Taylor & Francis
Taylor & Francis Group

Remote Sensing Letters

ISSN: 2150-704X (Print) 2150-7058 (Online) Journal homepage: https://www.tandfonline.com/loi/trsl20

A new method for shoreline extraction from airborne LiDAR point clouds

Sheng Xu, Ning Ye & Shanshan Xu

REMOTE SENSING LETTERS
2019, VOL. 10, NO. 5, 496–505
https://doi.org/10.1080/2150704X.2019.1569277

Taylor & Francis
Taylor & Francis Group

Check for updates

A new method for shoreline extraction from airborne LiDAR point clouds

Sheng Xu, Ning Ye and Shanshan Xu

College of Information Science and Technology, Nanjing Forestry University, Nanjing, China

ABSTRACT
This work proposes a method for the extraction of shorelines from airborne LiDAR (light detection and ranging) point clouds. In the beginning, water bodies are removed based on the flatness clue. Then, boundaries of lands are extracted by using a new minimum-cost boundary model. Finally, false boundaries caused by man-made objects and vegetations are removed in the refinement step, and true boundaries are regarded as shorelines. The main contribution is that the cost of boundaries is calculated through an energy function and minimized by the proposed minimum-cost model globally. Evaluation on five experimental scenes shows that the proposed method achieves the completeness of 92.5% and correctness of 90.7%, which are promising results in the shoreline extraction.

ARTICLE HISTORY
Received 8 August 2018
Accepted 22 December 2018

1. Introduction

Shoreline is the line of contact between lands and water bodies (Boak and Turner 2005), which plays an important role in the coastline management and engineering design (Feng et al. 2012; Choung 2009).

Nowadays, many approaches have been proposed for the shoreline extraction. Niedermeier, Romaneessen, and Lehner (2000) extract the shoreline information from synthetic aperture radar (SAR) images using a four-step method, including detecting all edges above a threshold, determining the boundary area between lands and water bodies, selecting local edges in the coastal area and joining the refined edge segments. Their accuracy is claimed to be sufficient to monitor and update the topography of large active areas. Di et al. (2003) investigate an approach for the shoreline extraction from high resolution satellite imageries using three steps: (1) segment homogeneous regions from images; (2) generate an initial shoreline based on their identified water body and (3) obtain final shorelines by a local refinement method. The work is capable of extracting shorelines from high-resolution satellite imageries with little human iteration.

Recently, LiDAR (light detection and ranging) point clouds are becoming a significant technique in 3D information extraction, and LiDAR data provide a new solution for the shoreline extraction. Liu, Sherman, and Gu (2007) segment a LiDAR digital elevation

CONTACT Shanshan Xu shanshanxus@gmail.com College of Information Science and Technology, Nanjing Forestry University, Nanjing, China

model (DEM) into a binary image, and then they use a chain of image processing algorithms to extract shorelines from airborne LiDAR data.

They derive spatially detailed shorelines from point clouds with minimal human intervention. Lee, Wu, and Li (2010) investigate shoreline extraction using the integration of LiDAR data and satellite imageries. They classify LiDAR data into water body points and land points by the means-shift algorithm, and trace boundaries between water bodies and lands to refine shorelines.

Although the above-mentioned methods achieve the shoreline extraction effectively, they conduct the extraction process in 2D space, which may lose 3D information and degrade the extraction accuracy. This work aims to propose an approach to extract shorelines from airborne LiDAR point clouds in 3D space without any imagery information and human interaction. In the beginning, water bodies are removed based on the flatness clue and disjointed lands are clustered into different groups. Then, boundaries of lands are optimized by the proposed minimum-cost model. Finally, false boundaries are eliminated from results and true boundaries are regarded as shorelines.

2. Water body removal

In most cases, the beam return from water regions is very minimal, because the laser energy is absorbed by water. To identify LiDAR returns associated with water bodies, we cluster points based on Euclidean distance. If the number of points in a cluster is less than a threshold n_p, this cluster will be removed. In the case of muddy and shallow water bodies, one can collect lots of beam returns. The water body detection is based on the work of Xu and Xu (2018). They provide a water point removal method based on the combination of the plane fitting and features filtering. The detection is based on the assumption that water bodies are presented as large horizontal planes in point clouds. False water body points are removed based on the elevation and density information. If the elevation of a point above the nearest extracted plane is larger than a threshold T_e, it will be marked as a non-water point. If the density of points from the extracted plane is larger than a threshold T_D, this region will be marked as the non-water body. Values of the above-mentioned n_p, T_e and T_D are set by users.

The obtained smallest water body depends on the value of n_p and its density. In our work, water bodies larger than 500 m $\times$ 500 m can be extracted effectively. After removing extracted water bodies, we use Euclidean clustering method (Rusu 2010) to cluster disjointed lands into different groups. The following sections will show how to obtain accurate boundaries of lands optimally.

3. Boundary optimization

There are three steps in the proposed boundary extraction. First, a testing method is used to detect candidate boundary points from input data. Second, the calculation of the boundary cost will be formulated by an energy function. Third, a new minimum-cost boundary model is used to optimize boundaries based on their cost.

In a convex hull, a point p fails to be a boundary point of a convex set S, if it lies in a triangle whose vertices are in S (Wang and Shan 2009). Therefore, boundary points can

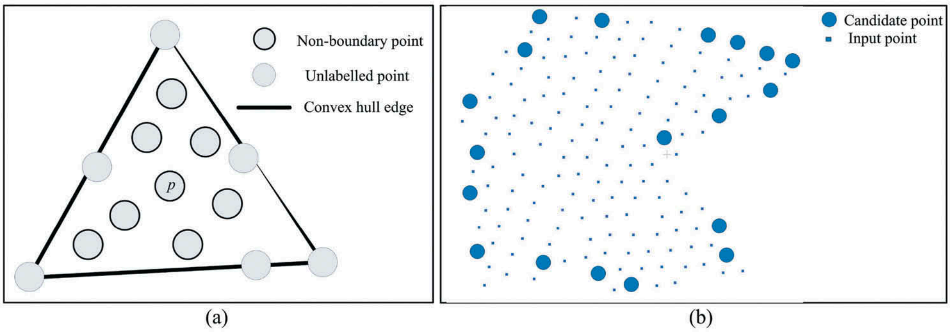

Figure 1. Candidate boundary point extraction. (a) Illustration of the testing algorithm. (b) Extracted candidate boundary points.

be achieved by removing non-boundary points. The idea of the testing algorithm is as follows.

In the initialization, all points are regarded as unlabelled points. As shown in Figure 1(a), if a point p is an unlabelled point, pick up its k-nearest neighbourhood points and construct a convex hull of (p, p_i), i.e., the hull consists of a set of points labelled p_i, $i = 1, 2, \ldots, k$. Then, label all points inside this convex hull as non-boundary points, and repeat the testing until no non-boundary points can be found. Finally, all unlabelled points are regarded as candidate boundary points. Figure 1(b) shows an example of the testing algorithm on a simple point cloud set.

There are three things worth noting. First, there may be missing points in the extraction of candidate points. This depends on the number of neighbouring points k. A small k incurs more complete boundary points but requires more time in the candidate point extraction. Second, if a candidate point is far from all other candidate points, i.e., larger than a user-defined threshold T_d, this candidate point is regarded as an error one and will be removed. Third, due to the fact that point clouds are uneven and unorganized, it is difficult to extract all boundary points of lands, and candidate points can generate different boundaries. Assume that there are five candidate boundary

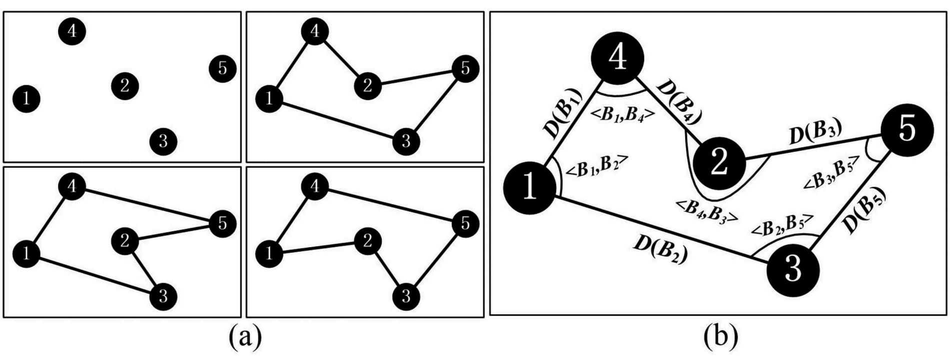

Figure 2. Cost calculation of different boundaries. (a) Different boundaries generated by candidate points (index: 1, 2, 3, 4, 5). (b) Weights of connections in the boundary optimization.

points as shown in the left corner of Figure 2(a), one can generate different boundaries as shown in the rest of Figure 2(a).

From Ockham's razor (Jefferys and Berger 1992): the law of parsimony, a simple boundary may be preferable in describing an object. To evaluate a boundary, we define the boundary value β as

$$\beta = \sum_{i=1}^{n} \left(D(B_i) + \sum_{\{B_i, B_j\} \in \mathrm{N}} \lambda \times \left| \cos(\frac{\langle B_i, B_j \rangle}{2}) \right| \right), \tag{1}$$

where n means the number of connections in the boundary, λ is a weight coefficient, $D(B_i)$ is the length of the connection B_i, N means the set of adjacent connections, and $\langle B_i, B_j \rangle$ is the angle between two connections. In the boundary optimization, boundary points are expected to be close to each other, i.e., $D(B_i)$ is small, and the angle of neighbouring connections is desired to be large, i.e., $\left|\cos(\frac{\langle B_i, B_j \rangle}{2})\right|$ is close to 0. For example, Figure 2(e) displays weights of all connections and the boundary value is calculated as

$$\begin{aligned} \beta = {} & D(B_1) + D(B_2) + D(B_3) + D(B_4) + D(B_5) + \lambda \times \left(\left| \cos\frac{\langle B_1, B_2 \rangle}{2} \right| \right. \\ & + \left| \cos\frac{\langle B_1, B_4 \rangle}{2} \right| + \left| \cos\frac{\langle B_4, B_3 \rangle}{2} \right| \\ & \left. + \left| \cos\frac{\langle B_3, B_5 \rangle}{2} \right| + \left| \cos\frac{\langle B_2, B_5 \rangle}{2} \right| \right). \end{aligned} \tag{2}$$

To obtain the boundary with the minimal value β, we propose a minimum-cost model to optimize Equation (1).

Before proceeding the proposed model, we need to define the node of a boundary model, which is the combination of any three points as {LMR}, i.e., the left, middle and right candidate point. Take Figure 2(a) as an example, nodes are formulated in the graph as shown in Figure 3. Each candidate point can be assigned as a middle point, therefore, nodes are in five columns. The node {*I JK*} is equal to the node {*K JI*}, where *I*,*J*,*K* are the index of candidate points, therefore, nodes are in five rows. Each black circle means a candidate point, and the combination of three circles means a node. Each blue line means a connection between nodes. If we add Source and Sink nodes in the formulated graph, each path from Source to Sink indicates a boundary for input candidate points. For example, the right corner of Figure 2 (a) is Source-{314}-{142}-{425}-{253}-{531}-Sink. Our task is to choose the minimum-cost path as the optimal boundary.

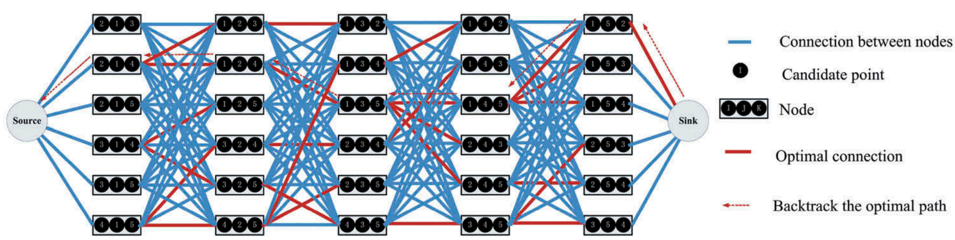

Figure 3. Graph used in the boundary optimization.

We weight each node {LMR} in Figure 3 as

$$\mathrm{cost}(\mathrm{L},\mathrm{M},\mathrm{R}) = \frac{\mathrm{dis}(\mathrm{L},\mathrm{M}) + \mathrm{dis}(\mathrm{M},\mathrm{R})}{2} + \lambda \times \left|\cos\left(\frac{\theta_{\mathrm{LMR}}}{2}\right)\right|, \tag{3}$$

where dis(L,M) is Euclidean distances of two candidate points and θ_{LMR} is the angle between the connection 'LM' and 'MR'. We have

$$\sum_{\{I,J,K\}\in \mathrm{P}} \frac{\mathrm{dis}(I,J) + \mathrm{dis}(J,K)}{2} = \sum_i D(B_i),$$

$$\sum_{\{I,J,K\}\in \mathrm{P}} \left|\cos\left(\frac{\theta_{\mathrm{LMR}}}{2}\right)\right| = \sum_{\{B_i,B_j\}\in \mathrm{N}} \left|\cos\frac{\langle B_i,B_j\rangle}{2}\right|, \tag{4}$$

where P is the node set in the graph. Therefore, the cost of a path from Source to Sink is equal to the boundary value in Equation (1). The path with the minimum-cost will be chosen as the optimal boundary of input candidate points.

The formulated optimization can be regarded as the shortest path problem, which can be efficiently solved by the dynamic programming approach (Eppstein 1998). The idea is to break the complex optimization into simpler sub-problems. Assume that we have found the optimal path from Source to each node. When adding a new node, the optimal path from Source to this new node should contain the optimal path from Source to its prior node. In Figure 3, red lines mean the optimal path from the prior node and the current node. The red arrow shows the obtained minimum-cost path backtracked from Sink to Source. The refinement step is to remove false boundaries caused by man-made objects and vegetations. If the Euclidean distance between a boundary point and the nearest water body is larger than a user-defined threshold T_{b}, this point will be removed from boundaries.

The shoreline extraction is not a well-addressed issue, because of many complicated factors, e.g., wave, erosion and sediment. One advantage of the proposed model is that the boundary of the shoreline can be modified by adding constraints in the energy function. We can balance terms by tuning coefficients or adding new terms based on the prior knowledge for different cases. For example, we can add the curvature information to smooth the boundary in erosion regions. Another advantage is that we extract shorelines in 3D space without the projection process, which may lose 3D geometric information.

4. Results and evaluation

This section shows performances of the proposed method on five experimental scenes located in US, including Estuary (Oregon), Wax Lake (New Orleans), Bowman Lake (California), Susquehanna River (Pennsylvania) and Canyon Stream (Washington). The 2D images of experimental scenes are shown in Figure 4. Description of input datasets is shown in Table 1.

Extraction results from ALS point clouds are visualized in Figure 5. The first experimental scene is Estuary. The challenge in the sea estuary is that boundaries of offshore and coastal wetlands are difficult to be marked from point clouds. As

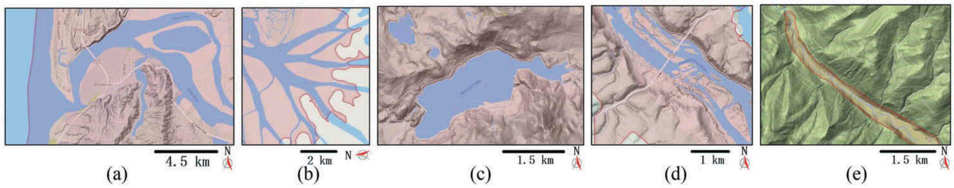

Figure 4. 2D images of experimental scenes from Google map. Input ALS point clouds for experiments are collected from the corresponding shaded areas in each scene. (a) Estuary. (b) Wax lake. (c) Bowman lake. (d) Susquehanna river. (e) Canyon stream.

Table 1. Description of experimental scenes.

Dataset	No. points ($\times 10^6$)	Area (km)	Density (points/m^2)	Survery Date	Location
Estuary	182	16×12	10.30	March 2007	Oregon, US (44°25 N, 124°04 W)
Wax Lake	232	5×6	14.27	February 2013	New Orleans, US (29°31 N, 91°26 W)
Bowman Lake	14	5×3	8.93	June 2014	California, US (39°27 N, 120°38 W)
Susquehanna River	24	3×3	1.37	January 2005	Pennsylvania, US (39°49 N, 76°19 W)
Canyon Stream	54	6×5	9.63	October 2006	Washington, US (48°00 N, 120°36 W)

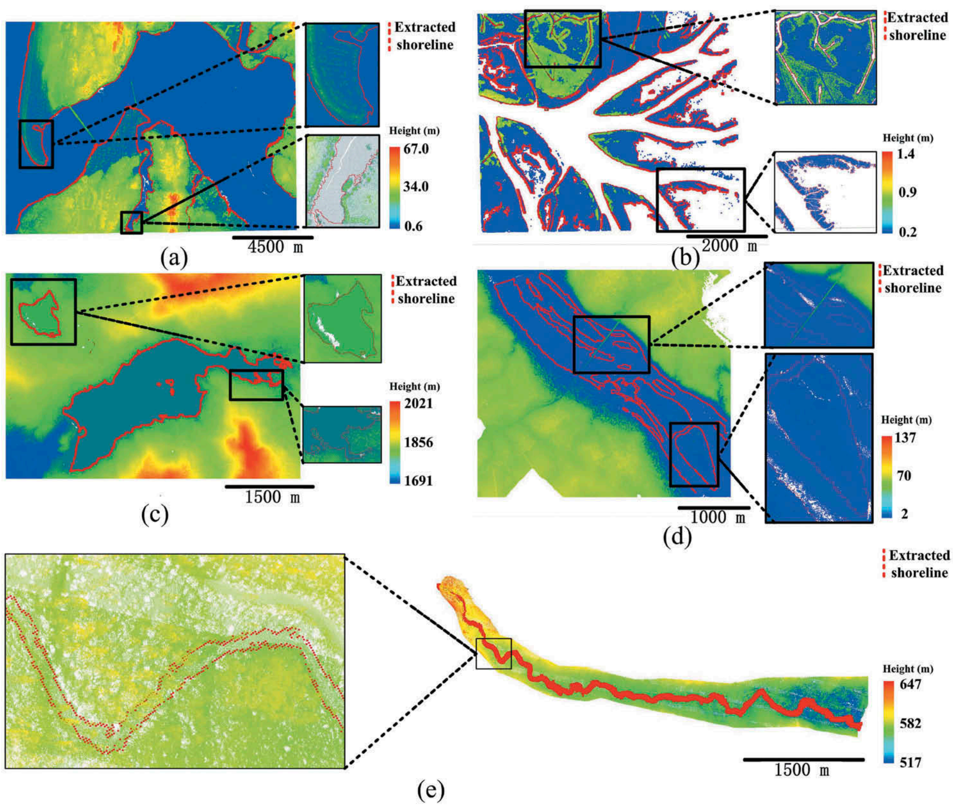

Figure 5. Visualization of shoreline extraction results on ALS point clouds. (a) Estuary. (b) Wax Lake. (c) Bowman Lake. (d) Susquehanna River. (e) Canyon Stream.

shown in Figure 5(a), the proposed method succeeds in obtaining the boundary of this estuary accurately, including the offshore area and the main tributary of the sea. The second experimental scene is Wax Lake (https://doi.org/10.5069/G9SF2T41). Although there is less elevation difference (around 1.2 m), we extract thin tributaries effectively as shown in Figure 5(b). The third experimental scene is Bowman Lake (https://doi.org/10.5069/G9V122Q1). This scene shows that the proposed method works well on the scene with a large elevation difference (over 300 m) and can achieve boundaries of small islands in the middle of the lake as shown in Figure 5(c). The fourth experimental scene is Susquehanna River (https://doi.org/10.5069/G9RV0KMG). This scene contains a large number of sandbars, which are detected by the proposed method successfully as shown in Figure 5(d). The last experimental scene is Canyon Stream (https://doi.org/10.5069/G9JM27JX). This one is the most difficult scene. It is even hardly to extract the ground-truth manually. Our performance on the last scene is shown in Figure 5(e).

The evaluation is based on the point level. There are three status of extraction results for a point, namely true positive (TP), false negative (FN) and false positive (FP). TP means the distance between the detected boundary point and the true boundary in the reference is less than 0.5 m. FN represents the distance between the detected non-boundary point and the true boundary in the reference is less than 0.5 m. FP represents the distance between the detected boundary point and the true boundary in the reference is larger than 1 m. Currently, we do not have survey data measured by humans for the evaluation. The reference data used in the evaluation is manually obtained through the point cloud visualization tool, e.g., CloudCompare (www.danielgm.net/cc/) in our work. We manually segment shorelines as the ground-truth, and then accumulate the number of TP, FN and FP points to calculate completeness and correctness in Equation (5) for the point-based evaluation.

$$\text{Completeness} = \frac{|\text{TP}|}{|\text{TP}|+|\text{FN}|}, \text{Correctness} = \frac{|\text{TP}|}{|\text{TP}|+|\text{FP}|}, \tag{5}$$

where $|\text{TP}|$, $|\text{FN}|$ and $|\text{FP}|$ are the number of TP points, FN points and FP points, respectively. The completeness measures the probability of ground-truth boundary points that can be extracted, and correctness measures the probability of extracted points that belong to ground-truth boundaries. The average completeness is over 92.5% and correctness is over 90.7% through five experimental scenes.

For the comparison, Niedermeier, Romaneessen, and Lehner (2000), Liu, Sherman, and Gu (2007) and Lee, Wu, and Li (2010) did not conduct the point-based evaluation. Therefore, we can only show the comparison of accuracy level in Table 2, e.g., 10-meter or 1-meter level, obtained by different methods in their experimental scenes. The column 'Data' shows the input data, i.e., satellite images or ALS point clouds. The column 'Resolution' shows average Euclidean distances between pixels or points. The column 'Level' shows the accuracy level and the last column 'No. scenes' shows the number of experimental scenes. Image processing methods are proposed for 2D space and cannot be used in unorganized ALS point clouds directly. Their accuracy level on images is over 10-meter level. Besides, image processing methods cannot address occlusion from bridges and trees. As shown in Table 2, the proposed method extracts shorelines in a high accuracy level, which indicates that airborne LiDAR data are very promising information in the shoreline extraction.

Table 2. Comparison of different shoreline extraction methods.

Methods	Data	Resolution	Level (m)	No. scenes
Niedermeier, Romaneessen, and Lehner (2000)	Image	12.5 m/pixel	31.0	1
Di et al. (2003)	Image	4.0 m/pixel	8.5	4
Liu, Sherman, and Gu (2007)	Point clouds	1.0 m/point	4.5	1
Lee, Wu, and Li (2010)	Point clouds	2.0 m/point	1.5	4
Proposed	Point clouds	1.0 m/point	1.0	5

5. Discussion

One shortcoming of the proposed method is the parameter setting in the processing as indicated in Table 3.

In the water body removal, n_p is the minimum number of points in the clustering. We remove clusters of fewer points as water bodies. A large n_p will remove more plane regions and lose small water bodies. T_e is the minimum distance between a non-water point and water bodies, and a large value will lose more non-water points. T_D is the minimum density of the non-water body. In ALS point clouds, points from water bodies are more sparse than lands, because of the quite different absorption rate. A large density threshold will remove false water bodies.

In the boundary optimization, k is the number of neighbouring points. A large value will obtain less candidate boundary points and a small value will decrease the accuracy of the candidate point extraction, because the convex hull requires points in the formulation for the testing process. T_d is the minimum distance between candidate points, and a large value will incur error candidate points. λ is the coefficient in the boundary value calculation. A small λ works well when there are plenty of candidate boundary points. A large λ is suggested when there are fewer boundary points and the boundary is desired to be smooth. T_b is the maximum distance between the boundary point and the nearest water body. A large value will bring error points in boundaries, and a small value will fail in the shoreline extraction when water bodies with high elevation difference.

We test all parameters in the range in Table 3 and use the suggested value to achieve the above-mentioned promising results. The proposed method is tested on different water bodies with different airborne laser sensors. Results of the shoreline extraction in the case of many mouths are shown in Figure 5(b). In this case, we assign each candidate point with an index and calculate the distance between candidate points to obtain dis(L,M) and dis(M,R). Results of the shoreline extraction in the case of different point distribution are shown in Figure 5(d). In this case, points around islands are dense and points in the middle of water bodies are sparse.

Table 3. Setting of parameters in the proposed method.

Section	Parameter	Range	Suggested Value	Unit
Water body removal	n_p	500–10,000	500	point
	T_e	0.5–5	2	m
	T_D	0.1–0.5	0.3	points/m^2
Boundary optimization	k	20–100	50	point
	T_d	0.5–3	2	m
	λ	1–10	3	N/A
	T_b	0.1–1	0.5	m

In terms of the execution time, the average cost time of the boundary optimization across sites (a)-(e) is 81.30 s. Experiments were done on a Windows 10 Home 64-bit, Intel Core i5-7200U 2.5 GHz processor with 16 GB of RAM and computations were carried on Matlab R2018a.

6. Conclusion

This paper proposes a method for the shoreline extraction from airborne LiDAR data, including (1) removing water bodies and segmenting disjoint land areas, (2) using a minimum-cost boundary model to optimally extract boundaries of lands and (3) refining boundaries as shorelines. The cost of boundaries is formulated by an energy function and minimized by the dynamic programming approach. The proposed extraction only uses airborne LiDAR point clouds and is conducted without any human interaction. Experiments on five typical scenes show that the achieved completeness is over 92.5% and correctness is over 90.7% which is competitive with other existing methods.

Funding

This work was supported by National Key R&D Program of China (2017YFD06009), National Natural Science Foundation of China (31770591, 41701510), China Postdoctoral Science Foundation (2016M601823).

ORCID

Sheng Xu http://orcid.org/0000-0002-9017-1510

References

Boak, E. H., and I. L. Turner. 2005. "Shoreline Definition and Detection: A Review." *Journal of Coastal Research* 214: 688–703. doi:10.2112/03-0071.1.

Choung, Y.2009. *Extraction of Blufflines from 2.5 Dimensional Delaunay Triangle Mesh Using LiDAR Data*. Columbus: The Ohio State University.

Di, K., J. Wang, R. Ma, and R. Li, 2003. "Automatic Shoreline Extraction from High-Resolution IKONOS Satellite Imagery". In *Proceeding of ASPRS 2003 Annual Conference, Anchorage, USA, ASPRS*.

Eppstein, D. 1998. "Finding the K Shortest Paths." *SIAM Journal on Computing* 28 (2): 652–673. doi:10.1137/S0097539795290477.

Feng, L., C. Hu, X. Chen, X. Cai, L. Tian, and W. Gan. 2012. "Assessment of Inundation Changes of Poyang Lake Using MODIS Observations between 2000 and 2010." *Remote Sensing of Environment* 121: 80–92. doi:10.1016/j.rse.2012.01.014.

Jefferys, W. H., and J. O. Berger. 1992. "Ockham's Razor and Bayesian Analysis." *American Scientist* 80 (1): 64–72.

Lee, I. C., B. Wu, and R. Li, 2010, March. "Optimal Parameter Determination for Mean-Shift Segmentation-Based Shoreline Extraction Using Lidar Data, Aerial Orthophotos, and Satellite Imagery." In *Annual Conference Baltimore*, Maryland, ASPRS.

Liu, H., D. Sherman, and S. Gu. 2007. "Automated Extraction of Shorelines from Airborne Light Detection and Ranging Data and Accuracy Assessment Based on Monte Carlo Simulation." *Journal of Coastal Research* 236: 1359–1369. doi:10.2112/05-0580.1.

Niedermeier, A., E. Romaneessen, and S. Lehner. 2000. “Detection of Coastlines in SAR Images Using Wavelet Methods.” *IEEE Transactions on Geoscience and Remote Sensing* 38 (5): 2270–2281. doi:10.1109/36.868884.

Rusu, R. B. 2010. “Semantic 3d Object Maps for Everyday Manipulation in Human Living Environments.” *KI-Künstliche Intelligenz* 24 (4): 345–348. doi:10.1007/s13218-010-0059-6.

Wang, J., and J. Shan, 2009, March. “Segmentation of LiDAR Point Clouds for Building Extraction.” *American Society for Photogrammer Remote Sensing Annual Conference*.35: 9–13.doi:10.1177/1753193409347428

Xu, S., and S. Xu. 2018. “A Minimum-Cost Path Model to the Bridge Extraction from Airborne LiDAR Point Clouds.” *Journal of the Indian Society of Remote Sensing* 46: 1423–1431. https://doi.org/10.1007/s12524-018-0788-9.