^{3D}LASSO: REAL-TIME POSE ESTIMATION FROM 3D DATA FOR AUTONOMOUS SATELLITE SERVICING

S. Ruel[(1)], C. English, M. Anctil and P. Church

[(1)] Neptec Design Group Ltd., 302 Legget Drive, Ottawa, Ont, Canada, sruel@neptec.com

ABSTRACT

The recent development of space flight ready 3D sensors, such as the Neptec Laser Camera System (LCS), allows 3D vision technology to be considered for autonomous missions. These active sensors provide their own illumination and have a small instantaneous field of view, making them immune to dynamic lighting. Harsh and dynamic lighting conditions have severely limited the use of 2D passive camera based space vision systems for mission critical applications. Autonomous robotic servicing missions, such as the Hubble Rescue Vehicle (HRV), will require vision systems that are capable of providing high accuracy pose estimates in real-time while being robust to changes in lighting conditions.

This paper describes the 3-Dimensional LCS Algorithms for Spacecraft Servicing On-orbit (^{3D}LASSO) system currently under development at Neptec. The project is funded by the Canadian Space Agency (CSA) under the Space Technologies Development Program (STDP). The ^{3D}LASSO system is designed to perform real-time tracking and 6 degree of freedom pose estimation of target spacecraft(s) from sparse and noisy 3D data. The approach is compatible with any sensor capable of providing 3D data. The algorithms have been successfully tested with Neptec's LCS in a variety of test scenarios. Tracking was performed using the random access capability of the sensor which is used to perform rapid, sparse sampling of the target object(s). The data obtained is aligned to a reference model of the target(s) using a newly developed faster version of the Iterative Closest Point (ICP) algorithm developed at Neptec. The pose estimate obtained is then used to compute the trajectory of the object(s).

Keywords: 3D, pose estimation, tracking, triangulation, LCS, on orbit, rendezvous, docking

1. INTRODUCTION

To date, the use of machine vision system for space robotics has been fairly limited. Extreme lighting conditions introduces significant issues to vision systems using video cameras. Even under these extreme conditions, machine vision systems can be successfully used in space provided that the operational conditions can be properly predicted and/or controlled. For example, the Space Vision System (SVS) developed by Neptec has been successfully used on over a dozen International Space Station (ISS) assembly missions. The SVS system used the video system of the Space Shuttle and/or Space Station to provide Shuttle arm operators with very accurate alignment cues during the installation of a new module. The system relied on cooperative targets installed on the station modules and surveyed pre-flight.

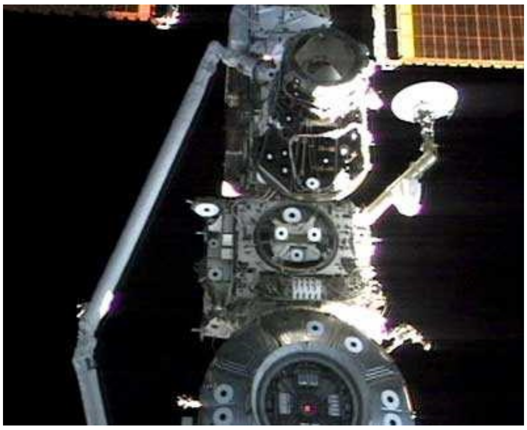

Fig. 1 Space Vision System (SVS) install view

Unfortunately, tasks such as autonomous rendezvous and docking, planetary probe landing and robot navigation require higher levels of autonomy. A vision system for these applications would need to be operational under a wider range of conditions. A system like SVS relied on human operation. The system could only be operational 80% of the time to avoid day/night transitions. As a result, proper mission planning was critical. Applications like satellite servicing deal with a non-cooperative objects that are already in orbit. Therefore, installing reference targets on the target spacecraft is not a possibility.

The recent development of flight certified three-dimensional sensors, such as the Neptec Laser Camera System (LCS), allows 3D vision approaches to be considered for autonomous missions. These active sensors provide their own illumination and have a small instantaneous field of view, making them immune to dynamic lighting. Since these sensors provide 3D data directly, there is no need to use reference markers on the target object(s).

This paper presents the 3-dimensional LCS Algorithms for Spacecraft Servicing On-orbit (^{3D}LASSO) system under development at Neptec. This system is designed to take full advantage of the extra information provided by 3D sensors such as Neptec's LCS to perform real-time tracking of non-cooperative targets for satellite servicing applications. Results from simulated data and real-time LCS data are also presented.

2. ^{3D}LASSO SYSTEM OVERVIEW

The 3-dimensional LCS Algorithms for Spacecraft Servicing On-orbit (^{3D}LASSO) project is co-funded by the Canadian Space Agency (CSA) under the Space Technologies Development Program (STDP). The objective is to design software that can perform real-time tracking and 6 degree of freedom pose estimation of target spacecraft(s) from 3D data.

2.1 System architecture

The ^{3D}LASSO system is made of three components: a system interface, a sensor interface and a tracking module (Fig. 2).

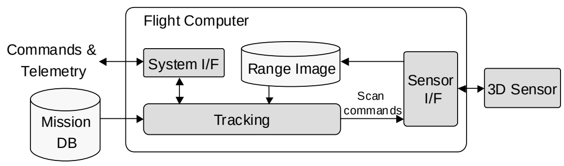

Fig. 2 High Level System Architecture

The system interface is responsible for accepting external commands and sending status and telemetry information. The heart of the ^{3D}LASSO system is the tracking module. The tracking module performs tracking and pose estimation of a target from data acquired with a 3D sensor. All mission specific configurations are stored in a mission database used by the tracking module. This module is portable and can be embedded in a flight computer.

The sensor interface layer accepts generic scan commands from the tracking module and converts them into sensor specific commands. The raw data acquired by the sensor is converted into a range image and sent to the tracking module. A range image, also called telemetric or 2.5D image, is defined as a list of 3D points (voxels) organized in a two dimensional grid. In contrast to a point cloud, a range image only provides information about the surface that the sensor can "see". This is typical of most 3D sensors which are perspective based and cannot penetrate the target surface (unlike radars or MRI, for example). Any sensor capable of generating a range image can provide data to the tracking system, making it sensor independent.

2.2 Mission Design and Analysis Tool (MDAT)

MDAT is a Windows application used for mission design, analysis and simulation. It provides an interface that allows the tracking module to be tested in various configurations. The application can be used in either simulated or real-time tracking mode. MDAT is made of four components: a mission database editor, a simulation environment, sensor interfaces and ^{3D}LASSO's tracking module (Fig. 3).

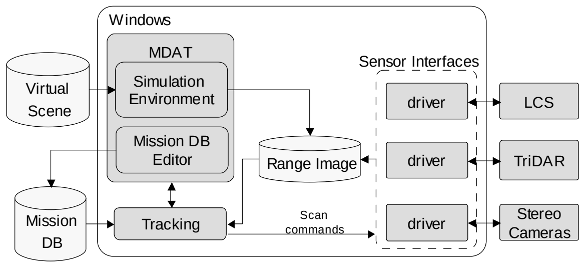

Fig. 3 MDAT Architecture

2.2.1 Mission database editor

The mission database editor allows the user to define the target and secondary targets, reference models, coordinate systems and setup all algorithm parameters. The mission database contains all mission specific configuration information.

2.2.2 Simulation environment

The simulation environment (Fig. 4) performs management and rendering of complex virtual scenes. Virtual scenes can be made up of 3D models, point cloud data, virtual sensors and lights sources arranged in a tree structure.

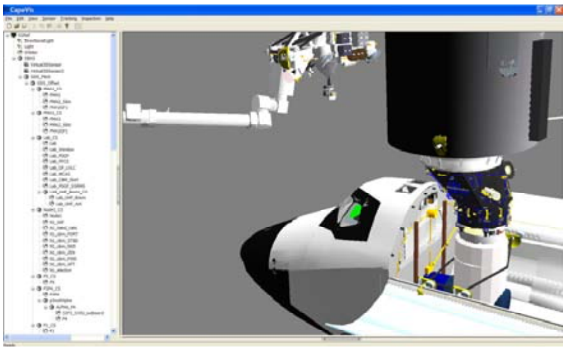

Fig. 4 Simulation Environment

The environment also provides sensor simulation capabilities that can generate synthetic range image data (Fig. 5). The synthetic data generated can be fed into the tracking algorithms to perform tracking simulations (MDAT simulated mode). This provides mission planners with a very powerful tool that can be used to test various operational scenarios and system settings for a particular mission. It can also be used for verification of scenarios that cannot be tested easily in a lab environment. The engine can currently simulate several types of sensors: LCS, stereo cameras and time of flight LIDARs.

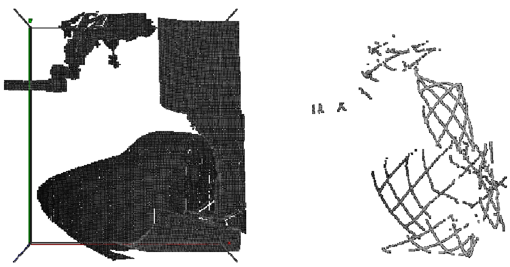

Fig. 5 Simulated sensor data, raster scan (left) and lissajous scan (right)

2.2.3 Sensor Interfaces

The MDAT tool is capable of performing real-time tracking from any sensor for which it has an interface driver. The ^{3D}LASSO common interface is used to send commands to sensors. The MDAT sensor interface is packaged into Windows drivers capable of accepting commands from the tracking module. The commands are then converted into sensor specific commands. The sensor raw data is converted into a range image that is input into the tracking module for pose estimation. MDAT currently supports the following sensors: LCS, TriDAR (Triangulation + LIDAR) and stereo cameras.

3. TRACKING APPROACH

The tracking module operates by constantly acquiring data from a 3D sensor and performing a pose estimate for each scan acquired. 3D sensors provide information about the geometry of a scene directly. Therefore, there is no need to add reference markers on the target. The geometry of the target becomes the reference marker. Pose estimation at each frame can be performed by registering new scan data to data from the last frame. By measuring the displacement between two consecutive frames, the target motion can be estimated. For most space applications, a priori knowledge about the scene object(s) is available. Therefore, pose estimation can be performed by registering new scan data to a reference model of the target. With this approach, the absolute pose of the target object and its trajectory can be obtained. This model-based approach is used by the ^{3D}LASSO system. Registration of the input range image to the reference model is performed using a newly adapted Iterative Closest Point [1,2,3,4] (ICP) algorithm developed by Neptec.

The registration is initiated with an estimate of the relative pose. For space applications, objects move quite slowly and therefore, the last pose estimate can be used as the initial guess. For faster moving objects, rate estimates can be used to predict the target location at the next frame. Once the ICP is initiated, points in the input range image are matched with the closest reference model surface element. The matched coordinates are then assumed to correspond to the same physical location on the object. A least squares algorithm operates on the set of coordinate pairs to calculate the best-fit relative pose estimation, from which a new set of matched coordinates can be found. This process is iteratively repeated until the pose estimation converges or a maximum number of iterations is reached. ICP is robust to range noise because the least-squares fitting of the coordinate pairs tends to average out to the correct pose. One characteristic of range data from 3D sensors (such as LCS or LIDAR) is that, although they can be quite noisy, the noise will generally average to the true shape of the target object. However, this noise makes it difficult to accurately calculate local features, such as curvature or edges [5]. Such local features can appear significantly different when measured from a different perspective or different range. Neptec's approach to ICP optimises the coordinate matching process making the algorithm significantly faster than traditional techniques (ref. Section 6.3).

3.1 Operational Modes

The tracking algorithms operate in 4 different modes: bearing, search, tracking and short-range multi-target tracking. Fig. 6 shows how the tracking algorithm enters each operational mode.

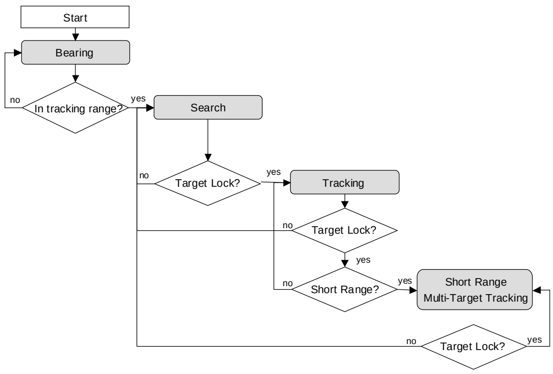

Fig. 6 Operational modes logical flow

Once initiated, the system enters the bearing mode. This mode only provides bearing, range and closing rates of the target relative to the sensor. Bearing is calculated by acquiring full field of view range images of the scene and calculating the centroid of the valid data relative to the sensor. By measuring the motion of the centroid over time, closing rates can be estimated.

The search mode calculates an initial guess of the target pose to initialise tracking. The initial guess can be obtained or calculated in numerous ways depending on mission requirements. Possible approaches are:

1. Object search algorithms like Neptec's Automatic Target Recognition (ATR) algorithms developed for the Canadian Defense [6,7].
2. Handover from long range systems
3. Through spacecraft navigational system
4. A priori scene knowledge (e.g. for space station operations)
5. Manual uplink

Once the target is found, the tracking is initiated. For larger objects, close range tracking is performed by specifying smaller components of the structure as secondary targets. This is configured on a per mission basis during the mission design process. Once in range, the system will cycle through all the targets that are in the field of view of the sensor.

4. NEPTEC LASER CAMERA SYSTEM (LCS)

The Laser Camera System (LCS) developed by Neptec is an auto synchronous triangulation random access scanner. Unlike time of flight 3D sensors that measure the time it takes for a laser pulse to travel to and from a target, LCS uses laser triangulation to infer range. This approach was developed by the National Research Council of Canada (NRCC) [8]. Fig. 7 shows the basic triangulation geometry for an auto synchronous laser scanner. In the simplest form, a laser beam is projected onto a surface and imaged by a detector offset by a baseline. The location of the laser spot image on the detector ("a" or "b") changes with the range of the surface ("A" or "B") as shown in the figure, and therefore the range to the measured point can be calculated from the spot location on the detector.

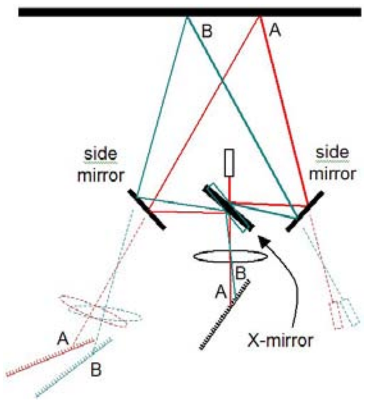

Fig. 7 LCS Triangulation Geometry

The accuracy of this type of sensor is inversely proportional to the square of the distance to the target. Triangulation sensors are therefore better suited for short-range applications that require high accuracy and high data acquisition rates. The space version of LCS uses a 1500nm laser source making it immune to solar interference. It flew on board Space Shuttle Discovery in 2001 on STS-105 [9,10].

Following the Columbia accident, LCS technology was selected to be part of the Orbiter Boom Sensor System (OBSS) that will perform inspection of the Space Shuttle's Thermal Protection System (TPS) before re-entry [11].

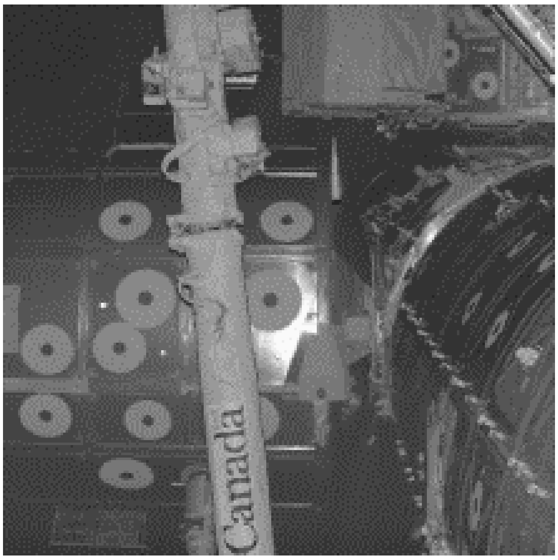

Fig. 8 LCS scan acquire during STS-105 demonstration flight

5. TRACKING WITH LCS

Active sensors like LCS and most LIDARs typically use scanning mirrors to acquire each data point independently. This process can be lengthy if the mirrors need to constantly stop and change direction, like in raster scan patterns. Model based recognition and pose estimation algorithms typically do not require very dense data sets in order to lock on target [6,7]. More important is the amount of geometric information contained in the data. For ICP, a sparse dataset is perfectly acceptable for the algorithm to converge provided the sampling is well distributed over the target surface.

The random access capability of LCS is used to acquire the tracking data at a rate that will allow real-time operations. When tracking with LCS, the algorithm module sends a command to the sensor to acquire the data using a rosette waveform (Fig. 9). Typically, 1024 pts are acquired in less than 200ms allowing for a ~5Hz frame rate. In order to maximize the number of measurements that hit the target, the waveform is aimed at the geometric center of the target. The waveform extents are also adjusted using the reference model to fit tightly around the target.

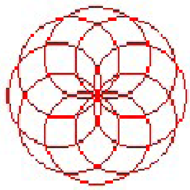

Fig. 9 Rosette Pattern used during LCS tracking

Other waveforms like lissajous can also be used, but the rosette provides better spatial coverage.

6. TEST RESULTS

The short-range accuracy and performance of the algorithm was evaluated using both simulated and LCS data. For lab testing, a 1/10th scale model of the Hubble Space Telescope (HST) was used as the target object. The bottom of the model was covered with silver Teflon tape, the same material covering parts of the HST. The top of the model was covered with aluminium foil. Since an active sensor is being tested, material properties of the target are important to replicate in order to obtain realistic results. A CAD model of the model was used in simulations to generate the synthetic tracking data in the MDAT software.

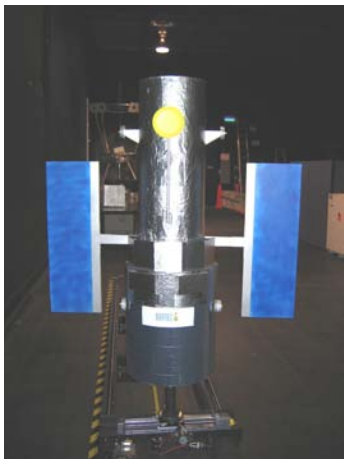

Fig. 10 1/10th scale mockup of Hubble Space Telescope used for lab testing.

6.1 Simulation Results

Six simulations were performed at distances of 2-7m, at one meter increments. At each distance, the object was rotated 1 degree per frame about the cylinder axis. Each simulation ran for 360 frames. Table 1 summarizes the results obtained for all ranges and poses tested.

	Mean Error	Std Dev. of Error
X (mm)	0.53	4.87
Y (mm)	3.33	2.79
Z (mm)	-3.36	6.25
Yaw (deg)	0.01	0.20
Pitch (deg)	-0.004	0.19
Roll (deg)	0.004	0.16

Table 1 Summary of simulation results

The average pose estimation error for the simulated model was well below 1 cm for position and 1 degree for attitude. The results also show a strong correlation with the target pose. The errors are quite small for most viewpoints except for views where the orientation of the target is close to 90 and 270 degrees. This is seen clearly from yaw measurements (Fig. 11)

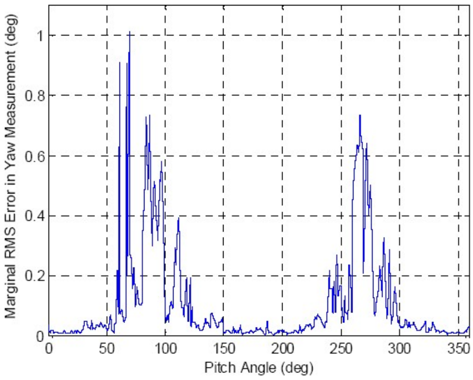

Fig. 11 Pose error in yaw measurements (RMS of all distances)

This phenomenon can be easily explained by looking at the geometry of the target visible from such viewpoints. At 90 and 270 degrees (Fig. 12), the solar panels are perfectly aligned with the sensor's optical axis and therefore not visible. This leaves very few features for the algorithm to "lock" onto and results in higher uncertainty in the measurement of the pose about the cylinder axis.

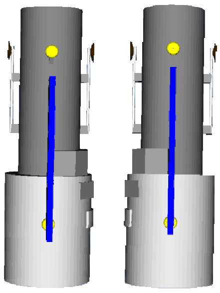

Fig. 12 Hubble model at 90 and 270 degrees

Results also indicate that accuracy is not dependant on the range of the target. This is expected in simulation since no noise was added to the data.

6.2 Lab results with LCS

For lab testing, relative pose errors were measured by computing the difference in rotation angle from discrete target positions at each range step (2,3,4,5,6 and 7m). At each range step, the model was rotated about its cylinder axis on a rotation stage from 0 to 360° in increments of 15°, thus providing relative truth data. At each position and orientation, the pose estimate is logged for 50 frames. The average and standard deviation of the pose over the 50 frames is then computed for every target position. By comparing the truth data with the mean pose change, a measure of relative tracking accuracy for the pitch angle (about the cylinder's axis) can be achieved. The Z error is measured by comparing the truth data to the measured range of the target. Table 2 presents a summary of the results.

	Average Pose Error	Std Dev. of measurement
Z (mm)	5.01	2.62
Pitch (deg)	0.46	0.24

Table 2 Summary of lab results

As seen in simulation results, the average pose estimation error was below 1cm and 1 degree. The results obtained during lab testing indicate that the pose accuracy is dependant on both target geometry and range. In contrast to the simulated data, LCS data noise increases with range which affects pose measurement (Fig. 13). Additional errors in LCS data can also be observed as the target moves outside the LCS calibration volume (LCS was calibrated for the 1-5m range only).

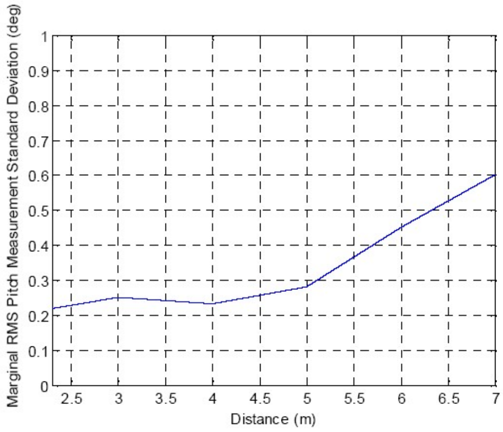

Fig. 13 Pitch error as a function of distance (RMS of all poses)

As shown in simulation testing, the accuracy of the pose measurement is dependant on the target geometry and the point of view to the target. Points of view that have a high quantity of geometric features will benefit from a better "lock" on the target and will therefore be more accurate. Some points of view may have high uncertainty due to a lack of visible features as seen at 90 and 270 degrees (Fig. 14).

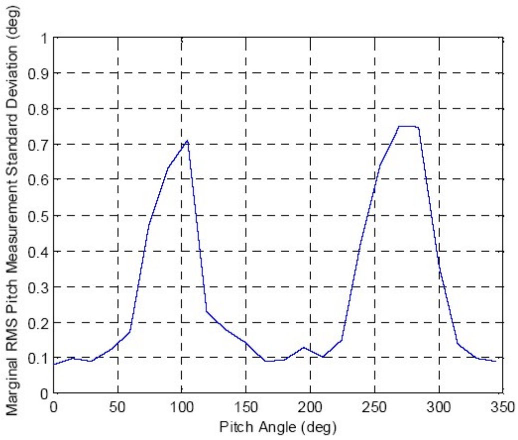

Fig. 14 Pitch measurement standard deviation as a function of pitch angle (RMS of all distance)

An important observation from the results presented is that there is a much stronger correlation between pose error and pitch angle than with object distance. This demonstrates that the algorithm is sensitive to the target geometry but tolerates noise and deviations from the reference model relatively well.

6.3 Processing Speed

All tests were performed using the MDAT software application running on a 1.6Ghz Pentium M laptop. The average algorithm execution time was 9.2ms.

7. REAL-TIME IMPLEMENTATION

As mentioned in section 2, the tracking module is portable and can run on a flight computer. This means that it can also be executed directly by the sensor assuming it has enough processing power. This would provide a complete tracking solution to the end user and also maximise the update rate by eliminating network overhead. It would also relax requirements on the spacecraft's flight computer.

In order to demonstrate this capability, the tracking module was ported to the OBSS - LCS sensor processing platform running the QNX operating system. This flight certified processor card is based on an embedded PowerPC 603e core with 256MB of EDAC memory. Using the same algorithm settings used for simulation and lab testing, the average processing time on this platform was measured at 148ms.

Since the LCS's data acquisition is performed by a pair of DSPs independent of the main processor, it is possible to execute the data acquisition and pose estimation in parallel. Typical LCS 1024 points rosette waveform acquisition time is < 200ms meaning that a final tracking update rate greater than 5Hz can be achieved.

8. CONCLUSION

Netptec is developing the 3-Dimensional LCS Algorithms for Spacecraft Servicing On-orbit (^{3D}LASSO) system under the Canadian Space Agency's Space Technology Demonstration Program (STDP). The algorithms are contained into a fully portable software library and can use any sensor capable of generating range images.

The Mission Design and Analysis Tool (MDAT) provides mission designers with a simulation environment where various mission scenarios and system configurations can be tested and validated. The tool can also perform real-time tracking with various sensors.

The algorithms were tested at short range with both simulated and lab data from the LCS sensor. The target object was a $1/10^{th}$ scale model of the Hubble Space Telescope. The model was positioned over a 2-7m range and rotated 360 degrees about its cylinder axis. This allowed the evaluation of algorithm sensitivity to varying target geometry and sensor noise characteristics. Simulated data was used to evaluate the absolute accuracy and lab data was used to evaluate relative accuracy. Average pose errors smaller than 1cm and 1 degree and as low as a few mm and 0.1 degree were observed. Poses that do not provide enough geometric information to the algorithm showed larger errors. This highlights the dependence of the system to object geometry. This is logical for model-based approaches, as various objects will provide different features to the algorithm. In extreme cases, an object such as a sphere would yield very poor tracking results, as there are no unique features in any orientation. Lab testing has also demonstrated that the pose accuracy is dependant on sensor noise. This was expected since LCS data noise is a function of range to target, sensor calibration limitations, and surface material properties. However, the results obtained clearly show that the viewpoint on the target has much more influence on the pose uncertainty than the range to target. This demonstrates that the geometry of the target dominates the system performance while sensor noise only drives the maximum tracking range for very good geometry. It also demonstrates that the algorithm is quite robust to sensor noise and deviations from the reference model.

Future space exploration initiatives will rely on robotic systems to perform more complicated tasks. As the required level of autonomy increases, vision systems will be expected to operate in a wide range of conditions. The recent development of space flight ready 3D sensors, like the Neptec Laser Camera System (LCS), allows 3D vision approaches to be considered for autonomous missions and eliminate the limitations of 2D machine vision systems.

9. REFERENCES

1. Besl, P. and McKay, N. "A Method for Registration of 3-D Shapes," Trans. PAMI, Vol. 14, No. 2, 1992.
2. Bergevin, R., Soucy, M., Gagnon, H., and Laurendeau, D. "Towards a General Multi-View Registration Technique," Trans. PAMI, Vol. 18, No. 5, 1996.
3. Chen, C., Hung, Y., and Cheng, J. "A Fast Automatic Method for Registration of Partially-Overlapping Range Images," Proc. ICCV, 1998.
4. Johnson, A. and Hebert, M. "Surface Registration by Matching Oriented Points," Proc. 3DIM, 1997.
5. Ferrie, F., and Lagard, J., "On computing stable surface descriptions from range images", Proc. 5th Int. Conf. on Image Analysis, Positano, Italy, Sept. 20-22, 1989.
6. Ruel, S., English, C., Melo, L., Berube, A., Aikman, D., Deslauriers, A., Church, P., and Maheux, J., "Field testing of a 3D Automatic Target Recognition and Pose Estimation Algorithm", SPIE Defense & Security Symposium, Orlando, Florida, April 12-16, 2004.
7. English, C. Ruel, S. Melo, L., Church, P. and Maheux, J., "Development of a Practical 3D Automatic Target Recognition and Pose Estimation Algorithm", SPIE Defense & Security Symposium Proceedings on Automatic Target Recognition XIV, Orlando, Florida, April 2004
8. F. Blais, & al., "Development of a real-time tracking laser range scanner for space application", Proceedings Workshop on Computer Vision for Space Applications, Antibes, France, September 22-24, 161-171 (1993)
9. Samson, C., English, C., Deslauriers, A., Christie, I., and Blais, F., "Imaging and tracking elements of the International Space Station using a 3D auto-synchronized scanner", SPIE AeroSense 2002, Orlando, Florida, April 1-5, 2002.
10. Samson, C., English, C., Deslauriers, A., Christie, I., Blais, F., and Ferrie, F., "Neptec 3D Laser Camera System: From Space Mission STS-105 to Terrestrial Applications", 2002 ASTRO Conference. Ottawa, Ontario, Canada. November 11-15, 2002
11. Deslauriers, A., Showalter, I., Montpool, A., Taylor, R., Christie, I. "Shuttle TPS inspection using triangulation scanning technology", SPIE 2005, Orlando, Florida, April, 2005.