2008 IEEE International Conference on
Robotics and Automation
Pasadena, CA, USA, May 19-23, 2008

High Quality 3D Laser Ranging Under General Vehicle Motion

Alastair Harrison and Paul Newman

Abstract— **This paper describes an end-to-end system capable of generating high-quality 3D point clouds from the popular LMS200 laser on a continuously moving platform. We describe the hardware, data capture, calibration and data stream processing we have developed which yields remarkable detail in the generated point clouds of urban scenes. Given the increasing interest in outdoor 3D navigation and scene reconstruction by mobile platforms, our aim is to provide a level of hardware and algorithmic detail suitable for replication of our system by interested parties who do not wish to invest in dedicated 3D laser rangers.**

I. INTRODUCTION

Laser measurement devices have become popular for use with mobile robotics platforms. They offer accurate range measurements of the vehicle's environment and high data rates. Though computer vision techniques such as stereo vision [1], [2] are increasing in popularity and performance, the laser range finder offers a higher level of robustness and is less dependent on surface texture to produce accurate results.

The SICK LMS200 2D Laser Measurement System is popular in robotics due to its low cost and high accuracy. It is primarily used as a 2D sensor in mapping and sensing applications [3], but has also seen use in 3D data acquisition applications which sweep the scanning plane over the environment. Whilst custom optical solutions are possible [4], there are two more usual techniques for generating 3D measurements. First, the device may be fixed on the vehicle with a vertical scanning plane and the vehicle's motion exploited to move the scanning plane [5], [6]. Alternatively the device may be coupled with an appropriate actuator which allows the scanning plane to be controlled independently of the vehicle's pose [7], [8], [9]. Use in this mode allows the collection of high-fidelity 3D measurements usually reserved for considerably more expensive devices, but the vehicle is required to be stationary during the sweep to ensure good coverage. We describe a system capable of gathering accurate point clouds with the vehicle either stationary, or moving at a useful velocity, by virtue of its rapid sweeping.

Though capable of 0.25° resolution and $15\mathrm{mm}$ accuracy, the LMS200 is designed primarily to be operated as a fixed sensor in industrial settings, so its use in mobile robotics can be problematic, particularly with regard to timing and data latency issues. This can have a significant impact on the quality of the data obtained from the device if not handled with care.

This work was supported by the UK EPSRC CNA program and Guidance Ltd

Both authors are with the Robotics Research Group, University of Oxford, Oxford - OX1 3PJ
{arh,pnewman}@robots.ox.ac.uk

For simplicity, the laser scanning system described in this paper couples the LMS200 with a low cost continuously nodding actuator rather than a servo. This does however present further timing challenges, since the nodding rate can be up to $180^\circ/\mathrm{s}$. It is vital that we have an accurate time stamp on the laser measurements so that the nodder elevation angle may be determined with sufficient accuracy. The usual method of determining data timestamps from the time of arrival in the serial buffer is entirely insufficient for this task, since even a small timing error can produce gross vertical position errors. The error may be estimated as $\zeta \approx r\sin(\tau\frac{d\gamma}{dt})$, where r is the measured range in meters, γ is the nod angle and τ is the timing error. Thus a modest 10 ms delay could cause an error of around $0.3\mathrm{m}$ at $10\mathrm{m}$ range. We shall describe a method of obtaining highly accurate correlation between laser data and nodder elevation angles.

With the vehicle stationary, the system is able to generate high quality point clouds of its environment. Coupled with a *stop-scan-start* paradigm and a 3D scan-matching engine we are able to produce 6DOF SLAM maps [10].

More attractive is the ability for the robot to move at a reasonable pace whilst simultaneously scanning its environment. The rapid nodding rate makes it possible to produce dense surface measurements with the robot travelling at speeds of around $0.5\mathrm{m}/\mathrm{s}$. Over short periods, the odometry can be assumed to be good, and small well registered point clouds constructed. With a skid steer vehicle, odometry errors are introduced when the vehicle turns, resulting in significant corruption of the generated point clouds.

In indoor environments with flat floors, we may employ 2D SLAM schemes to correct the odometry and maintain point cloud integrity. In outdoor environments the problem is much harder to solve. In addition to wheel slip, the vehicle is subject to low amplitude rolling and pitching movements on uneven ground. This results in further corruption of the generated point clouds - particularly when sampling surfaces greater than $10\mathrm{m}$ away. It also renders 2D SLAM schemes ineffective, since the scanning plane is constantly changing.

We present an interesting algorithm for inferring roll, pitch and yaw movements by examining the structure of the point clouds generated by the sensor during vehicle motion. This method is also used to correct yaw inaccuracies caused by wheel-slip.

We begin by discussing the physical set up of the 3D range finder in Section II. In Section III we describe the software calibration techniques employed to ensure accurate timing. Finally we present the algorithm for feature based roll, pitch and yaw correction of dynamically gathered point clouds in Section IV.

978-1-4244-1647-9/08/$25.00 ©2008 IEEE.

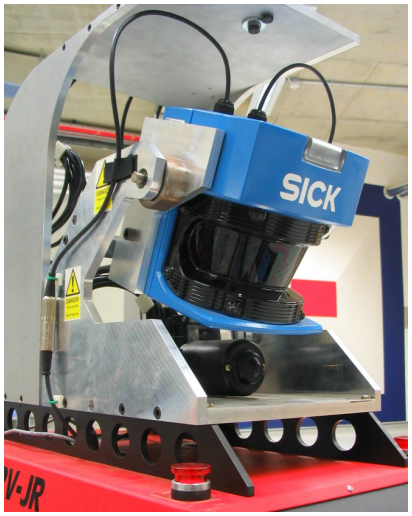

Fig. 1. The nodding 3D laser range finder mounted on a mobile robot

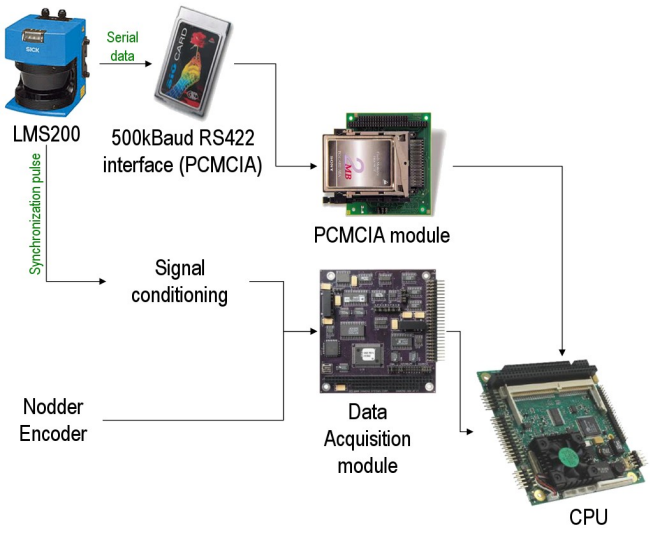

Fig. 2. The data flow for data acquisition. The signal conditioning step takes the 24V synchronization pulse from the LMS200 and converts it into a 5V logic signal that can be used to trigger sampling on the data acquisition module

II. MECHANICAL AND ELECTRICAL ASPECTS

A. Elevation Mechanism

The LMS200 is mounted in a nodding cradle with a quick return mechanism and powered by a motor running from a constant voltage source (ie not exactly constant velocity). The nod elevation profile is near sinusoidal, with a period of $1.2s$. The elevation range is from $44°$ up to $26°$ down, giving good, rapid coverage of the environment ahead. The maximum angular velocity occurs in the middle of the nod sweep and is around $180°/\mathrm{s}$. The apparatus is shown in Fig. 1. The nodder incorporates an analogue encoder for determination of the elevation angle.

The constant nodding scheme was chosen over a servoing scheme for simplicity, though it does place further demands on the data processing implementation, to ensure accurate timing. We shall discuss this in Section III

1) Data Acquisition Hardware: The system described here requires a computer with analogue data acquisition capabilities for sampling the nod angle encoder and a 500kBaud RS422 interface for communication with the LMS200. The precise set-up is not critical, but in this work we use a 1 Ghz Kontron MOPS-lcd-VE based PC104 stack with 1GB RAM, with a Diamond-MM-AT data acquisition module and a CSM PCMCIA RS422 serial card. The operating system is a standard Linux installation; a real-time operating system is not required.

B. Laser Configuration

The SICK LMS200 operates by sending out infra-red laser pulses and measuring the time taken for the reflected light to return. The measured range is proportional to the time of flight of the pulse. Inside the LMS200, the laser beam is reflected by a rotating mirror, allowing measurements to be taken over a $180°$ horizontal fan in front of the device. For the remaining $180°$ of the scan no useful measurements are taken because the mirror directs the beam inside the casing of the LMS200. The mirror rotates at 75Hz and measurements are taken at $1°$ intervals, corresponding to a rate of 13575 measurements per second.

Usually, the LMS200 begins taking measurements when the mirror is in the $0°$ position, stopping at the $180°$ position. Optionally scans may be successively offset by $0°$, $0.25°$, $0.5°$ and $0.75°$, to allow higher resolution coverage of the field of view. This mode therefore requires 4 scans to obtain full coverage. The LMS200 is capable of a number of other measurement modes, but the work described here uses the $180°$ scan/$0.25°$ angular resolution mode, with a maximum range of $32\mathrm{m}$ at $1\mathrm{mm}$ precision.

C. Serial Data Packet Capture

The LMS200 communicates over an RS422 serial interface. While the mirror is in the front half of its rotation, measurements are gathered in an internal buffer. Once the mirror reaches the back half of its rotation, the buffer contents are transmitted over the serial interface. If the mirror reaches the $0°$ position before the buffer contents have been fully transmitted then the following scan is discarded. In order for the device to transmit all scans, communications must be at the maximum rate of 500kBaud. This is a non-standard serial communications speed, requiring the use of either a USB-to-serial converter, or a special serial interface supplied by SICK.

In order to accurately locate each range measurement relative to the robot, it is necessary to know the precise elevation angle of the LMS200 when each measurement was taken. It is not sufficient to simply time-stamp the data from each scan when it arrives in the serial buffer of the client computer. Not only is there an unknown (and possibly varying) delay between the LMS200 gathering data and transmitting it, but there are also unknown latencies in the computer servicing its serial buffer and generating a time stamp. For an LMS200 at a fixed elevation angle, these

latencies do not introduce significant errors, but since the nodder described here has a maximum angular velocity of around $180^\circ/\mathrm{s}$, a timing error of 10 milliseconds could result in a 1.8° error in nodder elevation. For a range measurement at $10\mathrm{m}$ this would correspond to a position error of $\sim 0.31\mathrm{m}$.

The LMS200 has a capability to be synchronized with a second LMS200 in order that their mirrors are kept 180° out of phase, to eliminate laser interference [11]. The 'master' LMS200 outputs a 24V synchronization pulse whose falling edges coincide with the 0° position of the mirror. We use this pulse to trigger sampling of the nodder encoder, so that the precise elevation of the nodder is known at the beginning of each mirror sweep.

III. Software

The software for data gathering, calibration and production of point clouds is implemented in C++ and runs in real time on the robot.

A. Calibration

Before use of the data, two calibrations must be performed - one for the mechanical system (a one time off-line step) and one for the data acquisition system (an on-line step performed during start up). These proceed as follows.

1) Mechanical Calibration: The mechanical calibration is an interactive off-line step, which needs to be performed only once. It is necessary to accurately determine the minimum and maximum elevation angles $\gamma_{min}, \gamma_{max}$ of the nodder, so that they may be associated with minimum and maximum nodder encoder voltage readings. If the correspondence is inaccurate then vertical surfaces appear warped. For this calibration, it is necessary to place the nodding apparatus in a location with a completely level floor, facing a flat section of wall, though the wall does not need to be perpendicular to the floor. The calibration considers only the middle range value of each laser scan; effectively a vertical slice in front of the sensor. The first step is for the user to identify a set of points lying on a flat section of the wall, using a GUI. The task is then to optimize $\Delta = \gamma_{max} - \gamma_{min}$ such that the wall points are as collinear as possible. Applying a common offset to both values simply rotates the measured points about the nodding axis, so the absolute values of γ_{min} and γ_{max} are not yet of interest. A hierarchical search based optimization is applied to find the optimal Δ. Finally, γ_{min} is chosen such that a straight line lying through the floor points is made horizontal.

2) Temporal Calibration: In order to convert laser scans to 3D Cartesian measurements, we must know the angle of the nodder when the scan was taken. We sample from the nodder encoder and laser at regular intervals (as determined by the synchronization pulse), but there is an unknown constant offset $\hat{\tau}$ between the timestamps of each, due to different latencies in each system. This can be thought of as finding the alignment between a list of range values and a list of nodder elevation angles. The temporal calibration is performed in the sensor initialization phase and is designed to find the value of $\hat{\tau}$.

We begin by taking laser and nod angle measurements for a period T_{cal} (long enough to ensure a number of full nodder oscillations) with the vehicle stationary in a static environment. We consider only the mid points of the 2D laser scans; that is, a vertical slice directly in front of the sensor. If the y-axis is body-frame forward and the z-axis is body-frame up we have a time series of points lying in the Y-Z plane.

For a given trial offset τ, the measured points can be considered to be samples drawn from a continuous closed curve, traversed once for each full oscillation of the nodder. When $\tau = \hat{\tau}$ the curve will have zero enclosed area; the parts of the curve corresponding to upward and downward nod sweeps are coincident.

The curve may be parametrized in the Y-Z plane as $(y(t), z(t))$ where $0 < t < T_{cal}$ and its enclosed area given by

$$A = \left| \int_0^{T_{cal}} y(t) \cdot \frac{d}{dt} z(t) dt \right| \tag{1}$$

Since $y(t)$ and $z(t)$ are functions of range r and nod angle γ they may be rewritten as $y(r(t), \gamma(t))$ and $z(r(t), \gamma(t))$. Substituting these into (1) and noting that we wish to minimize the area, we find that

$$\hat{\tau} = \underset{\tau}{\operatorname{argmin}} \left| \int_0^{T_{cal}} y(r(t), \gamma(t+\tau)) \frac{d}{dt} z(r(t), \gamma(t+\tau)) dt \right| \tag{2}$$

In practice, the optimization to find $\hat{\tau}$ is relatively straightforward. Using the synchronization pulse from the LMS200 to trigger the sampling of the nodder encoder ensures that the correct nodder elevation angles are collected, so that they may be directly paired with a laser scan without interpolation. Given an upper bound on $\hat{\tau}$ (experimentally, latencies are always less than $100\mathrm{ms}$) we are left with only a small discrete set of values of τ which we must test.

It should be noted that data bytes from the LMS200 are occasionally dropped by the serial buffer if it is not serviced regularly enough[1]. We cope with this by enabling the LMS200's 'real time indices' option, which appends a scan counter to each of the data packets. This allows detection of dropped packets and adjustment of the timing offsets.

B. Constructing point clouds

Given an accurate elevation angle for every set of measurements received from the LMS200 it is an easy task to project the points into the vehicle frame. We do find it necessary to interpolate nod elevations across individual laser mirror sweeps, to account for the fact that it takes almost $7\mathrm{ms}$ for the LMS200 to sweep its beam through 180°.

Fig. 3 shows a typical calibrated point cloud taken from a stationary vehicle. With the laser system now fully calibrated

[1]The 500KBaud RS422 interface we use is simply a standard device retrofitted with a faster oscillator. The UART buffer therefore fills more rapidly and must be serviced more often to prevent overflow.

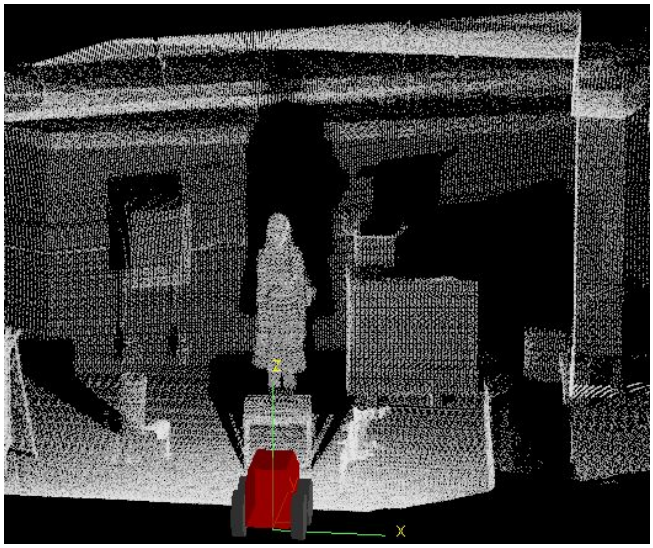

Fig. 3. A typical point cloud generated from a stationary vehicle. The longer the scanning period, the more dense the data becomes, by progressively filling in the gaps.

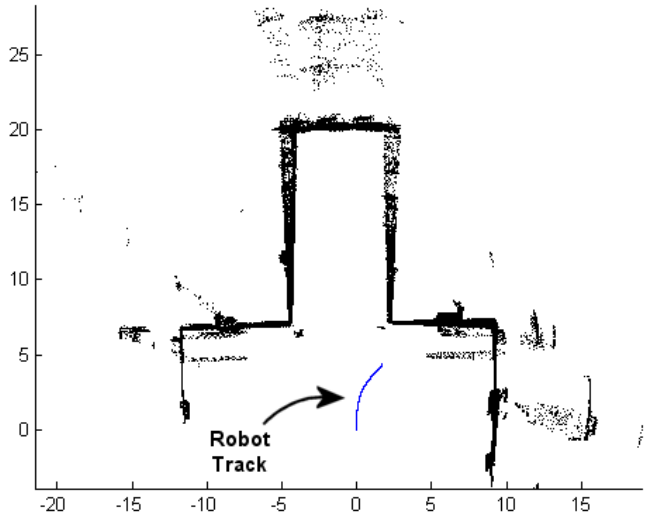

Fig. 4. A plan view of an uncorrected 3D point cloud gathered while the robot was moving. The robot's track is shown in blue. Notice that odometry errors (wheel slip) cause gross alignment errors in the point cloud. Units are meters.

we can proceed to high level processing, in particular to account for and correct small errors in reported vehicle motion as individual 2D scans are accumulated into 3D point clouds. We shall discuss this in the next section.

IV. DATA GATHERING ON A MOVING VEHICLE

The rapid rate of nodding allows the environment in front of the vehicle to be scanned (albeit in relatively low detail) once every 0.6s. Measurements over larger ranges are noisier than those at short range. Nyquist's sampling theorem limits the maximum forward velocity of the robot to obtain sufficiently dense and low-noise samples of the environment. We find experimentally that a speed of $0.5\mathrm{m/s}$ gives a good trade-off between data quality and coverage.

Fig. 4 shows a point cloud gathered over a 10 second period, while the robot was moving. There is significant noise in the data, particularly on surfaces seen from a distance. Even relatively small odometry errors can introduce gross misalignments over large ranges. Unmodelled pitching and rolling movements can cause similar data misalignments about their axes.

We now describe an algorithm which takes as input a point cloud with unknown odometry errors and produces a cleaned up point cloud, where rotations in all three principle axes have been corrected throughout the vehicle's trajectory. This algorithm is implemented in Matlab and is an off-line step. Our prior is that most man-made environments contain many vertical surfaces. Exploiting the knowledge that almost vertical features in the point cloud almost certainly *are* vertical allows us to infer useful information about roll and pitch, where the data would normally be too sparse to perform operations such as point cloud matching [12], [13].

A. Plane extraction

We start by splitting the point cloud, $\mathcal{X}$, gathered over a period T (usually around 10s), into a set of smaller, temporally contiguous point clouds:

$$\mathcal{X} = \{X_1, X_2, \ldots, X_N\} \tag{3}$$

where X_i is a set of 3D points collected over a suitable period, T_S. In our implementation, we choose T_S to be half the nodding period - the time taken for the nodder to transit between its elevation extents.

From each cloud, X_i we first extract lines from 2D scans using a RANSAC based technique [14], then cluster the lines over successive 2D scans to find a set of planes.

$$\Pi_i = \{\hat{\pi}_{i,1}, \hat{\pi}_{i,2}, \ldots, \hat{\pi}_{i,p}\} \tag{4}$$

is the set of normals of the near-to-vertical planes extracted from X_i. For each X_i we seek a rotation matrix $R(\phi, \psi)$ where ϕ and ψ are rotations about the x and y axes such that

$$\hat{\pi}.\hat{\mathbf{z}} = 0 \qquad \forall \quad \hat{\pi} \in \Pi_i \tag{5}$$

We wish to make all the near-vertical planes as vertical as possible by applying a single rotation to all of them. We do this by using the fminsearch function to minimize the cost function

$$F(\Pi_i) = \frac{\sum_{j=1}^{p} A_j \left[R(\phi,\psi)\hat{\pi}_{i,j}.\hat{\mathbf{z}}\right]^2}{\sum_{j=1}^{p} A_j} + w(\phi,\psi) \tag{6}$$

where A_j is the area of the plane whose normal is $\hat{\pi}_j$, such that

$$\phi_i, \psi_i = \underset{\phi,\psi}{\operatorname{argmin}} \left[F(\Pi_i)\right] \tag{7}$$

The regularizer $w(\phi, \psi)$ penalizes deviations away from zero in ϕ and ψ when the ensemble of planes in Π_i inhibit robust estimation of ϕ or ψ. Consider a case where all planes have similar normals, such that one of ϕ or ψ is in the kernel of F; it has little or no effect on the cost. Thus we wish to

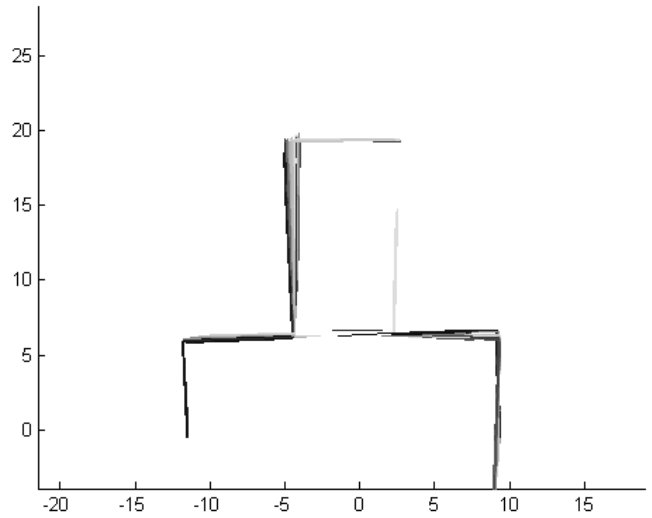

Fig. 5. A plan view of grouped planes after vertical correction. The shading shows how they move over time, representing accumulated error in yaw.

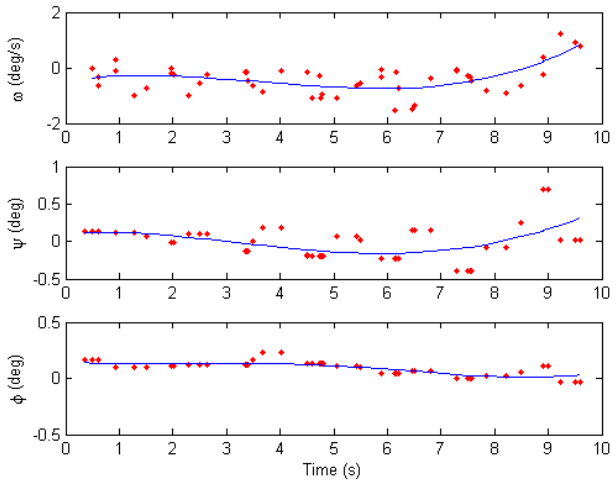

Fig. 6. Yaw rate, pitch and roll adjustments throughout the time period of $\mathcal{X}$. The corrections are smoothed with polynomial fits of degree 4.

prevent the optimization scheme from making unbounded modifications to that variable. This motivates setting

$$w(\phi,\psi) = (1-\alpha)\phi^2 + \alpha\psi^2 \tag{8}$$

$$\alpha = \frac{2}{\pi}\arctan\frac{\sum_{j=1}^{p} A_j\,|\hat{\pi}_{i,j}.\hat{\mathbf{y}}|}{\sum_{j=1}^{p} A_j\,|\hat{\pi}_{i,j}.\hat{\mathbf{x}}|} \tag{9}$$

The larger the plane, the greater the importance of its orientation. The area is hard to compute, so our implementation uses an approximation based on the quantity and average density of points supporting a plane.

B. Yaw slip estimation

After correcting for roll and pitch, we now introduce temporal groupings on planes, by searching for vertical planes which are similar through a sequence of contiguous point clouds, $X_k, X_{k+1}, \ldots X_{k+l}$ and assuming them to correspond to a single real-world planar surface. Plane similarity is determined by thresholding on angle between normals and distances to the origin for planes in consecutive clouds.

With zero slip about the yaw axis, these planes should be exactly coincident. In the presence of slip, the planes undergo rotational drift as shown in Fig. 5. Given a pair of normals π_a, π_b representing the same plane at consecutive times, the instantaneous slip rate may be derived as

$$\omega = \frac{\arccos\left(\pi_a^T \pi_b\right)}{T_S} \tag{10}$$

C. Reconstructing Point Clouds

At this point we have a discrete set of values of ϕ, ψ and ω for discrete points in time. By polynomial fitting we can obtain functions $\phi(t)$, $\psi(t)$ and $\omega(t)$ which are valid over the duration of the capture period T (Fig. 6). . Every individual 3D point x_i in $\mathcal{X}$ has a time stamp t_i and so in principal we can now retrospectively apply a correcting transformation $T(\phi(t), \psi(t), \omega(t), x(t))$. While the ϕ,ψ correction (which renders points in a gravity down frame) is simply a matter of pre-multiplication by a rotation matrix, $R(\phi(t_i), \psi(t_i))$, the ω (angular slip) correction requires more effort. For $0 < t < T$ we are in possession of the set of odometry measurements and thus interpolation allows us to express the odometric vehicle trajectory as a continuous function of time $\mathbf{x}_v(t)$. The corrected trajectory, $\hat{\mathbf{x}}_v(t)$ is deduced by integrating the yaw slip correction along the trajectory:

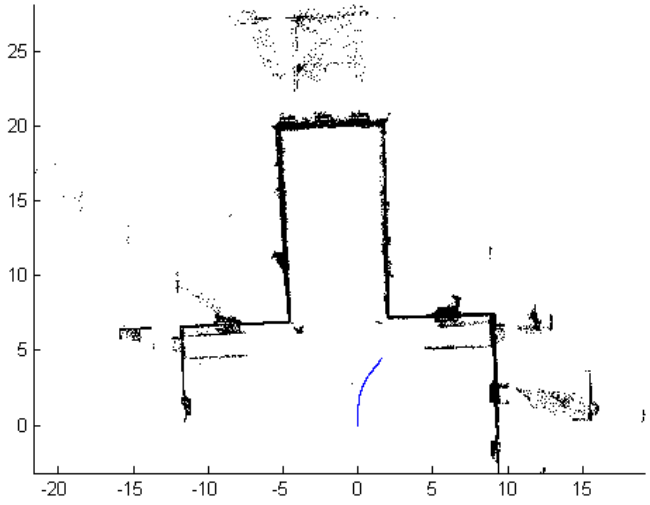

Fig. 7. The corrected version of the point cloud seen in Fig. 4. The cloud is noticeably crisper, as the rotation errors have been greatly reduced.

$$\hat{\mathbf{x}}_v(t) = \int_0^t \frac{d\mathbf{x}_v(\tau)}{d\tau} \oplus \begin{bmatrix} 0 \\ 0 \\ \omega(\tau) \end{bmatrix} d\tau \tag{11}$$

where $\oplus$ is the transformation composition operator. Now that we have a corrected vehicle orientation over the duration of the point cloud capture period, for each point x_i we can perform a final rotation around the yaw axis of the vehicle at time t_i to yield the final corrected point cloud $\mathcal{X}'$

Figs. 7 and 9 show two views of the final corrected point

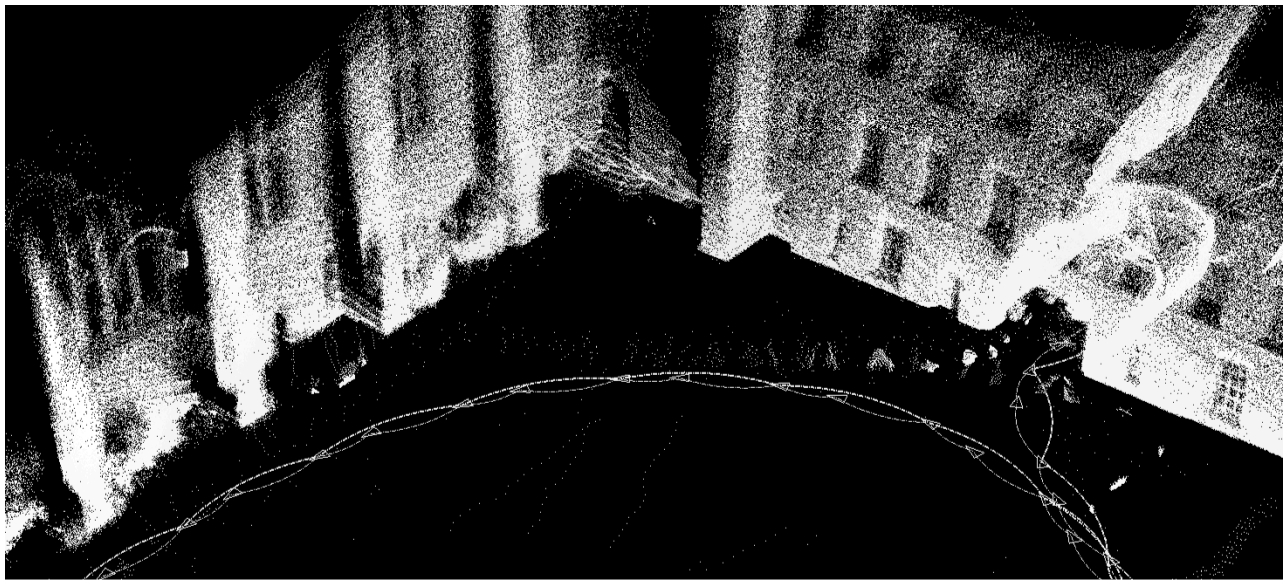

Fig. 8. A collection of corrected point clouds fused together in a SLAM framework.

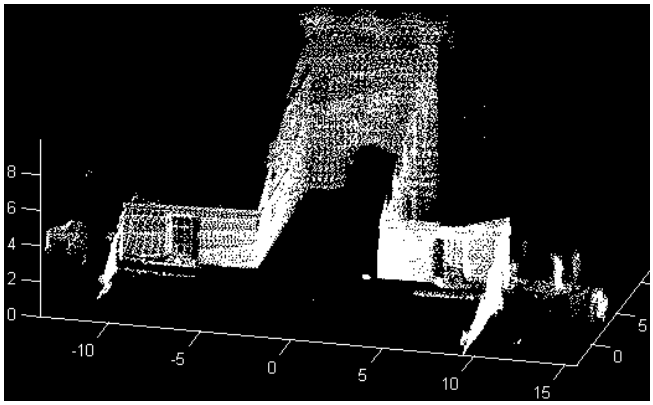

Fig. 9. Another view of the corrected point cloud shown in Fig. 7.

cloud. Fig. 8 shows a collection of corrected clouds rendered together as part of a SLAM map.

V. Conclusions

We have described the key issues in building a low-cost 3D laser range finder system for mobile vehicles. A careful treatment of timing allows an LMS200 to be moved in a dynamic fashion with very high rotation rates, yet the recovery of remarkably accurate point clouds is still possible.

When compared to schemes involving a fixed 2D scanner which is 'raked' along the environment, this capability opens up the possibility of making path planning decisions a long way in advance of the vehicle's arrival at a particular location, rather than just using the data for mapping.

With such rapid coverage of the environment ahead of the vehicle it becomes possible to gather high-fidelity data whilst travelling at speeds of around 0.5 $\mathrm{m/s}$. Because the environment is repeatedly scanned over short time periods, the evolution in location of tracked planar features can be used to enrich 2D odometry with roll and pitch information, as well as correct wheel slip errors.

References

[1] D. Scharstein and R. Szeliski, "A taxonomy and evaluation of dense two-frame stereo correspondence algorithms," *Int. J. Comput. Vision*, vol. 47, no. 1-3, pp. 7–42, 2002.

[2] R. I. Hartley and A. Zisserman, *Multiple View Geometry in Computer Vision*, 2nd ed. Cambridge University Press, ISBN: 0521540518, 2004.

[3] F. Lu and E. Milios, "Globally consistent range scan alignment for environment mapping," *Autonomous Robots*, vol. 4, no. 4, pp. 333–349, 1997.

[4] T. C. Ng, J. I. Guzman, and J. C. Tan, "Development of a 3d ladar system for autonomous navigation," in *Proc. IEEE Robotics, Automation and Mechatronics Conference*, 2004.

[5] A. Howard, D. Wolf, and G. Sukhatme, "Towards 3d mapping in large urban environments," in *IEEE/RSJ Int. Conf. on Intelligent Robots and Systems*, Sendai, Japan, Sept 2004, pp. 419–424.

[6] S. Thrun, D. Fox, and W. Burgard, "A real-time algorithm for mobile robot mapping with applications to multi-robot and 3d mapping," in *IEEE Int. Conf. on Robotics and Automation*, 2000.

[7] O. Wulf and B. Wagner, "Fast 3d scanning methods for laser measurement systems," in *Int. Conf. on Control Systems and Computer Sci*, vol. 1, Bucharest, Romania, July 2003, pp. 312–317.

[8] H. Surmann, K. Lingemann, A. Nüchter, and J. Hertzberg, "A 3d laser range finder for autonomous mobile robots," in *Proc. 32nd International Symposium on Robotics (ISR'01)*, 2001.

[9] D. Hähnel and W. Burgard, "Probabilistic matching for 3d scan registration," in *Fachtagung ROBOTIK*, 2002.

[10] D. M. Cole and P. M. Newman, "Using laser range data for 3D SLAM in outdoor environments," in *IEEE Int. Conf. on Robotics and Automation (ICRA'06)*, Orlando USA, May 2006.

[11] SICK, *Telegrams for Operating/Configuring the LMS 2xx Laser Measurement Systems*, SICK AG, Germany, Apr. 2003.

[12] A. W. Fitzgibbon, "Robust registration of 2D and 3D point sets," in *Proc. British Machine Vision Conf.*, 2001, pp. 662–670.

[13] P. J. Besl and N. D. McKay, "A method for registration of 3-d shapes," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 239–256, 1992.

[14] V. Nguyen, A. Martinelli, N. Tomatis, and R. Siegwart, "A comparison of line extraction algorithms using 2d laser rangefinder for indoor mobile robotics," in *Proc. Intelligent Robots and Systems*, 2005.