ISPRS Journal of Photogrammetry and Remote Sensing 64 (2009) 117–124

Contents lists available at ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

journal homepage: www.elsevier.com/locate/isprsjprs

# A multi-directional ground filtering algorithm for airborne LIDAR

Xuelian Meng [a,*], Le Wang [b], José Luis Silván-Cárdenas [a], Nate Currit [a]

[a] Texas State University-San Marcos, Department of Geography, 601 University Drive, San Marcos, TX 78666, United States
[b] The State University of New York at Buffalo, Department of Geography, Buffalo, NY 14261, United States

ARTICLE INFO

*Article history:*
Received 29 August 2007
Received in revised form
31 July 2008
Accepted 5 September 2008
Available online 10 October 2008

*Keywords:*
LIDAR
Ground filtering
Multi-directional

ABSTRACT

Automatic ground filtering for Light Detection And Ranging (LIDAR) data is a critical process for Digital Terrain Model (DTM) and three-dimensional urban model generation. Although researchers have developed many methods to separate bare ground from other urban features, the problem has not been fully solved due to the similar characteristics possessed by ground and non-ground objects, especially on abrupt surfaces. Current methods can be grouped into two major categories: neighborhood-based approaches and directional filtering. In this study, following the direction of the second branch, we propose a new Multi-directional Ground Filtering (MGF) algorithm to incorporate a two-dimensional neighborhood in the directional scanning so as to prevent the errors introduced by the sensitivity to directions. Besides this, the MGF algorithm explores the utility of identifying pattern varieties in different directions across an image. The authors conducted a comprehensive test of the performance on fifteen study sites and compared our results to eight other publicized methods based on the Kappa coefficients calculated from the error matrices reported by ISPRS. Overall, the MGF filter produces a promising performance in both urban and forest areas. The size and shape of non-ground objects do not pose significant influence on the performance of the MGF algorithm. The fact that MGF algorithm is robust to two commonly required parameters, slope and elevation difference thresholds, has added practical merits to be adopted in different landscapes.



## 1. Introduction

An airborne LIDAR system acquires dense point measurements using three-dimensional coordinates more directly than traditional surveying and mapping systems, e.g., photogrammetric systems (Shan and Sampath, 2005). Value-added LIDAR products, like DTMs, hydrologic models, three-dimensional urban visualization models, and transportation network models, increasingly demand accurate LIDAR surveys (Hill et al., 2000). Currently, LIDAR has two major advantages over photogrammetric systems: (1) the acquisition of vertical information over a large area is more cost-effective; and (2) there are fewer requirements for data preprocessing. In terms of DTM creation, LIDAR has taken the place of traditional photogrammetric methods and become the primary technique for producing regional or national DTMs in some countries, especially in Europe (Vosselman, 2000; Schickler and Thorpe, 2001; Elmqvist et al., 2001).

In raw LIDAR data, both bare-ground and non-ground objects, such as trees, buildings, vehicles, and electrical wires, generate backscatter. Non-ground points need to be identified and eliminated from LIDAR measurements before constructing value-added products like DTMs (Zhang et al., 2003; Vosselman, 2000). Likewise, ground points need to be removed to accurately identify non-ground objects, such as buildings and trees. In either case, an efficient and accurate ground filtering is required. Existing algorithms have achieved some success, but usually have difficulty along steep slopes or ridges. To this end, our goal is to develop a better ground filtering algorithm to facilitate DTM creation.

Ground filtering algorithms operate on either raw LIDAR point clouds or gridded elevation values (Sithole and Vosselman, 2004), which are derived by interpolation of raw data. Interpolation techniques include fitting a linear function (Passini and Jacobsen, 2002), a surface function (Kraus and Pfeifer, 2001; Okagawa, 2001; Haugerud and Harding, 2001), a morphology function such as smoothness (Kilian et al., 1996) or a local mean or minimum value. Merits and drawbacks for the algorithms using either type of input data have been reported (Zhang et al., 2003). Algorithms that work on raw LIDAR data (Zhang and Whitman, 2005; Elmqvist, 2002) require less preprocessing, and avoid

* Corresponding address: Texas State University-San Marcos, Department of Geography, 601 University Dr., ELA 392, San Marcos, TX 78666, United States. Tel.: +1 512 245 2170.

*E-mail addresses:* xm1001@txstate.edu (X. Meng), lewang@buffalo.edu (L. Wang), jlsilvan@txstate.edu (J.L. Silván-Cárdenas), currit@txstate.edu (N. Currit).

0924-2716/$ – see front matter © 2008 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS). Published by Elsevier B.V. All rights reserved.
doi:10.1016/j.isprsjprs.2008.09.001

errors introduced by interpolation. But searching for neighbors in an irregularly distributed point cloud can be time-consuming and troublesome, especially when users apply the algorithms to broader areas. Interpolating points into a regularly distributed grid can resolve this problem more effectively. In this paper, we have developed a multi-directional ground filtering algorithm using gridded elevation values that economizes the search for neighbors in multiple directions.

Most ground filters are based on the assumption that natural terrain variations are gradual, rather than abrupt. Therefore, ground elevation and slope should vary smoothly when moving from one ground point to another neighboring ground point. In contrast, the boundary between ground and non-ground points should exhibit an abrupt change in elevation and slope. The joint use of slopes, elevation differences, and local elevations can discriminate ground points from non-ground points (Zhang and Whitman, 2005; Vosselman, 2000).

Existing ground filtering algorithms calculate elevation differences and slopes based on pixels within a roving, two-dimensional window or along a scan line in a specified direction. Generally, neighborhood-based filters preserve the shape of non-ground objects but are insensitive to small-scale elevation changes on ground, like shrubs, low walls, and vehicles. Directional scanning approaches may effectively detect small objects through referencing to its intermediate neighbors but sometimes may generate artificial lines across the ground or objects.

The two-dimensional roving window technique compares the center point value to the mean or minimum value of its neighbors, or to a value estimated from its neighbors. If the center point value is above a predetermined threshold, the point is labeled as non-ground. The size of the neighborhood is critical for the performance of neighborhood-based filters (Kilian et al., 1996; Zhang et al., 2003). If the neighborhood size is smaller than the size of non-ground objects, points lying near the center of the objects will be wrongly labeled as ground points since their predicted values will not deviate much from the elevation of the center of the objects.

Whitman et al. (2003) develop an expanding window technique by gradually increasing the window size to remove non-ground objects of different sizes and avoid mislabeling ground pixels. Zhang and Whitman (2005), Zhang et al. (2003), and Vosselman (2000) report similar techniques and results. Other approaches such as the weighted window (Kraus and Pfeifer, 2001), multi-resolution or changing mesh size (Silván-Cárdenas and Wang, 2006; Zhang and Whitman, 2005; Kampa and Slatton, 2004) are alternative strategies for this problem.

The scan line technique creates an elevation or slope profile for each scan line, and identifies ground points based on the information along the profile. Sithole and Vosselman (2005) segment the profile into ground and non-ground points based on elevation differences along scan lines. Sithole (2001) proposes an adaptive filter to identify ground points based on the slope threshold of a profile. Brovelli et al. (2002) filter non-ground points by comparing the elevation of the points with the estimated value in a bilinear spline surface. The major drawback of most scan line approaches is that they are limited by the choice of filtering directions. Most existing directional scanning methods suffer dramatically when the ground surfaces present unique patterns in different directions along a scan line profile. To remedy this shortcoming Shan and Sampath (2005) develop a one-dimensional and bi-directional labeling (OBL) filter combining elevation and slope changes. Nevertheless, a bi-directional filtering algorithm only considers one-dimensional neighbors (i.e., those along a scan line) and does not take full advantage of neighborhood information.

In this study, we present an algorithm that combines the advantages of the directional and neighborhood-based scanning. Development of this algorithm explores the utility of identifying a variety of patterns in different directions across an image. Specifically, our MGF algorithm considers the slopes for neighboring pixels in up to four directions (i.e., parallel and perpendicular to a scan line), and the elevation difference between a pixel and the local minimum elevation within a two-dimensional neighborhood and the nearest ground pixel. A practical advantage of the MGF algorithm is that the object size and shape have no significant influence on the performance of the algorithm, which is especially critical for urban applications. Additionally, the MGF algorithm is robust to parameter selection based on experiments with and without an optimization process.

## 2. Data

The International Society for Photogrammetry and Remote Sensing (ISPRS) Commission III/WG3 provides LIDAR data for eight study sites with both first and last returns in urban and rural environments. ISPRS collected the raw LIDAR data using an Optech ALTM scanner and manually generated fifteen reference sites from sites 1–7 (www.commission3.isprs.org/wg3/). The authors selected these fifteen sites to test the performance of the MGF algorithm and compare the algorithm with other methods evaluated by ISPRS. Table 1 describes the characteristics of the study sites modified from Sithole and Vosselman (2004). Site 8 is not included due to lack of reference data.

Two preprocessing steps are necessary prior to applying the MGF algorithm: outlier removal and grid interpolation. Outlier elevation values, including random errors caused by birds, airplanes, or sensor noises, can be removed using a histogram examination and the Delaunay triangulation technique (Silván-Cárdenas and Wang, 2006). The elevation histogram distribution reveals the elevation range of ground and above-ground features, and points with elevations out of the range are usually outliers. The remaining outliers are removed if the elevations fall out of the range of their neighbors as defined by Delaunay triangulation. In this research, the threshold for high outliers is twice as high as the one for low outliers because many above-ground pixels from trees are much higher than their triangulation neighbors. For example, if the threshold is five meters, the range is from five meters below the local minimum to ten meters above the local maximum elevation. The last preprocessing step is to interpolate the irregularly distributed point clouds into grid pixels by assigning the elevation of the nearest point found within a specified distance to the output pixel. When no point is within the specified distance a no-data value is assigned to the pixel.

## 3. The multi-directional ground filtering (MGF) algorithm

The MGF algorithm filters ground points based on three criteria: (1) the slope measured in various scanning directions; (2) the elevation difference between each point and the nearest ground point; and (3) the elevation difference between each point and the minimum elevation in a local neighborhood. Slope is calculated between each point and the previous point in a particular scanning direction. Elevation differences are simple arithmetic differences. We examine various neighborhood sizes to test their influence on the performance of the MGF algorithm. We believe that using these three criteria will produce a robust ground filtering algorithm.

The first step in running this algorithm is to select a ground pixel near the first scan line. Our algorithm automatically selects the lowest pixel within a local neighborhood since ground is usually the lowest feature in the local environment. To avoid selecting an outlier pixel instead of a true ground pixel we remove all outliers prior to analysis and alter the size and location of the searching area.

After finding the ground seed, the MGF filter iterates repeatedly through the following steps to label points as ground, non-ground, or uncertain. We scan each line in the two, three or four of the four possible directions: (1) left to right, (2) right to left, (3) top to bottom, and (4) bottom to top.

**Table 1**
Study site features after Sithole and Vosselman (2003)

| Site | Pixel size (m) | Ref. data | Special features |
|---|---|---|---|
| City site 1 | 1 | samp11 | Steep slopes, mixture of vegetation and buildings on hillside, buildings on hillside, data gaps. |
| | 1 | samp12 | |
| City site 2 | 1 | samp21 | Large buildings, irregularly shaped buildings, road with bridge and small tunnel, data gaps. |
| | 1 | samp22 | |
| | 1 | samp23 | |
| | 1 | samp24 | |
| City site 3 | 1 | samp31 | Densely packed buildings with vegetation in between, buildings with eccentric roofs, open space with mixture of low and high features, data gaps. |
| City site 4 | 1 | samp41 | Railway station with trains (low density of terrain points), data gaps. |
| | 1 | samp42 | |
| Forest site 5 | 2 | samp51 | Steep slopes with vegetation, quarry, vegetation on river bank, data gaps. |
| | 2 | samp52 | |
| | 2 | samp53 | |
| | 2 | samp54 | |
| Forest site 6 | 2 | samp61 | Large buildings, roads with embankments, data gaps. |
| Forest site 7 | 2 | samp71 | Bridge, underpass, roads with embankments, data gaps. |

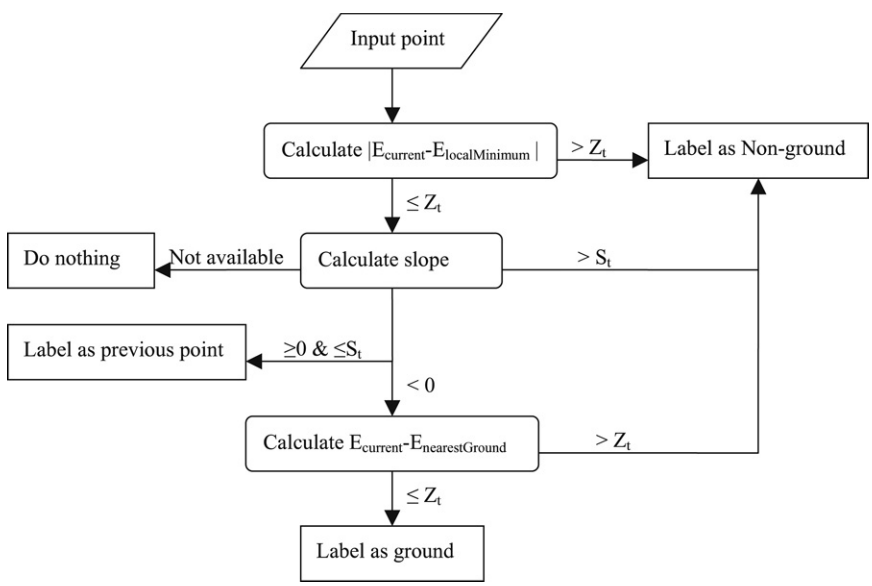


**Fig. 1.** Flow chart of labeling process of the MGF algorithm.

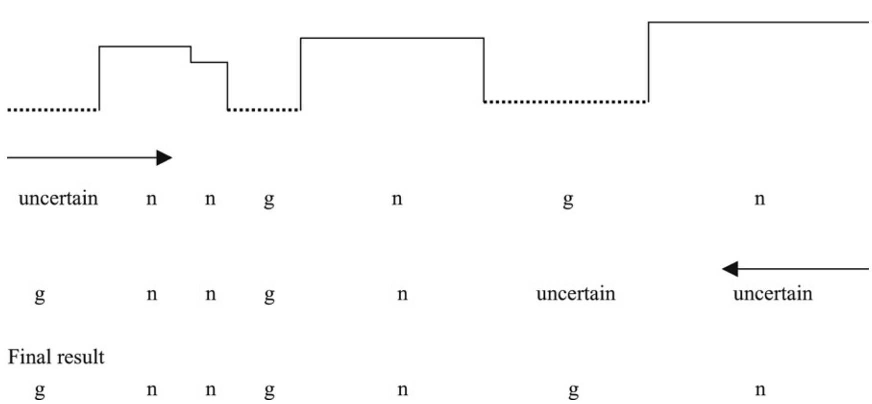


**Fig. 2.** The labeling process of the MGF filter given scanning directions from left to right and then from right to left. The dotted lines represent ground, and the others mean non-ground.

For each pixel, we illustrate the labeling process as shown in Fig. 1 and an example as shown in Fig. 2:

(1) Calculate the elevation difference between this point and its lowest local elevation.

a. If the elevation difference is greater than the elevation threshold, label this point as a non-ground point.
b. Proceed to the next step if the elevation difference is equal or smaller than the elevation threshold.

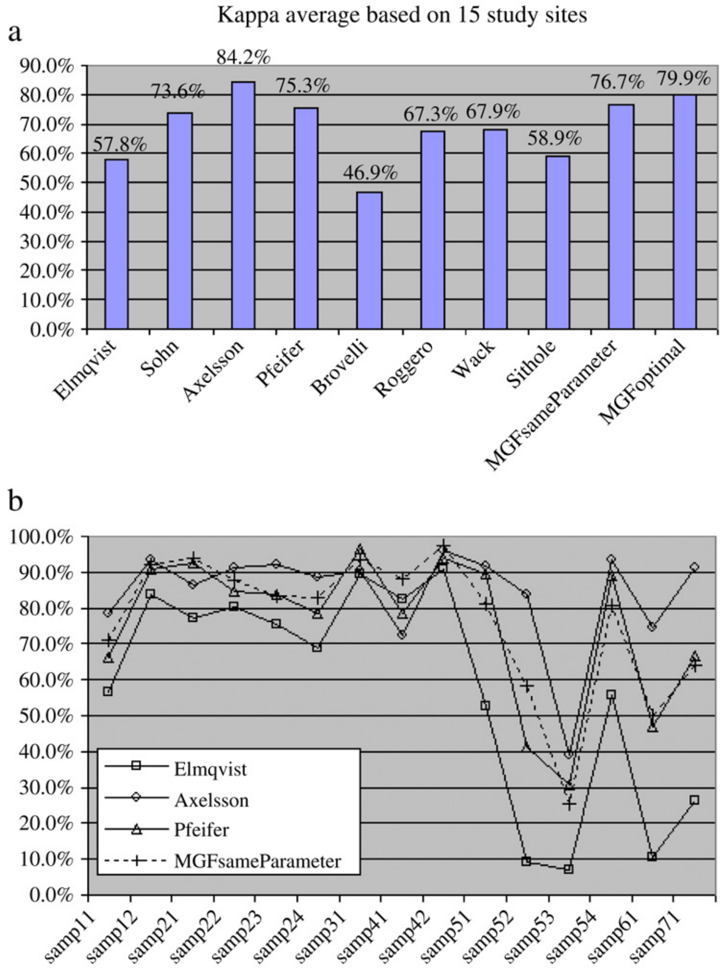


**Fig. 3.** Kappa averages on fifteen sites for nine filters (a) and Kappa values by sample sites for the MGF algorithm and three filters involved in the best performance (b).

**Table 2**
Parameters and kappa coefficients of the MGF algorithm

| Site | Pixel size | Slope | Elevation | Kappa |
|---|---|---|---|---|
| samp11 | 1 | 30 | 1.0 | 70.96 |
| samp12 | 1 | 30 | 1.0 | 92.28 |
| samp21 | 1 | 30 | 1.0 | 93.79 |
| samp22 | 1 | 30 | 1.0 | 87.83 |
| samp23 | 1 | 30 | 1.0 | 83.35 |
| samp24 | 1 | 30 | 1.0 | 82.83 |
| samp31 | 1 | 30 | 1.0 | 93.31 |
| samp41 | 1 | 30 | 1.0 | 88.27 |
| samp42 | 1 | 30 | 1.0 | 97.18 |
| samp51 | 2 | 60 | 2.0 | 81.18 |
| samp52 | 2 | 60 | 2.0 | 58.43 |
| samp53 | 2 | 60 | 2.0 | 25.60 |
| samp54 | 2 | 60 | 2.0 | 80.61 |
| samp61 | 2 | 60 | 2.0 | 50.16 |
| samp71 | 2 | 60 | 2.0 | 64.11 |

(2) Calculate the slope between the previous point in the scan line and this point.
   a. If the slope is greater than the slope threshold, label this point as a non-ground point.
   b. If the slope is positive and equal or less than the threshold, label this point with the same label as the previous point.
   c. If the slope is not available when there is no previous point, do nothing.
   d. If the slope is negative, proceed to the next step.

(3) Calculate the elevation difference between this point and its nearest ground point.

**Table 3**
Optimized parameters and Kappa coefficient for the MGF algorithm

| Site | Pixel size | Slope | Elevation difference | Kappa |
|---|---|---|---|---|
| samp11 | 1 | 60 | 1.0 | 70.96 |
| samp12 | 1 | 30 | 0.8 | 93.12 |
| samp21 | 1 | 45 | 0.9 | 95.40 |
| samp22 | 1 | 45 | 0.9 | 88.75 |
| samp23 | 1 | 30 | 1.6 | 87.56 |
| samp24 | 1 | 30 | 0.8 | 83.39 |
| samp31 | 1 | 30 | 0.5 | 97.45 |
| samp41 | 1 | 15 | 1.3 | 88.58 |
| samp42 | 1 | 30 | 1.1 | 97.25 |
| samp51 | 2 | 15 | 1.8 | 87.20 |
| samp52 | 2 | 30 | 2.7 | 65.57 |
| samp53 | 2 | 60 | 2.9 | 31.25 |
| samp54 | 2 | 30 | 0.9 | 92.71 |
| samp61 | 2 | 60 | 2.2 | 52.43 |
| samp71 | 2 | 45 | 1.3 | 67.36 |

**Table 4**
Kappa coefficients of the MGF algorithm on samp31 based on different window sizes

| Window size | Kappa coefficient |
|---|---|
| 3 | 97.45 |
| 5 | 96.45 |
| 7 | 95.01 |
| 9 | 93.83 |

   a. If the elevation difference to the nearest ground point is greater than the elevation threshold, label this point as a non-ground point.

**Fig. 4.** Error distribution for city sites 1–4. Each image is displayed at a unique scale.

   b. Otherwise, label this point as a non-ground point.

(4) Repeat steps 1–3 for each pixel in each scanning direction.

It is important to clarify that a previously ground-labeled point can change to non-ground if the slope or elevation difference along the current direction is larger than the threshold. Experiments prove that allowing status change generates higher performance. When searching for the nearest ground points, we target the smallest window that contains a ground point and then locate the nearest point to expedite the process.

## 4. Results and discussion

We apply the MGF algorithm to the fifteen urban and forest study sites provided by ISPRS and calculate the ground filtering accuracy using the Kappa Index of Agreement (Jensen, 2005). In the first section, we use identical slope and elevation threshold parameters for all urban sites and for all forest sites. We compare our results with eight other published ground filtering methods that were tested by ISPRS on the same datasets. In the second section, we test the sensitivity of our algorithm to the selection of slope and elevation thresholds by using an optimization process that incorporates ground truth data to determine the optimal thresholds for each site. The optimization demonstrates shows the potential performance of the MGF algorithm given well selected thresholds.

### *4.1. Comparative algorithm performance*

The fifteen study sites are subsets from two larger sites provided by ISPRS to generate ground truth for ground filtering algorithms. The nine urban sites (samp11 to samp42) are relatively flat with few steep slopes. We use 30° and 1 m for the slope and