

A Multi-Pass Generation of DEM for Urban Planning

¹Zheng Cui, ²Keqi Zhang, ³Chengcui Zhang, ¹Shu-Ching Chen

¹School of Computing and Information Sciences, Florida International University, USA

²Department of Earth and Environment, Florida International University, USA

³Department of Computer and Information Sciences, The University of Alabama at Birmingham, USA

^{1,2}{cuiz, zhangk, chens}@fiu.edu

³zhang@cis.uab.edu

Abstract—High quality DEM (Digital Elevation Model) is indispensable for a smart city nowadays. A large quantity of useful spatial information can be extracted from the big data acquired by surveying, imaging, and even mobile device with GPS system. This information is critical to many applications such as urban management, emergency event planning, and social location services. The extraction of non-ground features such as buildings, constructions and vegetation can be used to generate accurate boundary shape and three dimensional models. These outcomes can be utilized to provide useful information for the location service for mobile social networks. The DEM can be generated by surveying the area. LIDAR (Light Detection and Ranging) is a widely used remote sensing technology to survey the urban area. High resolution three dimensional data can be acquired from the LIDAR survey. How to separate the terrain surface and non-terrain features from three dimensional spatial data is the most challenging task for generating high quality DEM. In this paper, a multi-pass morphological filter is proposed to generate better DEMs on complex urban areas that have difficulties to use large filtering window sizes. Multi-pass filtering will use relatively small filtering window sizes in multiple rounds to avoid the cut-off problem commonly present when large window sizes are used in filtering. A priority boundary interpolation method is proposed and dedicated to interpolate the intermediate filtering result between each pass. A second pass of the morphological filter will use filtering window sizes in descending order to remove the remaining areas from large non-ground objects that cannot be removed with small window sizes. The filtering results demonstrate that the proposed method is able to effectively use relatively small filtering windows to filter large non-ground objects on complex terrains and avoid the cut-off problem.

Keywords—LIDAR; Digital Elevation Model (DEM); data filtering; urban planning.

I. INTRODUCTION

As LIDAR (Light Detection and Ranging) has become one of the most effective and reliable technology on surveying and digital elevation model (DEM) generation in recent years, many filtering methods have been developed to separate ground and non-ground features. The filtering results, such as terrain surfaces and building boundaries, are critical in many smart city applications including urban planning, emergency event planning, and social location services. For example, the flood inundation modeling in an

urban area requires correct terrain surface elevation information and building boundaries information [1]. Since LIDAR technology provides a way to acquire accurate three-dimensional information, there is a demand for high quality LIDAR filters for extracting terrain and non-terrain features and generating elevation surfaces and boundary information including building boundary polygons which can be used in commercial software, such as ArcGIS, or overlaid on web mapping service applications, e.g. Google Map. Fig. 1 shows the non-ground objects points from the filtering result and boundary polygons identified from these points. The boundaries of non-ground features such as buildings, constructions, and vegetation are very useful to provide accurate location services which would benefit the mobile users to get location information for their social network. Therefore, the quality of separation of ground and non-ground data points are critical for achieving accurate terrain and non-terrain features' boundaries. Many filtering methods have been developed to carry out this separation process. Among these methods, the mathematical morphological filtering method is one of the most widely used techniques in many filters [2][3][4][5] to generate high quality DEM. These DEMs can be further refined to suit many urban planning and management applications [6].

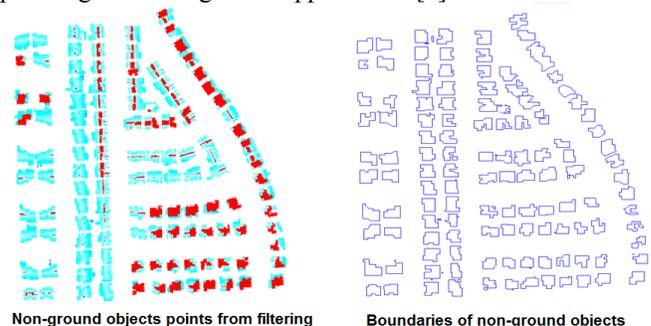


Fig. 1. Non-ground objects points and boundaries of non-ground objects

Among all the filtering methods, the progressive morphological filtering [5] is a very effective and efficient method to separate ground and non-ground objects from LIDAR data. Zhang et al. [5] use a moving filter window from small scale to large scale to filter out non-ground features. The main operation in this progressive morphological filter is the *open* operation. The elevation difference between the surface generated before and after the *open* operation would be compared with a predefined

threshold under each filtering window size. A point would be removed if its elevation difference is over the threshold. The progressive morphological filter is a very effective filtering method on many terrain types, especially on flat terrains. However, a common problem, called cut-off problem, would occur for morphological filters when the filtering window size is increased to a certain level. This is an intrinsic problem from the morphological filtering methodology. Even on some flat terrains, the cut-off problem could happen when the data set contains some large non-ground features or relatively low ground features. In order to avoid the cut-off problem on this kind of terrain, the window size used must be kept sufficiently small. However, this could cause incomplete filtering of large non-ground objects. To balance between over-filtering and under-filtering, a multi-pass filter is proposed to remove non-ground features in multiple rounds under relatively small window sizes so as to prevent the ground points being removed after the filtering window size is increased to a certain level. A sophisticated interpolation method, called priority boundary interpolation, is proposed to interpolate the intermediate result for further filtering of the remains of large non-ground features. The experiments show that the proposed method can achieve promising results, while the morphological filter itself has difficulty on these complex terrain data sets [7][8].

II. A MULTI-PASS MORPHOLOGICAL FILTER

In order to avoid the cut-off problem on some complex terrains, only relatively small filtering window sizes can be used. Otherwise, the cut-off problem is almost inevitable. As shown in “in press” [9], even with some local spatial smoothness justification “in press” [9], the morphological filter could still cause the cut-off problem on some complex terrains. In this case, a smaller filtering window size has to be used. However, it could cause incomplete filtering of some large non-ground features. Therefore, a multi-pass morphological filtering strategy is proposed in this paper to utilize relatively small window sizes to filter the data in multiple rounds so as to avoid the cut-off problem. And a new interpolation method is proposed to interpolate the intermediate filtering result for a second pass of filtering in order to remove the remaining large non-ground features that cannot be removed with small window sizes.

The general idea of this filtering process is as follows. First, the progressive morphological filter is used to filter the data set in the first pass with relatively small filtering window sizes, which could prevent some ground terrain points from being removed under large filter window size. Second, the priority boundary interpolation is performed to interpolate the empty grids that represent the filtered non-ground areas surrounding the remaining areas of large non-ground objects. Then, the second pass of morphological filtering will use a series of filtering windows from large to small sizes for removing the remaining large non-ground objects. If there are still non-ground features from large non-ground objects, this second pass filtering might be carried out until the remaining ground features are completely removed.

A. Multi-Pass Filtering

Since the proposed multi-pass filter utilizes the progressive morphological filtering [5] in each filtering pass which is based on grid data set, the input data set need to be gridded before processing. A representative point is chosen from the points within each grid, and the filter treats each grid as one data point. The filtering status of all the grids can be represented by a mark matrix. Each unfiltered point grid’s status will be initialized as either a real unfiltered point from the data set, or an interpolated unfiltered point. The general procedure of the proposed multi-pass filter is as follows.

1. $outputData \leftarrow MorphFilter(inputData, window_size_set1, threshold_set1);$
2. $interpData \leftarrow PQBoundInterp(outputData);$
3. $outputData \leftarrow MorphFilter(interpData, window_size_set2, threshold2);$

In Step 1, the procedure MorphFilter will carry out the progressive morphological filtering [5] on the input data with a series of window sizes and thresholds. The filtering windows are from small to large sizes, in ascending order. The elevation difference threshold series is pre-defined by the user. In order to prevent large ground features from being removed, the other critical parameter, the maximum filtering window size has to be limited to a certain level, which is determined by the ground terrain characteristics. A half window size is used as each step’s filtering window size parameter. The half window size is the number of grids extending to the left or right from the current grid. Commonly used maximum half filtering window sizes for 1 meter cell-size grid data set are usually less than 20, and thus the full window sizes are less than 41. The maximum filtering window size is determined by the resolution of the grid data set and terrain features. Fig. 2 shows the original complex data set with varying sizes of non-ground features (red areas). Since there are several large size buildings in the data set, only the use of a sufficiently large filtering window can completely remove them. However, a large filtering window could remove some ground surface points as well in areas not so flat or in cases of low elevation artificial constructions (e.g., channels). In order to avoid the cut-off problem when using the morphological filter, the filtering window sizes have to be limited under a certain level. Fig. 3 shows the filtering result when using the maximum filtering window size 20 on a 1-meter cell-size grid data. The result shows that there are a substantial amount of non-ground points from large buildings failed to be removed after Step 1, shown as the red areas in the green rectangles. Fig. 4 shows the filtering result when using the maximum filtering window size 80 on a 1-meter cell-size grid data. The result shows that it has significant cut-off problem in red rectangle area caused by large filtering window size. To avoid the cut-off problem under large filtering window, relatively small filtering window need to be used. However, the filtering remaining from large non-ground objects need to be handled.

In order to further remove the remaining non-ground features, a second pass filtering has to be carried out. However, the empty grids resulted from the removed areas

from Step 1 have to be interpolated, before the second pass filtering can be carried out. Since the goal of the second pass filtering is mainly to remove those large non-ground features, a good interpolation method is critical to the filtering. In the first pass filtering, a regular progressive morphological filtering is used in which the nearest neighbor interpolation method is used before filtering is applied. However, using the brute force nearest neighbor interpolation is likely to enlarge the remaining non-ground object area significantly and unnecessarily. Fig. 5 shows the result of directly applying the nearest neighbor interpolation without any constraint. As can be gleaned from this figure, the high elevation points from the boundary of remaining non-ground objects are undesirably involved in the interpolation. Consequently, in the interpolation result, the sizes of almost all the non-ground objects are unnecessarily enlarged in a significant way, causing potential cut-off problems to the second pass filter. Therefore, we must minimize the participation of high points in the interpolation, and use as many low elevation points in vicinity to interpolate empty grids as possible. A more sophisticated and dedicated interpolation method (in Step 2) is necessary for the second pass filtering (in Step 3). This interpolation and second pass filtering can be carried out repeatedly until all the non-ground features were successfully removed.

B. Priority Boundary Interpolation Method

In this paper, a new interpolation method called priority boundary interpolation method is proposed to interpolate the empty grids after the first pass morphological filtering. These empty grids will be interpolated with as many surrounding ground points as possible. In this way, the remaining non-ground features would be much easier to be removed from this interpolation result, because the elevation difference is more evident.

The essential idea of this priority boundary interpolation method is as follows. First, the empty grids are grouped into a set of non-overlapping connected grid sets by expanding connected empty grids. Second, each of these empty grid sets will be interpolated using the aforementioned method. For each empty grid set, its outside and/or inside boundary grids are first identified. The boundary grids of an empty grid set are shown in Fig. 6. The outside boundary grids are along the outside orange rectangle, and the inside boundary grids are along the inner orange polygon that goes side by side with the remaining non-ground area(s) (in red color) completely enclosed by that empty grid set. These outside and inside boundary grids will determine the interpolation result of the empty grids between them. Then the empty grid set can be interpolated according to the defined priority rule. The priority rule defines in what order the interpolation is carried out. The priority rule reflects the purposes of the interpolation for the specific filtering methods. For example, if the goal is to interpolate empty grids as low as possible, using the elevations of nearby ground points, the boundary points can be stored in an ordered queue according to the ascending elevation, and the empty grids can be interpolated along this order. The newly interpolated empty grids would

be treated as the new boundary points, because they formed the new boundary of the empty grid set.

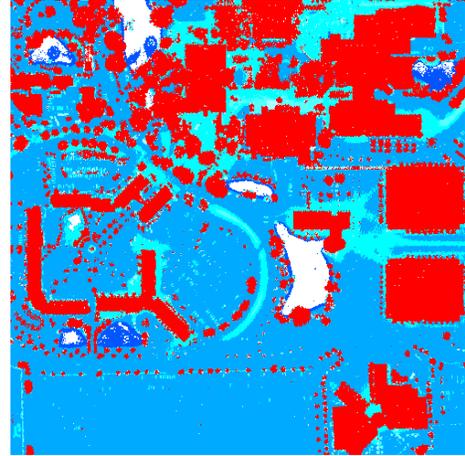


Fig. 2. The original data set

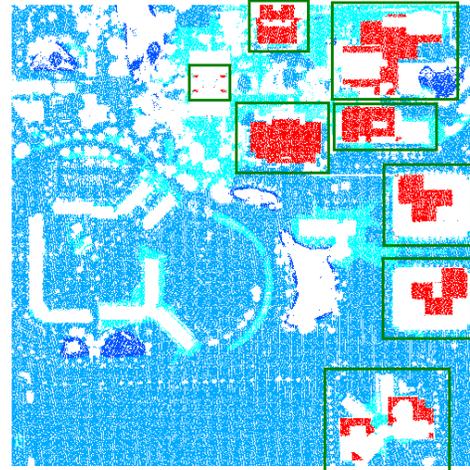


Fig. 3. The first pass filtering result

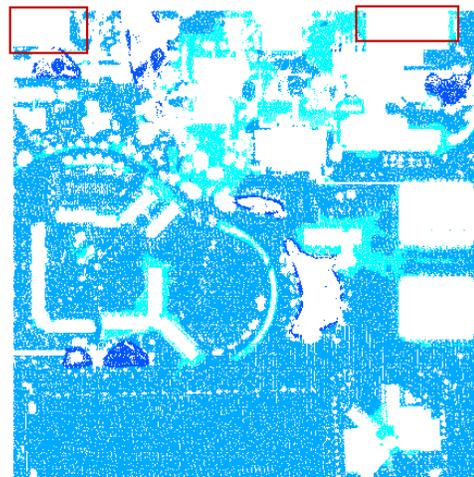


Fig. 4. Filtering result with cut-off problem caused by large filtering window

Each interpolated point will be assigned a priority value based on its elevation, and inserted into the corresponding

place in the priority queue. The used boundary points will be removed from the priority queue, and not be used in the future interpolation. By this means, the boundary points of the current empty grids set can be maintained in the priority queue until all the empty grids are interpolated. More algorithm details will follow.

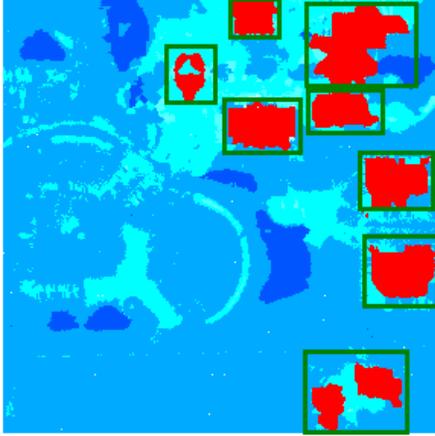


Fig. 5. The result of applying the nearest neighbor interpolation

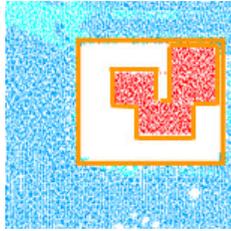


Fig. 6. Interpolation Boundary

During each step of progressive morphological filtering in the first and second pass, a moving window will be used to filter non-ground objects. A point classified as a non-ground point will be marked with the current filtering window size value. A mark matrix is used to record the filtering status for data grids. Each grid point's filtering status will be updated and stored in the mark matrix during filtering. The initial mark matrix contains only two kinds of values which indicate a real unfiltered data point or an empty grid, respectively. After interpolation, the empty grids will be interpolated by some method, and their mark values will be changed to "interpolated". Once a point is filtered, its mark value will be changed to the corresponding size of the filtering window under which it is filtered. At the end of the filtering process, those points with mark values indicating 'a real data point' or "interpolated" represent the points classified as true ground points. In other words, those points survive the filtering of non-ground objects. The algorithm description of this proposed priority boundary interpolation is as follows.

Algorithm description: *PQBoundInterp*

INPUT:

1. An array of input data points: *pts*
2. Minimum *x, y* coordinates: *min_x, min_y*
3. Grid size: *cellsize*

4. Threshold for the minimum number of connected empty grids in an empty grid set: *minConn*

OUTPUT:

An interpolated array of points: *interpPts*

1. $grdPts \leftarrow GridPts(pts, min_x, min_y, cellsize)$ // Gridded data set
2. Initialize *mark* matrix
3. $connMark \leftarrow findEmpConn(mark, pts)$; // Find connected empty grid sets
4. for $i = 1 \rightarrow size(connMark)$
5. $[row\ col] \leftarrow find(connMark == i)$; // Find the indexes of all the empty grids in Set *i*
6. If $size([row\ col]) < minConn$, then continue;
7. $edgeIndex \leftarrow findBoundGrid(mark, [row\ col])$; // Find the location indexes of boundary grids
8. $pq \leftarrow CreatePQ(edgeIndex)$; // Push each boundary grid into a priority queue based on the *z* value; lower *z* values have higher priorities
9. while the *pq* is not empty
10. $edge \leftarrow PopPQ(pq)$; // Pop the first element in the priority queue
11. $NBGrids \leftarrow interpolateNB(edge)$; // Interpolate its neighboring empty grids using the average value from adjacent non-empty grids and store the interpolated grids in *NBGrids*
12. $PushPQ(NBGrids, pq)$; // Push interpolated grids into the boundary grid priority queue
13. end while
14. end of for loop of all steps

Algorithm description: *findEmpConn*

INPUT:

Mark matrix: *mark*

An array of input data points: *pts*

OUTPUT:

Mark matrix for empty grid sets: *connMark*

1. Initialize *connMark* matrix with Unprocessed marks ("0") // *connMark* matrix has the same size as *mark* matrix.
2. $setID = 0$; // Initialize the empty grid set ID
3. $(rows, cols) \leftarrow size(mark)$;
4. for $i = 1 \rightarrow rows$
5. for $j = 1 \rightarrow cols$
6. if (*mark*(*i,j*) is an empty grid && *connMark*(*i,j*) is an unprocessed grid)
7. $setID \leftarrow setID + 1$;
8. $ptsQueue \leftarrow push(pts(i,j))$;
9. while (*ptsQueue* is not empty)
10. $pts(i,j) \leftarrow pop(ptsQueue)$;
11. $connMark(i,j) = setID$;
12. for $pts(m,n)$ in 4 or 8 neighboring grids of $pts(i,j)$
13. if (*mark*(*m,n*) is an empty grid && *connMark*(*m,n*) is an unprocessed grid)
14. $ptsQueue \leftarrow push(pts(m,n))$;
15. end if

```

16.         end for
17.     end while
18. end if
19. end for j
20. end for i

```

As in the algorithm description of *PQBoundInterp*, the input data will be gridded first with predefined minimum x, y coordinates and the cell-size (Line 1). The mark matrix is initialized with “real unfiltered data point” or “empty grid” marks. In Line 3, *findEmpConn* procedure will find all the empty grids and separate them into a set of non-overlapping connected grid sets. As long as one empty grid has at least one empty neighboring grid (either in a four or an eight neighboring grids), the current empty grid set will be expanded until no more connecting empty grids can be found. When checking connecting grids, four (only top, bottom, left, and right) or eight neighboring grids will be searched. In our own experience, searching four neighboring grids could sometimes avoid the inclusion of some unrelated empty grids in the expanding process. The *findEmpConn* procedure uses a *connMark* matrix to label each grid’s property. *connMark* has the same dimensions as that of *mark* matrix. All the grids are initially labeled as “0” (meaning unprocessed grids) in *connMark* matrix, while each empty grid will later be labeled with its set ID.

All the connected empty grids will be collected into the same set and assigned a unique ID number. After this empty grid set searching step, each empty grid will be assigned a positive group ID number and stored in the mark matrix *connMark*. Grids with the same ID number form an empty grid set. The non-empty grids in the *connMark* will still have their initial value “0”. Each empty grid set can be easily extracted by its set ID number. In Line 6, the total number of empty grids in each set is compared with a predefined threshold *minConn* to determine whether this empty grids set needs to be interpolated. User can use this threshold to eliminate those empty grid sets that are too small. If the *minConn* is 0, it will interpolate all the empty grid sets. Only the sets in which the number of connected empty grids is great than the minimum threshold will be interpolated. This would save computing time for interpolating small empty grid sets which are formed by filtering small non-ground objects in the first pass. These areas are not the targets for filtering in the second pass.

The procedure of the interpolation is illustrated in Fig. 7. The orange grids are the outside and inside boundary grids, while the white grids are empty grids. The blue grid is the point grid that has the lowest elevation among all the initial boundary grids. The red grids and the inside boundary grids belong to some remaining non-ground object that have a relatively high elevation and are surrounded by empty grids. The green grids are the empty grids adjacent to the blue grid. Starting from the blue grid, the two adjacent empty grids in green color can be interpolated, using the average elevation of its neighboring boundary grids; then the blue grid will be removed from the priority queue and the two interpolated grids will be pushed into the priority queue as the new boundary grids. The grid that has the current lowest elevation will be continuously popped up from the boundary grid

priority queue, and the above procedure will be repeated until there is no boundary grid left in the priority queue. At the end of each round, the newly interpolated empty grids will be treated as the new boundary grids which form new boundary lines of the remaining empty grids which are yet to be interpolated. By this means, empty grids will be interpolated from the neighbors of the lowest elevation grid to the highest, thus largely limiting the negative impact of the high elevation boundary points in the remaining non-ground objects (e. g., the inside boundary points in Fig. 7) to its adjacent grids only. If the empty grids near the remaining non-ground object have already been interpolated by the lower neighboring grids in the priority queue, the higher non-ground object points will not be used for interpolation at all. This way we can minimize the unnecessary expansion of non-ground objects, and consequently, the maximum filtering window size used in the second pass can be kept as small as possible. A sample priority boundary interpolation result is shown in Fig. 8. The result shows that the empty grids around the filtering remains have been interpolated with low ground points desirably. The areas in the green rectangles demonstrate that the interpolated empty grids tightly surround the remaining non-ground features output from the first pass filtering. This interpolation result greatly benefits the second pass filtering.

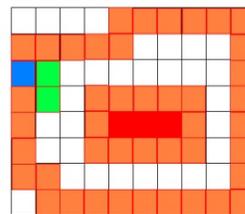


Fig. 7. Priority boundary interpolation illustration

C. The Second Pass Filtering

The second pass filtering in the proposed filtering strategy is also based on the progressive morphological filter. The filtering window sizes and thresholds are different from that of the first pass morphological filter. Since the targets of the second pass filtering are the remaining points from the large non-ground objects, the filtering window sizes are applied in the descending order. The starting filtering window size is normally the same as the last filtering window size in the first pass, which is also the largest filtering window size in the first pass. Then, the window size is reduced and another round of filtering is performed. The elevation difference threshold for each window size would be relatively large, because it is used for filtering out the remaining large non-ground objects. The elevation thresholds for this purpose are normally much higher than that of each window size in the first pass. The second pass filtering result is shown in Fig. 9. The result shows that almost all the remaining large non-ground objects have been successfully removed from the interpolation result shown in Fig. 8. The filter window sizes series is 20, 16, 8, 4, 2, and 1. The elevation difference threshold used in all rounds is 2.0 meters.

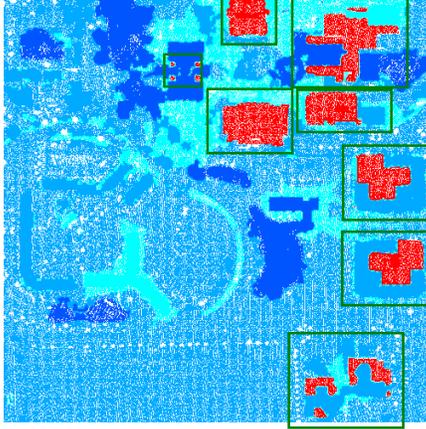


Fig. 8. Priority boundary interpolation result

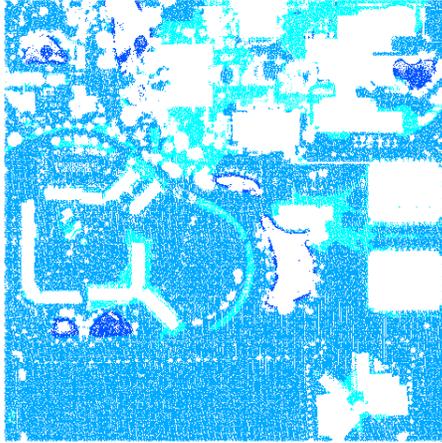


Fig. 9. The second pass filtering result

D. Computational Complexity

Based on the algorithm description of the proposed multi-pass morphological filter, the time complexity of the algorithm should be the same as that of the progressive morphological filter. Since the proposed filter is based on the progressive morphological filter, the major computation time for the progressive morphological filter is the *open* operation in addition to the interpolation. The time complexity for the *open* operation is $O(wN)$, where w is the window size of the morphological filter and N is the number of grids which is the product of the number of rows and columns. For M windows, the time complexity is the same as [5].

$$O\left(\sum_{k=1}^M w_k N\right)$$

The time complexity of generating a priority queue initially is $O(N)$ in the priority boundary interpolation, and the time complexity of insertion and removal in the priority queue are $O(\log N)$. The time complexity of searching all the connecting empty grids is $O(N)$. Therefore, The time complexity of the priority boundary interpolation is $O(N)$. Further, the overall time complexity of the proposed multi-pass filter has the same time complexity as that of the progressive morphological filter.

III. CONCLUSIONS

Experiments results show that the proposed multi-pass morphological filter is able to take advantage of the progressive morphological filter when filtering small to intermediate size non-ground objects. It also avoids using large filtering window on complex terrains with relatively large non-ground objects, especially large buildings, which can prevent the ground points from being removed under large filtering windows. The dedicated priority boundary interpolation method provides a desirable way to interpolate the filtering output from the first pass filtering and make the second pass filtering effective and efficient, avoiding the need for a big window size. The proposed method will provide more accurate filtering results in urban areas with complex non-ground features. It will benefit the applications “in press” [10] of urban planning and social location services with higher quality terrain surface and boundary information.

REFERENCES

- [1] G. Priestnall, J. Jaafar, and A. Duncan, “Extracting urban features from LiDAR digital surface models,” *Computers, Environment and Urban Systems*, 24 (2), pp. 65–78, 2000.
- [2] J. Kilian, N. Haala, and M. Englich, “Capture and evaluation of airborne laser scanner data,” *International Archives of Photogrammetry and Remote Sensing*, Vol. XXXI, Part B3, pp. 383–388, 1996.
- [3] B. Petzold, P. Reiss, and W. Stossel, “Laser scanning-surveying and mapping agencies are using a new technique for the derivation of digital terrain models,” *ISPRS J. Photogrammetry and Remote Sensing*, 54.2, pp. 95-104, 1999.
- [4] M. Morgan, and K. Tempfli, “Automatic building extraction from airborne laser scanning data,” *International Archives of Photogrammetry and Remote Sensing*. Amsterdam, vol. 33(B3): pp. 616-623, 2000.
- [5] K. Zhang, S.-C. Chen, D. Whitman, M.-L. Shyu, J. Yan, and C. Zhang, “A progressive morphological filter for removing nonground measurements from airborne LiDAR data,” *IEEE Transactions on Geoscience and Remote Sensing*, vol. 41, 4, pp. 872-882, April, 2003.
- [6] A. Elaksher, and J. Bethel, “Refinement of digital elevation models in urban areas using breaklines via a multi-photo least squares matching algorithm,” *Journal of Terrestrial Observation*: vol. 2: Iss. 2, Article 7, 2010.
- [7] X. Meng, N. Currit, and K. Zhao, “Ground filtering algorithms for airborne LiDAR data: a review of critical issues,” *Remote Sensing*, 2, 3, 833-860, March, 2010.
- [8] X. Liu, “Airborne LIDAR for DEM generation: some critical issues,” *Progress in Physical Geography*, 32, 1, 31-49, February, 2008.
- [9] Z. Cui, K. Zhang, C. Zhang, and S.-C. Chen, “A cluster-based morphological filter for geospatial data analysis,” accepted for publication, 2nd ACM SIGSPATIAL International Workshop on Analytics for Big Geospatial Data, in conjunction with 21st ACM SIGSPATIAL International Conference on Advances in GIS, Orlando, Florida, USA, November 5-8, 2013.
- [10] Z. Cui, K. Zhang, C. Zhang, J. Yan, and S.-C. Chen, “A GUI based LIDAR data processing system for model generation and mapping,” accepted for publication, 1st ACM SIGSPATIAL International Workshop on Interacting with Maps (MapInteract) 2013, in conjunction with 21st ACM SIGSPATIAL International Conference on Advances in GIS, Orlando, Florida, USA, November 5-8, 2013.