A NOVEL ANGULAR FILTER BASED LIDAR POINT CLOUD CLASSIFICATION

Partha P. Acharjee[1], George J. Toscano[1] and Venkat Devarajan[1]
[1]University of Texas at Arlington, Dept. of Electrical Engineering, Arlington, TX76019

ABSTRACT

We consider the problem of ground filtering of LiDAR point cloud by utilizing geometric properties of the scene area. We propose a new approach for detecting man-made object edges from the elevation profile using a novel angle filtering method. This method analyzes neighbors from two nearby tiers, which alleviates the need for multiple gradient calculations from different directions. A subsequent connected component and convex–hull analysis separate all planar surfaces from the detected edges. These separated planar surfaces provide information about objects' geometry. All objects are separately analyzed to reduce the error around the border region, which is prominent in several existing ground filtering algorithms. Experimental results are shown for a complex urban scene, where complicated building structure is present.

Index Terms— LiDAR, Bare earth model, Angular filtering, Point cloud classification

1. INTRODUCTION

An airborne LiDAR system acquires 3–D point clouds which also have registered intensity information of the earth. Better vertical accuracy and remote data acquisition framework of a LiDAR system gives it the edge over traditional mapping and survey system [1]. LiDAR data provides cost effective, less time consuming and more accurate vertical height and reflectivity information of the targeted surface. On the other hand, processing of huge 3–D point cloud demands higher computational time and more human interaction. In the process of developing any LiDAR product, manual classification and quality control consume 60-80% of the processing time, which motivates researchers to develop automated classification methods [2].

Bare earth model generation from LiDAR data is important to produce any LiDAR derivative product. Due to high application value of this problem, the model generation needs to be precise and efficient [3]. Over time, different filters have been introduced to extract bare-Earth points from LiDAR point clouds. ISPRS Working Group III/3 reported performance of twelve different bare-earth filtering methods and identified potential improvement scope, and direction for future research [4]. None of these reported methods treated individual objects separately. In the proposed method, all objects were separated and analyzed to deal with different types of special situations.

2. METHOD

In this paper, we utilize the geometric properties of planar surfaces to classify ground and non-ground areas. We are attempting to develop a framework to integrate all available LiDAR information to classify different objects. For this purpose, we develop an angular filtering method to detect edges from local neighborhood analysis. This proposed angle filtering approach detects edges, based on the surrounding neighbors from two adjacent tiers, which alleviates the need for single and multi-directional gradient calculations previously introduced in several ground filtering algorithms [5]. From the detected edges, subsequent object based classification is performed based on connected component and convex-hull analysis.

2.1. LiDAR acquisition and preprocessing

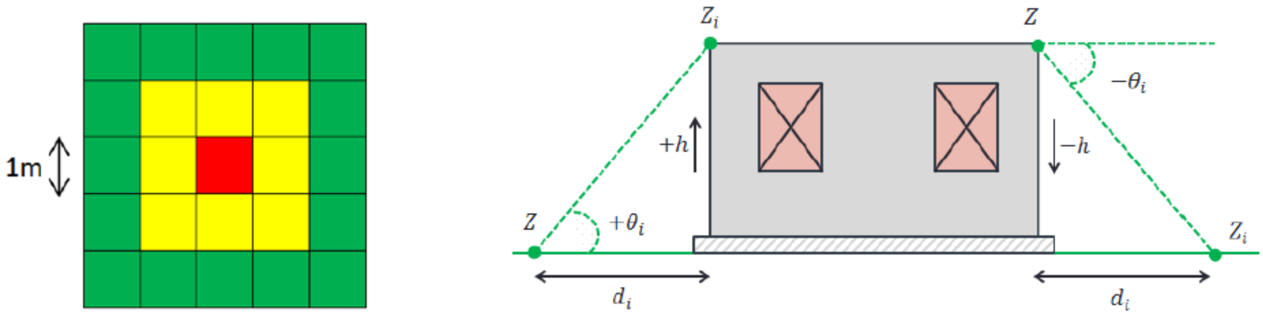
In this work, an urban dataset was used to show experimental results. An urban location consisting of complex building structures near Indiana University-Purdue University Indianapolis (IUPUI) campus was selected. This came from 2011–2013 Indiana Statewide LiDAR dataset, which is available online on open–topography website for public download. Point density for this dataset is 1.56 pts./m^2. WGS–84 and NAVD–88 coordinate systems are used for horizontal and vertical coordinate respectively. The preprocessing (Fig.03), as well as the algorithm development, was performed in MATLAB. For visualization, ArcGIS was used with Q-Coherent and other plug-ins. To rasterize the point cloud, $2m \times 2m$ area was selected as block size. Block elevations were calculated using inverse weighted distance (IDW) interpolation method as given in equation 1. Number of returns and return count were also stored for each raster.

$$Z_b = \frac{\sum_{n=0}^{N} Z_n/d_n}{\sum_{n=0}^{N} 1/d_n} \qquad (1)$$

where Z_b is the elevation of the block, Z_n is the elevation of n-th point and d_n is the distance of the n–th point from the center of the corresponding block. IDW was also used to interpolate the elevation data of those blocks, which had no returns e.g. water–body, data–gap, etc. In those cases, Z_n and d_n were obtained from eight nearest neighbors and Z_b represents the elevation of the no–data block. In short, pre–processing steps provided a raster consisting of elevation, intensity, return count and return number from the point cloud.

2.2. Angular filter

Rapid change of elevation profile is a key feature man-made object. It can be safely assumed that the bare earth elevation change is less abrupt than that for man-made objects in most cases. Thus, detecting elevation gradient is the preliminary step. For this purpose, an angular filter based elevation gradient detection method is proposed. Neighborhood elevation from all sides for a given LiDAR pixel (target block) was used as shown in Fig.01(a). A total of 24 neighboring pixels were selected from two adjacent rows and columns to accommodate two tiers of pixel neighbors. Two rows/columns were considered to increase the probability of capturing a possible ground/non-ground edge. In Fig.01(a), angle values were calculated for the red block; yellow blocks are first tier neighbors and green blocks are second tier neighbors.



(a) Neighbor two tier block system

(b) Angle filtering example

Fig. 01. (a) Total 24 neighbors are considered from two adjacent tiers. (b) Relationship of θ_i with elevation changes.

Angles between the target block and neighbors were calculated using equation 2. Elevation difference with a corresponding neighbor was divided by the center point distance to get the slope, from which the angle was calculated. Each neighbor provided an angle value. Let the i-th neighbor provide angle θ_i.

Two types of angular filtering can be performed based on the calculated angles.

1) Maximum angle filtering calculated the maximum of all θ_i and set this as the angle value for the target block. If any block has at least one higher elevation neighbor, its maximum angular filtering would provide a positive θ_{max} value (as see in Fig.01(b)). Therefore, maximum angular filtering would provide the slope with the highest elevated neighbor. Value of θ_{max} would vary from 0 to $+90$ degrees.
2) Minimum of calculated angles also was set as filtered angle for the target block. If it has at least one lower elevation neighbor, its minimum angular filtering value would be negative θ_{min}. Minimum angular filtering would provide the slope with the lowest elevated neighbor and θ_{min} would vary from 0 to -90 degrees.

It is clear that border pixels between high and low elevated areas would have the highest θ_{max} value, from maximum angle filtering, and the lowest θ_{min}value from minimum angle filtering. The only difference in minimum angle filtering is that the border pixel belongs to a higher elevation area. In maximum angle filtering, the border pixel belongs to a lower elevation area.

$$\theta_i = \tan^{-1}\left\{\frac{Z_i - Z}{d_i}\right\} \qquad (2)$$

$$\theta_{max} = \max\{\theta_i\}, \forall i$$

$$\theta_{min} = \min\{\theta_i\}, \forall i$$

2.3. Proposed implementation of bare earth model generation

2.3.1 Point cloud details*:* LiDAR data was obtained in LAS file format, where different parameters were provided for each point cloud. A few parameters were used for the rasterization process e.g. latitude, longitude, elevation, intensity, number of returns and return number.

2.3.2 Raster creation and interpolation: The whole algorithm consists of three major stages of operation named *pre-processing*, *object based analysis* and *fine tune*. All major stages and corresponding steps are shown in Fig.03. As described earlier, LiDAR point cloud was pre-processed first to obtain a rasterized elevation map. This pre–processing stage was performed in two steps: raster creation and interpolation, which were described in the pre-processing subsection. This pre-processed rasterized elevation map was used for subsequent object based analysis.

2.3.3: Object based analysis: At the very beginning, the physical properties of elevation were explored. It was assumed that rapid elevation change is a key property of non–ground blocks.

i) *Angle filtering*: Object based angle filtering is started with angle filtering, which is elaborated in Fig.04. First, angle filtering was performed on the elevation raster, which provided all potential edges. Minimum angle filtering was used to extract all the edges. As mentioned above, minimum angle filtering provides elevation difference of a block from its lowest elevated neighbor block. Additionally, the block, which has a very low elevation difference with all of its neighbors, can be considered as a point on a planar surface. In the next step, a threshold angle of $45°$ was applied, which means a neighbor x meters away can have a maximum of x meters elevation difference, to be considered to be in the same plane. Minimum angle filtering ensures that borders of adjacent but separate elevated area do not overlap. This characteristic facilitates the object separation process by connected component analysis. Therefore, this step provided all potential edges, which separated all non-ground objects.

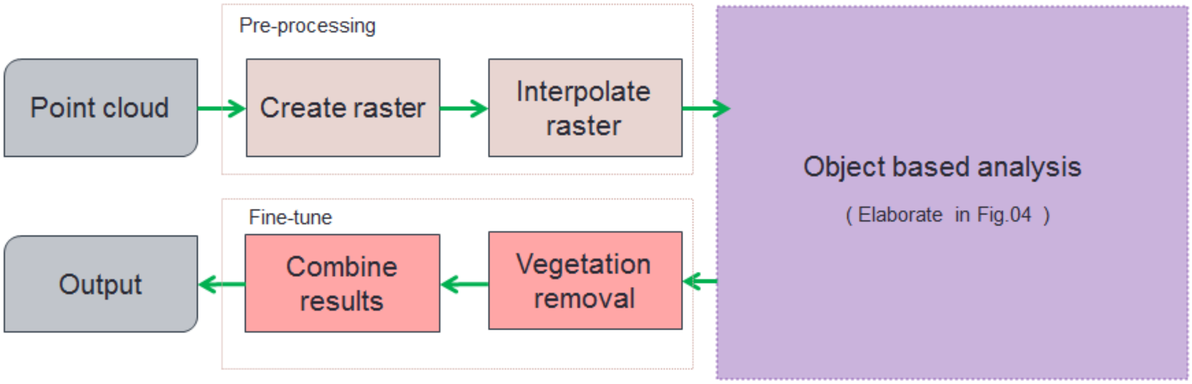


Fig. 03. Proposed algorithm with all major stages of operation.

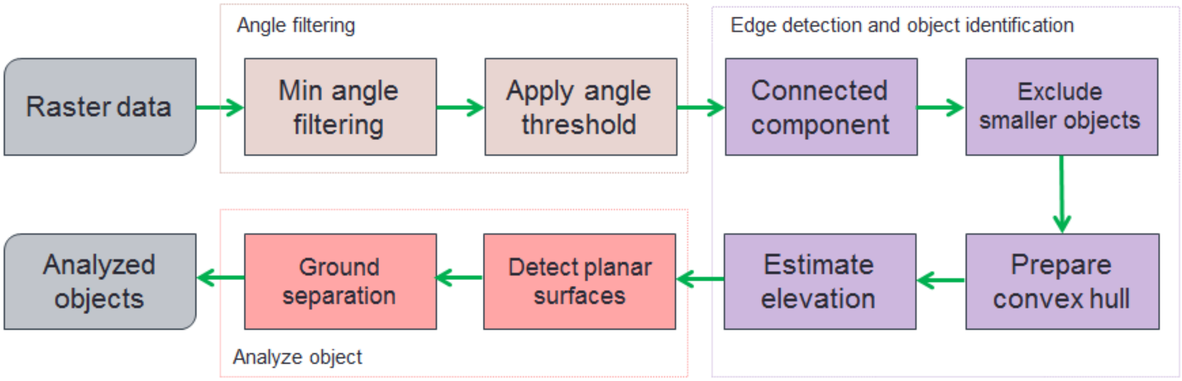


Fig. 04. Elaboration of object based analysis stage from Fig.03.

ii) *Edge detection and object identification*: In this step, all edges belonging to each object were identified. Edges from separated object were expected to be disconnected from each other because all edges were inside their own object. A connected component analysis was performed to connect all edge points from a single object as a single entity. Therefore, after performing connected component analysis, edges from each object were numbered in descending order based on the number of connected edge points. If at least n points were not connected to represent an object then it was excluded as being a smaller object. Now, the borders of all objects are well-defined and need extensive and separate analysis. A convex-hull is prepared for each object from its connected edges. The largest planar surfaces excluding all of these convex sets are potential ground points. An elevation model was estimated for each convex hull using all of these potential ground points.

iii) *Analyze objects*: The elevation estimation provides an approximate idea of the elevation beneath the object enclosed by a convex hull. Then, a rectangular box was taken to analyze each convex hull. Two criteria were used to distinguish ground and non-ground planar surfaces: 1) If the block is inside the rectangular box but outside the convex hull then it's a ground block 2) If the actual elevation of a block is outside a particular threshold of the estimated elevation then it is an object. By applying all these criteria, all planar surfaces were classified from each object. Surfaces were numbered according to the area of the surface. Planar surfaces from each object were helpful to build a 3–D model of the object. At the end of this stage, edge points were resolved and assigned to a particular surface according to their elevation. An edge point was assigned to its neighboring surface, which has the lowest elevation difference with that point's actual elevation. This completes the separation of ground and non-ground surfaces.

2.3.4: Vegetation removal: The numbers of returns were used to check all vegetation. If the lowest elevation of a vegetation block has below threshold elevation difference with its neighbor then it represents the bare earth elevation. If not, the elevation was estimated by the inverted weighted difference method. Finally, both results were combined to show three different classes: ground, man-made object and vegetation.

3. EXPERIMENTAL RESULTS

Experimental results are shown for an urban scene in this section. This urban scene was extracted from the Indiana statewide LiDAR dataset. Complex building structures were available to evaluate the performance of the algorithm in a complex urban scene.

In Fig.05, all edges extracted from different objects are shown. Angular filter provided all edges. Then connected component analysis combined the detected edges, which came from same object. Edges from larger objects are more reddish and smaller objects are more bluish. It can be seen that all building edges and smaller objects are visually distinguishable.

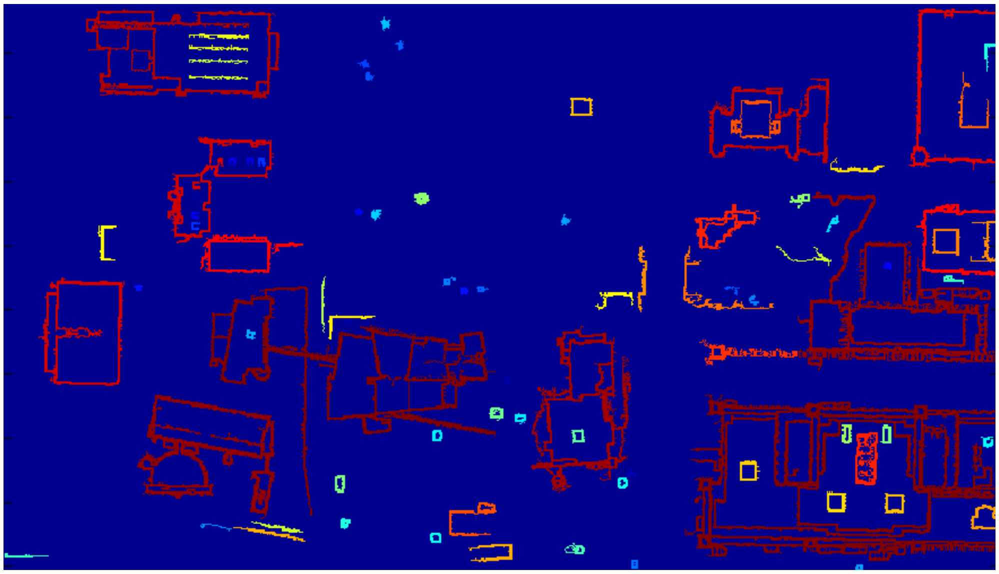


Fig. 05. Minimum angular filter followed by connected component analysis provide edges from different man-made objects

In Fig.06, all convex hulls are shown using light blue color. Connected edges from a single object were used to generate the convex hull, which contained the whole object. It is shown that all green connected edge lines are enclosed by their corresponding light blue convex-hull. Brown color dots on edge line are all convex-hull vertices. Once, all convex hulls were removed, the remaining blocks were considered potential bare earth blocks.

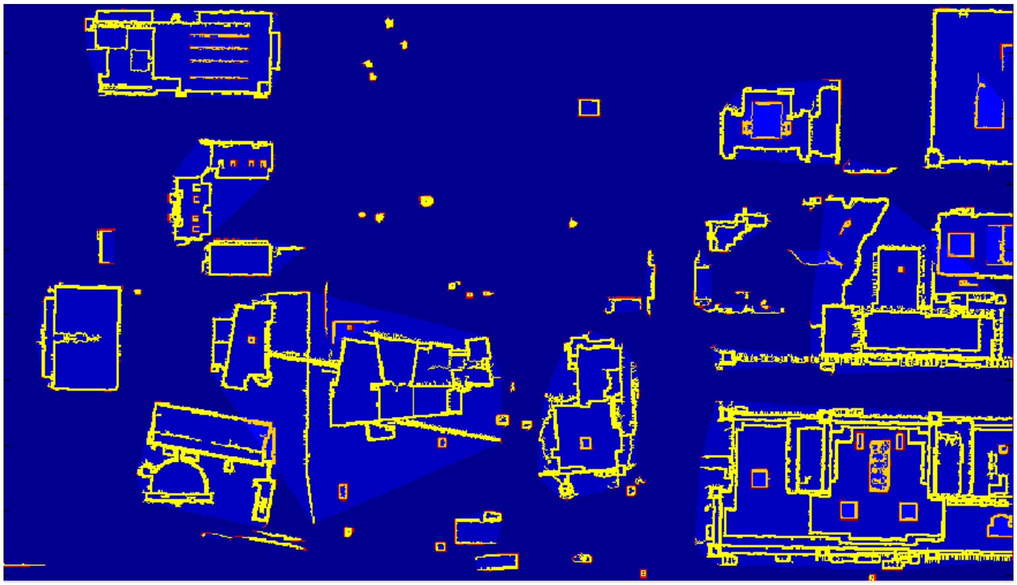


Fig. 06. Convex-hulls are shown in light blue color which was generated from detected edges from a particular object. Green color shows corresponding edge lines of that object.

In Fig.07, the elevation map of the remaining ground pixels is shown. Elevation beneath all convex hulls was approximated from elevation of available ground pixels using the IDW method. Thereafter, all objects were separately analyzed based on their actual elevation and estimated elevation at a particular pixel. Two criteria described in the proposed method section were used to find ground and non-ground pixels inside a convex hull.

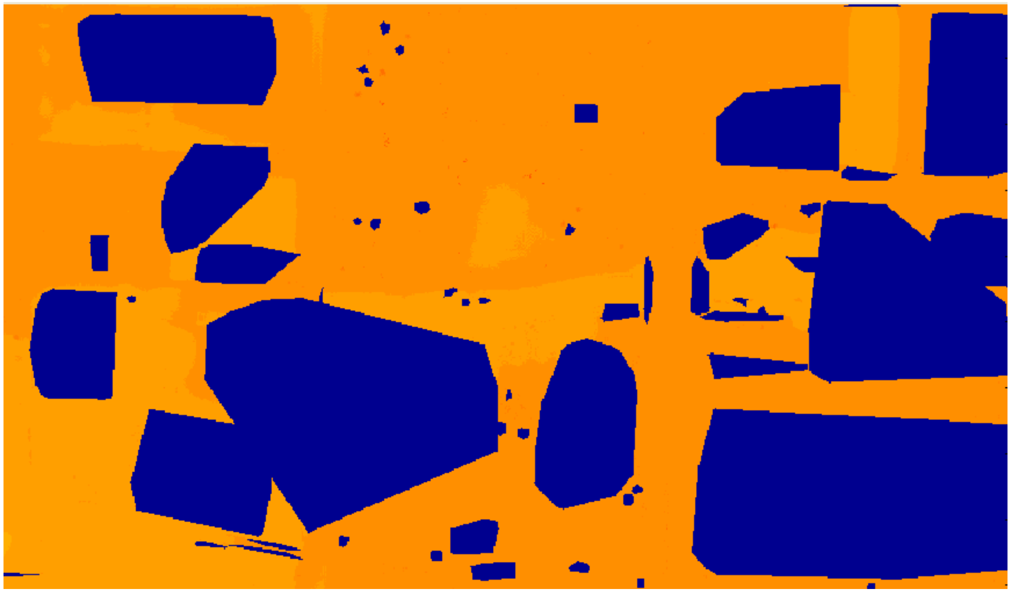


Fig. 07. Elevation map of ground points outside convex-hull (Orange), convex hulls (Blue)

In Fig.08, all detected planar surfaces from all objects are shown in different colors. Larger surface areas are indicated by reddish color. It is seen that all planar surfaces are visually distinguishable for each object. This surface information can be utilized for building and object modeling in future research. The edges were not resolved yet, which can be observed by blue lines between all surfaces. Minimum filtering ensured that detected edges belong to a higher elevation object. Therefore, all edges need to be combined with one of their neighborhood surfaces. In the next step, edges were assigned to a particular surface based on their elevation difference. In Fig.09, all planar surfaces are shown after combining edge line to their corresponding edges. It is seen that all blue lines between two surfaces from Fig.08 are removed in Fig.09. The final output shows all vegetation with detected man-made objects in Fig.10. All ground points are shown in green color, vegetation in brown color and man-made objects in blue color.

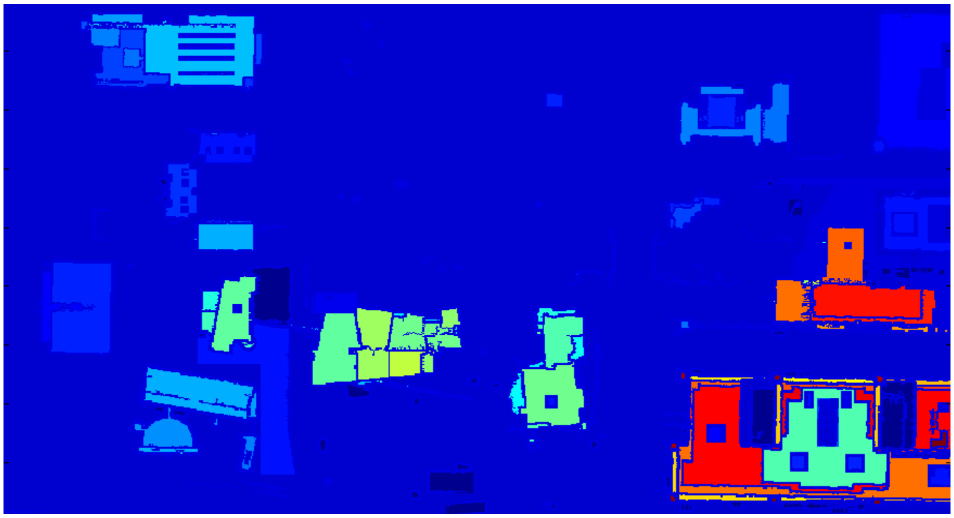


Fig. 08. Detected planar surfaces before combining edge lines to their corresponding surfaces.

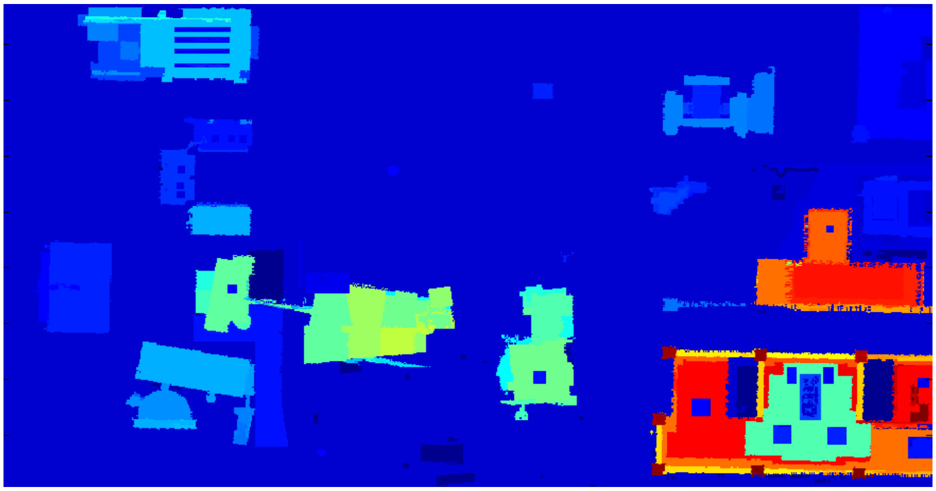


Fig. 09. Detected planar surfaces after combining edge lines to their corresponding surfaces.

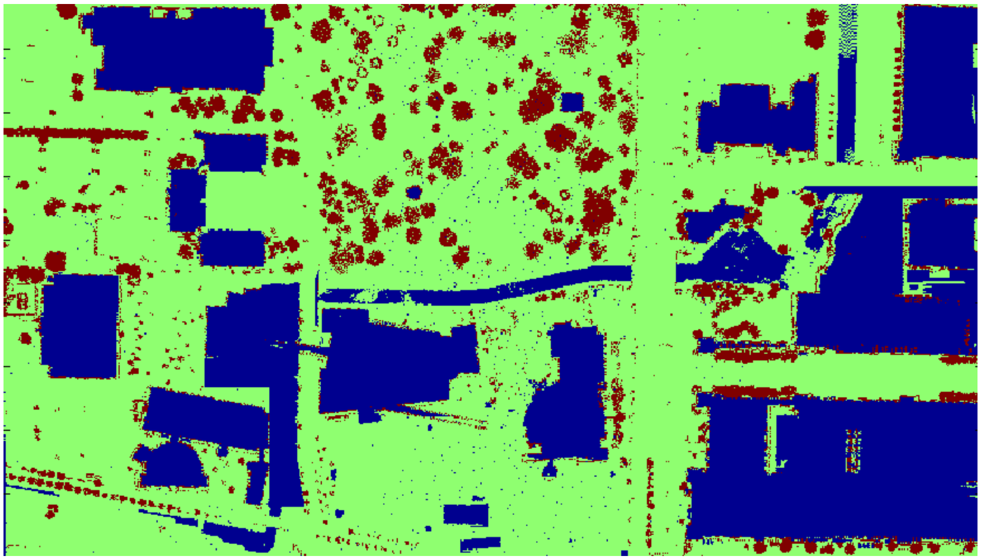


Fig. 10. Ground blocks in green, Man-made objects in blue and vegetation in brown are shown.

4. CONCLUSIONS AND FUTURE WORK

In this work, a ground filtering method was presented, powered by a novel angular filtering based algorithm. The proposed angular filtering algorithm calculates minimum/maximum slope between a targeted block with its neighbors from two adjacent tiers. Minimum/maximum angle filtering showed promising result to detect edges even in a complex urban scene. The angular filtering was followed by connected component analysis and convex hull analysis to detect man-made objects. Vegetation was detected from pulse return count at the end of the process.

Intensity and other statistical analysis will be useful for better performance of ground non-ground separation in future work. Reduce computational complexity, multi-grid data operation, extensive comparison with other leading algorithms using the same dataset, making python scripts public availability – these are few plans for our future work.

5. REFERENCES

[1] Jie Shan and Charles K Toth, Topographic laser ranging and scanning: principles and processing, CRC Press, 2008.

[2] Christian Briese and Norbert Pfeifer, “Airborne laser scanning and derivation of digital terrain models,” in Fifth Conference on Optical, 2001, vol. 3.

[3] Xuelian Meng, Le Wang, Jos Luis Silvan-Cardenas, and Nate Currit, “A multi-directional ground filtering algorithm for airborne LIDAR,” ISPRS Journal of Photogrammetry and Remote Sensing, vol. 64, no. 1, pp. 117–124, 2009.

[4] G. Sithole and G. Vosselman, “Experimental comparison of filter algorithms for bare-Earth extraction from airborne laser scanning point clouds,” *ISPRS journal of photogrammetry and remote sensing*, vol. 59, no. 1, pp. 85–101, 2004.

[5] X. Meng, N. Currit, and K. Zhao, “Ground filtering algorithms for airborne LiDAR data: A review of critical issues,” *Remote Sensing*, vol. 2, no. 3, pp. 833–860, 2010.