Direct Utilization of LiDAR Data in GNSS/IMU Processing for Indoor and Mobile Mapping Applications

Cosandier, D.[b], Martell, H[a]. and Roesler, G.[a]

[a]NovAtel Inc., AB, Canada, [b]Norient Systems, Germany

Key words: SLAM, LiDAR positioning, Indoor Mapping, Mobile Mapping, GNSS, INS

ABSTRACT:

Scanning LiDAR sensors have become a standard component in most mobile mapping systems, and they provide an impressive level of detail in 3-dimensions. To provide real-world coordinates of the LiDAR point cloud, a GNSS/IMU system is often used for the exterior orientation[1] (EO), and post-processing typically delivers optimal EO estimates. For producing the final point cloud, the navigation EO parameters can either be used directly or as an initial approximation for subsequent LiDAR processing, such as SLAM. Therefore, improving the accuracies of the EO values produced by the GNSS/IMU system is highly desirable.

This paper looks at the benefits of adding the scan-by-scan LiDAR matching directly into the navigation processing workflow. While LiDAR-only matching can be very sensitive to the surface geometry, the combined approach is much more robust due to the LiDAR-inertial coupled processing. In addition, for optimal results, this coupled approach requires accurate error modelling for each of the sensors being fused. Using both indoor and outdoor datasets collected with the Velodyne HDL-32 sensor, in areas where GNSS coverage is minimal or denied the algorithm is shown to significantly improve upon the GNSS/INS-only solution. In the case of the outdoor datasets, sections of the GNSS data were removed thereby providing – in addition to the reference trajectory – a trajectory with satellite outages, as typically occurs in urban canyons. The results from these periods show a significant improvement in the position computation with the addition of LiDAR data.

Obtaining indoor datasets with reference is difficult, but distinct improvements in point clouds are noticeable with the LiDAR navigation processing, and visual discrepancies can also be analysed for areas with multiple passes. It is shown that the combined GNSS/IMU/LiDAR processing reliably produces sub-metre and often sub-50-cm results in building interiors, which provides a good initial estimate for subsequent SLAM—benefitting the automatic finding of loop-ties.

1. INTRODUCTION

3-D mapping is now commonly performed with scanning LiDAR sensors in airborne and mobile mapping (ground) environments. In commercial mapping campaigns, EO parameters are often provided to the LiDAR sensor via a GNSS/IMU system. In configurations such as this, the integrity of the real-world laser-point coordinates are largely dependent on the accuracy and reliability of the position and attitude information obtained from the coupled GNSS/IMU. In airborne modes, this is usually not an issue because open-sky GNSS positioning has proven itself to centimetre level in a production mode context. Attitude information, while partially dependent on the quality of the IMU, also benefits from associated high accuracy position determination. However, mobile mapping at ground level has become increasingly important and common in recent years. The consumption of such maps is now taken for granted by the public even though GNSS-dependent mapping in high density urban areas is challenging. More recently, backpack systems are taking scanning LiDAR mapping systems indoors and into locations where GNSS reception is non-existent.

Commercial users are aware of the positioning issues associated with mobile mapping projects in areas where buildings and trees occlude or degrade signals from GNSS satellites. In severe urban canyon areas, a tightly coupled GNSS/IMU may only provide absolute positioning at the metre-level. In fact, without constant updates, the accompanying inertial system will drift relatively quickly depending on the quality of the system. Given the current tendency towards employment of MEMS IMUs in ground surveys, especially those of a backpack nature, the inertial sensor may accumulate metres of error in a matter of minutes of free-mode operation. The loss of accuracy in the EO parameters may consequently produce point cloud information which is degraded or even unacceptable to the end-user in a worst-case scenario.

[1] Position and orientation of a sensor in a reference frame

This paper addresses the problem of obstructed GNSS signals in ground mapping projects by proposing a post-processed solution, whereby the LiDAR range data collected in a GNSS/IMU/LiDAR survey is utilized in an Iterative Closest Point (ICP) solution as relative coordinate updates to the inertial sensor. The system currently utilizes scan-to-scan-matching, which is robust and requires little or no user-input. In cases where subsequent scans do not match due to lack of overlap, too many dynamic objects or poor geometry, the system puts more emphasis on the inertial solution.

The focus of this paper is to determine the benefit of adding LiDAR to the navigation post-processing. We exploit the information obtained from GNSS/INS/LiDAR systems by combining a commercial GNSS/INS post-processing package with CloudMatcher, a custom-developed ICP scan-matching application. This data is employed here in reconstructing trajectory information from production-level surveys performed in several GNSS-restricted indoor and outdoor environments, and the results are presented with respect to available reference data.

The organization of the paper is as follows. Section 2 presents methodology, Section 3 shows the workflow and covers operational issues, Section 4 exhibits results, while Section 5 outlines conclusions.

2. Methodology

This section outlines the process of utilizing LiDAR and navigation (GNSS/IMU) data as inputs to compute a precise, high-rate trajectory that can be used to geo-locate the LiDAR point cloud. This description focuses on the LiDAR integration and scan-matching algorithm.

2.1 GNSS/INS Processing

Utilizing GNSS and IMU measurements to position and orient LiDAR point clouds has been commonplace since the advent of LiDAR. NovAtel's Inertial Explorer (IE) is an example of a software package that accepts raw GNSS/IMU data and outputs accurate, high-rate trajectory information, as explained by Kennedy and Martell in [1]. Recently, IE was modified to accept position, velocity and attitude (PVA) updates, which are derived from external sensors, such as camera systems or LiDAR. The "PVA" file supports two update types: relative updates, a difference between epochs, and absolute updates, a coordinate update in an established reference frame. Obtaining absolute updates has additional complexity, as they require pre-surveyed control points; however, relative updates, which are the focus of this paper, are extracted from scan-to-scan or image-to-image data, and they provide local-level[2] position (r), attitude (R) and velocity (v), as well as the associated covariance (C). Scans are generally captured at a rate higher than 1 Hz, but they are integrated together to form a 1 Hz update. An illustration is provided in Figure 1.

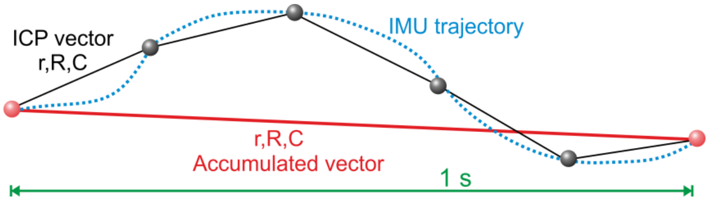

Figure 1: PVA vector formation presented in 2 dimensions

The PVA file is an input into the navigation Kalman filter and will be used as an update for the filter. A Kalman filter provides optimal results when precise error modeling is provided for each sensor used. This means that the PVA updates must also have a realistic covariance matrix (C) assigned.

In post-processing, the Kalman filter can process the data in both forward and reverse chronological order. Unfortunately, the position trajectory is subject to step-effects caused by Kalman filter updates. Furthermore, position accuracy during outages grows at a 2nd order rate. An essential part of post-processing is the backsmoother, which recursively processes the data and updates the error filter states as it proceeds. With proper error modeling, the backsmoother can improve accuracies significantly. The backsmoother's ability to compensate for errors is directly related to the quality of the error modeling.

[2] True-north, z-up Cartesian coordinate system centered about the current position.

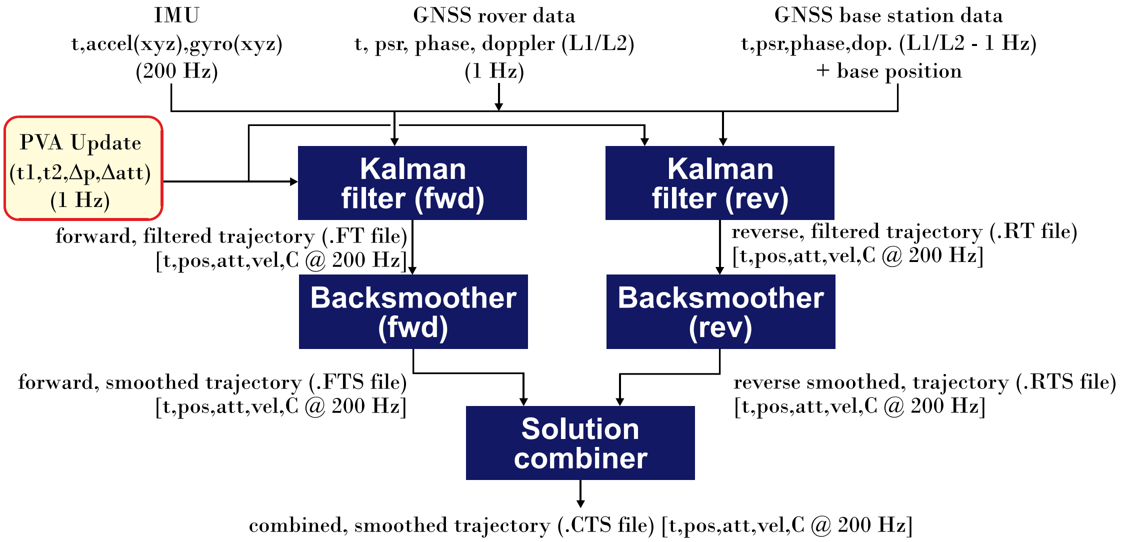
IMU
t,accel(xyz),gyro(xyz)
(200 Hz)
GNSS rover data
t, psr, phase, doppler (L1/L2)
(1 Hz)
GNSS base station data
t,psr,phase,dop. (L1/L2 - 1 Hz)
+ base position
PVA Update
(t1,t2,Δp,Δatt)
(1 Hz)
Kalman filter (fwd)
Kalman filter (rev)
forward, filtered trajectory (.FT file)
[t,pos,att,vel,C @ 200 Hz]
reverse, filtered trajectory (.RT file)
[t,pos,att,vel,C @ 200 Hz]
Backsmoother (fwd)
Backsmoother (rev)
forward, smoothed trajectory (.FTS file)
[t,pos,att,vel,C @ 200 Hz]
reverse smoothed, trajectory (.RTS file)
[t,pos,att,vel,C @ 200 Hz]
Solution combiner
combined, smoothed trajectory (.CTS file) [t,pos,att,vel,C @ 200 Hz]

Figure 2: GNSS/INS Post-Processing Workflow

The entire navigation process is shown in Figure 2, where the Kalman filter and backsmoother have been introduced. The final step is to combine the forward and reverse data streams into a single combined trajectory.

2.2 Scan-Matching Algorithm

The goal of the LiDAR-matching algorithm is to correlate pairs of LiDAR point clouds to compute the relative translation and rotation between the point clouds. The relative LiDAR scan matches must be highly accurate—on the order of 1 cm between scans.

The method used to compute the relative LiDAR vector and attitude is that of Iterative Closest Point (ICP)—first proposed as the point-to-point variant by Besl and McKay in [2]. The point-to-plane algorithm used here was later introduced by Zhang in [3]. It can be described as follows: For each point in cloud 'A', the ICP finds the closest matching point in cloud 'B' using a 6-parameter transformation (position + attitude) between the two, which is initially approximated by the navigation EO data. Using these matched point-pairs, the 6-parameter rigid body transformation is updated by solving a system of equations. Since the "closest point" is not necessarily the correct point-match, the algorithm must iteratively transform point cloud 'B', re-find matches and solve for a new transformation.

ICP typically has three major issues:

a) Errors in position and attitude are induced by moving objects and indistinct objects such as foliage.
b) Converges to a false minimum—especially if given poor initial estimates.
c) A failure to converge at all due to a lack of "matchable" features in one or more axes.

For the first issue (a), the blunder detection algorithm is a critical part of the ICP, as covered by Rusinkiewicz in [4]. Essentially, there are three tests that may be applied:

i) The point-to-plane distance for a given match must be less than a tolerance—known as the distance tolerance
ii) If surface normals are formed on both point clouds, then the angle between the surface normals pairs should be less than a tolerance
iii) If the same point in cloud 'B' is matched to several points in 'A', these are considered duplicates and all but the closest match is rejected. This is common in the boundary areas of the point clouds.

Together these tests remove the majority of incorrect matches. Typically, half the points remain after this stage, but this is not a problem given that many thousands of points are used to compute 6 unknowns.

Issue (b), convergence to a false minimum, is largely mitigated by the fact that the navigation system is also providing a precise relative position and attitude estimate. Hence, ICP is "fine-tuning" this solution.

A failed convergence described by (c) is a major problem for LiDAR-only SLAM systems because it forms a discontinuity in the relative accumulated trajectory. Fortunately, the GNSS/INS processor can work independently of the LiDAR updates and will still provide a solution.

2.3 Forming Point Cloud Pairs

LiDAR sensors can operate at different scan rates, for example 10, 15 or 20 Hz. Each scan provides many tens or hundreds of thousands of points, but most are not utilized —especially with tilted LiDAR mounts. Matching single scans tends to be less accurate than combining multiple sweeps[3] into a single point cloud.

For each pair or n-tuple of sweeps, a "reference time" is needed. The point cloud coordinates will be relative to the position at this time. Point cloud matches are collected and integrated until 1.0 second has elapsed. The point clouds straddling the whole second are referenced to this time and are used to output the PVA data to file for subsequent navigation processing iterations. This process is illustrated in Figure 3.

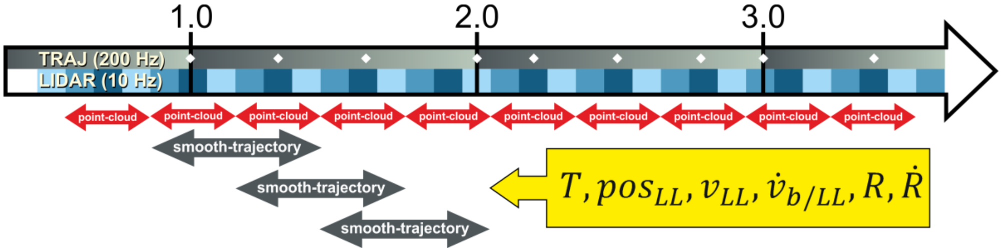

Figure 3: Illustration of point cloud formation (10 Hz input with window size of 3)

The data interval between point clouds depends on the maximum speed. For slow or walking applications, a lower data-rate (such as 2 Hz) can be used. In this case, only a sub-set of the LiDAR data is needed unless the number of sweeps is increased. This increases computational cost but can improve accuracies. For faster vehicle platforms, all of the data should be used to maximize overlap. In tilted LiDAR implementations, where the LiDAR unit is mounted at an angle, there may not be enough overlap between point clouds for accurate scan-matching. In such cases, it is advised to keep the speeds below a threshold to maintain sufficient overlap in the scans[4]. LiDAR units mounted with the rotation axis near vertical (i.e. ~0° tilt) will provide the best ICP results, and very steep tilt angles provide the poorest.

2.4 Error Modeling

Assigning realistic covariances to the relative PVA outputs is crucial because the Kalman filter is fusing multiple data inputs including IMU, GNSS, LiDAR-odometry and possibly vehicle odometry. In the case of ICP, the EO parameters from each cloud-pair will have a varying accuracy depending on the sensor precision and scene geometry. Using the covariance estimated by the 6-parameter ICP solution solver, a robust and representative full covariance estimate is computed for each PVA record. The input standard deviation should be calibrated for a given sensor, but otherwise remains constant from project to project. Figure 4 shows the estimated errors (SD values) for Y (along-track), X (across-track) and Z (height) for a half hour section of HDL-32 outdoor urban data. The estimated ICP errors can be verified against the GNSS\INS solution. This can be accomplished by using surveys that provides various LiDAR geometries and reasonable GNSS signal tracking for reference.

[3] Also known as scans or frames.

[4] For a 45° LiDAR mount angle, such a maximum speed is ~65 km/h or 40 MPH.

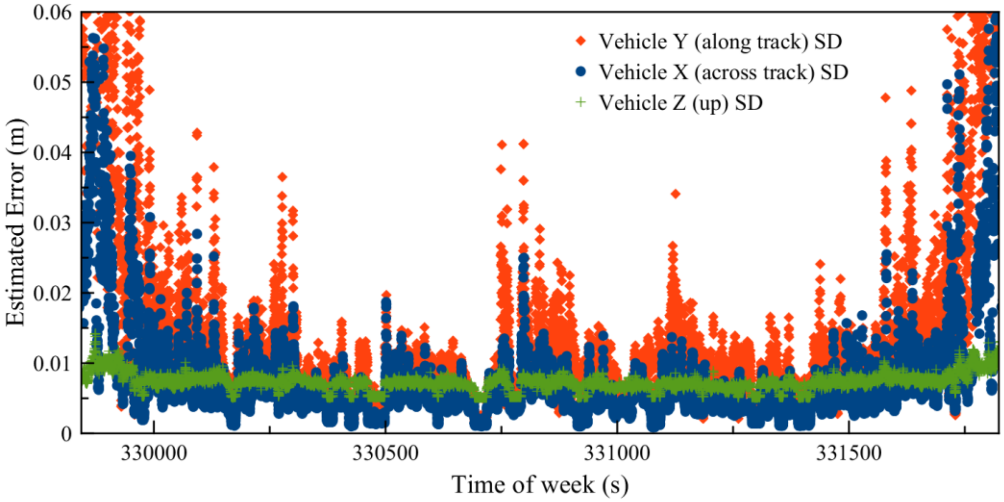

Figure 4: Estimated ICP errors from a sample outdoor scene

3. Workflow and Operational Issues

This section discusses practical issues and shows the workflow of the entire LiDAR-GNSS-IMU processing outlined in Section 2.

3.1 LiDAR Processing Workflow

Figure 5 shows the process for obtaining precise EO values given the raw data inputs. The procedure is to iteratively execute first the GNSS/IMU processing and then, in a subsequent step, perform the LiDAR processing in order to output a PVA file. This PVA file is used to update the next GNSS/IMU computation cycle. Optionally, the LiDAR-derived PVA file may only cover sections where GNSS information is degraded or absent. Typically, the entire GNSS/INS/LiDAR solution sequence converges within two to four iterations. The last step in the processing is to re-export the LiDAR point could using the final trajectory.

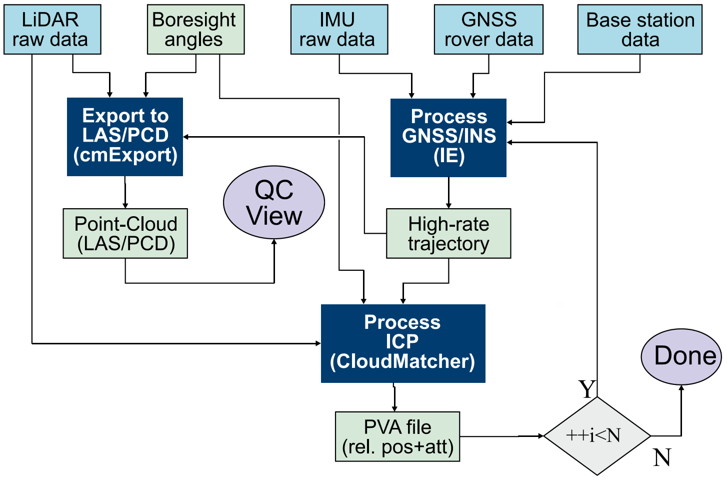

Figure 5: Entire LiDAR/GNSS/IMU Processing Workflow

3.2 GPU Acceleration and Point-Thinning

ICP scan-matching is computationally intensive given the large number of LiDAR points. For example, an HDL-32 may produce several billion points over four hours of data collection. The original implementation of the ICP algorithm employed here would require more than 6 hours to process one iteration of this data set. From the investigation, it was determined that the majority of the CPU time was spent searching corresponding point clouds to find the nearest neighboring (NN) points, using the FLANN library, developed by Muja and Lowe in [6].

Three methods were investigated to reduce the ICP processing time:

a) Utilize faster processing hardware, such as a GPU;
b) Employ a different algorithm other than the KD-tree for the nearest neighbor searches;
c) Reduce the number of points in a way that does not affect accuracy.

Ultimately, all three of these approaches were utilized to improve processing speeds. Many LiDAR users have a graphics card for visualization. These can be utilized through NVIDIA's CUDA (or alternately OpenCL) API. Algorithms that can be parallelized efficiently may run much faster on a GPU than on a multi-core CPU. In this investigation, it was determined that the KD-tree algorithm, whether FLANN or custom-developed, is not optimally suited to the architecture of the GPU. The authors investigated employing Morton codes for the purposes of improving memory caching speeds (described by Morton in [7]). This algorithm utilized the computational power of the GPU and delivered significant speed improvements for the nearest neighbor searches (shown in Figure 6). Finally, the Morton point coding permits a fast and efficient point reduction scheme, typically halving the point count while also improving ICP accuracy and throughput.

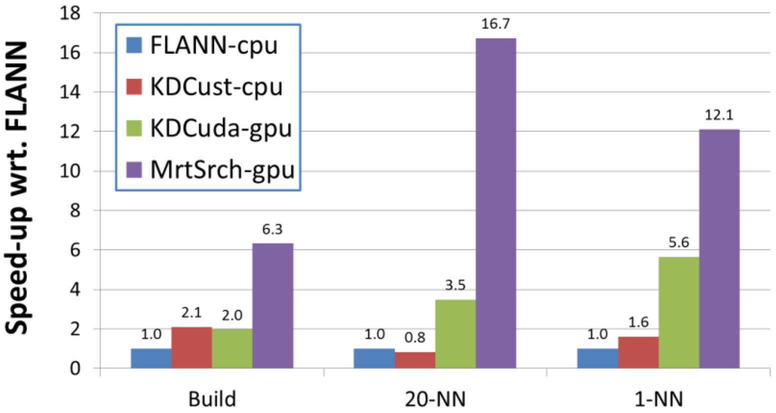

Figure 6: Speed improvement with Morton Search on GPU

The 4 hour survey described in section 4.2 now processes in ~45 minutes per iteration (or ~5 times real-time), and there are still further opportunities to improve the processing efficiency.

3.3 Boresight Determination

For any LiDAR application that utilizes an IMU, a precise boresight[5] is an essential prerequisite. The combined ICP/inertial processing is especially sensitive to these angles. It should be noted that a verification of the IMU-LiDAR sensor boresight angles is required for new LiDAR installations. The process for determining the boresight angles is implemented in two steps.

1) Convert a piece of the LiDAR data to an absolute point cloud. The goal is to generate a visually correct point cloud. Focusing on the correct representation of vertical surfaces will allow the roll angle to be corrected. For this purpose, the CloudCompare program has been used here (see [9]).
2) In stage two, the ICP is run with a reliable fixed-integer GNSS solution. Errors in the pitch or heading boresight will cause a speed-dependent ramp in the residuals, and by iteratively minimizing these residuals, the angles can be computed.

4. Test Results

The results presented here are based on production datasets contributed to us by commercial LiDAR operators. We have included two urban mobile mapping projects undertaken in Madison, Wisconsin, one indoor/outdoor building survey in Toronto, Ontario and one UAV LiDAR-mapping survey flown in the Los Angeles, California area.

The equipment utilized in all cases consisted of a NovAtel SPAN GNSS/INS equipped with dual frequency receiver and MEMS IMU. The LiDAR sensors for all surveys were the HDL-32, manufactured by Velodyne. The

[5] The angular offsets between the IMU and LiDAR measurement frames

GNSS/INS/LiDAR post-processing software employed was a version of NovAtel's Inertial Explorer software loosely coupled to the CloudMatcher ICP program. Similar MEMS inertial units[6] were deployed for all tests.

Experimental results for the mobile mapping and aerial surveys were obtained by utilizing the original data with full GNSS coverage and selecting an area where accuracy is optimal. This provides a reference for the next stage of the test where GNSS data was fully omitted from parts of datasets to simulate a complete outage, such as would be anticipated in urban canyon or under dense foliage conditions. Outages of 100 to 400 seconds were added, and solutions with and without LiDAR scan-matching are shown either visually as point clouds, as position error plots, or in tabular form.

Deriving truth information for the indoor data set was problematic given that no GNSS information was available. For this survey, indirect information is relied on, such as ghosting effects in the point cloud geo-referenced from the ICP/INS procedure.

4.1 First Mobile Mapping Test (20 Minute Operation Time)

This is the first mobile mapping test-survey provided by Mandli Communications and covers the downtown area of Madison, WI, largely consisting of two to six story structures. This is not a "true" urban canyon, but the data within the LiDAR field of view has similar building profiles to cities with taller structures and is still representative; furthermore, there are also many moving objects, such as vehicles, pedestrians and cyclists, while foliage is also abundant. Hence, such data is ideal for prototyping the software, given that truth is available via the GNSS data. Like many other mobile mapping systems, the LiDAR sensor was mounted with a 45° tilt angle, so along-track ICP accuracies are poorer than those in the other axes.

The GNSS/IMU data was post-processed in tightly-coupled mode to create a reference trajectory. Accuracy of the resulting reference trajectory was estimated to only be 15 cm for the time period selected owing to signal interference in the downtown environment. Using this first pass as truth, some GNSS data was then removed in a second step to induce an outage lasting for 300 seconds. This is shown as the grey area in the left of Figure 7. It can be seen that INS-only processing for the 5 minute outage period results in errors with respect to truth of up to 0.75 m. Finally, in a third pass, LiDAR updates were introduced during the GNSS outage period. This resulted in a significant accuracy improvement. In fact, the maximum error was almost halved by employment of ICP-derived updates.

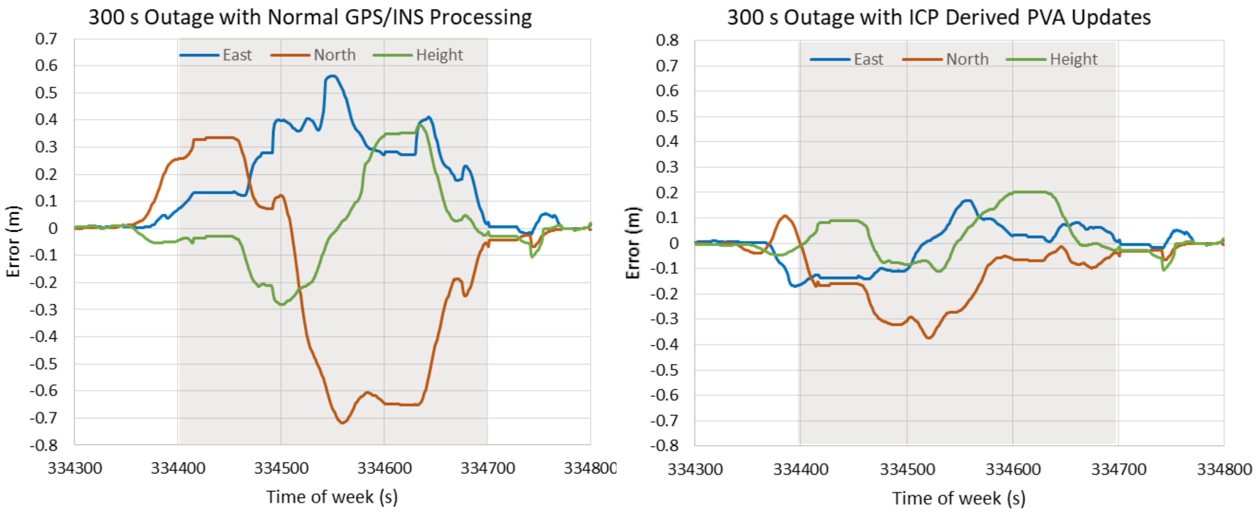

Figure 7: Accuracy improve of adding LiDAR scan-matching (on induced outage)

4.2 Second Outdoor Mobile Mapping Test (4h 20m Observation Time)

In late fall of 2017, Mandli provided a longer survey collected with the same sensor, also through Madison, WI. Given the length of the data, the GPU accelerated version of CloudMatcher was used. Stand-alone GNSS/INS performed reasonably well here. For example, Figure 8 shows 5 vehicle passes over the same location; this is from an area with a few signal obstructions. The lines appearing to go into the page are intensity plots of the same white road lines mapped on the multiple vehicle runs. Any discrepancies larger than several centimetres

[6] With accel. bias of 0.01-0.05 m/s^2 and gyro angular random walk (ARW) of ~0.6 °/√hr

would show up as ghosting. For instance, vertical errors, would show up as a distinct series of lines stacked on top of one another. The consolidation of the 5 line images into one indicates that point precision is at cm level in this region.

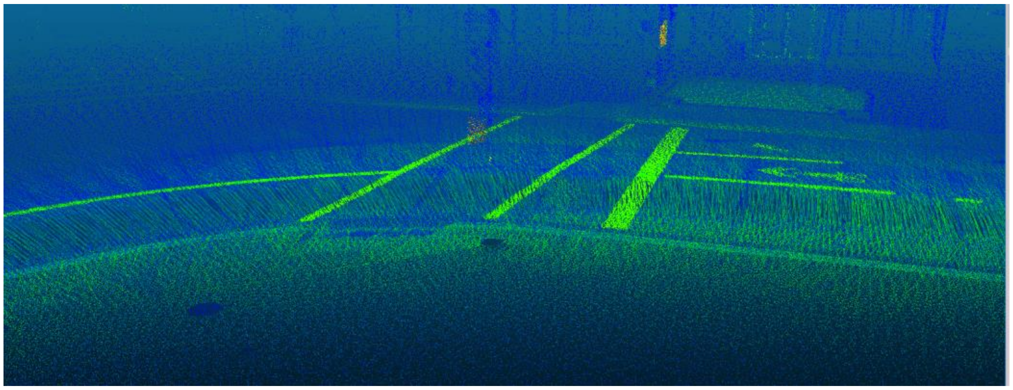

Figure 8: LiDAR intensity values for 5 vehicle passes (curb view)

By carefully examining the position and orientation standard deviations and GNSS position quality, 19 time-locations possessing high accuracy (<2 cm) were selected. These form the truth points for the subsequent INS LiDAR-aided processing. Surrounding each of these points, a 100 second GNSS outage was inserted. The truth-point is at the middle of the outage interval, where errors tend to be maximum. The entire dataset was processed with and without LiDAR scan-matching. This can be seen in Table 1, accuracy improved from half metre to decimetre-level, and this is a reduction in error of ~3 times horizontally and ~5 times vertically. The benefit to the z-axis (elevation) is due to the substantial number of the ICP points on the ground. This data set shows that LiDAR processing is a useful tool to compensate for missing or severely occluded GNSS signals.

Outage/ Test-point	Error with NAV-only (no LiDAR)			Error with LiDAR scan-matching		
	East (m)	North (m)	Elev (m)	East (m)	North (m)	Elev (m)
1	0.09	-0.85	0.11	0.02	0.03	-0.06
2	0.73	-0.04	0.68	0.35	0.02	0.18
3	0.54	-0.52	0.23	0.03	-0.01	0.06
4	-0.32	-0.01	-0.34	0.07	-0.05	-0.15
5	-1.10	-0.35	-0.52	-0.04	0.11	-0.11
6	0.10	-0.51	-0.35	0.08	0.24	-0.17
7	0.04	-0.73	0.15	0.05	0.05	0.06
8	0.18	-0.13	-0.16	0.03	0.02	-0.03
9	-0.61	1.03	-0.11	0.39	-0.27	0.16
10	0.05	0.11	-0.13	0.12	0.04	-0.05
11	0.07	0.26	-0.41	-0.16	0.17	-0.09
12	-0.04	1.20	0.39	0.10	0.14	0.07
13	-0.10	0.26	0.09	0.00	-0.12	0.10
14	0.00	-0.04	0.28	-0.07	-0.21	-0.01
15	0.37	0.16	0.63	0.11	0.19	0.00
16	0.04	-0.14	1.59	0.11	-0.04	0.15
17	-0.23	0.02	0.28	0.20	-0.30	0.09
18	0.29	-0.09	0.48	0.20	0.01	0.07
19	-0.49	0.40	-0.17	-0.12	-0.02	-0.10
RMS (m)	0.47	0.50	0.50	0.16	0.14	0.10

Table 1: Table of errors for test points with and without LiDAR updates

4.3 Indoor Mapping Example

Determination of truth for indoor datasets can be problematic owing to the complete absence of GNSS coverage. It should be noted that in this survey, GNSS was used to initialize the position, while initial attitude is determined by walking in a straight line for 100 m outdoors. However, inside the building only the INS system, whose position continually drifts, determines the position. Hence, examining the benefit of LiDAR scan-matching is appealing. The majority of indoor environments with walls and ceilings provide good geometry for the LiDAR scan-matching; however, long corridors and hallways can be challenging.

The indoor survey in question was mapped with a GNSS/INS/LiDAR system attached to a cart. Again, the system started outdoors to provide initialization of position and attitude, and then moved indoors.

Due to the lack of truth, verification of the accuracy is made from the intrinsic nature of the point cloud depiction. Using CloudCompare to visually evaluate the ghosting effects and general sharpness of the point cloud, insight into the precision of the indoor map is obtained. Also, the estimated position standard deviations provide clues to the overall accuracies. Although not always correct in the absolute sense, they do provide a good reference for evaluating the benefit of LiDAR, as they are highly correlated to the actual errors. Figure 9 shows the estimated position standard deviation values including the LiDAR and non-LiDAR portions.

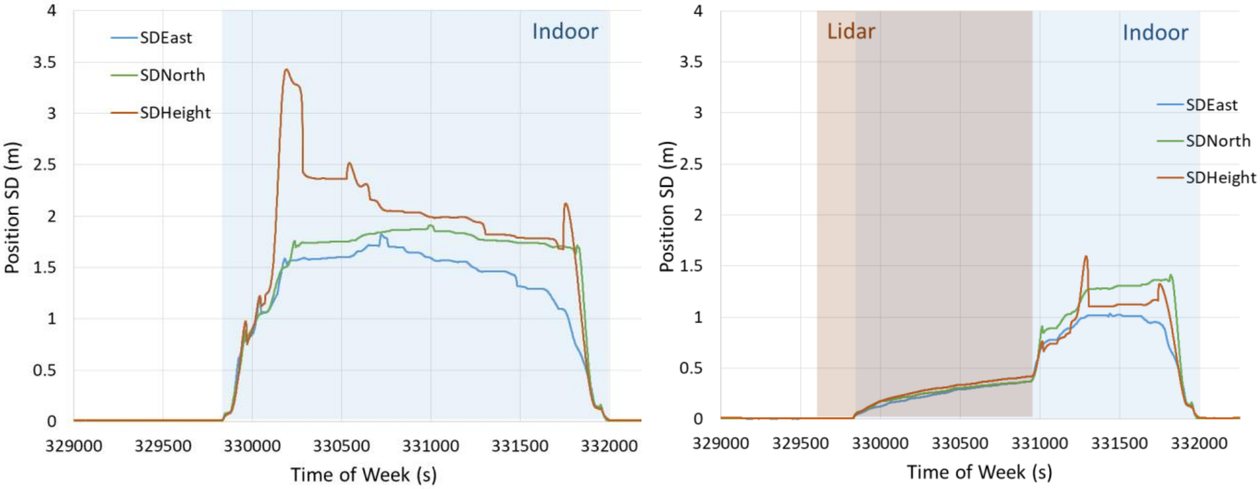

Figure 9: Estimated position standard deviation of the entire dataset

For this test, a GNSS/INS/LiDAR system was wheeled throughout a building for a period of 35 minutes. In the given test, LiDAR measurements were made for only the first 12.5 minutes of the experiment. LiDAR data collection began at the south end of the building and was discontinued part way through the traverse. For the first portion of the experiment, the inertial system was updated with zero velocity updates and relative coordinate vectors derived from ICP. For the remainder of the survey, the inertial process was aided by ZUPTs and constraints only. The drift in the trajectory following elimination of the LiDAR-aiding can be clearly seen in the given image. There should be an obvious difference between a point cloud geo-referenced by INS-only and point clouds registered from INS updated by ICP. This is illustrated in the side by side example in Figure 10

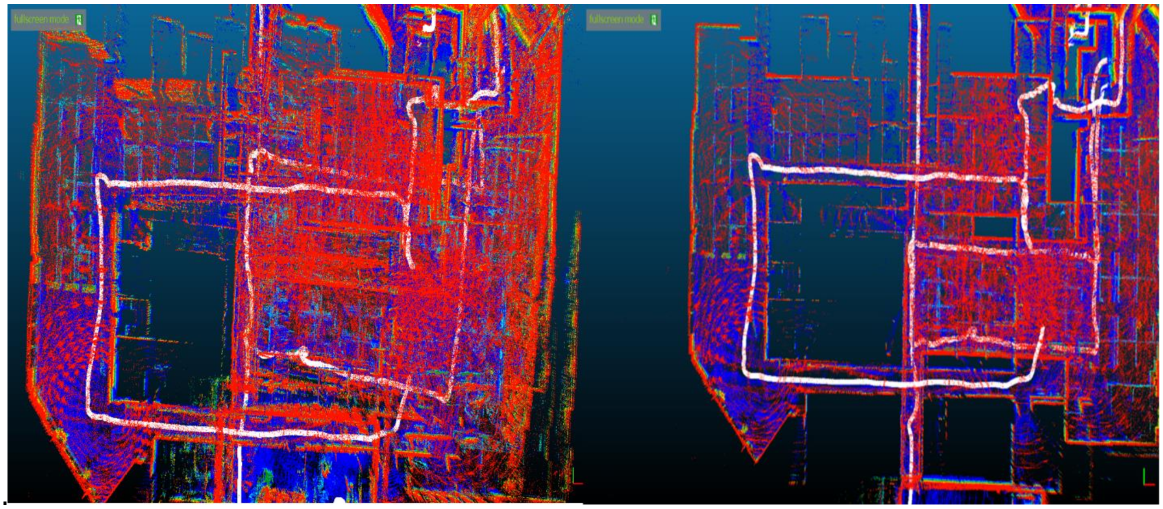

Figure 10: Left side INS-only, right side INS aided with LiDAR

The left-hand side of Figure 10 depicts a point cloud that was geo-referenced with only a MEMS IMU. The right-hand side of the image renders the same point cloud data, but it includes the ICP updates with the MEMS IMU. The differences are striking. The IMU-only image portrays a trajectory which drifts significantly over time as the cart is pushed through the visible hallways. Some question remains on how accurately the right-hand side of the figure has registered the point clouds utilizing LiDAR data. Without truth data, we examine the quality of the final point cloud.

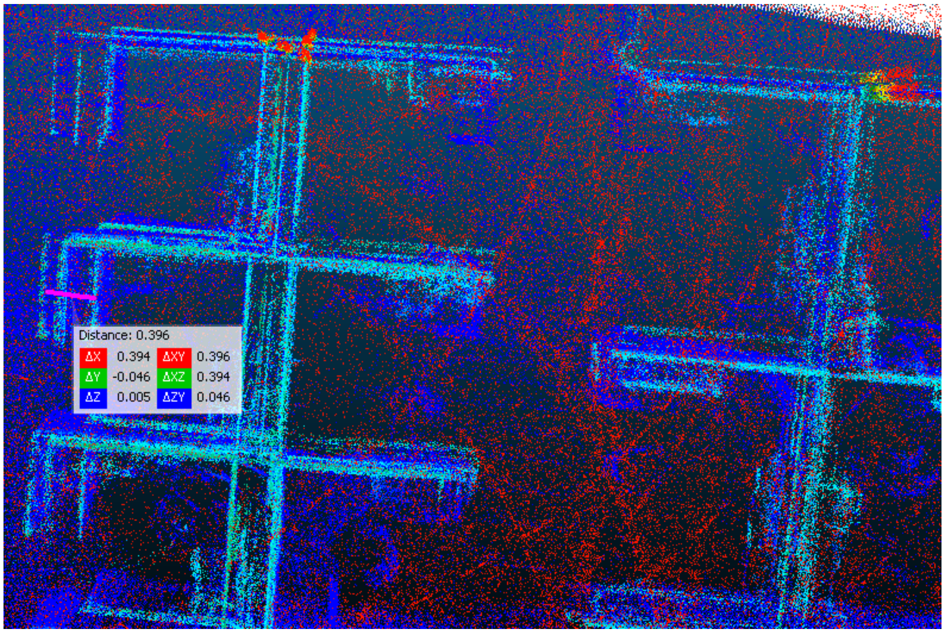

Figure 11: Measuring ghosting effect in indoor survey

Figure 11 shows the maximum ghosting effect observed when ICP updates were used. The spatial separation between multiple passes shows ~0.40 m horizontal difference in the registration of the two cubicle walls. A SLAM algorithm that automatically detects loop-ties should have little difficulty in reducing this difference. With INS-only, the discrepancy is 5-6 m, which means that rooms or cubicles may be matched incorrectly with loop-ties.

4.4 GNSS-Denied Airborne Drone Example

Airborne mapping is typically synonymous with high quality GNSS data and stands to gain little from LiDAR odometry. However, there are certain cases where overhead structures may be an issue and GNSS signals can also be jammed. One issue ICP has with such data is a lack of geometry in the across and along-track directions, making this a challenging dataset for the ICP algorithm.

In this survey, the high quality GNSS data results in an accurate reference trajectory. Figure 12 shows the point cloud geo-referenced with full GNSS coverage. This visualization shows consistency between the flight-lines. To stress the software, an outage was added to almost the entire LiDAR collection. Only 20 seconds of GNSS at the beginning and end of the dataset are retained. This induces an outage of 6 minutes and 20 seconds, which is long for an airborne project. During this period, the position error increases to 10 m. This is distinctly visible in the point cloud shown in Figure 13. Finally, ICP processing is added. For 26% of the solutions, the ICP did not converge at all, and this is to be expected given the geometry for the ICP. This tests how well the software deals with ICP convergence failures. For those epochs that do succeed, the height update is especially beneficial to the GNSS/INS processing as illustrated in Figure 14. The horizontal accuracy improves from 10 m maximum error to 0.85 m with ICP, and the height accuracy improves from 2 m of error to 0.3 m.

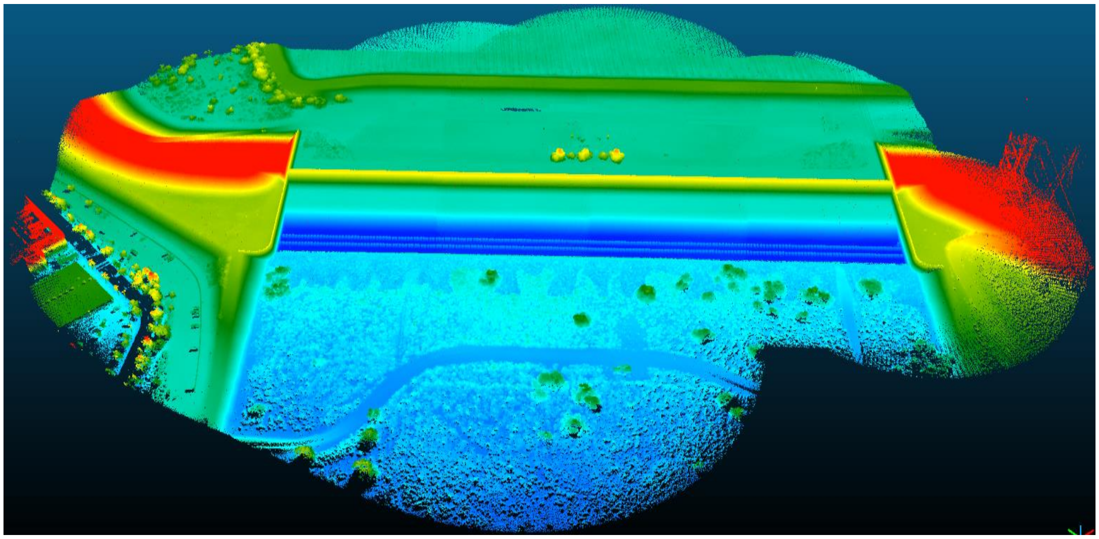

Figure 12: Point cloud using full GNSS coverage

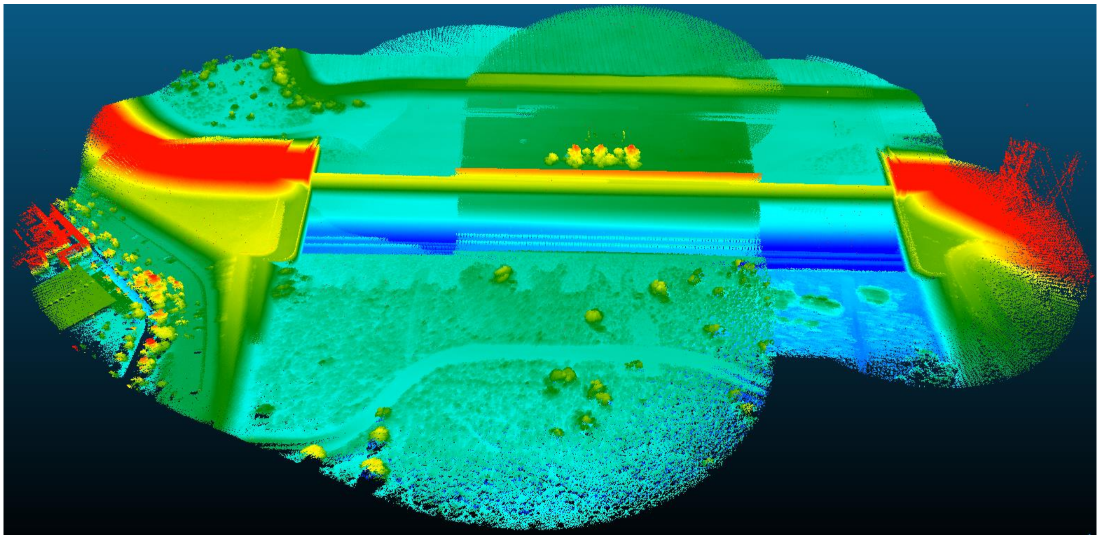

Figure 13: Point cloud with over 6 minutes of GNSS data removed (no ICP)

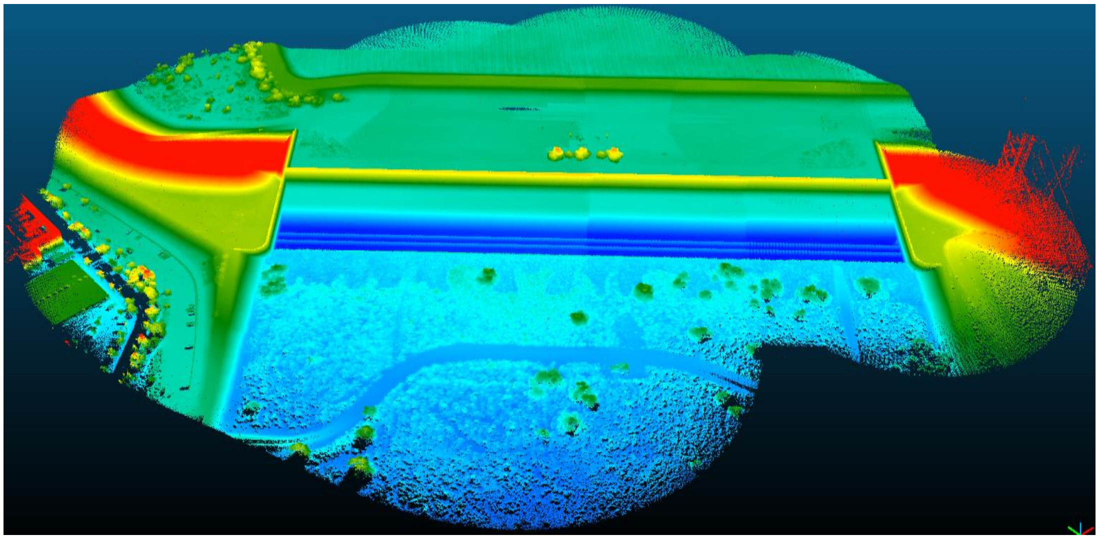

Figure 14: Point cloud with same outage with ICP updates added

5. CONCLUSIONS

In summary, this paper describes the integration of GNSS/INS and LiDAR measurements for the purposes of improving post-processed sensor EO trajectory determination, especially in GNSS-denied environments such as urban canyons or indoors. Under these circumstances, users who rely on the afore-mentioned EO information for geo-referencing of point clouds could certainly benefit from the enhanced positioning depicted here.

The GNSS/INS/LiDAR workflow has been described in the context of a commercial GNSS/INS post-processing package. LiDAR measurements are utilized in an in-house developed ICP algorithm and the resulting scan-to-scan translation vectors employed to update the GNSS/INS computation in a loosely-coupled fashion. We also characterize the significant effort made to improve robustness and computational efficiencies in the ICP procedure.

Field test results were obtained from production-style datasets shared with us from organizations in the LiDAR industry. We have included mobile mapping, indoor and aerial UAV examples and show the differences in coordinates between data sets with and without GNSS outages. It can be seen that even in the complete absence of GNSS signals for hundreds of seconds, the inclusion of terrestrial LiDAR measurements in an ICP procedure enables us to maintain maximum estimated errors of some 0.50 m. It is acknowledged that this is of limited benefit in an open-sky survey, but is a 2-4 times improvement over the 1-2 m RMS accuracy we observe in a severe downtown urban situation where MEMS INS have been employed. Additionally, GNSS/INS positiong can be subject to sudden changes in the output due to the appearance and re-appearance of satellites. These discontinuities can be ameliorated through integration of LiDAR. Finally, we show that hosting of the ICP process on readily available graphics cards, makes processing a practical task.

6. ACKNOWLEDGEMENTS

The authors would like to express their appreciation to Teledyne-Optech, Toronto, Ontario, Mandli Communications, Fitchburg, Wisconsin, and Phoenix Lidar Systems, Los Angeles, California for their kind permissions to use the GNSS/INS/LiDAR data sets presented in this paper.

7. REFERENCES

[1] Kennedy, S., Martell, H. (2006) Architecture and system performance of SPAN NovAtel's GPS/INS solution,DOI10.1109/PLANS.2006.1650612

[2] P.J. Besl, H.D. McKay (1992). A method for registration of 3-D shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence

[3] Zhang, Z. Int J Computer Vision (1994) 13: 119. https://doi.org/10.1007/BF01427149

[4] Szymon Marek Rusinkiewicz (2001). Efficient variants of the ICP algorithm. In Proceedings of the International Conference on 3-D Digital Imaging and Modeling, pages 145–152

[5] Pomerleau F., Colas F., Ferland F., Michaud F. (2010) Relative Motion Threshold for Rejection in ICP Registration. In: Howard A., Iagnemma K., Kelly A. (eds) Field and Service Robotics. Springer Tracts in Advanced Robotics, vol. 62. Springer, Berlin, Heidelberg

[6] Marius Muja and David G. Lowe (2009). "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration", in International Conference on Computer Vision Theory and Applications (VISAPP'09)

[7] Morton, G. M. (1966), A computer Oriented Geodetic Data Base; and a New Technique in File Sequencing, Technical Report, Ottawa, Canada: IBM Ltd. (see also https://en.wikipedia.org/wiki/Z-order_curve)

[8] Li, S., Simons L, Bhaskar, J., Abbasinejad, P.F., Owens, J.D., Amenta, N. (2012). kANN on the GPU with Shifted Sorting, High Performance Graphics (2012), C. Dachsbacher, J. Munkberg, and J. Pantaleoni (Editors)

[9] See http://www.cloudcompare.org/