Digital Terrain Model Generation using LiDAR Ground Points

Andi Zang
Nokia Technologies, Chicago
425 W Randolph St
Chicago, Illinois 60606
andi.zang@nokia.com

Xin Chen
HERE, a Nokia company
425 W Randolph St
Chicago, Illinois 60606
xin.5.chen@here.com

Goce Trajcevski
Dept. of EECS, Northwestern University
L360 Tech, 2145 Sheridan Rd
Evanston, Illinois 60208
goce@eecs.northwestern.edu

ABSTRACT

As the trend of autonomous self-driving cars is becoming more of a reality, High-quality navigation methods and tools become a paramount. This, in turn, is crucially dependent on High-definition maps, for which one of the enabling tools is high resolution *Digital Terrain Model (DTM)* – the role and values of which have already been demonstrated even in the settings of manned cars. Traditional DTM generation methods have insurmountable barriers in creating centimeter-level resolution. In this paper, we propose a novel method for fully-automated, high precision DTM generation using the database generated and maintained in our existed dataset, and with no additional overheads in terms of extract labor and equipment cost. The input data is a point cloud captured by the vehicle-mount LiDAR devices which, naturally, has extremely large volume. We show how with Ground Points Processing and DTM Generation steps, we can generate a centimeter-resolution DTM and, as our experiments demonstrate, when compared to DTM form U.S. Geological Survey (USGS) and altitude data from a third party surveying dataset, our proposed DTM indeed provides a higher precision.

Keywords

Point cloud processing, Digital Terrain Model, LiDAR, GIS

1. INTRODUCTION AND MOTIVATION

With the developments of GPS-based autonomously driving cars and trucks[14] [12], the demand for high-quality navigation and HD maps is ever increasing and, consequently, high resolution *Digital Terrain Modeling* (DTM) is a paramount. To survey a large area from a county to a country, most of DTM providers, e.g., U.S. Geological Survey (USGS) and Earth Remote Sensing Data Analysis Center (ERDAC), use flying platforms such as remote sensing satellites and airborne laser scanner. These acquisition techniques, while enabling a large territory to be covered, have certain notable disadvantages for their use in the context of autonomous vehicles. Firstly, the data captured by flying platforms has low precision and introduces measurement errors – typically, the resolution of these kinds of DTM are meter-level. Secondly, due to objective surveying methodologies constraints, such as cost and operating altitude, system error and random error are hard to reduce. Thirdly, the data captured is originally a Digital Surface Model (DSM), which contains buildings, trees and other objects which, in turn, can distort the representation of the ground-level that is essential for navigation purposes.

UrbanGIS'15 November 03-06 2015, Bellevue, WA, USA

ACM 978-1-4503-3973-5/15/11 ...$15.00.
DOIs: http://dx.doi.org/10.1145/2835022.2835024

The computation of DTM relies on two basic methodologies: it can be automatically computed from DSM [6] or additional labor is needed to manually remove the objects occluding the terrain. Thus, a high precision altitudes information can be measured, verified and/or corrected by manual survey, however, such "control points" require a lot of manpower and resources, and can only provide a coverage of a small area with which they are co-located.

Since the airborne LiDAR cannot penetrate through obstacles that cover the underlying terrain (e.g., trees in a forest), the idea to use ground-based LiDAR to generate DTM has been proposed in creating the ground-level DTM [7][11] [16]. We note that there are other methodologies for removing non-surface objects (cf. [13] and [10]), however, these approaches can only be applied in the settings in which which non-surface object areas are much smaller than the surrounding/detected surface area. Essentially, the techniques described in [13, 10] rely on sensors that are fixed on ground, which is the main reason for the limitation to generate only small clean-regions. Ground LiDAR also suffers from the problem of being blocked by non-surface objects and in most cases, an interpolation is used to generate the data for the non-covered ground data. Similarly to the the case for airborn LiDAR, there are techniques [2, 5] that can solve the problem for the settings in which missing region area is much smaller than the known region area. However, in many realistic scenarios, non-surface objects are not small objects – they can even include buildings and other large size objects.

In this paper, we address the problem of creating high resolution DTM using ground based LiDAR points in a large area, and our approaches are especially amenable to urban areas. The proposed methodologies can create centimeter-level using our dataset automatically, with no need for manual interventions. The main contributions of this work are summarized as follows:

1. We extract ground points from huge size of point cloud data, this step offers reliable input data for DTM generation.
2. Bridges, tunnels and overpasses can be effectively detected in our method, which makes the proposed DTM approach much more accurate.

3. We design a region-function based region interpolation method, which can restore the missing details accurately.
4. We provide experimental observations which provide a quantitative demonstration of the advantages of the proposed methodology.

The rest of this paper is structured as follows. In Section 2 we present the preliminary background and introduce the notation that will be used throughout the rest of the paper. Section 3 presents the issues and solutions related to ground points processing and Section 4 discussed in detail the aspects of the DTM generation. Experimental results are presented in Section 5, which is followed by Section 6 in which we overview the state of the art and position the proposed work in that context. We summarize the paper and outline directions for future work in Section 7.

2. PRELIMINARIES

The field of DTM emerged as a technology attempting to improve the quality of geological photogrammetry [1], and it was within DTM framework that DEMs – 2D discrete functions of elevation – became the main source of information on topography. In addition to the problem-domain addressed in this work, DTM is widely used to solve various multiscale problems of geomorphology, hydrology, remote sensing, soil science, geology, geophysics, geobotany, glaciology, oceanology, climatology, planetology, and other disciplines-see reviews [4][9]. Developing a DTM for urban regions, especially ones involving mountainous terrain presents a plethora of challenges. Complex geomorphological shapes, trees/forests, along with unevenly shaped and sized buildings, make the classic methods inadequate and non-applicable.

Figure 1: All drives plot in San Francisco area.

Due to its natural topographical and urban features, the city of San Francisco has been subject to a variety of studies which, in turn, have provided plenty of credible DEM resources – for example: USGS[15], National Oceanic and Atmospheric Administration (NOAA)[8], as well as third party survey data. While we can use each of them to cross-validate our results, in our dataset we have a point cloud captured by acquisition vehicles, as follows: for each point, we record its position is World Geodetic System 1984 (WGS84) [3] coordinate, which contains latitude, longitude and altitude information in fifteen digits of precision. Along with these, there are other information-items recorded such as, e.g., time stamp, LiDAR reflected intensity and device information – however, throughout this paper we only utilize the spatial position information.

Given a particular acquisition vehicle AV_i, we use the term *Drive(AV_i)* to denote each acquisition operation by that vehicle. The LiDAR device keeps detecting objects while the vehicle is moving. Typically, the number of points recorded along a strip of ten meters road in urban regions is approximately three millions, whereas for the highway, the corresponding number is approximately one million. The respective recorded points represent the surfaces of not only road, but also other objects such as vehicles, trees, pedestrians, buildings. In our dataset, 248 Drives drive inside or through San Francisco Region, with the total drive distance of 34,278 km.

The first major challenge addressed in this paper is how to handle the extremely large sized datasets of points collected during the recording/acquisition. An additional challenge stems from the fact that, while the acquisition vehicles drive on road surface, it may be the case that sometimes the roads are not on the terrain surface. An Example is shown in Figure 1 which shows cases of both below terrain surface (tunnel) and above terrain surface (bridge). Such points may introduce various kinds of errors and cause a degree of a misleading in the outcome of the development of DEMs. Thus, the second major challenge is how can we detect and remove such data/points. The last challenge tackled in this paper occurs when one misses the elevation information of the non-road regions. In such settings, one needs to interpolate and estimate elevation information in the missing regions.

3. GROUND POINTS PROCESSING

We re-iterate that the data size of LiDAR point cloud is extremely large – as an example, the generation of only 10 meters urban road chunk occupies approximately 100 MBs. Using entire point cloud from all the target Drives requires unacceptably large amount of computational resources – however, we note that the raw point cloud contains every spatial entity captured, such as buildings, trees and vehicles. Thus, eliminating redundant information from the point cloud data and focusing on the subset of relevance for DTM generation can significantly speed-up the overall process.

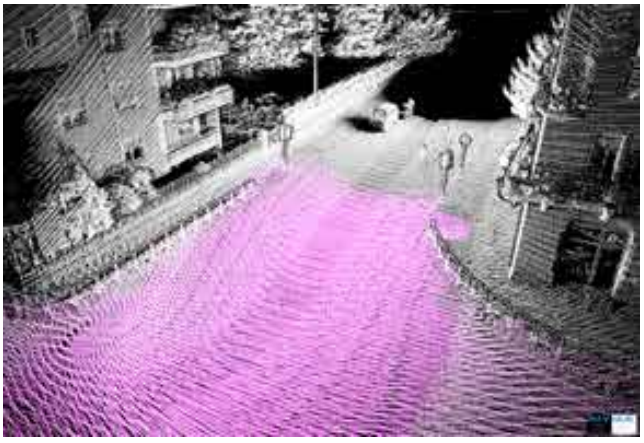


Figure 2: Road surface points(purple) are extracted in street view point cloud

3.1 Road Surface Detection

The first step towards in ground points processing is to retain the road surface points and filter out all the other points. This step in not a novel contribution of the present paper and has already been solved as part of the projects in our road features extraction

pipeline – however, we describe it here for completeness. The basic idea of this step is to first find the trajectory points (which is, the points right under the acquisition vehicle with a certain time interval) recorded with the correlated point cloud data, and then use region growing method to search neighbor points until a non-surface structure is intersected. These may include curbs, guard rails, building, other vehicles and grass. Then the connected ground points will be filtered out from entire point cloud. An illustration of extracted surface points in 3 chunks point cloud is shown in Figure 2.

3.2 Ground Points Selection and Projection

The number of road surface points after road surface detection is still too large for practical applications – however, it provides the superset (i.e., candidates) of all the ground points needed. To further reduce the number of points, a sampling process needs to be applied, and the sampling rate depends on the DTM resolution.

Let x and y denote the location in two dimension DTM coordinate, and $DEM(x, y)$ denote the altitude information at location (x, y). Given the upper left latitude and longitude of DTM in WGS84 coordinate, we have:

$\begin{bmatrix} lat & lon \end{bmatrix}$, resolution $\begin{bmatrix} rlat & rlon \end{bmatrix}$ are latitude per pixel and longitude per pixel and the DTM size $\begin{bmatrix} height & width \end{bmatrix}$, we can project ground point to our model by using Equation 1:

$$DTM(x,y) = \begin{bmatrix} height \\ width \end{bmatrix} + \frac{(P - \begin{bmatrix} lat \\ lon \end{bmatrix})}{\begin{bmatrix} -rlat \\ rlon \end{bmatrix}'} \tag{1}$$

Depending on the DTM resolution, typically, we have millions of pixels for a given one meter/pixel resolution urban model, with almost two orders of magnitude larger input-set of points. This, in turn, implies that some points will inevitably be projected onto a same pixel in the model. These points which are collectively assigned to a particular pixel have height difference distribution and for each road surface pixel the altitude information is assumed to have a distribution normal distribution, to capture the impact of the errors of measurements instrumentation [**?**]. As often done in the literature, we select the median value for of all these road-surface points to determine the value associated with a particular pixel, with a bounded error.

4. DTM GENERATION

Our GPs data have been aligned and corrected when stored in system, so the point cloud from each drive has nearly no device error. To develop the urban DTM, only using road region altitude information directly is not enough. There are still two problems need to solve. First of all, some of the road surfaces are not built right on terrain surface, for example, tunnels and bridges. At same time, flyover (multi-layer bridge) will also confuse the DEM development. Second, we do not have the altitude information "off-road", for example, buildings, parks and other un-drivable regions. So that we need to estimate the missing altitude information in these regions.

4.1 Bridge, Tunnel, Flyover Detection

Bridges, tunnels and flyovers will always give us wrong terrain altitude information under or above them. Hence finding them then correcting these kinds of road structures are important and necessary before interpolate the missing regions. To detect a multi-layer bridge structure, we need to analyze the histogram and standard deviation of the GP set which project onto one pixel, instead of selecting the median value of GP set directly. The solution is for each GP set $P(x, y) = \{GP_1, GP_2, GP_n\}$, we calculate a standard deviation map

$$S(x,y) = STD(P(x,y)) \tag{2}$$

In map S, higher value means the road passes this pixel may have multiple layers, other pixels which have lower standard deviations are only have altitude errors. In Figure 3, we visualize a normalized standard deviation intensity map. The brighter pixels are the chunks where bridge goes over the road which can been validated from Figure 3b, a 3D map from HERE 3D map. The red pixel in Figure 3a is the pixel which has the most standard deviation, the normalized values in a 7 by 7 neighborhood region are show in figure 3c. The reason causes this result is we have several vehicles acquire LiDAR data on both two layer drives. Figure 3d shows the altitude histogram of points which project onto red pixel. Comparing to the neighbor road altitude information, we can select the median value of left peak as the altitude information at this pixel.

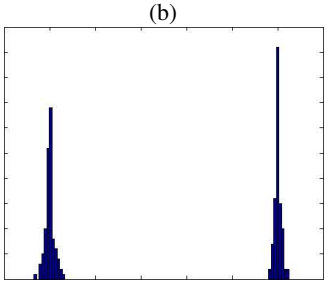
Bridge and tunnel detection need extra DEM from other resources, such as USGS. We assume that although the altitude information at a certain location are different from our DTM and other DTMs because of the error, the altitude differences (relative terrain) in a small area still have referential meanings. Based on this assumption, we can use other DTMs to help us detecting bridges and tunnels.

(a)

(b)

0.0500	0.0855	0	0	0	0	0
0	0.0741	0	0	0	0	0.0902
0	0.0741	0.0500	0	0	0.0903	0.0708
0	0.0821	0.7392	1	0.1009	0.0794	0.0660
0	0.8462	0.7706	0.6977	0.0768	0.0641	0.0500
0	0.0859	0.7447	0.7321	0.1795	0.0499	0.0500
0.0915	0.0914	0.5655	0.6489	0.1756	0.0500	0.6758

(c)



(d)

Figure 3: A bridge sample with its normalized standard deviation intensity map(a), the real view of bridge(b), intensities around red pixel(c) and altitude histogram of red pixel(d)

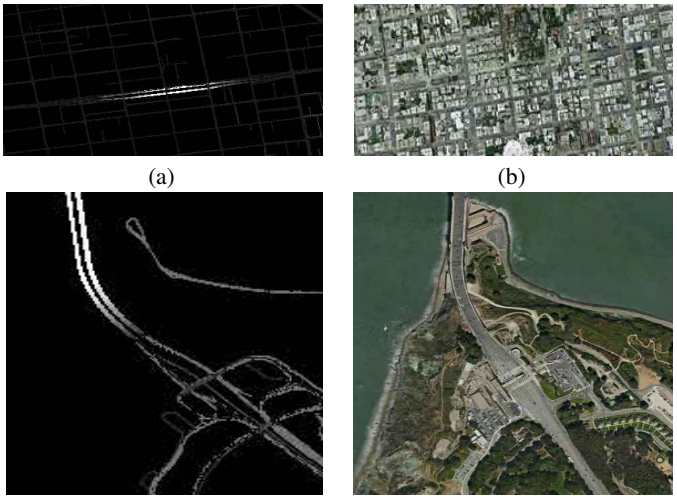


Figure 4: A bridge sample with its normalized standard deviation intensity map(a), the real view of bridge(b), intensities around red pixel(c) and altitude histogram of red pixel(d)

In a small region, the DTMs from two development methods have measurement error, hence the altitude difference of each pixel should almost the same. In Figure 4, we visualize two examples of the normalized altitude differential maps and original satellite images. Giving a region, altitude difference of each pixel should around a certain value, while the difference of tunnel or bridge pixels will show significant contrast. For example, in Figure 4a, we normalize the intensities of a region which contains a tunnel. The majority of altitude differences are round 0.3 meter and follow normal distribution as the most dark gray road pixels show, while the highlighted pixels have more than 30 meters difference. In satellite image Figure 4c, we validate our detection. Figure 4c and d show a bridge example.

When one of these structures are detected, we need to correct the road pixels of this structure. For each pixel which has significant altitude difference compares to the majority of altitude difference in one region, we refresh their value with following equation 3:

$$DTM(x, y) = DTM_r(x, y) - \delta h \tag{3}$$

Where DTM_r is a third party reference model and δh is the majority of altitude difference.

4.2 Block classification and Interpolation

We have two methods to fill up the blank regions in each close road loop (block) which are depended on their types. The first method is called terrain transfer. The idea is copy a local DTM inside the closed loop from USGS or other DTM resources then paste to our DTM with correcting the altitude shift. Altitude shift is the majority of altitude differences in the local region of known pixels. The second one is use bilinear interpolation to fit a plane.

The interpolation method is based on the block classification. For the pixels inside each block boundary, we can fit a plane in 3D space: row, column and altitude. Block can be classified by the standard deviation of the distances to fitted plane of all known pixels in the block from referenced DEM source. If the standard deviation is larger than a threshold, we can assume this region may contain a hill or a basin which will be classified as park block then use copy-paste interpolation. Otherwise, the block will be classified as resident block then use bilinear interpolation. In Figure 5a, we colorize the class of each block, red for resident block, green for park block and white pixels are road.

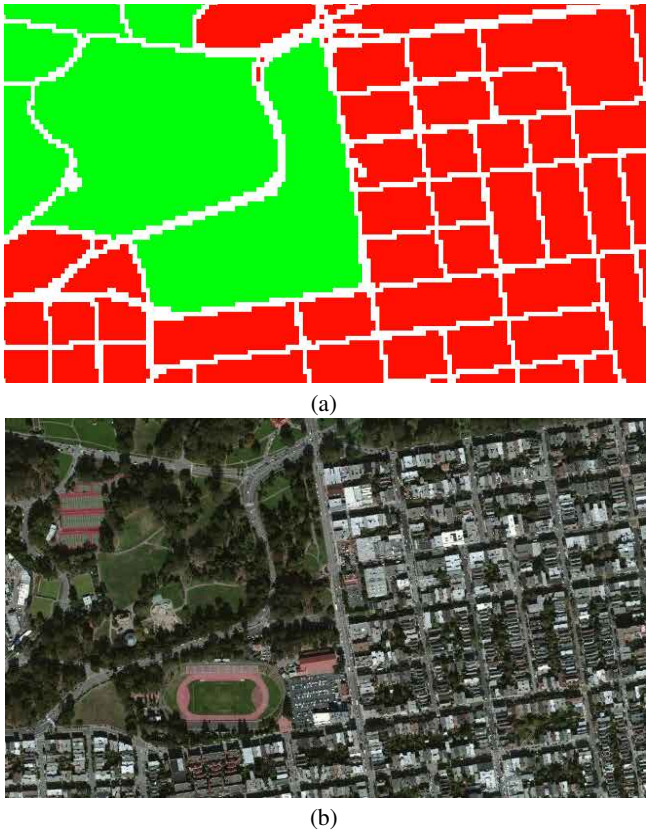


Figure 5: Visualization of the block types of a small region(a) and related aerial image(b).

5. EXPERIMENTAL RESULTS

We tested our algorithm on the data pertaining to San Francisco urban region. The latitude range from south to north is 37.7067789 to 37.811017 and longitude range from west to east is -122.519654 to -122.349709. There are total of 248 drives completed through the said urban region which, in turn, generates 30,000 km of road-segments data in the LiDAR point cloud. To reduce the data size while keeping it consistent with USGS DTM resolution, we extracted and subsampled ground points from the original data by 1 meter per pixel in order to develop a 5 meter per pixel resolution DTM. With this, the number of final ground points became 12,681,310. To cross validate our DEM and USGS DTM, we also imported 102 control points (CP102) from a third party artificial survey. In San Francisco, we detected 6,887 blocks and 6,376 of them are classified as resident blocks. Figure 6 shows the type of each block, while blue region is sea mask, and dark regions are the regions with no closed loop, we will interpolate this kind of region by using copy-paste method. After interpolation, our DTM height map is shown in figure 7a compares to USGS DTM 7b.

Similar to clock problem, although we prove our DEM develop-

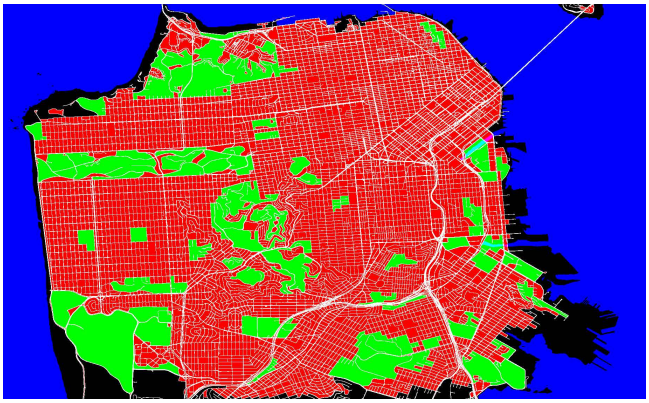


Figure 6: Visualization of the block types of San Francisco: sea(blue), resident(red), park(green) and undetected(black).

ment is close to USGS, we have altitude difference between these two models. So we need "the third clock" to cross validate them. We compare each control point from CP102 with our and USGS DTMs, and visualize the comparisons in figure 8. Green boxes are the point close to CP102 with 0.5 meter difference; red ones mean the altitude at that pixel, ours or USGS altitude is higher than CP102; otherwise we use blue boxes. The table 1 presents the detail altitude information at each control point. From figure 9, we count the percentage of total number of points which below some altitude differences, and prove that our DEM is close to âĂIJabsoluteâĂİ altitude. In statistics, 55% of points in our DTM is closer to CP102 while USGS is 45%, 69% of our points have lower than 1 meter altitude difference while USGS is 64%.

6. CONCLUDING REMARKS AND FUTURE WORKS

In this paper, we proposed a DEM development method using LiDAR point cloud. Evaluation results demonstrated that we achieved better accuracy and faster processing time compared with known classic DEM development approaches. The success of this method is based on two main aspects: firstly, since the vehicle-mount LiDAR is close to the terrain, our method is able to filter out non-ground objects and retain only the ground points; secondly, due to multiple times acquisition, we were able to minimize the measurement errors. At the time being, we are planning to focus our effort on one big challenge that needs to be addressed – namely, we need additional DEM sources to further correct our DEM. At the current settings, our method can develop HD DEM for the cities already available in the HERE point cloud dataset. However, as part of the future extensions of this work, when high precision GPS devices are mounted on most of personal vehicles, we plan to investigate how to efficiently generate HD DTM via coupling LiDAR data with other real-time data sources.

7. REFERENCES

[1] C. W. Bater and N. C. Coops. Evaluating error associated with lidar-derived dem interpolation. *Computers & Geosciences*, 35(2):289–300, 2009.

[2] M. A. Brovelli, M. Cannata, and U. M. Longoni. Lidar data filtering and dtm interpolation within grass. *Transactions in GIS*, 8(2):155–174, 2004.

[3] EUROCONTROL and IfEN. *WGS84 IMPLEMENTATION MANUAL*. 1998.

[4] I. V. Florinsky. *Digital terrain analysis in soil science and geology*. Academic Press, 2012.

[5] K. Kraus and N. Pfeifer. Advanced dtm generation from lidar data. *International Archives Of Photogrammetry Remote Sensing And Spatial Information Sciences*, 34(3/W4):23–30, 2001.

[6] Z. Li, C. Zhu, and C. Gold. *Digital terrain modeling: principles and methodology*. CRC press, 2004.

[7] J. Lovell, D. L. Jupp, D. Culvenor, and N. Coops. Using airborne and ground-based ranging lidar to measure canopy structure in australian forests. *Canadian Journal of Remote Sensing*, 29(5):607–622, 2003.

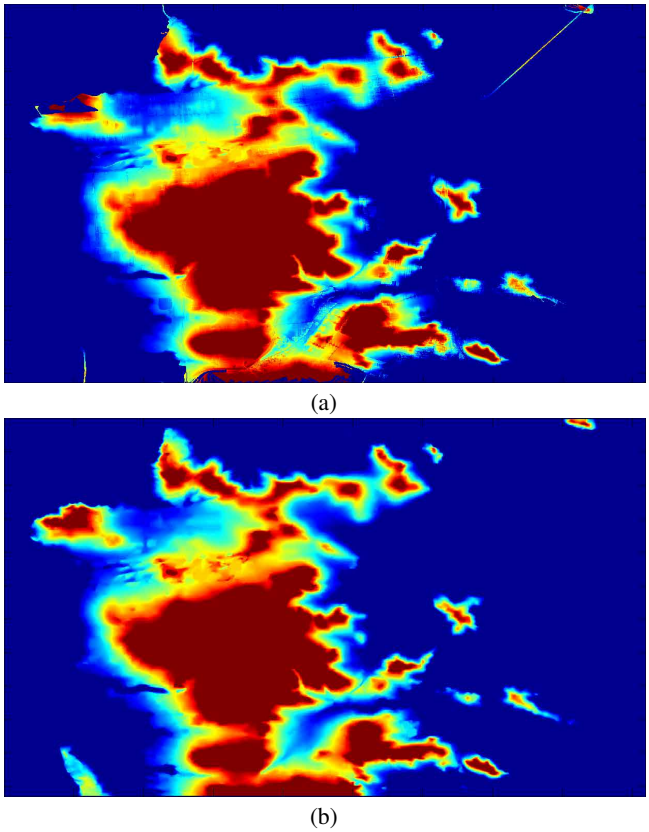
[8] NOAA. Dem data provided by the noaa 2009-2011 ca coastal conservancy coastal lidar project from their web site at http://coast.noaa.gov/.

[9] R. J. Peckham and J. Gyozo. *Digital terrain modelling*. Springer, 2007.

[10] R. L. Perroy, B. Bookhagen, G. P. Asner, and O. A. Chadwick. Comparison of gully erosion estimates using airborne and ground-based lidar on santa cruz island, california. *Geomorphology*, 118(3):288–300, 2010.

[11] S. E. Reutebuch, R. J. McGaughey, H.-E. Andersen, and W. W. Carson. Accuracy of a high-resolution lidar terrain model under a conifer forest canopy. *Canadian Journal of Remote Sensing*, 29(5):527–535, 2003.

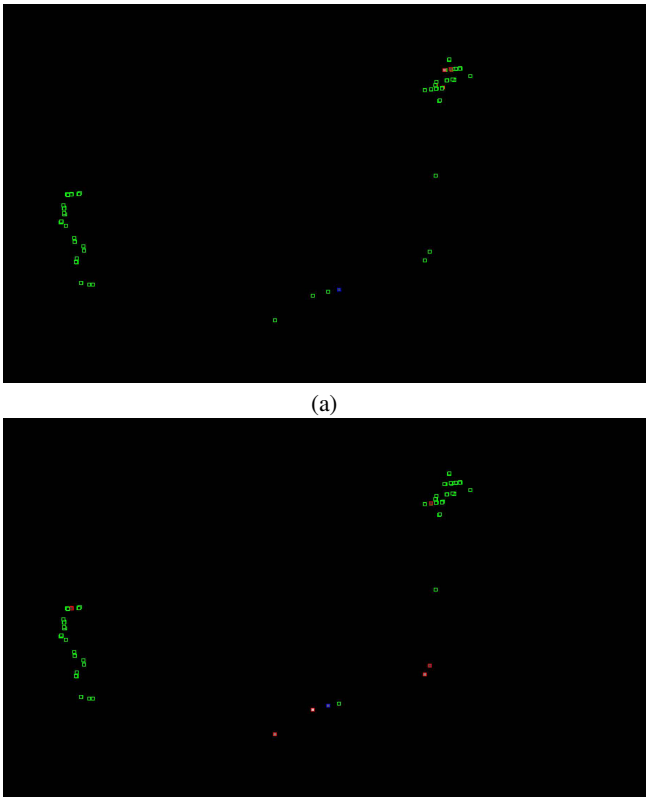
[12] B. Schoettle and M. Sivak. A survey of public opinion about autonomous and self-driving vehicles in the us, the uk, and australia. 2014.



(a)

(b)

Figure 7: Our DTM height map(a) and USGS DTM height map(b).



(a)

(b)

Figure 8: Accuracies of our DTM (a) and USGS DTM (b) compare to CP102: match(green boxes), our/USGS DTM is higher(red boxes) and lower(blue boxes).

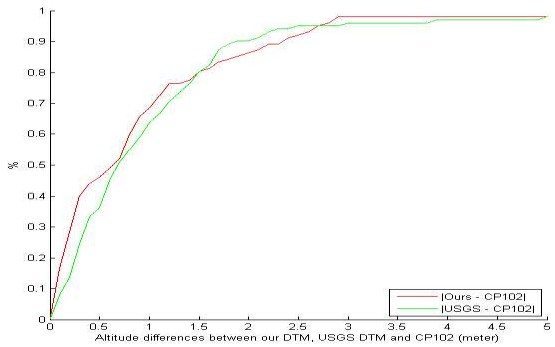


Figure 9: Altitude differences statistics between our DTM(red line)/USGS(green line) and CP102.

[13] M. Sharma, G. B. Paige, and S. N. Miller. Dem development from ground-based lidar data: A method to remove non-surface objects. *Remote Sensing*, 2(11):2629–2642, 2010.

[14] D. Thomas. Driverless convoy: Will truckers lose out to software?. bbc business report, 2015.

[15] USGS. Dem data provided by usgs national elevation dataset from their web site at http://ned.usgs.gov/.

[16] J. Xiong, Y. Fang, B. Jin, and Z. Zhao. Automated dtm generation in urban areas with airborne lidar data. In *Intelligent Human-Machine Systems and Cybernetics (IHMSC), 2012 4th International Conference on*, volume 1, pages 181–184. IEEE, 2012.

X	Y	CP102	Ours	USGS	X	Y	CP102	Ours	USGS
511	1236	-4.373	-4.1085	-5.5767	835	247	-8.911	-9.0733	-10.093
737	1219	-10.513	-10.257	-7.5845	835	255	-6.611	-6.7074	-8.2585
763	1205	-14.572	-14.721	-9.2058	835	257	-5.881	-6.1521	-7.6087
850	960	12.341	9.659	11.603	830	224	-13.22	-13.257	-14.847
856	929	22.322	22.086	18.486	830	223	-13.172	-13.35	-14.861
868	885	19.145	18.915	27.163	191	1306	-29.607	-27.574	-29.437
941	777	33.841	33.714	38.788	193	1307	-29.354	-28.527	-29.437
607	176	-11.683	-11.984	-12.36	193	1305	-29.307	-27.68	-29.374
567	195	-6.031	-6.6798	-3.869	194	1294	-28.855	-27.873	-28.831
565	196	-5.935	-6.831	-3.6631	194	1294	-29.04	-27.873	-28.831
566	197	-5.936	-6.0948	-3.4722	193	1294	-29.075	-27.608	-28.828
567	196	-5.953	-6.8033	-3.8089	193	1294	-29.159	-27.608	-28.828
566	216	3.965	3.5429	3.1162	194	1286	-28.492	-26.113	-28.281
565	218	4.882	4.8025	4.1296	196	1281	-28.335	-25.618	-27.913
564	217	4.976	4.2134	3.622	196	1281	-28.171	-25.618	-27.913
565	218	5.035	4.8025	4.1296	195	1281	-28.227	-25.893	-27.915
563	220	5.084	5.1412	5.1447	194	1280	-28.143	-26.726	-27.911
566	218	5.013	4.505	4.132	194	1279	-28.224	-25.386	-27.92
567	183	-6.817	-7.0624	-7.7012	197	1264	-27.359	-25.23	-26.397
566	184	-6.857	-6.8564	-7.4463	197	1262	-27.418	-25.444	-25.973
567	185	-6.872	-6.9713	-7.2187	197	1261	-27.101	-21.057	-25.729
569	186	-5.657	-7.5192	-6.7848	227	1269	-22.25	-22.452	-21.975
568	186	-6.843	-7.5737	-6.9691	228	1267	-22.04	-20.841	-21.488
606	175	-11.923	-11.953	-12.503	228	1267	-21.99	-20.841	-21.488
608	174	-12.417	-12.783	-12.959	228	1268	-22.249	-21.368	-21.749
609	175	-12.009	-12.14	-12.709	225	1288	-26.075	-23.641	-25.483
599	173	-10.959	-11.674	-12.618	226	1289	-25.979	-25.269	-25.577
626	178	-13.706	-14.142	-13.449	227	1289	-25.736	-25	-25.426
626	175	-14.164	-14.329	-14.517	225	1284	-25.61	-24.914	-24.658
622	175	-12.833	-13.64	-14.164	232	1238	-14.967	-13.482	-13.48
650	165	-20.092	-20.482	-20.343	243	1237	-15.971	-13.072	-14.765
650	167	-19.859	-20.569	-19.6	243	1235	-15.745	-13.072	-13.963
648	168	-19.355	-19.411	-19.292	242	1235	-15.758	-13.644	-13.892
647	168	-19.239	-19.394	-19.1	252	1239	-17.376	-16.778	-17.036
647	167	-19.824	-19.838	-19.462	252	1237	-17.152	-16.594	-16.485
660	180	-16.229	-16.148	-16.271	253	1237	-17.204	-16.11	-16.503
707	205	-5.618	-5.4471	-7.0605	249	1257	-19.232	-10.682	-19.505
707	206	-5.351	-5.4422	-6.1485	250	1254	-19.053	-18.177	-19.045
708	206	-5.31	-6.2701	-5.9686	252	1254	-19.097	-19.041	-19.285
708	205	-5.704	-5.7565	-6.7734	255	1223	-14.451	-13.403	-12.936
696	204	-6.546	-6.8003	-7.2256	254	1223	-14.407	-14.174	-12.381
721	230	6.97	6.9197	7.0414	256	1205	-11.914	-10.764	-10.81
733	232	7.231	6.2772	8.8334	257	1205	-11.961	-11.28	-11.122
734	232	7.044	5.9996	6.3618	257	1205	-12.051	-11.28	-11.122
757	211	-1.954	-2.0649	-0.93676	289	1246	-22.36	-22.531	-21.825
769	211	-0.453	-0.39786	-1.5466	287	1248	-22.407	-21.611	-21.823
767	211	-0.442	-0.29495	-1.3932	215	1335	-29.244	-28.189	-29.436
767	209	-0.364	-0.64899	-1.8803	167	1275	-28.804	-27.088	-28.502
768	209	-0.396	-0.34413	-1.8717	167	1274	-28.447	-27.129	-28.285
835	250	-8.115	-8.1603	-9.2252	167	1274	-28.748	-27.129	-28.285
835	247	-8.867	-9.0733	-10.093	165	1275	-28.742	-28.429	-28.502

Table 1: Comparison table of CP102, our DEM and USGS DEM at certain DEM location (X, Y).