STATE OF CALIFORNIA • DEPARTMENT OF TRANSPORTATION

TECHNICAL REPORT DOCUMENTATION PAGE

TR0003 (REV 10/98)

ADA Notice

For individuals with sensory disabilities, this document is available in alternate formats. For information call (916) 654-6410 or TDD (916) 654-3880 or write Records and Forms Management, 1120 N Street, MS-89, Sacramento, CA 95814.

1. REPORT NUMBER	2. GOVERNMENT ASSOCIATION NUMBER	3. RECIPIENT'S CATALOG NUMBER
CA16-2777		

4. TITLE AND SUBTITLE	5. REPORT DATE
Precision Mapping of the California Connected Vehicle Testbed Corridor	11/01/2015
	6. PERFORMING ORGANIZATION CODE

7. AUTHOR	8. PERFORMING ORGANIZATION REPORT NO.
Dr. Jay Farrell and Dr. Matthew Barth	

9. PERFORMING ORGANIZATION NAME AND ADDRESS	10. WORK UNIT NUMBER
Center for Environmental Research & Technology University of California, Riverside 1084 Columbia Avenue Riverside, CA 92507	11. CONTRACT OR GRANT NUMBER

12. SPONSORING AGENCY AND ADDRESS	13. TYPE OF REPORT AND PERIOD COVERED
California Department of Transportation 1120 N Street Sacramento CA 95814	Final Report
	14. SPONSORING AGENCY CODE

15. SUPPLEMENTARY NOTES

This work was performed by University of California Riverside, Center for Environmental Research & Technology, in cooperation with the State of California Business, Transportation and Housing Agency and California Department of Transportation.

16. ABSTRACT

In this project the University of California Riverside mapping sensor hardware was successfully mounted on an instrumented vehicle to map a segment of the California Connected Vehicle testbed corridor on State Route 82. After calibrating the sensor platform, the instrumented vehicle was driven repeatedly up and down the corridor of interest, collecting raw point cloud data on the on- board data servers. Then, after a sufficient amount of data was collected, the raw point cloud data was processed, and feature extraction algorithm was used to extract the precise positions of stop bars on the selected intersection.

17. KEY WORDS	18. DISTRIBUTION STATEMENT
Precision maps, LIDAR, Point Cloud, Feature extraction, Connected Vehicle, Geometric Intersection Description, MAP Messages,	This document can be distributed without any restrictions

19. SECURITY CLASSIFICATION *(of this report)*	20. NUMBER OF PAGES	21. COST OF REPORT CHARGED
None.	25	

DISCLAIMER STATEMENT

This document is disseminated in the interest of information exchange. The contents of this report reflect the views of the authors who are responsible for the facts and accuracy of the data presented herein. The contents do not necessarily reflect the official views or policies of the State of California or the Federal Highway Administration. This publication does not constitute a standard, specification or regulation. This report does not constitute an endorsement by the California Department of Transportation (Caltrans) of any product described herein.

For individuals with sensory disabilities, this document is available in braille, large print, audiocassette, or compact disk. To obtain a copy of this document in one of these alternate formats, please contact: the California Department of Transportation, Division of Research Innovation, and Systems Information, MS-83, P.O. Box 942873, Sacramento, CA 94273-0001.

UCR | College of Engineering- Center for Environmental Research & Technology

Precision Mapping of the California Connected Vehicle Testbed Corridor

FINAL REPORT

Prepared for:

California Department of Transportation

By:

Center for Environmental Research and Technology
University of California at Riverside

November 2015

Disclaimer

The statements and conclusions in this report are those of the contractor and not necessarily those of the California Department of Transportation. The mention of commercial products, their source, or their use in connection with material reported herein is not to be construed as actual or implied endorsement of such products.

Acknowledgments

The authors acknowledge the funding support from the California Department of Transportation (Caltrans). We are grateful for the assistance received from the Caltrans Project Manager, Asfand Siddiqui. In addition, we thank the following individuals for their assistance in this project: Dr. Haiyu Zhang and Dr. Yiming Chen who were the key students that worked on this project.

1. Introduction

1.1. Background

As part of a sponsored Exploratory Advanced Research Program project of the Federal Highway Administration, researchers at UC Riverside have developed both hardware and software tools to precisely map features on the roadway down to decimeter-level accuracy. This precise mapping of roadway features is a necessary component of any ITS application that relies on lane-level positioning of vehicles. Further details on the mapping capability are provided below.

Over the last several years, there were a number of research programs that have used the California Connected Vehicle Testbed Corridor, including:

1) The MMITTS (Multi-Modal Intelligent Traffic Signal System) project being carried out by the UC Berkeley PATH program; and
2) The Advanced Signalization Phase II project (part of a FHWA EAR program) being carried out by UC Riverside and UC Berkeley.

These projects have required that a precise map be available for the different applications. Vehicles will not only need to know which lane they are in, but they will also need to know their precise distance away from the intersection. As part of the mapping process, various roadway features will be used, including lane markings, stop-bars, intersection features, etc.

In this exploratory project, the UCR mapping sensor hardware was mounted on an instrumented vehicle to map a segment of the California Connected Vehicle testbed corridor. After calibrating the sensor platform, the instrumented vehicle was driven repeatedly up and down the corridor of interest, collecting raw point cloud data on the on-board data servers. Then, after a sufficient amount of data was collected, the raw point cloud data was processed, extracting various features and their precise positions.

1.2. Site Location

A section of El Camino Real arterial in Palo Alto/Mountain View, San Francisco Bay Area was used for a number of ITS and connected vehicle experiments. This corridor section consists of 11 signalized intersections for a total length of 2 miles (3.2 km). There are three through lanes plus left turn lanes on each signalized intersection approach. El Camino is a major arterial with substantial commercial, retail and residential/commuter traffic. The corridor section location is shown in the maps of Figure 1.

Traffic signals are currently set up in a coordinated actuated regime. During the project, the different traffic signal controllers were being upgraded to support Signal Phase and Timing information and Dedicated Short Range Communications (DSRC).

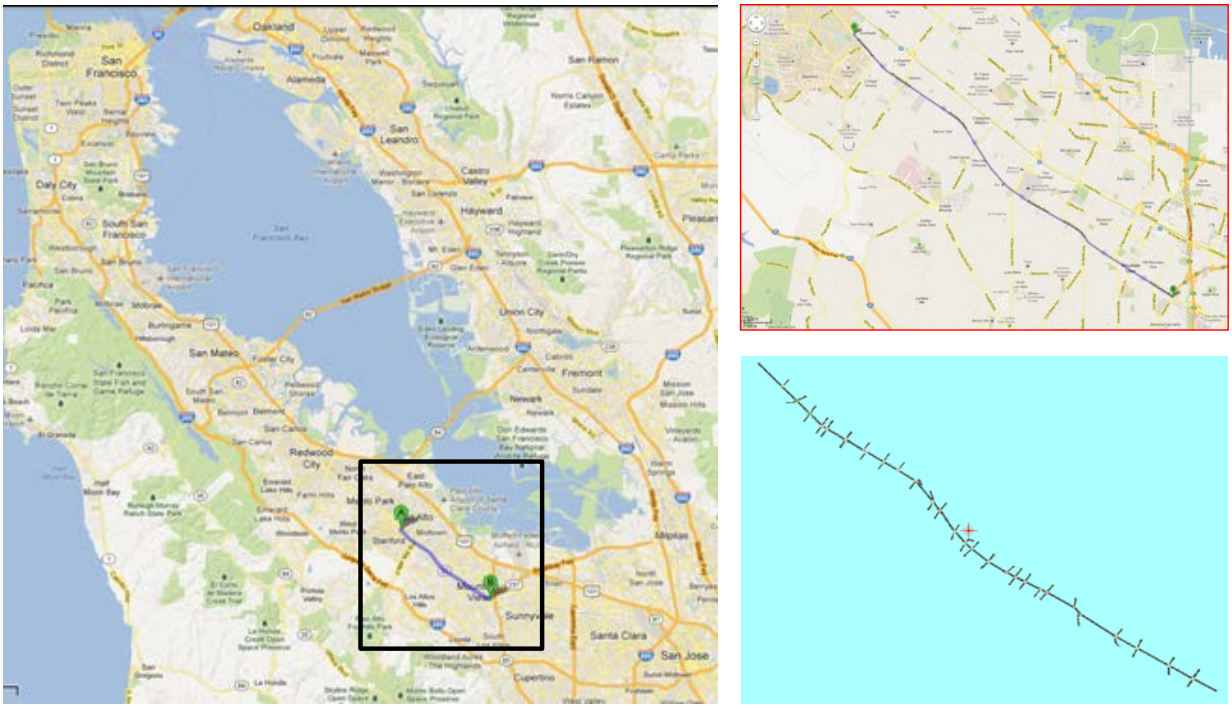


Figure 1: Location of Northern California ITS Test Bed Corridor

1.3. Mobile Positioning and Mapping System

The University of California-Riverside's CE-CERT has developed a Mobile Positioning & Mapping System (MPMS) mobile test-bed platform which collects positioning and mapping data from a variety of sensors and combines them to provide accurate, available and continuous intelligence on the state of the MPMS moving vehicle and on the surrounding areas, yielding more accurate and precise location detail and associated feature maps. This is achieved through a combination of global positioning satellite (GPS) technology, feature-based aiding sensors (vision, RADAR, LIDAR) and high-rate kinematic sensors (INS, ENS) to capture and process multiple location and feature-based signals and to bridge data gaps whenever sensor reception is interrupted. The MPMS is shown in Figure 2.

Data captured through the MPMS integrated vehicle telematics can be utilized to create a coordinate-based map of features accurately surveyed down to the decimeter level, enabling feature-based sensors to be more effectively used as navigation aides. This mobile mapping can be performed at normal arterial roadway speeds, enabling much faster data collection than what would be possible with conventional surveying techniques. Integration of data from high-rate and aiding sensors increase the accuracy and reliability of sensor fusion algorithms to accommodate asynchronous and latent sensor measurements.

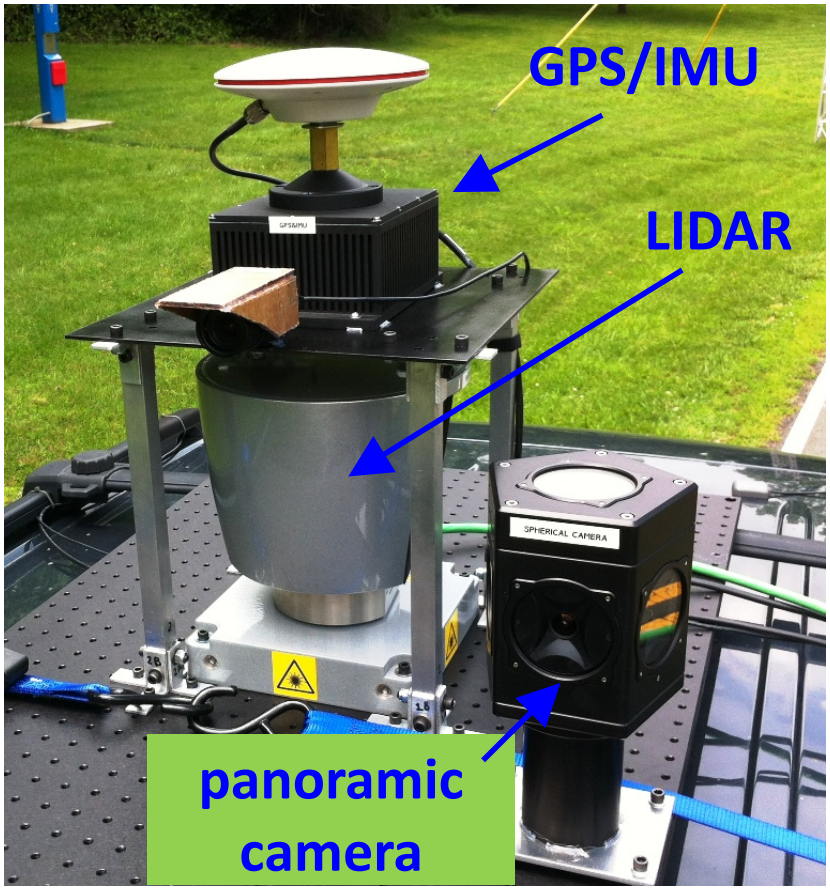


Figure 2: Mobile Positioning and Mapping System

2. Data Collection

2.1. Calibration

Prior to the data collection in the field, the MPMS was prepared and calibrated. The preparation included configuring the different sensors on the mobile platform. The entire system then underwent a complete calibration, so that the different sensors were aligned and measurement parameters are established in a real-world setting.

The MPMS was then transported up to the California ITS testbed site, mounted on a test vehicle, then be recalibrated. After this short calibration process, the data collection process began. The vehicle with the system was driven repeatedly over the test site at different times of the day, with varying levels of congestion. At the end of each data run, the raw data was examined to determine the data validity which guided the data collection process for the remaining days. Once enough data were gathered for the test site, the raw data collection was complete.

2.2. Offline Processing

The logged raw data were then processed using three major blocks as shown in Figure 3. Raw GPS, IMU, LiDAR and/or Camera measurements are input to the offline processing system. The data preparation block is the common step for all road feature extraction algorithms. In this block, the vehicle trajectory is estimated by smoothing the whole log of GPS and IMU measurements. Then the raw LiDAR data are calibrated with the factory parameters, filtered by distance, converted to global coordinate frame using the optimized vehicle trajectory and extrinsic parameters with respect to IMU. Finally, the time-series of LiDAR point clouds are stored into 151m by 151m blocks based on the North and East coordinates. The second block generates the bird's eye view intensity image of selected intersection regions and then enhances the intensity image using Morphological operations. In the third block, image processing algorithms are utilized on the intensity image to extract the stop bars as straight lines and then find the ends of each stop bar.

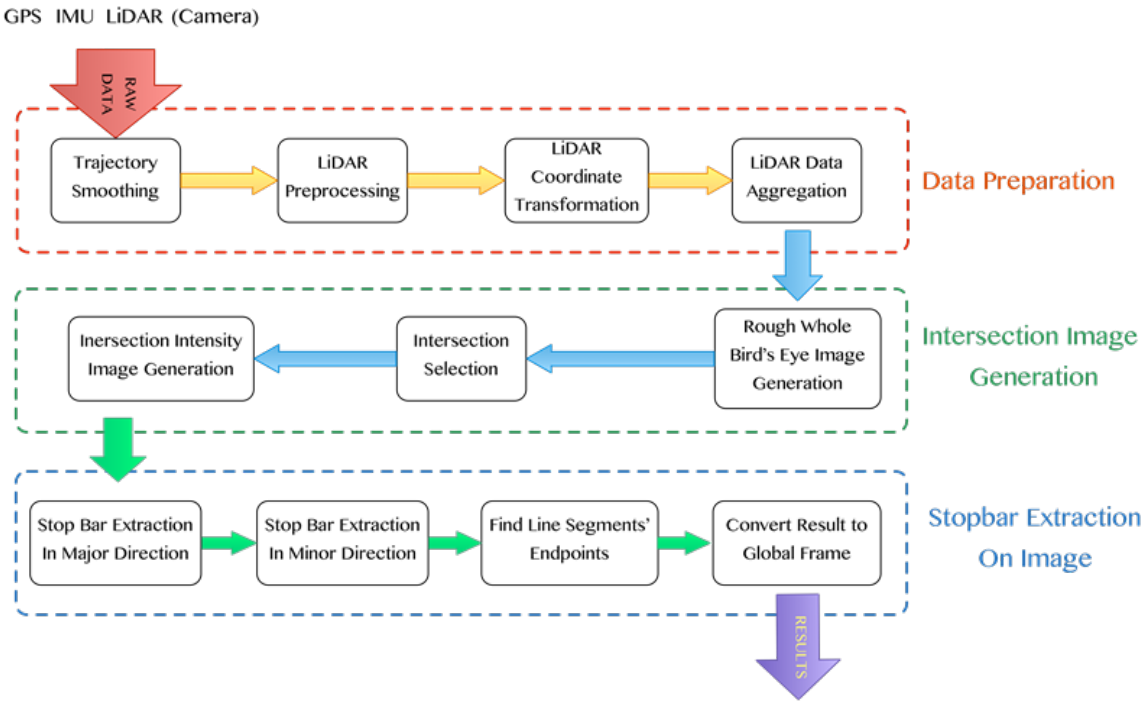


Figure 3: Data processing flow chart of our LiDAR based Mobile Mapping System

2.3. Data Preparation

The first fundamental step for all mobile mapping systems is the acquisition of an accurate mobile platform trajectory. Most commercial systems rely on a real time positioning solution by integrating GPS with IMU or Dead Reckoning using EKF. Post processing kinematic solutions (PPK) of GPS post processing could provide most accurate positioning accuracy in the popular MMS systems. Besides, commercial systems also utilize expensive IMUs for most accurate positioning.

The nature of the mapping problem does not require real time positioning solutions, such that offline post processing which may take more computation time is also viable. *Smoothing* is a non-causal operation on the data using past, present and future measurements to estimate the platform pose at each time. Compared to *Filtering* which only uses past and present measurement to estimate the present pose, Smoothing in nature has lower levels of uncertainty and greater accuracy. In this section, a Smoothing And Mapping (SAM) algorithm from [1] is briefly described. It is an extension to the Square Root Smoothing and Mapping [2].

In this algorithm, the raw dual-frequency GPS measurements that includes pseudorange, Doppler and carrier phase and raw 6-DOF IMU measurements are used as input. The mathematical model of GPS and IMU measurements have already been developed. The system kinematic model and error model have also been developed in previous work. The state of the system at each time step is defined by:

$$\hat{\mathbf{x}}_B = \left[{}^{G}\hat{\mathbf{p}}_B^T \quad {}^{G}\hat{\mathbf{v}}_B^T \quad {}^{G}\hat{\boldsymbol{\theta}}_B^T \quad {}^{B}\hat{\mathbf{b}}_a^T \quad {}^{B}\hat{\mathbf{b}}_g^T \right]^T,$$

where $\hat{\mathbf{x}}_B$ is vehicle states reported by the navigation system. The state vector is continuously estimated by numerically integrating the following equations as the mechanization equations

$$\begin{aligned}
\dot{\hat{\mathbf{x}}}_B(t) &= f\big(\hat{\mathbf{x}}_B(t), \hat{\mathbf{u}}(t)\big): \\
{}^{G}\dot{\hat{\mathbf{p}}}_B &= {}^{G}\hat{\mathbf{v}}_B \\
{}^{G}\dot{\hat{\mathbf{v}}}_B &= {}^{G}_{B}\hat{\mathbf{R}}\,{}^{B}\hat{\boldsymbol{a}}_{iB} - 2\,{}^{G}\hat{\boldsymbol{\Omega}}_{iE}\,{}^{G}\hat{\mathbf{v}}_B \\
{}^{G}_{B}\dot{\hat{\mathbf{R}}} &= {}^{G}_{B}\hat{\mathbf{R}}\big({}^{B}\hat{\boldsymbol{\Omega}}_{iB} - {}^{B}\hat{\boldsymbol{\Omega}}_{iE}\big) \\
{}^{B}\dot{\hat{\mathbf{b}}}_a &= \mathbf{0}_{3\times1} \\
{}^{B}\dot{\hat{\mathbf{b}}}_g &= \mathbf{0}_{3\times1}.
\end{aligned}$$

The smoothing algorithm optimizes the trajectory over the entire driving period. The states, IMU measurements (as input) and GPS measurements (as observation) of the entire period is

$$\begin{aligned}
\mathbf{x}_{1:M} &= [\mathbf{x}_1^T \quad \mathbf{x}_2^T \quad \cdots \quad \mathbf{x}_M^T] \\
\mathbf{u}_{1:M} &= [\mathbf{u}_1^T \quad \mathbf{u}_2^T \quad \cdots \quad \mathbf{u}_M^T] \\
\mathbf{z}_{1:M} &= [\mathbf{z}_1^T \quad \mathbf{z}_2^T \quad \cdots \quad \mathbf{z}_M^T].
\end{aligned}$$

The joint probability of $\mathbf{x}_{1:M}$ and $\mathbf{z}_{1:M}$ can be defined based on total probability rule as

$$P(\mathbf{x}_{1:M}, \mathbf{z}_{1:M}) = P(\mathbf{x}_0)\prod_{i=1}^{M} P(\mathbf{x}_\mathrm{i}|\mathbf{x}_{\mathrm{i}-1}, \mathbf{u}_{\mathrm{i}-1})\prod_{i=1}^{M} P(\mathbf{z}_\mathrm{i}|\mathbf{x}_\mathrm{i}),$$

where $P(\mathbf{x}_0)$ is a priori on the initial state, $P(\mathbf{x}_\mathrm{i}|\mathbf{x}_{\mathrm{i}-1}, \mathbf{u}_{\mathrm{i}-1})$ is the vehicle kinematic and mechanization model in probability form, and $P(\mathbf{z}_\mathrm{i}|\mathbf{x}_\mathrm{i})$ is the GPS measurement model in probability form. According to Bayes' rule,

$$P(\mathbf{x}_{1:M}|\mathbf{z}_{1:M}) = \frac{P(\mathbf{x}_{1:M}, \mathbf{z}_{1:M})}{P(\mathbf{z}_{1:M})}.$$

The denominator is irrelevant to $\mathbf{x}_{1:M}$, such that the maximum likelihood estimate is

$$\hat{\mathbf{x}}_{1:M} = \arg\max_{\mathbf{x}} \prod_{i=1}^{M} P(\mathbf{x}_\mathrm{i}|\mathbf{x}_{\mathrm{i}-1}, \mathbf{u}_{\mathrm{i}-1})\prod_{i=1}^{M} P(\mathbf{z}_\mathrm{i}|\mathbf{x}_\mathrm{i}).$$

Given the assumption that state and observation noises follow Gaussian process, and working with log-likelihood, the maximum likelihood estimate can be transformed to

$$\hat{\mathbf{x}}_{1:M} = \arg\min_{\mathbf{x}} \sum_{i=1}^{M} \left(\|\mathbf{x}_\mathrm{i} - \boldsymbol{\Phi}(\mathbf{x}_{\mathrm{i}-1}, \mathbf{u}_{\mathrm{i}-1})\|^2_{\boldsymbol{Q}_d} + \|\tilde{\mathbf{z}}_\mathrm{i} - \mathbf{h}(\mathbf{x}_\mathrm{i})\|^2_{\boldsymbol{R}_k}\right).$$

It becomes a nonlinear least square problem and could be solved with numerical method. The detailed solution to this problem, and the solution of integer ambiguity to GPS carrier phase measurements can be found in [1].

The raw LiDAR measurement of the i[th] of the 64 lasers is $[R_i \quad \delta_i \quad \varepsilon \quad int]^T$, which represents the raw distance, the angle of the i[th] laser with respect to LiDAR's x-y plane, the rotation angle of the whole LiDAR housing and finally the intensity. We utilize a transformation of the raw LiDAR measurement to the 3D coordinate in LiDAR coordinate frame. In practice, the 64 laser emitter and receivers are not aligned in a vertical line, so that several calibration parameters are involved in the actual transformation.

1. Horizontal rotation correction is the offset of the actual rotation angle of the i[th] laser from the whole LiDAR housing's rotation angle.
2. Vertical rotation correction is the offset of the actual vertical angle of the i[th] laser from the nominal vertical angle δ_i.
3. Horizontal offset is the offset of laser measurement origin from z-axis in the x-y plane.
4. Vertical offset is the offset of laser measurement origin from x-y plane in z direction.
5. Distance offset is the bias of the i[th] laser's raw distance measurement.

In the LiDAR preprocessing block, the raw LiDAR measurements are converted to 3D Cartesian coordinates with the calibration parameters and methods provided by the manufacturer. Before the transformation, a simple distance filter is applied to remove detections that are closer than 1m or farther than 75m from the LiDAR. The 3D coordinates are then passed into the next block to be transformed to global coordinate frames.

The LiDAR points in LiDAR coordinate frame are stored as a list with the time reference when the laser point detection took place. To convert the LiDAR measurement to global coordinate frame, we need to find the corresponding vehicle pose of the specific laser detection. The pose is obtained by interpolation of the two states in the smoothed trajectory whose times are closest to the given LiDAR time step. The extrinsic calibration parameters between LiDAR and body frame are denoted as the translation vector ${}^{\boldsymbol{B}}\mathbf{T}_{LB}$ and rotation matrix ${}^{\boldsymbol{B}}_{\boldsymbol{L}}\mathbf{R}$. The transformation equation is

$$ {}^{G}\mathbf{p}_L^k = {}^{L}_{B}\mathbf{R}\left({}^{B}_{G}\mathbf{R}^{i_k}\left({}^{L}\mathbf{p}_L^k - {}^{G}\mathbf{p}_B^{i_k}\right) + {}^{B}\mathbf{T}_{LB}\right), \tag{1} $$

where ${}^{L}\mathbf{p}_L^k$ is the position of k[th] LiDAR detection in LiDAR frame, ${}^{G}\mathbf{p}_B^{i_k}$ is the position of the body frame at the time step of the k[th] LiDAR detection and ${}^{B}_{G}\mathbf{R}^{i_k}$ is the rotation matrix that represents body frame's attitude.

The preprocessed data of the entire run is around 4 gigabytes, which hardly fits the computer memory. It is partitioned to pieces of 40 megabytes which contains 1000 cycles. Each partition is converted to global frame separately, and then passed into the next block.

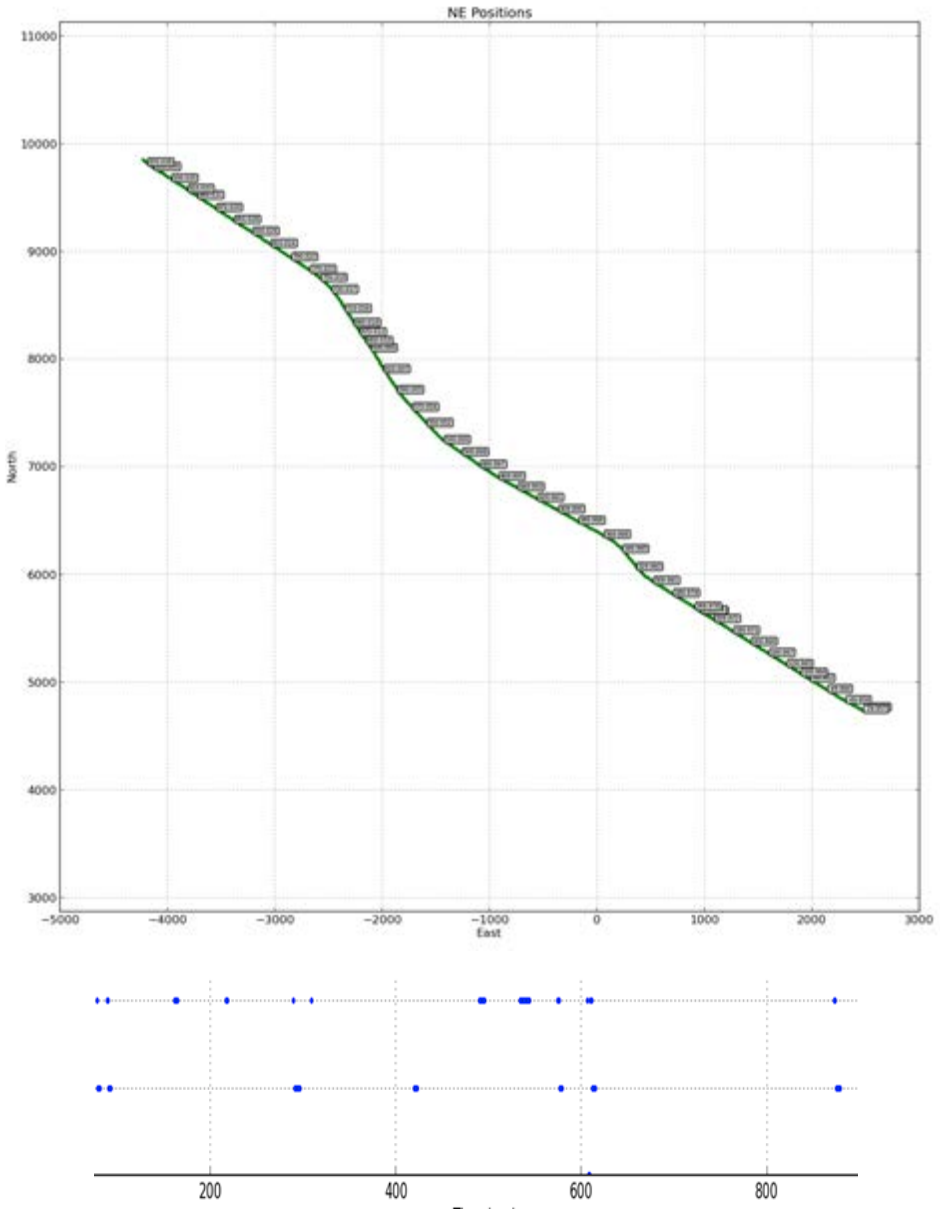


Figure 4: The first plot (a) is the plot of the estimated north and east coordinates of the trajectory. The second plot (b) is the number of satellites above predefined elevation.

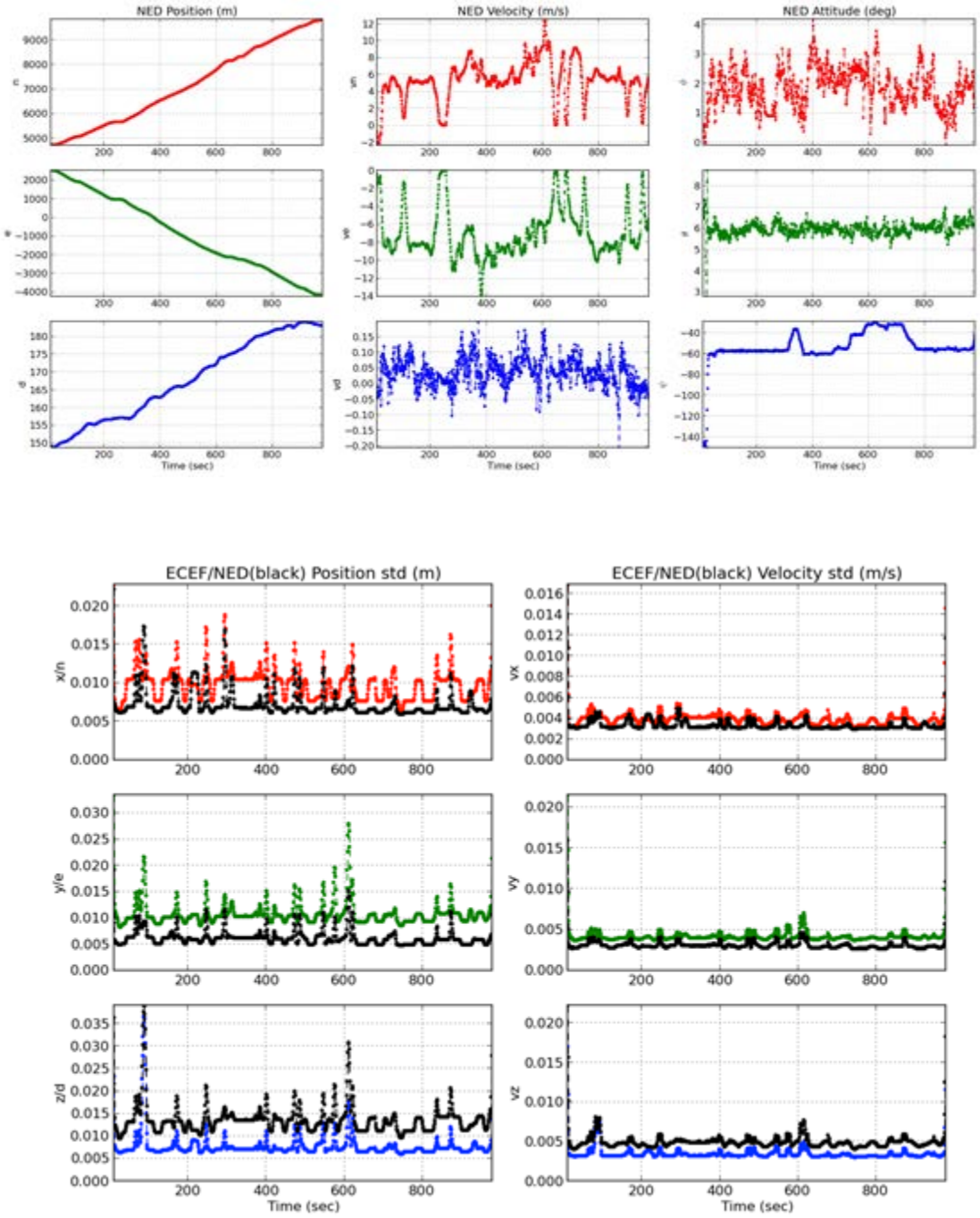


Figure 5: The first three rows are the smoothing results of the position, velocity and attitude in NED frame. The bottom three rows show the standard deviations of the position and velocity estimates. In general, the overall positioning standard deviation is below 2cm. If we refer to Figure 4, it can be seen that the large position standard deviation happens when the number usable satellites are low.

The LiDAR data in global frame from the previous block need to be stored into database or simply data files on hard drive, because different applications could implement different algorithms on these preprocessed data. Instead of using the whole trajectory of LiDAR data, the feature extraction algorithms

usually process only small sections of data to fit the limited memory. The simple way of storing all LiDAR data into a single file has several disadvantages. Firstly, the file would be too large in size due to the large amount of LiDAR data, so that it could not easily be loaded into memory, and it also takes too long. Secondly, obtaining specific LiDAR point requires traversing the whole file, which requires unnecessary time cost. So a distributed storage architecture is designed and implemented in our system.

To support the following procedures which extract the intersection region, the LiDAR data is stored into several non-overlapping blocks of data files. Each block contains all the LiDAR points that falls into the north and east boundaries of the block. In our system, each block is a box with infinite length along down-axis. The box covers 151m by 151m region in North-East plane. The size of the box is selected such that the LiDAR points from each LiDAR scan cycle (full 360̊) would fall into at most four such blocks in the worst scenario to make reduce the memory usage. The index of each block is the coordinate of its North and East corner.

The LiDAR points in the block can be simply stored as plane list. If the processing algorithms are based on 3D point cloud algorithms, the unstructured list is enough. In our algorithm, the next step requires the creation of a 2D bird's eye view of intersection, so a matrix like structure is selected. The 151m by 151m block is partitioned into 10cm by 10cm cells in North-East plane. Each cell stores a list of LiDAR points within the NE region.

2.4. Intersection Image Generation

After LiDAR data aggregation and storage, a 2D intensity image in bird's eye view of the complete trajectory is built. In the previous step, the NED coordinate of each LiDAR point is stored in the block file. According to the definition of a local tangent frame, the origin of the local tangent frame is close to the test field. In this case, the North-East plane could be a rough approximation of the actual earth surface.

The full image is created by merging images of each block file. The single image of each block file contains 1510 by 1510 pixels. In this step, the intensity of each pixel is the average intensity of all LiDAR points that falls into the corresponding 10cm by 10cm cells. The intensity of LiDAR detection is in the range from 0 to 255, which could be directly used as grayscale value in the image. The result of a single run along the test route, as shown below.

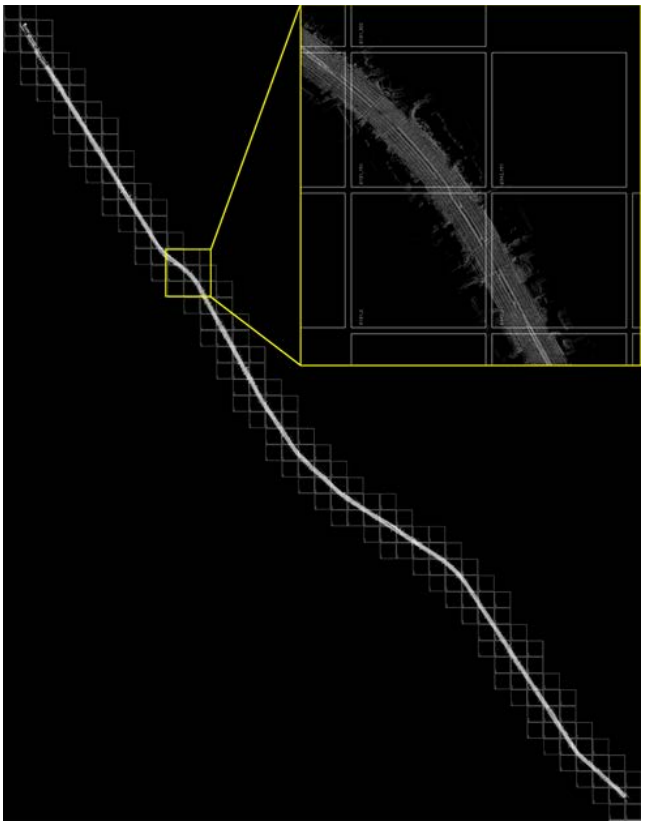


Figure 6: The bird's eye view intensity image of the whole trajectory is demonstrated. The north direction is pointing down, and the east direction is pointing right. Each small square represents the region of a block file.

The intersection region that needs to be processed is selected by detecting the mouse click position on the full image. The results of the selection are the lower and upper boundary in both North and East positions. Though the intensity image is already generated in the previous step, the focus for the intersection intensity image is on the road surface. The intensity image using average cell intensity did not filter out the above-road-surface objects like trees and other vehicle and would introduce noises to the image. The method used here utilizes the fact that the accumulated LiDAR point cloud on the road surface should occupy the lowest dense layer of points in each cell.

3. Data Processing

Before the operation, the LiDAR point cloud from different runs (in our experiment, two runs on both directions) are preprocessed and put into the same block files. In this way, the stored point cloud would have better coverage on both directions of the road surface. In the intensity image generation algorithm, the LiDAR points in each cell is sorted by their down coordinates. Then we traverse from the bottom point up to connect consecutive points with height difference smaller than a threshold. The threshold is chosen to be 5cm. Each connected cluster is defined as a layer, and the bottom layer has high probability to be the road surface layer. The intensity of the cell is represented by the median of the intensity of the bottom layer. The results are shown in Figure 7.

As an exercise of feature extraction, we have applied image processing algorithms to reliably extract stop bars on the intensity image generated from previous steps. The algorithm also uses the vehicle trajectory as an indicator of road area. In the following description, we define the major direction as the direction of the vehicle trajectory, and minor direction as the direction perpendicular to the major direction.

The essence of the stop bar extraction is to detect straight lines with high intensity using the specific characteristics of the stop bars. One key characteristic of the stop bars in the major direction is that they are mostly perpendicular to the driving direction. However, the lane markers on the minor direction have a very similar property. So the intensity image is first masked using the vehicle trajectory on both sides to remove the lane markers on the minor directions. The driving route of the mapping platform always crosses the stop bars in the major direction, so the intersection image is masked with the inflated trajectories in both directions. The masked image is shown in Figure 7b.

Even though the camera images cover every pixel in the FOV, the LiDAR images do not guarantee that every cell which is represented as pixels in the intensity image has LiDAR detection in it. The LiDAR scans in nature are sparse. This results in a large number of jagged edges in the intensity image. Image enhancement is thus necessary. Image morphological operations and smoothing are common techniques for image enhancement. At this stage, Gaussian smoothing is applied to the intensity image. The resulting image is Figure 7c.

It is difficult to extract edges of stop bars directly on the whole intersection image, because there are many other straight lines that has similar properties. So the idea is to roughly extract the stop bars first, and then fit straight lines in the affinity of the stop bars. A Probabilistic Hough Transform (PHT) is applied to do the initial extraction. PHT algorithm is an efficient straight line voting algorithm. It detects and extracts straight line segments from binary images. There are three parameters in the PHT algorithm, voting threshold, minimum line length and maximum line gap.

The binary image is created by thresholding the grayscale image of masked intersection. The threshold is chosen to such that road surface is removed and bright stop bars are mostly reserved. After the initial step, 8-connected component labelling is applied to the image, and all connected components that has pixels smaller than 30 are removed. The binary image is shown as Figure 7d.

The parameters of PHT algorithm are loosely chosen such that multiple line segments could be extracted on the stop bars. The line segments are filtered based on their angle, and clustered based on special

proximity, in order to find one line segment on each stop bar. The PHT results before and after clustering are shown as Figure 7e and 7f.

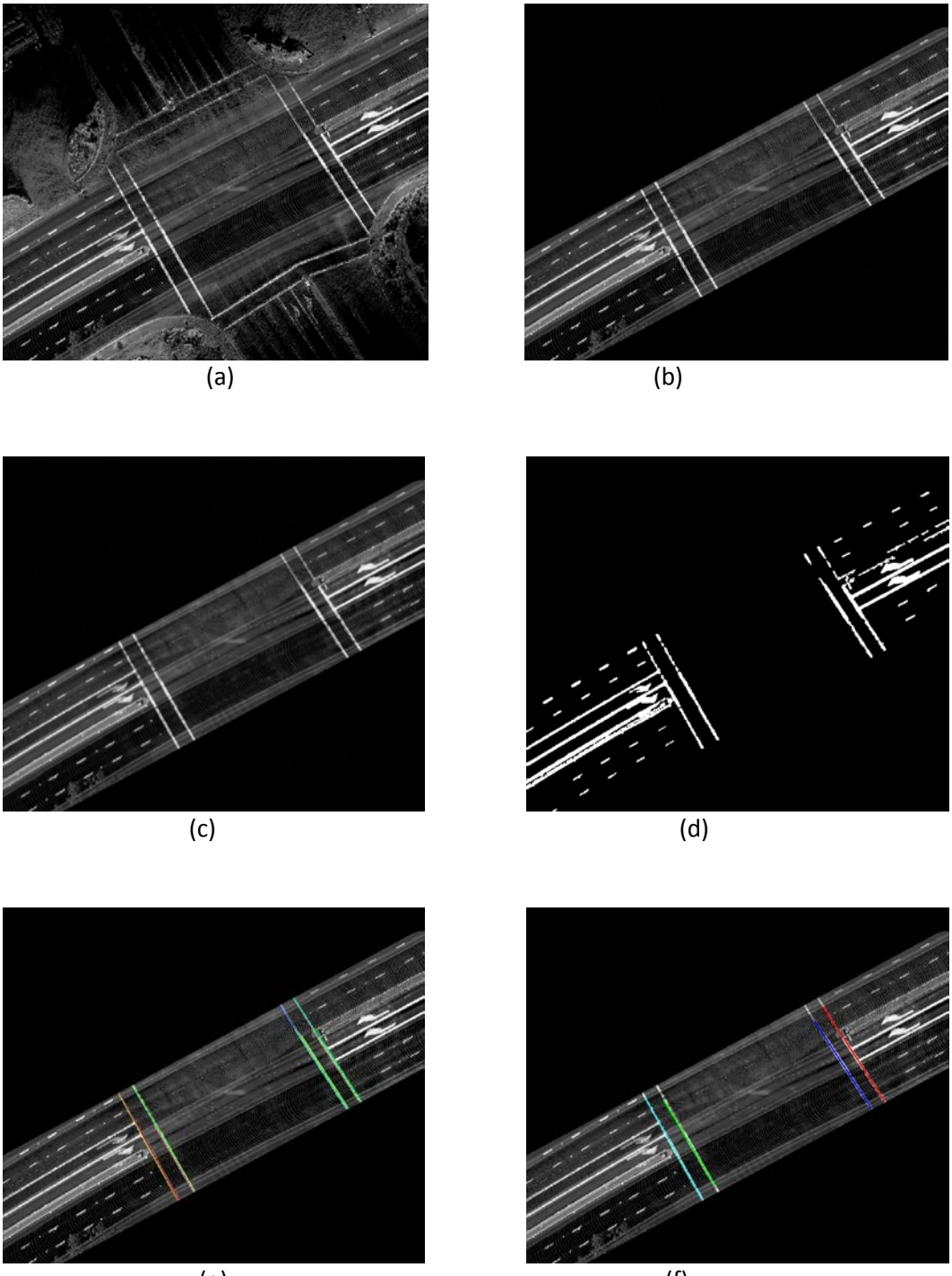


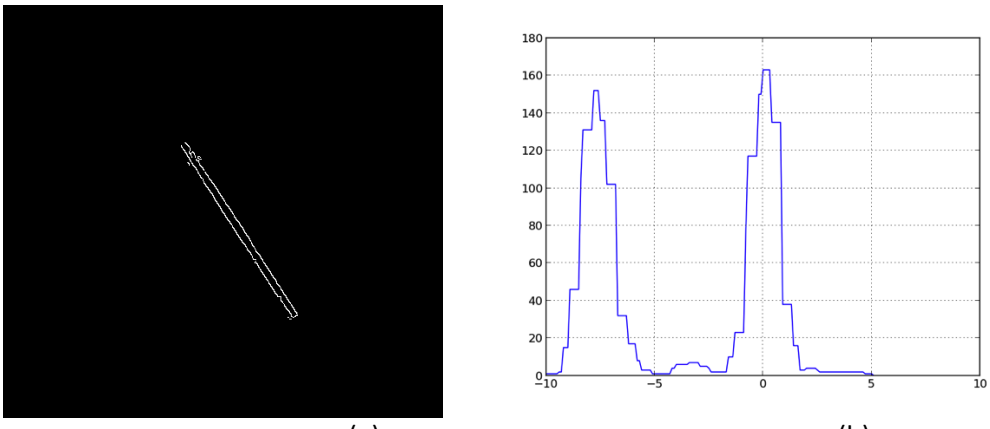
Figure 7: (a) is the original image; (b) is the image masked by trajectories; (c) is the smoothed image of (b); (d) is binary image of (c); (e) shows the line segments of PHT algorithm overlaid on the original image; (f) shows the final line segments after clustering.

The line segments detected with PHT does not guarantee accuracy or exact line-ends. Therefore, we developed an algorithm to fit straight lines on both edges of each stop bar. To generate the edge image, some preprocessing needs to be done. One is image closing, and another is Gaussian smoothing. Image closing is a morphological operation that combines erosion and dilation. Image closing operation closes holes within the image, both on the dark surface area and the bright bar areas. The Gaussian smoothing is used to smooth the jagged edges of stop bars.

After the image preprocessing step, a Canny edge detector is applied to the image. Canny edge detector is a commonly used algorithm to detect edges in a grayscale image. The algorithm can be found in many computer vision textbooks. The thresholds are also loosely set to reserve more edges because only edges around the extracted stop bars would be used for line fitting.

The edge image is masked with inflated stop bar line segments using bitwise and operation. A single stop bar extracted from PHT algorithm is inflated to 15 pixels in width. The results covers the whole stop bar region, but removes everything else. The result preserves only edges of the current stop bar with noise. One of the resulting edge images is shown in Figure 8a. Hough transform is a discrete voting algorithm for straight line extraction. In the edge image around a single stop bar, ideally, the Hough transform line detector would return two peaks in the Hough space. However, sometimes when two edges of one stop bar are close to each other, the Hough peak extractor would result in false detection. So in our algorithm, the largest peak in the Hough space is selected. The straight line of the other edge is extracted as follows.

The straight line parameters of the edge that is already extracted is defined as a tuple (ρ, θ), where ρ is the shortest distance from the origin to the line, and θ is the angle of the normal vector of the line. The edge image is correlated with straight lines with parameters $(\rho + \varepsilon, \theta)$ where $\varepsilon \in [-9.9, -9.8, \dots 0, 0.1, \dots, 9.9]$. The correlation is calculated as the number of ones on the line. This kind of correlation could generate two peaks on the two edges of the stop bar which represent two parallel straight lines. The two peaks of the correlation curve is detected using continuous wavelet transform (find_peak_cwt function in Scipy library). The correlation curve of Figure 8a is shown in Figure 8b.



(a) (b)

Figure 8: (a) is the edges around one of the stop bars (b) is the correlation plot.

After the stop bars edges of the major directions are extracted, the original intensity image is masked using the extracted edges to remove lane markers in the major direction. Only the area between the stop bars on both sides of the intersection is kept. The result is shown in Figure 9. The same operations are then applied to the masked intensity image. After all the stop bar edges are extracted, they are plot onto the original intensity image, as shown in Figure 10.

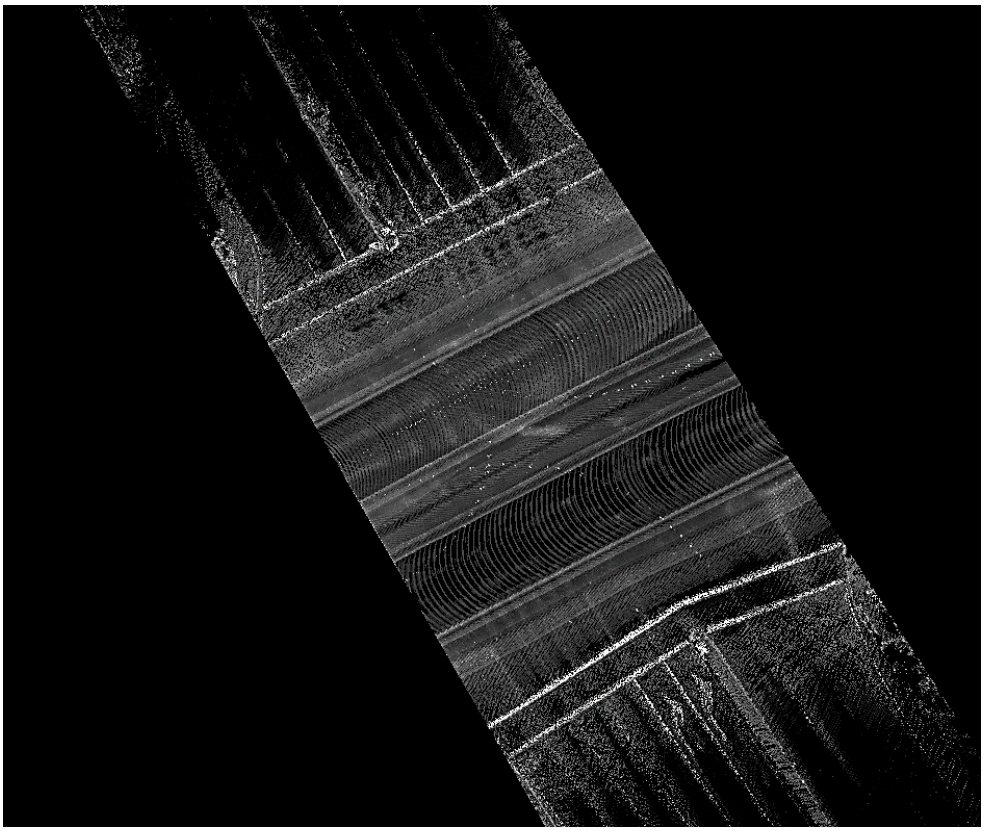


Figure 9: The intensity image is cropped with the stop bar in major direction. The area between stop bars on both sides are kept. The lane markers behind the stop bars are removed.

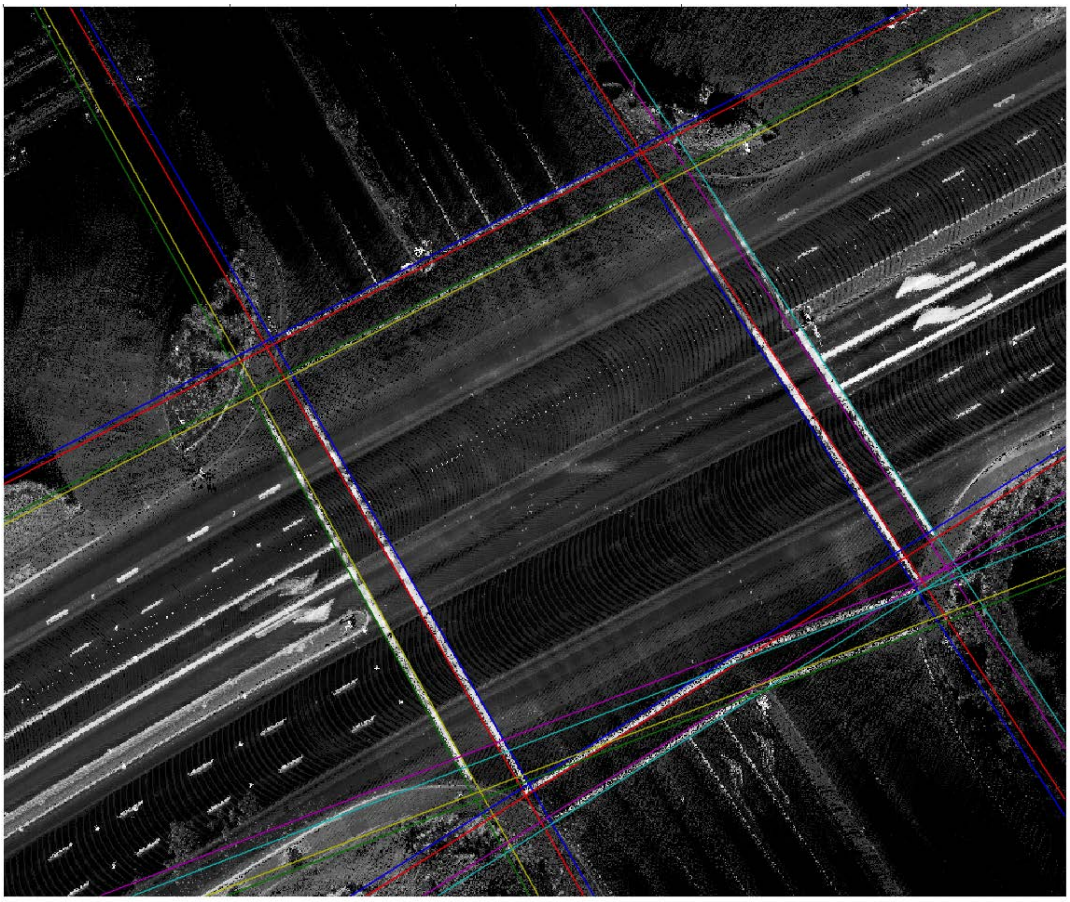


Figure 10: The results of the stop bar edge extraction are overlaid on the original intensity image.

The results are straight lines with infinite length. Our target is the accurate representation of the stop bars, so the endpoints from the straight lines need to be determined. The first step is the combination of edges. The extracted edges from the same stop bar are combined and averaged to obtain the straight line estimate of the stop bar itself. The average straight lines are around the middle line of the stop bars, so most of the pixels along those lines have high intensity. This fact is utilized in the following steps.

The second step calculates the intersections of all pairs of lines from the first step. The intersection points on a single line is sorted by the x or y coordinate. By connecting consecutive intersection points that are already sorted, a set of mutually exclusive line segments are generated. These line segments are guaranteed to be on certain stop bars.

It can be seen from Figure 11 that some of the line segments generated from step 2 are not actually on the image. We use the fact that an effective line segment should go through bright strips and both sides of the strip should be dark areas. A similar shift and correlate algorithm is applied. The algorithm is demonstrated in Figure 12.

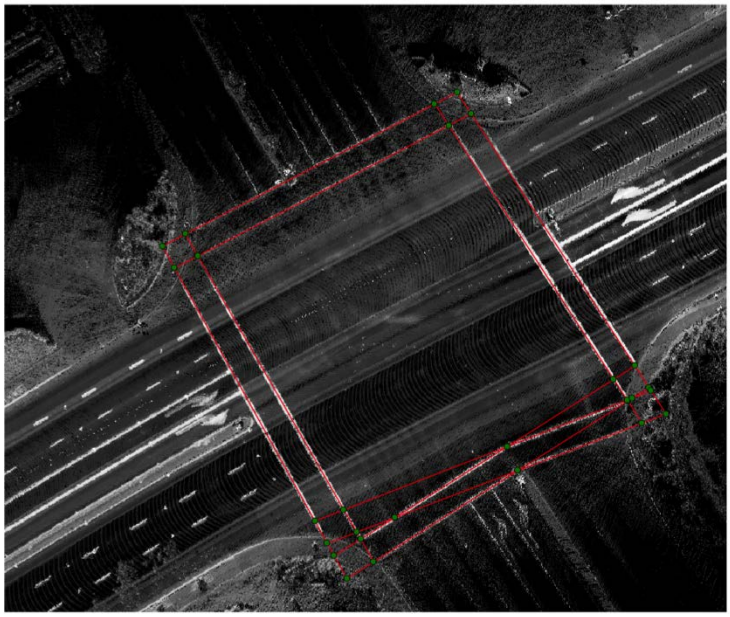


Figure 11: The extract stop bar center lines are shown in red. The intersections are green dots.

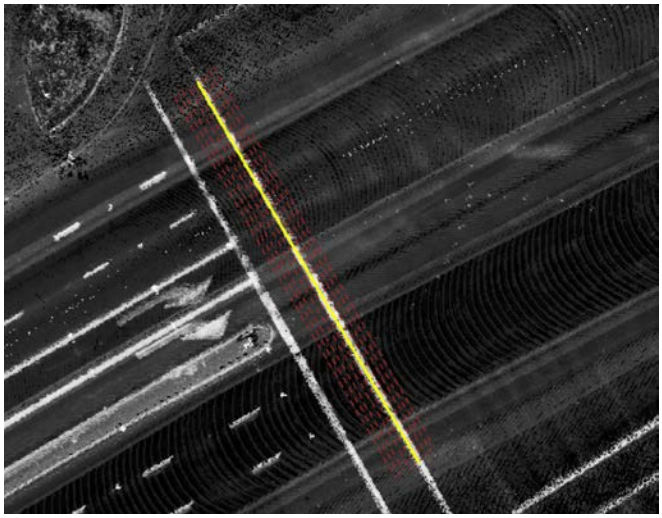
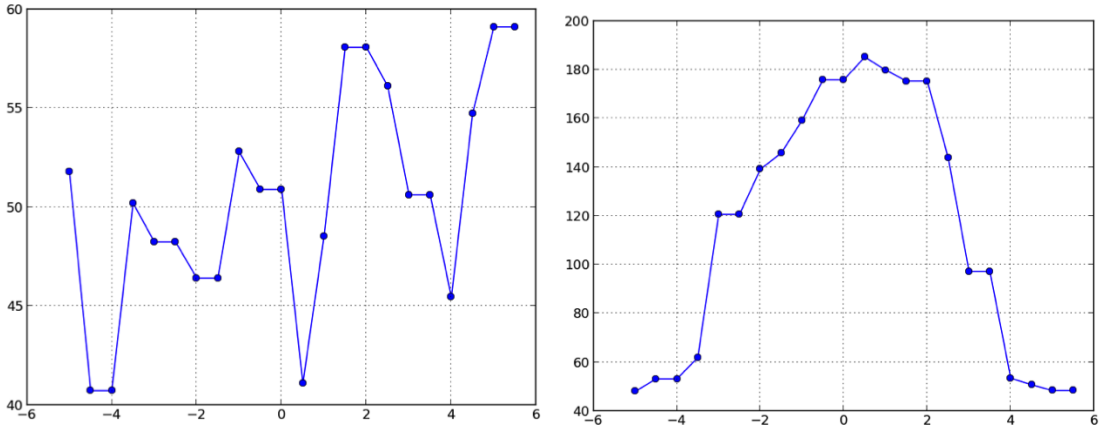


Figure 12: Demonstration of shift and correlate algorithm. The yellow line represents the stop bar center line. The dashed red lines represents the lines shifted in the perpendicular direction.

The stop bar center line is shifted in the perpendicular direction for $\varepsilon \in [-5, -4.5, \dots 0, 0.5, \dots, 4.5]$. The intensities of the pixels on the shifted line segment is averaged, and recoded as y. The plot of (ε, y) has different shape on and off the stop bar as shown in Figure 13. The average intensity on the stop bar would be large, and the average intensity off the stop bar would be small. So the correlation plot of the effective line segments has a hill like shape, while the plot of the invalid line segments have a random shape. To discriminate between these two conditions, a 2nd order polynomial is fitted to the plot. The polynomial coefficient of the effective stop bar has the following property (if we define the polynomial as $y = ax^2 + bx + c$:

1. The coefficient a that defines the curvature is smaller than -1. The coefficient a of an invalid line segment is usually on the level of 0.01.
2. The symmetry axis of the fitted parabola -- $-b/2a$ – is close to 0.

If both conditions are satisfied, the line segment to be tested is identified as valid. After all segments are tested, the valid segments on the same straight line which shares common endpoints are merged together. The results are shown in Figure 14.



(a) (b)

Figure 13: (a) is the correlation plot of the line segments not on the road surface. (b) is the correlation plot of the line segments on the road surface.

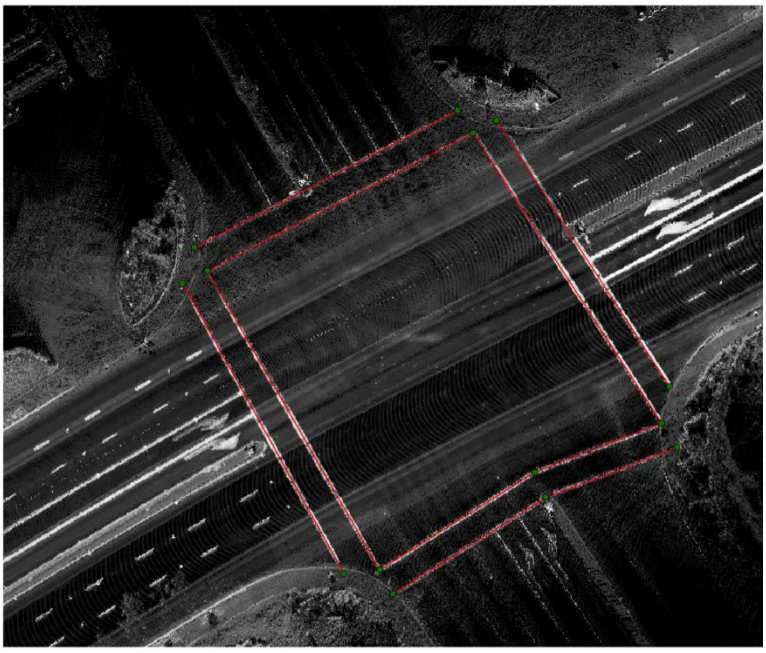


Figure 14: The extracted stop bar center lines after verification are shown in red. The intersections are green dots.

The end point of the stop bar center line is used to find the endpoint of stop bar edges. A line perpendicular to the centerline and passes through the endpoint intersects with both edges, and the intersections are considered as the endpoint of the stop bar edges.

Our system is unique in the positioning subsystem by estimating mapping platform trajectory using smoothing algorithm for carrier phase DGPS and IMU measurements. The storage of the vast amount of LiDAR point cloud data is challenging and is solved with a distributed storage system. An image processing based intersection stop bar extraction algorithm is proposed and verified with field test data. On a well-structured intersection, the proposed algorithm provides accurate and reliable extraction of stop bar edges.

4. Conclusions

In this exploratory project, the UCR mapping sensor hardware was successfully mounted on an instrumented vehicle to map a segment of the California Connected Vehicle testbed corridor. After calibrating the sensor platform, the instrumented vehicle was driven repeatedly up and down the corridor of interest, collecting raw point cloud data on the on-board data servers. Then, after a sufficient amount of data was collected, the raw point cloud data was processed, extracting stop bars as an example feature along with their precise positions.

This proof-of-concept project was successful in showing that it is possible to collect precise map information from a mobile sensor platform. The key next steps are to formalize the process and to extend the work with more elaborate feature extraction algorithms.

5. References

[1] Vu, A., J. Farrell, and M. Barth, *Centimeter-Accuracy Smoothed Vehicle Trajectory Estimation.* Intelligent Transportation Systems Magazine, IEEE, 2013. **5**(4): p. 121-135.

[2] Dellaert, F. and M. Kaess, *Square Root SAM: Simultaneous localization and mapping via square root information smoothing.* The International Journal of Robotics Research, 2006. **25**(12): p. 1181-1203.