DEM Generation Using Adaptive Point Cloud Filtering Algorithm

Miao Wang* Yi-Hsing Tseng Fu-Chen Chou

KEY WORDS: Airborne LIDAR, DEM, Morphological, 3D raster, Octree

ABSTRACT

DEM generation is the primary application of airborne LIDAR. The major process of generating DEM from airborne LIDAR is to filter out off-terrain points from the point cloud data. This paper proposes an adaptive filtering algorithm for DEM generation based on the morphological filtering theory. To make the algorithm adaptive, a 3D rasterized octree structure is used to organize point cloud data, so that the trend surface and local slope of ground surface can be estimated. Several data sets covering the areas featured with different topographic characteristics were tested to show the feasibility of the proposed algorithm. The test results show that it works very well on the data of rural or residential areas. However, some over-filtering problem may happen on the data of rough terrain covered with dense vegetations. To assure the quality of DEM product, manual check and editing is still necessary against improper filtering results.

1. INTRODUCTION

Airborne LIDAR is able to capture accurate point positions of 3D ground surface in the speed of several thousands points per second. The large amount of collected data points densely distributed on the scanned ground surface and features, and is called point cloud data. The primary application of airborne LIDAR is DEM generation (Ackermann, 1999). The core procedure of generating DEM from airborne LIDAR is to filter out off-terrain points caused by reflections of laser pulses on vegetation and buildings.

This paper presents a new filtering strategy which is featured of the use of a 3D raster structure and a workflow of an iterative procedure. The principle of the proposed algorithm is to peel off off-terrain points iterative. To make the algorithm adaptive, a 3D raster structure is employed to organize point cloud data, so that the trend surface and local slope of the ground can be estimated. Based on 3D rasterized point cloud data, the filter concepts of slope based, block-minimum, and surface based are employed in each run of the iteration. Trend surface estimation plays an important role in this scheme, i.e., the surface based concept is used. For trend surface estimation, only the points contained in the lowest non-empty raster cells are used, i.e., the block-minimum concept is applied. A rough filtering will be perform first to get rid of points of large ground objects, and then followed a detailed filtering by using the slope-based filter proposed by Vosselman (2000). The size of the raster cells will be reduced in each run to refine the estimated trend surface, so that off-terrain points are gradually filtered out.

2. ADAPTIVE MORPHOLOGICAL FILTERING

2.1 Slope Based Morphological Algorithm

The design of morphological algorithm is based on an idea that the height difference of any two points within the nearby region should be smaller than a defined value, $\Delta h_{\max}$. It means that a measure point will be filtered out, if the height difference between this point and any other points within the nearby area is larger than $\Delta h_{\max}$. In other words, a kept point should fulfill the following condition:

$$DEM = \{p_i \in A \mid \forall p_j \in A : h_{p_i} - h_{p_j} \leq \Delta h_{\max}(d(p_i, p_j))\} \tag{1}$$

* Ph. D. student, Department of Geomatics, National Cheng Kung University, Tainan, Taiwan, monsterr@seed.net.tw

If the nearby region, or called search region, is defined by a kernel function, k(Δx, Δy), then the eroded surface, the bald ground surface, can be represented by the kept measure points. Therefore, the eroded surface is defined by:

$$k(\Delta x, \Delta y) = -\Delta h_{\max}(d(p_i, p_j)) = slope \times d \tag{2}$$

Equation (1) then can be redefined as:

$$DEM = \{p_i \in A \mid h_{p_i} \le e_{p_i}\} \tag{3}$$

The kernel function may be adjusted adapting to various local slopes, and the algorithm becomes a slope-based filter (Vosselman, 2000). Figure 1 shows the concept of a slope-based filter. A point will be filtered out, if there is any other point under the cone, formed by the point, a specified slope, and the search region. For example, if the slope of a terrain is considered smaller the 30%, the kernel function can be:

$$k(\Delta x, \Delta y) = -\Delta h_{\max}(d) = -0.3d \tag{4}$$

The filtering works with a predefined size of search region. For each test point, as the centre of the search region, all other points in the search region will be compared. Searching for the points contained in the search region is the most time consuming task in the filtering process. The larger search region used, the more computation effort needed. To improve the efficiency of point searching is requisite for most LIDAR application cases.

2.2 Idea of Adaptive Filtering

It is in a dilemma as to determine the size of search region and the slope threshold for a filtering case. A small search region maintains small topographical undulations well, but points on large objects, such as buildings, may not be clearly filtered out. Figure 2 shows an example of this case. The points in the central part of the building were wrongly kept. On the other hand, a large search region would filter out most of unwanted points, but the possibility of wrongly filtering out wanted points of ground surface is increased too. Similarly, the use of a small slope threshold may filter out some wanted terrain relief, and some low vegetation points may be kept if a large slope threshold is used. Therefore, a time consuming manual editing is frequently required after the filtering process for a case of real application to obtain accurate DEM.

A proper search region and slope threshold can be pre-determined if the trend surface of terrain can be estimated before filtering. An adaptive filtering algorithm is therefore can be implemented by providing a predicted local slope derived from the estimated trend surface.

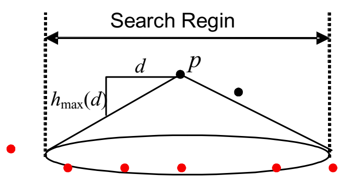

Figure 1. The concept of slope-based filtering.

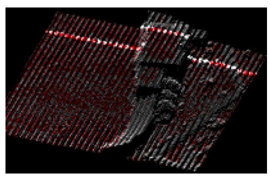

Figure 2. Points on large objects are not clearly filtered out due to the use of a small search region.

3. ATAPTIVE POINT CLOUD FILTERING ALGORITHM

Point cloud data of LIDAR are distributed evenly and sub-randomly in the horizontal space. It is hard and inefficient to deal with such unorganized data in computer, especially to find points within a specified search region for the filtering algorithm. So a 3D raster structure is employed to organize point cloud first. Point searching in a raster structure can be very efficient, and it makes the filter to be adaptive by providing an estimated local slope of each raster cell based on the contained points.

3.1 3D Raster Structure

LIDAR point cloud distribute in 3D space. A simple 3D raster structure is to divide the 3D space into regular rectangular cells along each axis. The dimensions of each cell can be denoted as dX, dY, and dZ. For most airborne LIDAR cases, dZ should smaller than dX and dY in order to separate multi-layered data. Setting dX equal to dY is recommended. During the divide process, the cells are organized with octree structure. The cells contain no point inside it is not recorded to save storage space. The octree structure can transfer to 3D raster index easily. So neighboring points can be search quickly in the same or neighboring cell via the octree structurized 3D raster structure. Figure 3 shows an example of 3D raster structure.

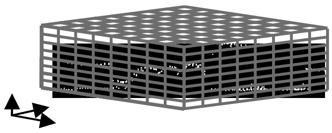

Figure 3. A 3D raster structure

3.2 Trend Surface Estimation

Trend surface estimation can be much easier after the point cloud data are organized in a 3D raster structure. First of all, points of the upper layers can be quickly excluded from trend surface calculation by keeping the points contained in lowest non-empty cells. The trend surface then can be formed by calculating a best-fit plane for each lowest non-empty cell.

It is preferable to use a raster structure with large dX and dY but small dZ for the first step, so that the points belong to large buildings or large canopy of trees can be filtered out at the beginning. The Z dimension of the raster cell should be smaller than the height of ground objects we want to filter out. Figure 4 shows an example of using a raster structure with dX=dY=5 m and dZ=1 m. Some roof and tree canopy points were kept in this case. They can be filtered out clearly when dX=dY=15 m and dZ=1 m (Figure 5). However, a large horizontal dimension of raster cell will tend to exclude some points of ground surface, especially in a sloping area as shown in Figure 6.

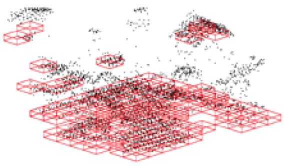

Figure 4. The lowest non-empty cells in a raster structure with dX=dY=5 m and dZ=1 m.

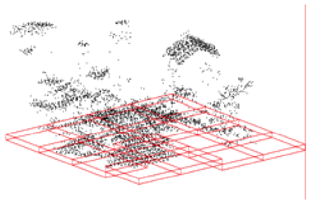

Figure 5. The lowest non-empty cells in a raster structure with dX=dY=15 m and dZ=1 m.

The purpose of trend surface estimation is to determine the local slope of the ground surface when a location is given. Therefore, it can be simplified as to determine a locally best-fit plane according to the given position. Based on the 3D raster structure, we can easily find the points belonging to the ground surface nearby the given position. For example, it is convenient to find the points contained in the 3 by 3 lowest non-empty cells with the centre cell containing the given position for the calculation of the best-fit plane. This procedure can be combined with the filtering process to pre-estimate the local slope, so as it becomes an adaptive filter.

The least-squares estimation can be applied to determine the best-fit plane of the located ground surface points. It is reasonable to assume that the best-fit plane would be mostly close to a horizontal plane and far away from a vertical plane. Therefore, the best-fit plane can be represented by a slope-intercept form as:

$$Z=AX+BY+C \tag{5}$$

By minimizing the sum of square distances of the points from the unknown plane, the parameters of the best-fit plane can be estimated. According to Eq.(5), the normal vector of the best-fit plane will be (A, B, -1). Applying the cosine law, the angle between the normal vector and the vertical vector can be calculated as:

$$\theta = \arccos\left(\frac{\vec{n}\cdot\vec{z}}{|\vec{n}|\times|\vec{z}|}\right) \tag{6}$$

This is equal to the angle between the best-fit plane and the horizontal plane (Figure 7). The local slope, therefore, can be calculated as tangent θ.

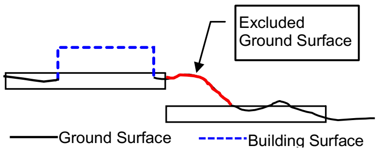

Figure 6. A large horizontal dimension of raster cell will tend to exclude some points of ground surface

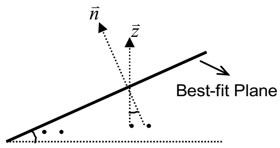

Figure 7. The angle between the best-fit plane and the horizontal plane.

4. ADAPTIVE FILTERING ALGORITHM

Although the trend surface estimation provides prior information of local slope for the filter, the determination of trend surface still depends on the size of raster cell. Using a large raster cell, the estimated trend surface would not follow some small undulations of ground surface as the circled areas shown in Figure 8. However, when a small raster cell is applied, the estimated trend surface may follow the surface of large ground objects as the circled area shown in Figure 9. In order to cope with this problem, an algorithm of hierarchically peeling off the points of ground objects is designed. In other words, the algorithm is an iterative process. In each step of iteration, a rough filtering and a detailed filtering are performed to gradually filter out the unwanted points.

4.1 Rough filtering

The rough filtering is performed based on the estimated trend surface using a raster structure with relatively large horizontal cells, denoted as (DX, DY). In the beginning of the iteration, a largest cell size can be set to surely filter out the largest ground objects in the application area. Then it gradually reduces the cell size in each step of iteration. The idea of the rough filtering is to remove all of the points which are higher than the estimated trend surface by $difZ$, which is defined as:

$$difZ = |z_i - (A*x_i + B*y_i + C)| \tag{7}$$

Figure 10 shows the idea of the rough filtering. The $difZ$ is a threshold value of the rough filter. Considering the height of common low buildings and trees, setting $difZ$ from 1 to 3 meters would be a good choice.

4.2 Detail filtering

The detailed filtering is a slope-based filtering, but the local slope is estimated from the trend surface determined from the raster structure with smallest horizontal raster cells, denoted as (dX, dY). In order to maintain very small undulations of the ground surface, adding a small value σ_h into the kernel function (Eq. 4) is suggested. Therefore, the threshold of the height difference employed in our algorithm is:

$$\Delta h_{\max} = slope_{local} \times d + \sigma_h \tag{8}$$

The flow chart shown in Figure 11 depicts the overall work flow of the proposed adaptive filtering algorithm. Although iterative process is suggested in general, non-iterative process is optional for flat area to save computation time.

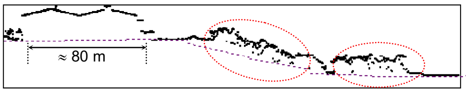

Figure 8. The estimated trend surface may not follow small undulations of ground surface when a large raster cell is used

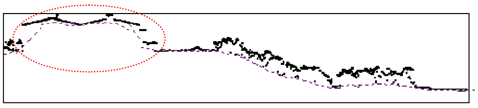

Figure 9. The estimated trend surface may follow the surface of large ground objects when a small raster cell is used

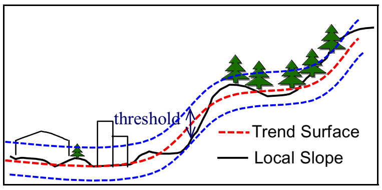

Figure 10. The idea of the rough filtering

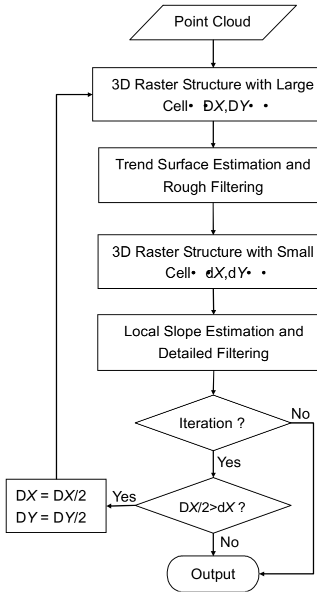

Figure 11. The overall work flow of the proposed adaptive filtering algorithm.

5. EXPERIMENTS

5.1 Test Data

The test data were extracted from an airborne LIDAR data set collected with Leica ALS40, in Hsinchu, Taiwan, on April 14th, 2002. The area covered by the test data is 1.5 by 1.5 km^2 and contains some flat and hilly areas covered by buildings or vegetation. The point cloud contains some multiple echoes, and the point density is about 1.7 points/m^2. The total point number is more than 4.3 million. Figure 12 shows the perspective view of the point cloud.

5.2 Results

The test data were processed iteratively with the beginning large cell (DX, DY) = (30m, 30m), the small cell (dX, dY) = (2m, 2m), and the threshold ΔZ = 1.5m. Figure 13 shows the kept points after filtering, in which the white areas were covered by the removed points. Figure 14 shows the DEM interpolated using the kept points.

The test results not been checked rigorously. A preliminary check was done by visually investigating a specified profile, in which the kept ground points are shown in red and the removed off-terrain points are shown in black. Figure 15 shows a profile on a flat area with some buildings and trees on it. Figure 16 shows a profile on a ramp area with some trees and bushes. Figure 17 shows a profile of a hilly area with densely distributed trees. In those figures, the filtering results seem to be correct. However, there are still some profiles propose questionable improper keeping or removing points. The circled parts shown in Figure 18 and 19 are some examples. In the future, a rigorous and systematic check requires a quantitative analysis of the test results.

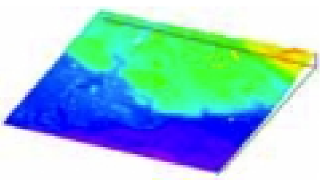

Figure 12. The perspective view of the test data.

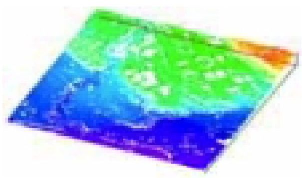

Figure 13. The kept ground points after filtering.

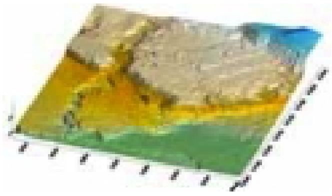

Figure 14. Generated DEM using the kept points

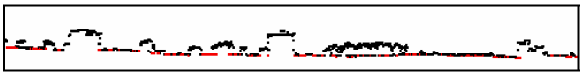

Figure 15. A profile on a flat area with some buildings and trees on it.

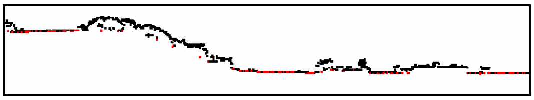

Figure 16. A profile on a ramp area with some trees and bushes.

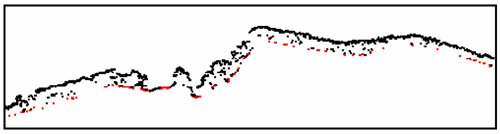

Figure 17. A profile of a hilly area with densely distributed trees.

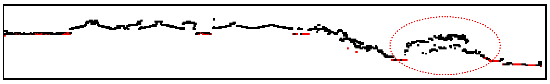

Figure 18. A filtering result with some wrongly removed points.

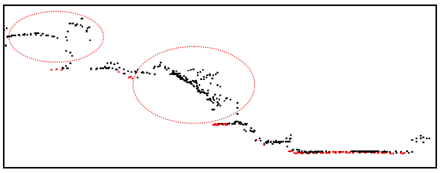

Figure 19. A problematic filtering result.

6. CONCLUDING REMARKS

This paper proposed a new filtering procedure for DEM generation from airborne LIDAR data. This filtering algorithm is featured of the use of a 3D raster structure of point cloud data and the combination of several filter concepts in an iterative procedure. The principle of the algorithm is to peel off off-terrain points iteratively. It is discovered that organizing the point cloud data in a 3d raster structure improves the efficiency of the filtering process. An adaptive filtering algorithm can be easily implemented based on the raster structure. The filtering results of the test data were visually checked in the diagrams of profiles of kept and removed points. Although most visually checked profiles show reasonably correct results, there are still some profiles propose questionable improper keeping or removing points. A quantitative analysis and error check is required in the future to judge the filtering algorithm rigorously.

The filtering of complex urban landscapes, hilly areas, or dense vegetation areas still poses many problems. A filtering algorithm may be adjusted to deal with one situation, but may create other problems for some opposite cases. Although making the filtering adaptive to the local distribution of points can solve some of the problems without creating new problems, it is still hard to reach a perfect filtering. As suggested by Sithole and Vosselman (2003), filtering using segmentation, data fusion, and understanding of the context of the landscape being filtered might be the ways to make a filtering algorithm more robust. However, in the meantime, manual check and editing is still recommended for a real project of DEM generation.

ACKNOWLEDGEMENTS

This research project was sponsored by the National Science Council of Republic of China under the grants of NSC 93-2211-E-006-066. We sincerely express our gratitude to the Council of Agriculture for the provision of airborne LIDAR data.

REFERENCES

Ackermann, F., 1999. Airborne Laser Scanning - Present Status and Future Expectations, ISPRS Journal of Photogrammetry & Remote Sensing, 54: pp. 64-67.

Sithole, G. and Vosselman, G., 2003. Comparison of Filtering Algorithms, ISPRS working group III/3 workshop `3-D Reconstruction from airborne laserscanner and InSAR data', Dresden, Germany, http://www.commission3.isprs.org/wg3/.

Vosselman, G., 2000. Slope Based Filtering of Laser Altimetry Data, International Archives of Photogrammetry and Remote Sensing, Amsterdam, pp. 935-942.