in various environments are presented in Section 6. Finally we conclude and discuss some future work in Section 7.

2 Related Work

There has been various work on visual odometry for planetary rovers with promising results. Semi-sparse terrain maps were constructed and matched successively to obtain a vision-based state estimate in [4]. 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 [12] 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 [9] to estimate visual odometry in outdoor unstructured terrain, with around 4% error over 25 metres. [7] evaluated a similar algorithm on the Marsokhod robot on many runs totaling several hundreds of metres and achieved about 2% translation error.

[6] 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, [11] discussed the use of inertial sensors to estimate camera ego-motion and to augment stereo tracking on rough terrain. [13] 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.

In our work, we employ both a stereo vision system and a LIDAR system. The visual odometry from stereo vision can complement rover dead reckoning continuously, while the LIDAR scan matching can estimate the relative rover location infrequently to improve the wheel and visual odometry.

Most of the previous work used vision systems for localization only, whereas we also use the vision system for 3D modeling. Recently, [17] 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.

(a) *(b)*

Figure 1: *Rover system with (a) a stereo camera (b) a LIDAR.*

3 MDR Rover System

The MDR R&D rover system shown in Figure 1 consists of a 4-wheel rover developed by the University of Toronto Institute of Aerospace Studies (UTIAS). The batteries of the rover are inside the tires to keep the centre of gravity low. We have developed other hardware and software components to provide the capability of monitoring and controlling the rover in its environment remotely.

3.1 Architecture

A Bumblebee stereo camera from Point Grey Research (PGR)[1] has been integrated into the rover system, as shown in Figure 1(a), with the following features:

- Stereo image capture up to 7 pairs per second
- 8-bit 640x480 greyscale image
- 70 degrees horizontal and 40 degrees vertical field of view
- Firewire interface

To increase the effective field of view of the camera, it is installed on top of a pan-tilt unit, for capturing multiple images.

A LIDAR device, the ILRIS-3D time-of-flight laser rangefinder from Optech[2], has been integrated with the rover, as shown in Figure 1(b). ILRIS-3D can

[1] www.ptgrey.com
[2] www.optech.ca

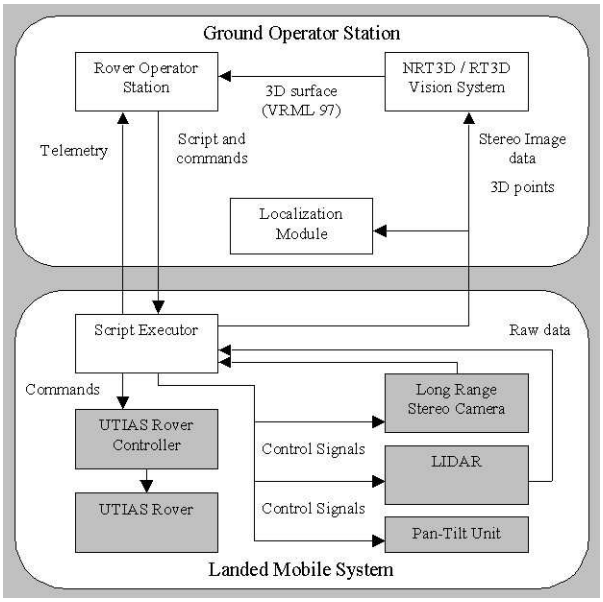


Figure 2: *MDR rover system architecture.*

capture 3D information of the environment, with the following features:

- Data capture rate up to 2000 points per second
- Each data point consists of X, Y, Z coordinates and intensity value
- Data point accuracy in the range of 5 mm
- 40 degrees horizontal field of view

To increase the effective field of view of the LIDAR, a turntable has been installed on the rover to rotate the LIDAR and create panoramic scans by registering multiple scans together.

There are four major software modules in this system: Rover Operator Station (ROS), Non-Real-Time 3D (NRT3D) Vision System, Script Executor and the Localization module. ROS, NRT3D and the Localization module run on a Ground Operator Station (GOS) computer with dual Pentium IV 2.4GHz processors. The Script Executor runs on the rover computer with a Pentium III 700 MHz processor. The two computers communicate via wireless ethernet and all modules communicate with each other through TCP/IP. Figure 2 shows a graphical view of the whole system architecture.

The ROS is a standalone graphical user interface station for the operator to create mission scripts, rehearse and execute these scripts. The operator also observes the rover's status and its environment remotely. It is based on the Remote Tele-operation of Robotics (RTR) Operator Station framework [14], which is designed for scripted control of robotic equipment from a remote location.

A screenshot of ROS is shown in Figure 3. The left window displays a 3D view of the scanned environment and a virtual model of the rover placed at its estimated location. The right top window displays a Digital Elevation Map, a top down view of the scanned scene. The white icon represents the rover location and the map is colour coded to indicate the height. The right bottom window displays the rover telemetry including: estimated location and heading, orientation of the pan and tilt unit, rover attitude, state of the batteries and readings obtained from sonars. Telemetry is updated automatically with information received from the rover.

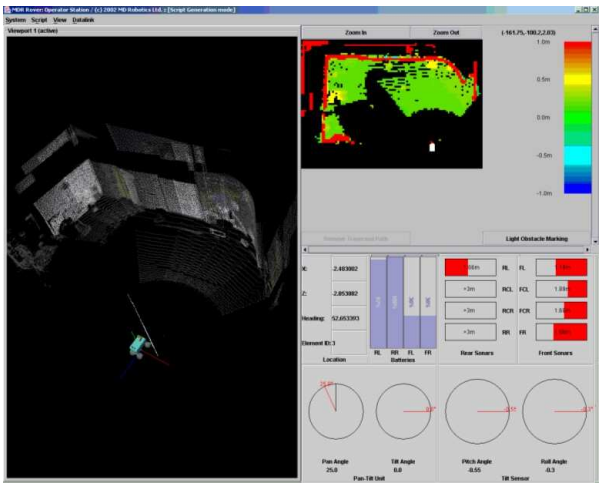


Figure 3: *Rover Operator Station user interface.*

With the stereo camera attached to the rover, the NRT3D Vision System takes the stereo image data and generates 3D models of the rover's environment. The images may be acquired from a stationary or moving rover. In the case where a LIDAR is attached to the rover, the Localization module analyses the 3D data captured by the LIDAR and estimates the motion of rover between scans. In the current architecture, vision processing is performed on the GOS computer due to the limited computational resources available on the rover. Therefore, the transfer of images or LIDAR scans via the wireless ethernet is a bottleneck of the current system.

The Script Executor [10] on the rover controls all the physical components of the rover and drives the rover in the way the user commands. It receives and executes scripts and commands from ROS. It also sends the telemetry and image or LIDAR data back to ROS and NRT3D.

The rover system can use different ways to estimate its current position and orientation. It can measure how much each wheel has traveled from the wheel encoder readings and calculate the 2D motion of the rover. With the stereo camera, NRT3D can track image features found in the stereo images captured at run-time and use them to estimate the 3D motion

of the rover. In the case where a LIDAR is used, the Localization module can find out the positional and orientation information by aligning the 3D points generated by the LIDAR.

3.2 Operations

The MDR R&D rover system can be used in situations where there is no prior knowledge of the rover's working environment. The rover uses stereo camera, LIDAR and sonar to sense the environment, and builds 3D model of the environment during execution. The steps of this process are as follows:

1. The rover scans the environment using stereo camera or LIDAR and transmits the raw data to the GOS computer
2. 3D models of the scene are created and sent to ROS for display
3. Depending on the configuration:
   - With the stereo camera, NRT3D computes the visual odometry to update rover's wheel odometry
   - With the LIDAR, the Localization module aligns two LIDAR scans and estimates the relative pose between these locations
4. Operator plans a script as a series of way-points through ROS
5. Operator verifies the script by rehearsing it in a simulator and transmits it from ROS to the rover
6. Script Executor running on the rover receives and executes the script
7. The rover moves by following the specified way-points using wheel odometry
8. Telemetry is transmitted from the rover to the GOS computer
9. Repeat the steps above.

4 Stereo Vision System

Visual odometry relies on extracting tie points in image sequences and computing their motion between frames. Extracting and matching image corners [3] is less computationally intensive, but the appearance of corners changes depending on the viewpoint and corners can only be tracked between a small number of frames, which increases cumulative error. Using higher level features enables matching the same features over longer sequences, which reduces this error at the expense of additional computation.

Tracking features in monocular images provides the motion estimate up to an unknown scale factor and the solution has singularities for some trajectories or scenes. Using stereo images is better as it provides Euclidean metrics and is always well conditioned regardless of the motion and scenes.

4.1 System Architecture

Figure 4 shows the architecture of the NRT3D Vision System. Images are captured by the stereo camera and we compute dense stereo to obtain 3D data. The system does not require any external sensors for computing the motion as it automatically extracts and tracks natural tie points in the images. The recovered camera motion is used to integrate 3D data computed using dense stereo algorithms from the sequences. The 3D data is converted to surface meshes, which are enhanced by mapping the texture.

This 3D modeling process achieves significant data compression, typical from 25:1 to 100:1, allowing the transfer of data as compact meshes instead of raw images.

4.2 Dense Stereo

In our earlier work, we used the 3DAware PCI card from Tyzx [3] for dense stereo computation. It consists of a DeepSea chip, which is an optimized hardware implementation of the Census stereo algorithm [18]. 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 its own stereo camera and for images up to 500x450 resolution. We now use the higher quality Bumblebee stereo camera from PGR with 640x480 image resolution. We run PGR's optimized Triclops library for correlation-based dense stereo on the processor.

4.3 SIFT Features

We chose a high level set of natural visual features called Scale Invariant Feature Transform (SIFT) as the tie points to compute visual odometry. SIFT was developed by Lowe [8] for image feature generation in object recognition applications. 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 landmarks for robust matching when the cameras are moving around in an environment, as the landmarks are observed from different angles, distances or under different illumination.

The SIFT features are determined by identifying repeatable points in a pyramid of scaled images. Feature locations are identified by detecting maxima and minima in the Difference-Of-Gaussian pyramid. A subpixel location, scale and orientation are associated with each SIFT feature.

Sufficiently distinctive features are required to match scenes in the map. In order to achieve high

[3] www.tyzx.com

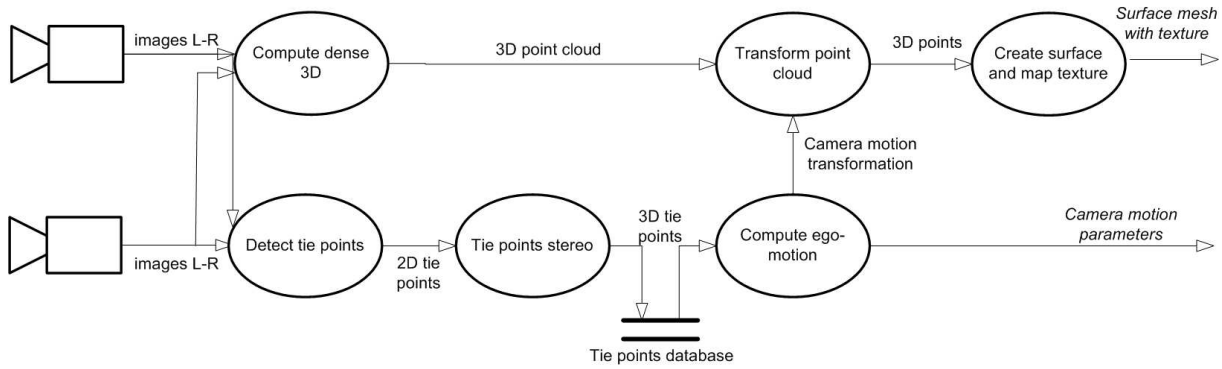


Figure 4: *Stereo vision system architecture.*

specificity, a local feature vector [8] is formed by measuring the local image gradients at a number of orientations in coordinates relative to the location, scale and orientation of the feature.

4.4 FPGA Implementation

The high computational requirements of vision algorithms often limit the amount of distance and science 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 Array (FPGA) for some intensive image processing to offload the processor.

For this work, we have implemented SIFT extraction on a Virtex II Xilinx FPGA as it is 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. A high level environment was needed.

System Generator 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 a 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.

(a) *(b)*

Figure 5: *(a) Stereo matched SIFT features. (b) SIFT features that are matched to the database.*

Since the software uses floating point operations, testing was required to convert the software implementation to work with fixed point operations. Furthermore, many of the routines in the software version needed to be modified to make the hardware implementation efficient.

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.

4.5 Visual Odometry

The SIFT features in the left and right images are stereo matched using the following criteria: epipolar constraint, disparity constraint, orientation constraint, scale constraint, local feature vector constraint and unique match constraint [16]. The subpixel disparity for each matched feature can then be computed. The stereo matched SIFT features in a lab scene are shown in Figure 5(a), where the length of the line is proportional to the disparity. We can see that all the matches are consistent and correct. Typically, we obtain hundreds of SIFT 3D features.

Subsequently we can compute the 3D position (X, Y, Z) of each stereo matched SIFT feature, us-

ing the following equations:

$$X = \frac{(u-u_0)I}{d}; \qquad Y = \frac{(v_0-v)I}{d}; \qquad Z = \frac{fI}{d}$$

where (u, v, d) are the SIFT image location and disparity, (u_0, v_0) are the image centre coordinates, I is the interocular distance and f is the focal length.

Since the rover moves in very rough terrain, we would like to recover the 6 degrees of freedom visual odometry, i.e., camera ego-motion, when the cameras are moving in 3D. We employ a Simultaneous Localization And Mapping (SLAM) approach that uses the SIFT features to localize and simultaneously build a database map [16]. Instead of frame to frame matching, we match the SIFT features at each frame with the database to reduce error accumulation. [13] reported a 27.7% reduction in navigation error when multi-frame tracking is used, rather than considering each pair of frames separately.

The local image vector is used to match SIFT features, which are highly distinctive to facilitate matching with very few false matches. We can then find the camera movement that would bring each projected SIFT feature into best alignment with its matching feature. A weighted least squares procedure is carried out taking into account the feature uncertainty. Figure 5(b) shows the SIFT features that are matched to the database. The line connecting the previous position to the current position is analogous to optical flow and we can see that all the matches are consistent and correct.

4.6 3D Modeling

As the rover moves around, dense 3D data is obtained relative to the camera position at each frame. The 3D data is backward transformed to the initial camera coordinates frame based on the estimated camera pose.

To generate triangular meshes as 3D models, we employ a voxel-based method [15], 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 are accumulated into voxels at each frame. It takes a few seconds to construct the triangular mesh at the end, which is dependent on the data size and the voxel resolution.

In order to create a photo-realistic reconstruction of the scene, texture mapping is required. A texture mapped mesh is much more visually appealing and easier to interpret. Greyscale images from the stereo camera are used for texture mapping. In general, other modalities such as colour images or IR images can be used for texture mapping, as long as the colour/IR camera has been calibrated with the stereo vision system.

Figure 6: *An example of a typical LIDAR scan.*

As each triangle may be observed in multiple texture images, we want to select 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. To find the best texture, we look at all the texture images and find the one that gives the largest area upon 2D projection according to the camera pose.

Moreover, we need to take into account any occlusion. For example, if there is an object in front of the triangle, then the image captured when the camera is facing the triangle directly should not be used, as the texture will be for the object in front. In this case, we will need to look for the next texture image that gives the second largest area upon projection.

5 LIDAR System

One advantage of using range data from LIDAR instead of stereo cameras for localization is that 3D measurements are readily available without the need to perform any feature extraction and 3D reconstruction from the images. The accuracy of stereo vision relies heavily on lighting conditions, which can often be unfavourable in an environment like Mars. In addition, the dense nature of LIDAR range data provides a more precise representation of the scene and can result in better registration.

Figure 6 is an example of a typical LIDAR scan. Note the narrow wedge shape that corresponds to the 40 degree field of view of the LIDAR and lower sampling density for surfaces further away.

5.1 Rover Localization

The scan is represented by a set of points in 3D space. Let

$$\mathbf{p}_i(f) = [X\ Y\ Z\ 1]^T$$

represent the location of point i in homogenous coordinates in the scan f, and

$$P(f) = [\mathbf{p}_0(f)\, \mathbf{p}_1(f)\ \cdots\ \mathbf{p}_{N-1}(f)]$$

be the set of points in the scan where N is the total number of points in the scan, then

$$P(f+1) = \begin{bmatrix} R(f+1,f) & \mathbf{t}(f+1,f) \\ \mathbf{O}^T & 1 \end{bmatrix} P(f)$$

where $R(f+1,f)$ is a 3x3 rotation matrix and $\mathbf{t}(f+1,f) = [t_x t_y t_z]^T$ is a translation vector. Together the set $\{R(f+1,f), \mathbf{t}(f+1,f)\}$ describes the relative pose of the LIDAR between f and $f+1$.

The problem of rover localization using 3D range scans is then to estimate the relative pose $\{R(f+1,f), \mathbf{t}(f+1,f)\}$ given $P(f)$ and $P(f+1)$. Then the estimates, $\hat{R}(f+1,f)$ and $\hat{\mathbf{t}}(f+1,f)$, can be used to correct the odometry error and provide a more accurate estimate of the rover's pose.

5.2 ICP Algorithm

Iterative Closest Point (ICP) algorithm was introduced by Besl and McKay [1] to register two sets of 3D data. We employ a variant of the ICP algorithm [5] to estimate the relative rover location by aligning 3D range scans of natural terrain. The algorithm has the following features:

1. Closest point matching is performed between two sets of 3D points. There is no need to pre-process one set of data to generate a mesh representation.
2. The Euclidean distance between two 3D points is the measurement used for determining the nearest neighbour.
3. A kd-tree data structure is used to accelerate the search for the closest point.
4. Rejection of corresponding point pairs as outliers is based on a threshold for maximum distance between the corresponding points. This threshold can be pre-defined or determined dynamically by statistical match filtering.
5. The registration between the two sets of corresponding points is done by least squares estimation where rotation is represented as quaternions.
6. Apart from a maximum number of iterations and a maximum RMS error, the iteration terminates when the convergence rate of the RMS error is sufficiently slow. This criterion is added to optimize between accuracy and computation time.

One of the characteristics of range data is the large number of data points that each scan contains. In order to reduce computation time, the data is first sampled before being passed to ICP. Since the data is collected in the order of the LIDAR scan lines, uniform sampling is a reasonable choice of sampling technique in preserving most of the scene information.

A LIDAR scan of natural terrain typically contains of a large number of data points lying on the surface of the ground. Very often, these points do not provide unique information for the alignment of two range scans, and they even sometimes bias the results due to the unique pie shape of the data. Hence an additional pre-processing step for the registration is to remove the flat surfaces by fitting the data to a plane using Random Sample Consensus (RANSAC) [2]. By removing the points that belong to this plane, it further reduces the size of the data sets to be processed by the ICP algorithm.

The wheel odometry provides an initial estimate, which is required by ICP. Once the relative location is found, multiple 3D scans can be registered together as a complete 3D model.

5.3 Wheel Odometry Correction

To correct wheel odometry, the ICP algorithm was applied to consecutive pairs of scans. The procedure is as follows:

1. The wheel odometry at $f = 1$ is used as initial estimate for the rover's relative pose for ICP, which will give $\hat{R}(1,0)$ and $\hat{\mathbf{t}}(1,0)$.
2. For each consecutive pair of scans, the ICP algorithm refines the estimates, $\hat{R}(f+1,f)$ and $\hat{\mathbf{t}}(f+1,f)$ and use them to correct the rover's pose with respect to $f = 0$, that is, computing $\hat{R}(f+1,0)$ and $\hat{\mathbf{t}}(f+1,0)$.
3. Then for registering $P(f+1)$ and $P(f+2)$, the corrected pose $\{\hat{R}(f+1,0), \hat{\mathbf{t}}(f+1,0)\}$ and the wheel odometry at $f+2$ provide an initial estimate to initialize the ICP algorithm.

This procedure attempts to correct the rover's pose at each location so that errors in the wheel odometry will not accumulate over a long distance of travel.

6 Experimental Results

Most of the experiments with LIDAR were conducted at a local quarry, which allowed us to use the full range of the LIDAR. Experiments with stereo vision system were conducted at the University of Toronto's MarsDome. Additional experiments were conducted at the MDR parking lot and laboratory.

The first system, wheel odometry, uses wheel encoders to estimate the rover location and heading. This system provides continuous estimates of the rover location. The second system, visual odometry, operates while the rover is moving and uses sequences of stereo image to compute the rover motion. This system provides continuous estimates of the rover position. The third system uses range data from LIDAR operating on a stationary rover and scanning the environment. Two scans obtained from two different locations are matched to provide information about

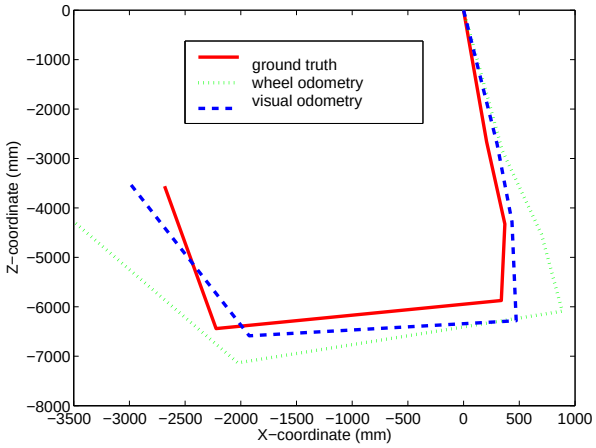


Figure 7: *Wheel odometry, visual odometry and ground truth.*

the rover pose between scans. This system provides discrete rover positions at the scan locations.

Experiments have been carried out to compare these three localization approaches. The obtained measurements have been compared with ground truth measurements collected using a surveying Total Station in the quarry and parking lot experiments.

6.1 Wheel Odometry

The wheel odometry of the rover is computed using the wheel encoder readings in each wheel. The implemented algorithm assumes that the rover travels on a flat surface and all its wheels have the same radius. Several tests on wheel odometry have been performed on different surfaces.

Figure 7 shows the results generated in a test run on the paved surface of the parking lot. The red solid line in the figure indicates the locations of the rover observed by the Total Station and the green dotted line corresponds to the wheel odometry estimation. The results show that the wheel odometry can provide an approximate estimation of the position and heading of the rover. Wheel odometry works best on hard surfaces with good traction when there is minimum wheel slippage.

Figure 8 shows an illustrative example of wheel odometry error accumulation. The rover was following a straight line motion of around 30m across the MarsDome gravel surface. The scans were performed from the stationary rover and successive scans were integrated using wheel odometry only.

Various experiments indicate the rate that the wheel odometry accumulates error depends on the type of surface. For hard, flat surfaces that provide consistent traction for all wheels (lab floor), this rate is small. For surfaces that do not provide good contact (MarsDome gravel), the orientation error reaches

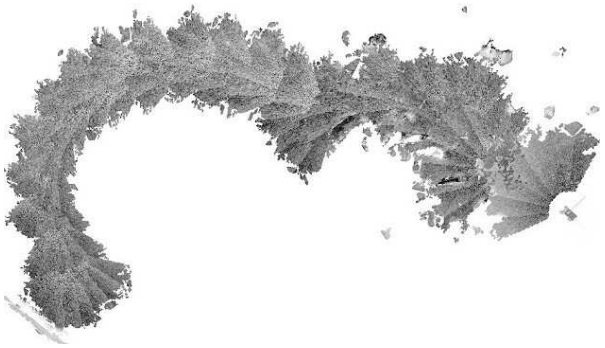


Figure 8: *Wheel odometry error during straight line motion of 30m on MarsDome gravel surface.*

Figure 9: *A screenshot of a 3D model created in the lab.*

180 degrees for 30m traveled. It is expected that for soft surfaces (sand), the error accumulation rate is even higher.

6.2 Stereo Vision

The blue dashed line in Figure 7 shows the rover position estimate generated by visual odometry. The results show that visual odometry can be used to estimate the position and heading of the rover.

The cumulative error in the estimated position over the distance of 4.5 m for wheel odometry is over 24% and for visual odometry 3.6%. These results are consistent with other experiments conducted and similar to results reported in [9, 7]. Thus visual odometry is a viable alternative to wheel odometry for rover localization.

Apart from visual odometry, the NRT3D Vision System also generates a photo-realistic 3D model. Figure 9 shows a screenshot of a 3D model reconstruction of a lab environment using the stereo vision system.

By using a combination of dedicated hardware and software, NRT3D can achieve near real time performance. With our earlier Tyzx stereo camera of 500x450 resolution, the pure software system runs at 2 Hz. With the Tyzx dense stereo card, it improves to 4 Hz. With both the Tyzx dense stereo card and the SIFT FPGA, the system runs at 7 Hz.

Figure 10: *Rover experiments at a local quarry.*

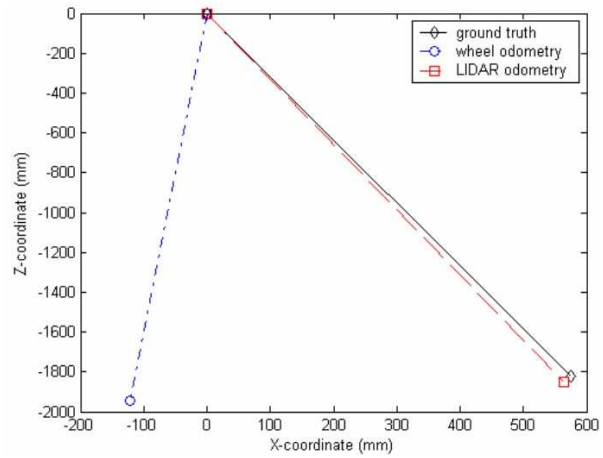


Figure 11: *Estimated rover path for LIDAR test 1.*

6.3 Scanning LIDAR

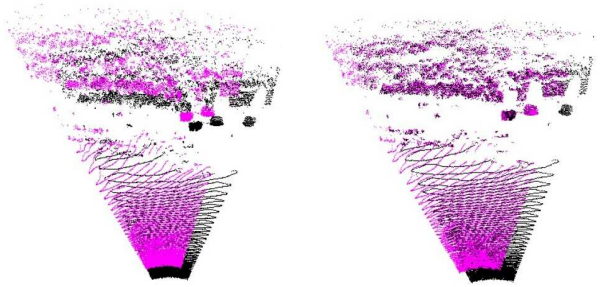
During experiments conducted at a local quarry, as shown in Figure 10, range data from the LIDAR installed on the rover and ground truth locations of the rover were collected. The rover was operated from the ROS and executed scripts generated by the operator.

We have collected sets of single 40 degree scans taken at several way-points along the rover's path. Figure 11 compares the test results of using wheel odometry and using LIDAR localization. It can be seen that the rover path estimated with LIDAR is much closer to the ground truth.

In addition, Figure 12(a) and Figure 12(b) show the aligned scans before and after applying a correction computed by the ICP algorithm. The improvement can be seen qualitatively by inspecting the aligned scans.

Another LIDAR test result is shown in Figure 13 and it indicates that the pose estimate using LIDAR data has not improved over the wheel odometry.

The ICP algorithm finds corresponding point pairs and computes a least squares estimate of the rigid transformation between them, so the accuracy de-



(a) *(b)*

Figure 12: *Alignment of two LIDAR scans (a) with wheel odometry (b) after ICP correction.*

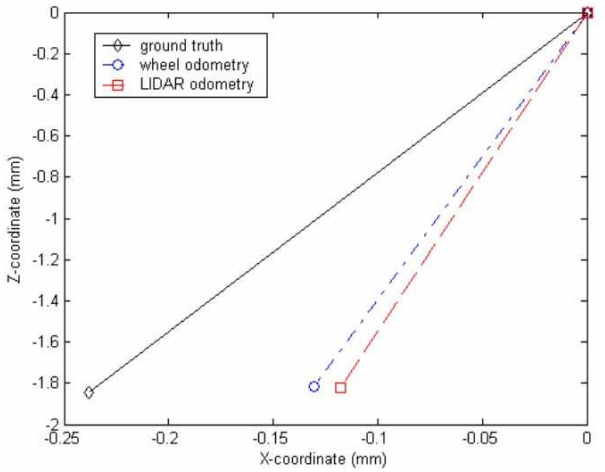


Figure 13: *Estimated rover path for LIDAR test 2.*

pends on a number of factors:

1. The initial estimate used to initialize the ICP algorithm
2. The number, location and orientation of geometrically identifiable features in the scene
3. The size of the overlapping region between consecutive scans

Since the initial estimate of the 3D pose is given by the wheel odometry, which may not be very accurate, and sometimes even erroneous, more work is needed to investigate in techniques to improve the initial estimate of the pose.

Moreover, in this experiment, the motion of the rover was such that the size of the overlapping regions between scans was very small, due to the narrow field of view of the LIDAR. Therefore, there are not enough common elements between the pairs of scans for establishing a match. By combining several scans of the scene with the turntable at every rover location, this will provide a wider field of view for matching.

The rover platform used in the experiments proved useful in laboratory and some outdoor tests. How-

ever, the load of the commercial rangefinder did not allow the rover to move reliably on soft surfaces such as gravel in the MarsDome. The size and mass of the LIDAR required us to lower the rover suspension and the full range of the LIDAR was restricted by this low vantage point.

6.4 Discussions

There are various trade-offs between using stereo and LIDAR based localization systems for planetary rovers.

An advantage of LIDAR is in the much longer operating range and depth of field as compared with stereo vision systems. The ability to operate without reliance on ambient illumination simplifies planning rover operations. However, LIDAR requires much longer acquisition time making it difficult to operate from moving vehicles. The fast scanning mechanisms inherent in LIDAR require additional power.

The computational cost of processing LIDAR data is lower than in the case of stereo, which requires an additional step of finding corresponding regions. The mapping from internal coordinates systems to 3D is of similar complexity for both technologies.

The disadvantage of using LIDAR is that it requires more time (seconds to minutes) to scan the scene than to capture a camera image. For stereo vision, images can be captured continuously while the rover is moving. As the changes in the scene are minimal between successive frames, the problem of non-overlapping regions in the scene is not significant.

On the other hand, it is only practical to perform a LIDAR scan every so often and the rover has to remain stationary over the course of the scan. In a typical operation scenario, a scan is performed while the rover is at the current location. Then it is commanded to move to the next location and another scan is obtained there. Since the motion between scans is larger, the wheel odometry may contain a large error and provide a poor initial estimate for the ICP algorithm. Moreover, significant changes in the scene could present another set of challenges for multiple view registration.

7 Conclusions

Conducted experiments have demonstrated operation of a rover with two different imaging systems: stereo and LIDAR based. The stereo vision system has been used for computing visual odometry as well as reconstructing photo-realistic 3D models. The LIDAR based system has been used for rover localization by matching 3D scans. Performance of both localization systems has been compared with wheel based odometry system and ground truth measurements.

Figure 14: *New rover with on-board vision processing capability.*

The experiments have shown that wheel odometry can be used only for short traversals as the error accumulates quickly. The accumulation rate depends on the type of surface and is lower for hard surfaces. Even for such surfaces, it may reach over 20% of the distance traveled. We observed orientation errors of 180 degrees for 30m traversals on rough surfaces, such as MarsDome gravel.

Visual odometry allows estimation of position and orientation better, as it accumulates distance error at a rate of several percent. Performance of visual odometry depends on the presence of visual features in natural environments. Performance of localization by matching LIDAR scans depends on the presence of 3D features (rocks, mountains) that can be detected in successive scans and matched.

We are currently moving the vision processing to an on-board computer of an ATRV Jr., a larger rover as shown in Figure 14, which will allow more extensive outdoor tests in the future. This will reduce requirements on transmitting all data (images or LIDAR scans) to the GOS computer for processing and enable more autonomous operations as the position estimates will be available to the rover. The bandwidth requirements and communication delays will be reduced as well.

Future work includes investigating the synergies between the stereo and LIDAR technologies by deploying both systems on the larger rover so that they can complement each other. This will lead to successful long range autonomous operations.

Acknowledgements

We would like to thank UTIAS, Tim Barfoot, Shawn Greene, Nazar Abbas, Chris Jen, Kirk Harasym and Lucas Szajek for their contributions and useful discussions. The project was partially funded

by Technology Partnerships Canada and the experiments at the quarry were carried out under the MSL program funded by the Canadian Space Agency.

References

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

[2] M.A. Fischler and R.C. Bolles. Random sample consensus: a paradigm for model fitting with application to image analysis and automated cartography. *Commun. Assoc. Comp. Mach.*, 24:381–395, 1981.

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

[4] 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.

[5] P. Jasiobedzki, J. Talbot, and M. Abraham. Fast 3d pose estimation for on-orbit robotics. In *Proceedings of International Symposium on Robotics (ISR)*, pages 434–440, Montreal, Canada, May 2000.

[6] S. Lacroix and A. Mallet. Integration of concurrent localization algorithms for a planetary rover. In *Proceedings of International Symposium on Artificial Intelligence and Robotics and Automation in Space: i-SAIRAS*, Canadian Space Agency, St-Hubert, Quebec, Canada, June 2001.

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

[8] D.G. Lowe. Object recognition from local scale-invariant features. In *Proceedings of the Seventh International Conference on Computer Vision (ICCV'99)*, pages 1150–1157, Kerkyra, Greece, September 1999.

[9] 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, San Francisco, April 2000.

[10] H. Ng. Software architecture of the R&D rover system. Technical Report TN-R&D-2002-050, MD Robotics, 2002.

[11] K. Nickels and E. Huber. Inertially assisted stereo tracking for an outdoor rover. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, pages 3078–3083, Seoul, Korea, May 2001.

[12] C.F. Olson and L.H. Matthies. Maximum likelihood rover localization by matching range maps. In *Proceedings of International Conference on Robotics and Automation*, pages 272–277, Leuven, Belgium, May 1998.

[13] C.F. Olson, L.H. Matthies, M. Schoppers, and M.W. Maimone. Robust stereo ego-motion for long distance navigation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR) Volume 2*, pages 453–458, South Carolina, June 2000.

[14] J. Richmond. ROSA ground station: Implementation of the RTR operator station. Technical Report TN-R&D-2002-032, MD Robotics, 2002.

[15] 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.

[16] S. Se, D. Lowe, and J. Little. Mobile robot localization and mapping with uncertainty using scale-invariant visual landmarks. *International Journal of Robotics Research*, 21(8):735–758, August 2002.

[17] 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.

[18] 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.