Evaluation and Improvements on Row-Column Order Bias and Grid Orientation Bias of the Progressive Morphological Filter of Lidar Data

Kody Potter
Utah State University

Follow this and additional works at: https://digitalcommons.usu.edu/etd

Part of the Civil Engineering Commons

Recommended Citation
Potter, Kody, "Evaluation and Improvements on Row-Column Order Bias and Grid Orientation Bias of the Progressive Morphological Filter of Lidar Data" (2011). All Graduate Theses and Dissertations. 882.
https://digitalcommons.usu.edu/etd/882

This Thesis is brought to you for free and open access by the Graduate Studies at DigitalCommons@USU. It has been accepted for inclusion in All Graduate Theses and Dissertations by an authorized administrator of DigitalCommons@USU. For more information, please contact digitalcommons@usu.edu.
EVALUATION AND IMPROVEMENTS ON ROW-COLUMN ORDER BIAS AND GRID ORIENTATION BIAS OF THE PROGRESSIVE MORPHOLOGICAL FILTER OF LIDAR DATA

by

Kody Potter

A thesis submitted in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE in

Civil and Environmental Engineering

Approved:

_________________________  ________________________
Robert T. Pack            David G. Tarboton
Major Professor           Committee Member

_________________________  ________________________
James A. Bay              Byron R. Burnham
Committee Member          Dean of Graduate Studies

UTAH STATE UNIVERSITY
Logan, Utah

2011
ABSTRACT

Evaluation and Improvements on Row-Column Order Bias and Grid Orientation Bias of the Progressive Morphological Filter of Lidar Data

by

Kody Potter, Master of Science
Utah State University, 2011

Major Professor: Dr. Robert T. Pack
Department: Civil and Environmental Engineering

This thesis reviews algorithms that have been developed for classifying lidar data and identifies a progressive morphological filter for evaluation and improvement. Two potential weaknesses evaluated include the row-column order bias and grid orientation bias.

Four different row-column orderings were developed to test for bias associated with the order choice. Moreover, a method rotating the filter grid to a series of angles was developed for testing bias associated with grid orientation. Measures of success of the improvements include Type I and II errors, where results are compared with a hand-produced “truth” dataset. Two datasets, one urban, the other rural, were selected for testing the modified filters. The results are presented and discussed for each algorithm.

It was found that the four row-column orders all classified the dataset exactly the same. After the erosion and dilation functions were completed, the same surface profiles
and elevations were produced regardless of row-column ordering. The filter windows used by the algorithm were found to create a rectangular area where the minimum and maximum values within that area were always selected. Therefore, it was found that the row-column orders did not create a bias in the classification.

However, grid orientation was found to greatly affect results. Misclassification problems occurred at ridgelines, mounds, and along roads with ditches and steep slopes running along them. Grid angles running parallel to these objects were found to avoid these errors. Buildings also created errors, but were minimized with grid angles crossing them at 45 degrees. The selected angle directions significantly affect the classification results in all cases. Therefore, the grid orientation bias was verified.

Two new methods of combining the results from the various angles have been developed. The first method used the best two classifying angles to combine the results. Best results were found in datasets with terrain objects positioned in similar directions for this method. The Multiple Angle method used all of the angle classifications to combine the results. This method performed best on datasets with terrain objects oriented in numerous directions. More accurate terrain models and better overall classification results have been generated using these methods.

(114 pages)
ACKNOWLEDGMENTS

I would like to give special thanks to my wonderful wife, Marcia, for her encouragement to select this thesis path, along with her support and patience through it all. I couldn’t and wouldn’t have done it without you; love you and thanks for everything. Also like to thank family, friends, and classmates for their support and friendship through this college experience and in providing me with the opportunities and means I have had to get to this point.

I wish to thank my major professor, Dr. Robert Pack, for the opportunity of performing this thesis research. I’d also like to thank him for his friendship, guidance, and suggestions, and for being great to work with through this process.

I would also like to thank my committee members, Dr. David G. Tarboton and Dr. James A. Bay, for their help, support, and suggestions that improved the depth of my research and success.

I would also like to thank the USTAR LASSI Service Center in providing me with the Lidar datasets in which the algorithms were tested.

Kody Potter
# CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>ABSTRACT</td>
<td>iii</td>
</tr>
<tr>
<td>ACKNOWLEDGMENTS</td>
<td>v</td>
</tr>
<tr>
<td>LIST OF TABLES</td>
<td>viii</td>
</tr>
<tr>
<td>LIST OF FIGURES</td>
<td>ix</td>
</tr>
<tr>
<td>INTRODUCTION</td>
<td>1</td>
</tr>
<tr>
<td>REVIEW OF EXISTING LIDAR CLASSIFICATION FILTERS</td>
<td>3</td>
</tr>
<tr>
<td>Linear Prediction Filters</td>
<td>3</td>
</tr>
<tr>
<td>Slope-Based Filters</td>
<td>5</td>
</tr>
<tr>
<td>Segmentation Filters</td>
<td>7</td>
</tr>
<tr>
<td>Morphological Filters</td>
<td>10</td>
</tr>
<tr>
<td>Filters Implemented by NCALM</td>
<td>15</td>
</tr>
<tr>
<td>OBJECTIVES</td>
<td>16</td>
</tr>
<tr>
<td>ALGORITHM CHANGES</td>
<td>19</td>
</tr>
<tr>
<td>Existing Algorithm</td>
<td>19</td>
</tr>
<tr>
<td>Row-Column Order Modifications</td>
<td>24</td>
</tr>
<tr>
<td>Grid Rotation Modifications</td>
<td>25</td>
</tr>
<tr>
<td>EXPERIMENTAL SETUP</td>
<td>29</td>
</tr>
<tr>
<td>Methodology</td>
<td>29</td>
</tr>
<tr>
<td>Row-column order</td>
<td>33</td>
</tr>
<tr>
<td>Grid rotation</td>
<td>34</td>
</tr>
<tr>
<td>Test Datasets</td>
<td>35</td>
</tr>
<tr>
<td>RESULTS AND DISCUSSION</td>
<td>39</td>
</tr>
<tr>
<td>Input Parameter Selection</td>
<td>39</td>
</tr>
<tr>
<td>Daniel Forest parameters</td>
<td>43</td>
</tr>
</tbody>
</table>
USU Campus parameters .............................................................................................................. 46

Results of Row-Column Order Analysis .................................................................................. 51
Results of Daniel Forest Grid Rotation Analysis .................................................................... 55
Development of Methods to Remove Grid Bias ...................................................................... 63

Description of methods ............................................................................................................. 63
Results ........................................................................................................................................ 68
USU Campus ............................................................................................................................. 78

CONCLUSIONS .......................................................................................................................... 93

RECOMMENDATIONS .............................................................................................................. 97

REFERENCES ............................................................................................................................ 100
## LIST OF TABLES

<table>
<thead>
<tr>
<th>Table</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Input parameter summary of the existing progressive morphological filter with a basic description of how to select each parameter</td>
<td>23</td>
</tr>
<tr>
<td>2</td>
<td>Type I and II error analysis with the algorithm classification at the top right side and the actual or true point classification at the bottom left side</td>
<td>32</td>
</tr>
<tr>
<td>3</td>
<td>Central values and sensitivity ranks for the Daniel Forest and USU Campus datasets with 1= Most Sensitive and 6=Least Sensitive</td>
<td>44</td>
</tr>
<tr>
<td>4</td>
<td>Selected input parameters from the sensitivity plots for both Daniel Forest and USU Campus datasets</td>
<td>50</td>
</tr>
<tr>
<td>5</td>
<td>Error analysis results of the four row-column orders of the Daniel Forest dataset</td>
<td>51</td>
</tr>
<tr>
<td>6</td>
<td>Comparison of the best two angles 240 and 320 degrees, Best Two Angle Both (BTAB), Best Two Angle Either (BTAE) and Multiple Angle (MA) combining methods for the Daniel Forest dataset</td>
<td>77</td>
</tr>
<tr>
<td>7</td>
<td>Comparison of the best two angles 230 and 315 degrees, Best Two Angle Both (BTAB), Best Two Angle Either (BTAE) and Multiple Angle (MA) combining methods for the USU Campus dataset</td>
<td>90</td>
</tr>
</tbody>
</table>
### LIST OF FIGURES

<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Existing progressive morphological filter flowchart.</td>
</tr>
<tr>
<td>2</td>
<td>The different four row-column orders tested with variables defined as follows: Row Erosion (RE), Column Erosion (CE), Row Dilation (RD) and Column Dilation (CD).</td>
</tr>
<tr>
<td>3</td>
<td>The grid rotation algorithm flowchart.</td>
</tr>
<tr>
<td>4</td>
<td>Edge misclassification error block and profile example.</td>
</tr>
<tr>
<td>5</td>
<td>Colored lidar point cloud of a portion of the T.W. Daniel Experimental Forest.</td>
</tr>
<tr>
<td>6</td>
<td>Colored lidar point cloud of a portion of the USU Campus.</td>
</tr>
<tr>
<td>7</td>
<td>Blocking of Daniel Forest dataset at 110 m block sizes.</td>
</tr>
<tr>
<td>8</td>
<td>USU Campus blocks at 80 m and 1.79 scaled buffer maps.</td>
</tr>
<tr>
<td>9</td>
<td>Daniel Forest 1.59 scaled buffer blocks map.</td>
</tr>
<tr>
<td>10</td>
<td>Sensitivity plot for Daniel Forest slope parameter with the black bar set to the “central value” of 0.2 and the red arrow set to optimum selected input parameter of 0.13.</td>
</tr>
<tr>
<td>11</td>
<td>Sensitivity plot for Daniel Forest maximum window size parameter with the black bar set to the “central value” of 20 m and the red arrow set to optimum selected input parameter of 35 m.</td>
</tr>
<tr>
<td>12</td>
<td>Sensitivity plot for Daniel Forest grid cell size parameter with the black bar set to the “central value” of 50 cm and the red arrow set to optimum selected input parameter of 26 cm.</td>
</tr>
<tr>
<td>13</td>
<td>Sensitivity plot for USU Campus slope parameter with the black bar set to the “central value” of 0.2 and the red arrow set to optimum selected input parameter of 0.25.</td>
</tr>
<tr>
<td>Page</td>
<td>Description</td>
</tr>
<tr>
<td>------</td>
<td>-------------</td>
</tr>
<tr>
<td>14</td>
<td>Sensitivity plot for USU Campus maximum window size parameter with the black bar set to the “central value” of 30 m and the red arrow set to optimum selected input parameter of 32 m. .......................... 48</td>
</tr>
<tr>
<td>15</td>
<td>Sensitivity plot for USU Campus base of exponential function parameter with the black bar set to the “central value” of 2 and the red arrow set to optimum selected input parameter of 3. ................................ 48</td>
</tr>
<tr>
<td>16</td>
<td>USU Campus sample area 2 sensitivity plot of maximum window size. Black bar set to the “central value” of 30 and the red arrow set to the optimum input parameter of 90. ................................................................. 50</td>
</tr>
<tr>
<td>17</td>
<td>Elevation tracking of a single lidar point after each function: Row Erosion (RE), Column Erosion (CE), Row Dilation (RD) and Column Dilation (CD). The four orders are shown in A through D. ............... 52</td>
</tr>
<tr>
<td>18</td>
<td>Order 1 profile tracking from the center of Block 13 in the Daniel Forest Dataset ........................................................................................................................................... 53</td>
</tr>
<tr>
<td>19</td>
<td>Order 2 profile tracking from the center of Block 13 in the Daniel Forest Dataset ........................................................................................................................................... 54</td>
</tr>
<tr>
<td>20</td>
<td>Order 3 profile tracking from the center of Block 13 in the Daniel Forest Dataset ........................................................................................................................................... 55</td>
</tr>
<tr>
<td>21</td>
<td>Order 4 profile tracking from the center of Block 13 in the Daniel Forest Dataset ........................................................................................................................................... 56</td>
</tr>
<tr>
<td>22</td>
<td>Error analysis graphs showing the effect of rotation on the A) Type I Errors, B) Type II Errors, C) Percent Correctly Classified and D) Weighted Scores from angles of 0 to 355 degrees for the Daniel Forest dataset. ............................................................................................................ 58</td>
</tr>
<tr>
<td>23</td>
<td>A ridgeline and road were found to have high numbers of Type I errors. ........................................................................................................................................................................ 59</td>
</tr>
<tr>
<td>24</td>
<td>Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis grid angle of 0 degrees azimuth in the Daniel Forest ........................................................................................................................................................................ 60</td>
</tr>
<tr>
<td>25</td>
<td>Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis grid angle of 120 degrees azimuth in the Daniel Forest ........................................................................................................................................................................ 61</td>
</tr>
</tbody>
</table>
Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis grid angle of 240 degrees azimuth in the Daniel Forest

Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis grid angle of 0 degrees azimuth in the Daniel Forest

Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis grid angle of 30 degrees azimuth in the Daniel Forest

Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis grid angle of 320 degrees azimuth in the Daniel Forest

Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis of the Best Two Angle Both (BTAB) combination method of the Daniel Forest

Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis of the Best Two Angle Either (BTAE) combination method of the Daniel Forest

Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis of the Multiple Angle (MA) combination method that uses 55% of angles from the Daniel Forest.

The Multiple Angle (MA) combination method error analysis graphs.
Error analysis graphs showing the effect of rotation on the A) Type I Errors, B) Type II Errors, C) Percent Correctly Classified and D) Weighted Scores from angles of 0 to 355 degrees for the USU Campus dataset. ........................................................................................................ 80

The location of the misclassification analyze for the USU Campus dataset is specified by the red box. ........................................................................................................ 81

Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis grid angle of 0 degrees azimuth in the USU Campus. ........................................................................................................ 82

Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis grid angle of 230 degrees azimuth in the USU Campus. ........................................................................................................ 83

Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis grid angle of 315 degrees azimuth in the USU Campus. ........................................................................................................ 84

Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis of the Best Two Angle Both (BTAB) combination method of the USU Campus. ........................................................................................................ 85

Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis of the Best Two Angle Either (BTAE) combination method of the USU Campus. ........................................................................................................ 86

The Multiple Angle (MA) combination method error analysis graphs. ........................................................................................................ 87

Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis of the Multiple Angle (MA) combination method that uses 80% of angles from the USU Campus. ........................................................................................................ 89
INTRODUCTION

Light Detection and Ranging (lidar) data is used in a variety of applications: mapping, forestry, floodPLAIN delineation, urban planning, and other earth-related sciences. Lidar datasets are made up of 3D point clouds that represent various types of objects. Often, it is of particular interest to classify ground and non-ground points. Non-ground points are defined as lidar points returned from vegetation, buildings, vehicles and any other points that do not represent the natural ground or bare-earth. Widely used Digital Terrain Models (DTMs) only include ground data points and therefore necessitate that the non-ground points be removed. A variety of filter algorithms have been developed to not only separate ground, but to also break the non-ground features into finer classifications. These filters have been developed to automate the process of developing DTMs as much as possible. Even with these filters there is still the need to manually check and edit the filtered data to correct misclassifications. Human editing of the data increases time and cost to the overall process. The better a filter algorithm can automatically classify the points the less human interaction it will take to develop an accurate DTM. Algorithms need to be improved to better automate this process.

The approach to this research was to first review the variety of filters that have been developed and then explore opportunities for improving weaknesses identified within the algorithm.

First, the three primary types of filters which include (l) linear prediction, (2) slope-based filters, and (3) segmentation filters will be reviewed. It will be shown that each type of filter has a set of strengths and weaknesses associated with it. Second, the
particular weaknesses of row-column orderings and grid orientation within the existing progressive morphological filter (Zhang et al., 2003) will be fully described and the ways and means to its improvement developed and discussed.
REVIEW OF EXISTING LIDAR CLASSIFICATION FILTERS

The purpose of lidar classification filters is to remove the non-ground lidar data points from the surface to generate a bare earth DTM. Moreover, they can be used to classify buildings, roads and structure within vegetation. These algorithms first started to be reported in the literature in the mid 1990’s and can be found as regular contributions to the literature ever since.

**Linear Prediction Filters**

Kraus and Pfeifer (1998) first developed the linear prediction algorithm that classified ground points using linear least squares interpolation in a rectangular grid system. This is done by estimating an average surface derived from all points in the lidar point cloud, including both ground and non-ground points. Negative and positive residuals from the estimated surface are then used to assign weights to the data points. The smaller the residual of a point, the higher the weight will be that is assigned to that point because it is more likely to be a ground point. Points within a given threshold using the weights are maintained. A new estimated surface is calculated and the process is repeated until the estimated surface does not change. Points contained in that surface are classified as ground. Schickler and Thorpe (2001) added a triangular irregular network (TIN) grid system for versatility in handling varying point densities. Break lines, curve and slope constraints were also added into the calculation of the weight values for the data points. Lee and Younan (2003) also used the original linear prediction algorithm from Kraus but added the comparison of the filtered points to the original point
measurements. Filtered points that matched the original point measurements were removed which removed spurious peaks and helped smooth the DTM.

An elevation threshold with an expanded window filter uses a linear prediction filter with a slight variation (Zhang and Whitman, 2005). A grid is developed into arrays of cells in rows and columns covering the dataset. The algorithm places the data points into the cells with minimum elevation points in each cell. The slope of each point is calculated according to the surrounding lidar points. Height differences between lidar points and its neighboring points are also calculated and used as parameters for classification. Points that do not meet the threshold values for slope and height difference are removed from the dataset. The remaining points are re-gridded with increased cell sizes and the process continues until there are no further points removed from the dataset. The ending group of points is used to create the DTM.

Zhang and Cui (2007) developed an iterative polynomial fitting filter for the National Center of Airborne Lidar Mapping (NCALM). First the data points are assigned to cells (about 2 meters) within a grid. The lowest points are selected from the arrays of cells within a large moving window and the points are used for an initial interpolated ground surface. The window size is then reduced and the lowest point within the window is selected as a candidate ground point. If the elevation difference between the point and the interpolated surface is below the threshold, it is classified as a ground point and added to the interpolated surface. The window size continues to decrease until it is smaller than the cell sizes. In the end, the interpolated surface is fitted to only the classified ground points.
The first of two polynomial surface filters used by Zhang and Cui (2007) recovers missed ground points. This is done by comparing the elevations of candidate point to the final interpolated surface. If the elevation difference is less than the specified threshold for the point, it will be classified as a ground point. Commission classification errors occur when non-ground points are misclassified as ground points by the algorithm. The second polynomial surface filter fixes these errors by comparing the new proposed surface with the previous surface to find the best fitting curve between the two. The curve is selected according the minimum distances between the two curves. When the proposed surface is selected this means missed ground points will have be added.

Slope-Based Filters

Slope-based filters use the basic idea that large height (elevation) differences between two points that are close together in the x and y directions (hence a higher “maximum local” slopes) imply that the higher point is a non-ground point and should be classified as such. Vosselman (2000) used this approach and found that prior knowledge of actual terrain slope is important for the setting of the maximum local slope parameter for the function. The parameters of distance between points and the angle between two points are also used. This maximum local slope filter has been implemented by Zhang and Cui (2007) for NCALM using 2D rectangular grid data structures.

Sithole (2001) modified Vosselman’s algorithm by adding a continuously changing maximum local slope threshold that adapts to the slope of the surrounding terrain. This enabled filter to adapt between flat and steep terrain within a dataset and
thereby improve the classification results. Meng (2005) added local regression and Shan and Sampath (2005) added varying height thresholds and extractors for noise errors to the algorithm. Most recently the “Climbing and Sliding” method by Shao and Chen (2008) was implemented using general, incremental and maximum slope parameters. These parameters take into account the terrain continuity, curvature on mountain tops and adding of objects. A back-selection step was also added to recover any missed ground points by reapplying the slope algorithm to points that are sometimes not searched in the initial processing. Thresholds are increased in the back-selection step to recover points along break lines in the dataset.

The commercial software package “Terrascan” (Soininen, 2010) has used a slope-based filter for many years. It estimates a triangulated irregular surface (TIN) model for the beginning ground surface. It then uses the parameters of maximum building size, terrain angle (maximum allowed ground terrain slope), iteration angle (maximum angle between points) and iteration distance (maximum distance between points) to classify ground points from the TIN.

An adaptive TIN filter was implemented by Zhang and Cui (2007). It modified Terrascan’s slope-based TIN filter (Soininen, 2010) by adding a “mirror point” parameter to the algorithm. This point is mirrored from the potential (candidate) ground point in question and the parameters of the slope, distance and angle are computed from the mirror point to the TIN surface. If the parameters used by Soininen (2010) of the mirror point are less than the thresholds, then the candidate point is classified as ground. The
addition of the mirror point helps in classifying the ground points in steep sloped areas by taking into account the terrain on the opposite side of the candidate point.

**Segmentation Filters**

With segmentation filters, lidar data points are first grouped together using a variety of techniques (segmentation). Similar groups are then pieced together into continuous segments. The continuous ground segments are then connected together to form a DTM. The process of starting with one segment and adding to it other segments is definite as a surface growing algorithm.

The initial segmentation step can be accomplished by statistical clustering algorithms. Filin (2002) and Lohmann (2002) first used attributes of slope and height difference to do the initial clustering and segmentation. Both attribute values were determined by using the neighboring points around the point of inspection. Filin (2004) then introduced the use of surface trend and surface curvature as the basis for segmentation. Filin and Pfeifer (2005, 2006) developed and implemented a slope adaptive neighborhood for extracting groups. In the meanwhile, Roggero (2002) used a surface growing algorithm which uses change of curvature for the basis of segmentation. Beginning at a point, the surface is grown until the curvature change exceeds a threshold. Nardinoocchi, Forlani, and Zingaretti (2003) also used a surface growing algorithm but added a point classification step after the data was segmented. This classification was based on the geometric and topological relationships of the points included in the segmented surface.
A scan line algorithm that bypasses the need for statistical clustering has been used by Sithole and Vosselman (2005). Thin pieces of the point clouds are defined as profiles and are used for this filter. Their algorithm groups the data by slicing the point cloud in different directions to come up with surface profiles which are then broken down into their component line segments. The line segments from all profiles are then combined by connecting them together in two dimensions to create surface segments. The surface segments are then connected together to develop the DTM. This is done based on the shape of the line segments, along with the fitting of a plane for every point (using its nearest neighboring points to fit the plane) and removing points from the ground segments if the standard deviation of the errors is greater than the set threshold.

Tovari and Pfeifer (2005) used the grouping of similar points based on slope and height as explained above to create segments. The segments were assigned a weight value based on the residuals from an estimated ground surface. The weight function is the same used by Kraus and Pfeifer (1998) in their linear prediction algorithm, but here Tovari applies it to segments instead of individual points. The function runs until the estimated surface stops changing significantly between each iteration and the resulting segments are classified as the overall ground surface. Jingue and Ming (2006) used an edge detection method to segment the ground data. Seed points are used as the beginning points in the surface growing algorithm. These points are assumed as ground points in the algorithm and other points are added to them to create segments. Therefore, it is very important to make sure these points are actual real ground points to create ground segments. These seed points were selected in the following way to begin the surface
growing algorithm. Outliers and vegetation points were removed from being selected as seed points by using a k-nearest neighbor method threshold and comparing the first and last lidar return pulses, respectively. The k-nearest neighbor method removes these points by classifying them in the same group as the greatest amount of neighboring points. The seed points for the surface growing algorithm were then selected from the remaining points and the ground surface was developed. Edges of buildings and breaklines points were selected and segments for buildings were also created.

Another approach used by Akel, Filin, and Doytsher (2007) used orthogonal polynomials to approximate the shape of the terrain and then classify (segment) the points according to the distance from that shape. This initial step is akin to a linear prediction approach. Results of the classification were then fine tuned with the use of a surface growing algorithm based on the normal direction of the TIN model to connect the segments and construct a DTM.

Most recently, Lu et al. (2008) used a TIN model of an entire dataset to segment the triangles of the TIN together, instead of the individual data points. The triangles are then classified according their “up angle.” This is the angle between a vector pointing to the sky and the normal vector of the triangle. Triangles with large and small “up angles” are classified as steep and flat, respectively. The triangles are then segmented into non-ground (steep triangles) and ground (flat triangles). The detecting of buildings is done by using a relative height difference between surrounding ground segments. Segments higher than the threshold are classified as buildings and removed. Trees are detected by segmenting the data according to a threshold for the triangles steepness of slope.
Triangles above the threshold for slope are classified as trees and removed. In the end the bare-earth triangulation is left.

**Morphological Filters**

Mathematical morphology is used for the filtering of lidar data by using erosion and dilation operations. The operations of erosion and dilation find the minimum and maximum elevation points within a window, respectively. In this context, a window is defined as either a one-dimensional (1D) interval on a line or a two-dimensional (2D) rectangle on a surface. With grid data structures, the line is a row or column in the grid. The window is successively centered at each cell in the grid dataset and the data within the surrounding window is analyzed. The erosion operation selects the lowest data point in the moving window and classifies it as ground. The higher elevation points within the window are either removed or replaced with minimum elevation values to create a smoothed DTM of the lowest values in the dataset. The dilation operation similarly uses a moving window. It uses the points removed from the erosion operation and finds the maximum points. This step restores shapes of objects that were lost in the erosion step and creates a smoothed DTM of the highest values in the dataset. Both the removal and replacement of data points are usually based on a specified elevation threshold. When the elevations are greater than the threshold the points are removed from the ground surface.

Eckstein and Munkelt (1995) defined an “opening function” as an erosion operation followed by the dilation operation. A closing function was then defined as the
reverse order of the operations, dilation then erosion. In this case the dilation uses the original data as input and the erosion function uses the dilated surface as its input. They used these opening and closing functions in a dual-rank algorithm with a fixed window size. The dual-rank algorithm uses two successive rank operators to select when the opening and closing functions are to be used. This is done by sorting the neighboring points (fixed window size), around the point to be classified according to elevation, and assigning a rank value to each data point. The algorithm runs through a loop using this rank value beginning with 1 and going to the highest rank value (which is total number of points). When the rank value is 1, the opening function is performed. The next iteration uses the results from the previous opening function to classify the points, until the highest rank value is reached and the dilation function is performed. The results are the classification of those points within the neighborhood or window around the selected point. The process is repeated until it has covered the entire dataset.

Kilian, Haala, and Englich (1996) first tested the use of several different sizes of windows in filtering the data beginning with the smallest. He found that fixed window sizes either lose ground details in removing too many ground points by selecting one minimum elevation point within an oversized window, or buildings and vegetation are not removed within an undersized window. Zhang et al. (2003) later developed an automatic, gradually changing window size with elevation thresholds to overcome the problems of optimizing the window sizes with the previous filters. Once a specific window is moved over the entire dataset the window size is increased and the process is repeated until the maximum window size parameter is met. By starting with a small
window size and increasing it gradually the ground details are preserved and the removal of larger buildings and vegetation are accomplished. Zhang divided the data into cells contained in rows and columns of a grid. Each grid cell was assigned the minimum elevation value of the data points contained within the cells. Cells with no data points were assigned elevations using the nearest neighbor method. Next Zhang used a 2D opening function within his filter to ensure removal of the non-ground points from the dataset. A minimum number of parameters are used in this algorithm yielding effective overall generated DTMs in different types of terrain by increasing the amount of correctly classified ground points. This so called progressive morphological filter by Zhang et al. (2003) has been modified for improvement and compared to other filters by several people. Arefi and Hahn (2005) reconstructed the filter by exploring the use of a different dilation operation called a geodesic. They use a mask surface limit for the geodesic dilation operation and no window size parameters. The mask surface is made up of the last pulse lidar data returns to form the upper limit in which a surface can be created in the dilation operation. This algorithm’s use of dilation focuses on finding the non-ground points.

Zhang and Whitman (2005) compared the progressive morphological filter to the Maximum Local Slope filter by Vosselman (2000) and the Elevation Threshold with Expanding Window filter by Whitman and Zhang (2003). It was observed by Zhang and Whitman that the progressive morphological filter was the only filter able to classify the tops of sand dunes correctly as ground terrain. In order to do this, the grid used for classification was rotated until it was generally parallel with the tops of the sand dunes.
A 1D opening filtering operation was then run along the dunes. The tops of the dunes were maintained as ground points. The 1D filter was used in order to avoid the filter misclassifying points when running perpendicular to the sand dunes. The problems arose with the window size parameter removing the top of the dunes from the DTM. This discovery leads to the question of potential grid bias in this algorithm and other types of terrain that have not been explored.

The progressive morphological filter in 1D and 2D by Zhang et al. (2003) was applied with an added data rotation angle and number of rotations parameters by Zhang and Cui (2007) in his work for NCALM. The purpose of the parameters was to rotate the data in order to better classify narrow straight linear features. Details of or specific applications for these parameters are not disclosed.

Zaksek and Pfeifer (2006), on the other hand, took the approach of improving the elevation difference threshold used within a morphological filter. This was done by using approximated data trend surfaces instead of the horizontal surfaces used by Zhang. Zhang’s surfaces assume horizontal terrain and do not take into account whether classification is going up or down sloped areas. Zaksek’s two data trend surfaces are estimated by using the first and last lidar echo data from the given pulses. The first echoes from a given laser pulse (upper surface) are estimated to represent non-ground points. The lower surface (last echo) is estimated to be along the ground. These data trend surfaces take into account the position of a potential ground point on a sloped surface. If the neighboring points are up the slope from the potential ground point then elevation threshold will be positive. Points down slope will have negative elevation
thresholds and perpendicular to the slope points will have a zero threshold. These three
elevation threshold positions are used to better classify points in steep terrain. The
elevation difference threshold is taken from the lower data trend surface for
classification. Points that fall below the threshold are classified as ground and points
above are removed. New data trend surfaces are estimated with remaining echo data and
the process iterates until the surface stops changing to form a bare-earth model.

Chen’s (2007, 2009) focus was to change the methods so the constant slope
assumption used by others would not be needed. One method that was implemented
helped fill in missing data areas, specifically in laser absorbent water areas. This was
accomplished by finding the lowest (last) lidar echo return within a set window area and
assigning that elevation value to the missing grid cells. The second method detects and
removes lower outlying points from the dataset that would have been classified as ground
points by the morphological filter. A specified height distance below the surrounding
ground surface points is used as a parameter to find low points. A parameter for the area
of surrounding points is also set to maintain an accurate ground surface when measuring
distances below the surface for the outlying low points. These two methods replaced the
constant slope parameter in Zhang’s progressive morphological filter.

Zhang et al. (2003) is a key paper because of its use of few parameters and
because it serves as a foundation for the rest of the developed algorithms in this class of
filter.
Filters Implemented by NCALM

The National Center for Airborne Laser Mapping (NCALM) funded a project in which Zhang and Cui (2007) developed some software to implement six particular algorithms into Airborne LIDAR Data Processing and Analysis Tools (ALDPAT). These algorithms include:

1. The progressive morphological filter in 1D and 2D (Zhang et al., 2003)
2. An elevation threshold with expand window filter (Zhang and Whitman, 2005)
3. An iterative polynomial fitting filter (Zhang and Cui, 2007)
4. Two polynomial surface filters (Zhang and Cui, 2007)
5. The maximum local slope filter (Vosselman, 2000)
6. An adaptive TIN filter modified from Terrascan’s slope-based TIN filter
   (Soininen, 2010)

It is clear from the literature that no one filter works best for any given combination of terrain, vegetation and man-made features. Sithole also observed this in his comparison of filtering algorithms. He concluded that the effectiveness of filters can be greatly dependent on the type of terrain within the dataset; whether it is flat, urban or steep mountainous terrain (Sithole and Vosselman, 2004). This allowed the opportunity to choose one of these many filters to evaluate and improve.
OBJECTIVES

From the literature review, it was found that the progressive morphological filter algorithm developed by Zhang et al. (2003) has potential weaknesses in two areas. The first recognized potential weakness in the algorithm was the unexplored effect of the order of implementation of the row-column erosion and dilation functions. The second potential weakness was the bias of the grid positioning. The objective of this work has been to test and improve this filter by characterizing and possibly overcoming these two potential weaknesses, keeping in mind that success would minimize human interaction.

The identification of the potential row-column ordering bias came from Zhang’s paper when he gave the option of using erosion and dilation functions in either or both the x (row) and y (column) directions. He used the 2D opening operation in the order of column first and row second. The following reason was given for his approach in the paper, “The opening operation was applied to both x and y directions at every step to ensure that the nonground objects were removed.” No further explanation of using this order over another was given. There were several other orders of the row and column directions to be placed within the erosion and dilation functions. This lead to the potential row-column iteration bias associated within the approach, based on which order was being used. The possible ordering bias was not explored or recognized in previous papers. It was theorized that the misclassification of ground points in steep terrain (Zaksek and Pfeifer, 2006) were caused by this potential row-column ordering bias.

In this thesis, four separate algorithms with different orderings of the erosion and dilation functions in the row and column directions have been developed. The objective
has been to test and compare the four algorithms to evaluate the potential bias associated with them.

Regarding the second potential weakness, the data rotation angle parameter implemented by Zhang and Whitman (2005) and in NCALM (Zhang and Cui, 2007) illustrated that this filter had a possible natural grid orientation bias. The classification of the sand dunes by Zhang and Whitman demonstrated a problem with alignment of the grid according to the features in the terrain. Using the term PM to mean progressive morphology, Zhang explains the problem and solution this way:

The straight, long, and elevated shape of a sand dune allows itself to be identified by the PM filter. However, some data preprocessing is required to make the PM filter work. The PM filter removes features with steep slopes within a window. This problem can be solved by rotating an overlain mesh so that it is parallel to ridges of the sand dunes. (p. 318)

Once the grid was parallel with the sand dunes, Zhang found that the tops of the sand dunes were correctly classified as ground points. It was hypothesized that other possible features could be affected such as ridges, peaks, cliffs and buildings. It was important that this problem be tested and improved to better classify the ground points at the tops of linear features and to thereby generate better DTMs. Hence, the objective has been to test for the potential negative effects of the grid bias in more detail and develop a more robust solution.

The approach has been to implement the classification algorithm using numerous grid orientations. This was done by selecting a given rotation angle increment and rotating the grid multiple times at that angle. Each time the grid was rotated to a new position, the eroding and dilating of the dataset’s surface was different because the windows were oriented in a new direction. Final DTMs were generated for each grid
direction. Two proposed methods for combining these results have then been developed and presented.

Two “truth” datasets were created for use in testing the algorithms. Each dataset was extensively hand-edited and used to compare and measure the success of the different algorithms. Developing these two “truth” datasets was done by manually stepping through the datasets, using profiles and 3D viewers and manually classifying the data points. Aerial photographs of the datasets were also used to better classify the ground points correctly.

To test the accuracy of the algorithms, “truth” datasets have been compared to the results of the filtered datasets. An error analysis of the test results for both the grid bias and row-column bias was used according to the two types of classification errors that were found. Type I errors occur when ground points are classified as non-ground points and Type II errors occur when non-ground points are classified as ground points. A method for balancing these error types and measuring an overall algorithm’s success has been developed and used.

The results are given and interpretations of those results are discussed. Conclusions for the two tested bias issues of the progressive morphological filter are then given. Recommendations of how to best use the algorithm in practice are stated along with future research topics.
ALGORITHM CHANGES

Existing Algorithm

The evaluation and improvement of the progressive morphological filter algorithm developed by Zhang et al. (2003) has been the focus of efforts in this study. Modifications to the algorithm have been implemented using MatLab following six general steps as explained below. A flowchart of the steps in the algorithm is given in Figure 1.

(Step 1) The first step of the algorithm reads a raw lidar point data file of x, y, and z coordinate values. A surface grid is then constructed with a specified grid cell size (c) so as to encompass the points. The grid cell size is selected according to the point density of the dataset. The more lidar points contained within the grid, the better the classification results. An elevation is then assigned to each cell. Grid cells containing more than one lidar point were assigned the z coordinate value of the minimum elevation lidar point. Cells with no lidar point measurements were given interpolated elevation coordinate values by using the nearest-neighbor interpolation from the neighboring points.

(Step 2) Once the surface grid was constructed, the morphology functions of erosion and dilation were then implemented. The algorithm uses profile sections aligned with the grid to classify the points. The grid structure enables profiles of data points to be easily extracted along each row and column. The 1D algorithm uses only the row or
(1) Import raw x, y, z coordinates and produce surface grid

(2) Erosion & dilation Functions (2D)

(3) Elevation difference < Elevation threshold (dhT)

(3.1) True
(3.2) False
(3.2.1) Remove points from surface grid
(3.2.2) True
(3.2.3) False

(4) Increase window size (w_k) and calculate elevation threshold (dhT)

(5) Window size (w_k) < maximum window size (w_max) ?

(5.1) True
(5.2) False

(6) Extract ground points and develop DTM

Figure 1. Existing progressive morphological filter flowchart.
column directions to extract profiles and classify the points using a line window shape. The windows represent segments of the profile in which sections of the extracted profiles are evaluated. The 2D algorithm uses both the row and column directions to extract profiles in both directions to classify the points using a square window shape.

Zhang introduces both 1D and 2D approaches but only implements the 2D algorithm with column direction first, followed by the row direction. The window shape size and surface grid elevations are used as inputs for the erosion function. The window shape is defined as the area in which the erosion function is incrementally processed along the extracted profile. For the first iteration, the initial window size, from Equation 1 in Step 4 with k equal to zero, and the original surface grid are used as inputs. In the subsequent iterations, the calculated window size from Step 4 and surface grids from the previous iteration are used. The eroding of the surface grid is then done by extracting the profiles along the rows and columns. Within each window area the minimum elevation is selected and placed in the grid cell that was located in the center of the window. The window is then moved along all of the grid cells until the entire area is covered and minimum elevations determined.

The eroded surface grid and the given window size are then given as the input for the dilation function. Maximum elevation values within the same given window area are assigned to the grid cells during dilation by moving the window over the dataset in the same way as selected for the erosion function.

(Step 3) The output surface from the dilation function and the input surface to the erosion function are used to decide whether a point was to be classified as a ground or
non-ground point. The elevation differences between the two surfaces are calculated and tested against the elevation threshold difference parameter. Each grid cell is tested individually. If the elevation difference is less than the threshold the point is classified as ground and if greater it is classified as non-ground. For the first iteration, the elevation threshold is set to the initial elevation threshold (dh0) specified by the user. This parameter is set to be close to the lidar measurement error.

(Step 4) The progressive part of the filter comes into play through increasing the window size and elevation difference threshold. Small window sizes preserve the ground points over gradually changing terrain; while larger window sizes remove buildings and vegetation by selecting the minimum elevation points on both sides of such features. The window size is progressively increased in order to remove both small and large non-ground objects from the data surface. The window size (wk) is increased using the equation

\[ w_k = 2b^k + 1 \]  

where (b) is the base of the increasing exponential function and (k) is the iteration step starting at 0 and increasing until the window size (wk) becomes greater than the maximum window size (wmax). The elevation threshold (dhT) is calculated based on the following conditional statements found in Equation 2:

\[ dhT_k = \begin{cases} 
\text{dh0,} & \text{if } w_k \leq 3 \\
\text{s}(w_k - w_{k-1})c + dh0, & \text{if } w_k > 3 \\
dh_{\text{max}}, & \text{if } dhT_k > dh_{\text{max}} 
\end{cases} \]  

where (dh0) is defined as the initial elevation difference threshold, (s) as the slope terrain, (c) as the grid cell size and (dhmax) as the maximum elevation difference threshold. The
The slope \( s \) parameter is set to be near the estimated average slope in the dataset. The maximum elevation difference \( (dh_{\text{max}}) \) can be set close to the smallest building height in urban areas or the height of the tallest trees in rural forested areas. Both of these calculated parameters are used in the following iterations.

(Step 5) The algorithm continues to iterate until the window size becomes larger than the predetermined maximum window size \( (w_{\text{max}}) \) selected by the user. This parameter is usually set to be slightly larger than the largest building in the dataset.

(Step 6) The ending surface grid after all of the iterations yields grid cells that are classified as ground and non-ground. Grid cells containing \( x,y,z \) points classified as ground are then extracted and used to develop the DTM.

Table 1 summarizes the parameters used in the algorithm. Variables, units and a brief general description of how to select each parameter are given.

Table 1. Input parameter summary of the existing progressive morphological filter with a basic description of how to select each parameter

<table>
<thead>
<tr>
<th>Input Parameters</th>
<th>Variable</th>
<th>Units</th>
<th>Parameter Selection Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Grid Cell Size</td>
<td>( c )</td>
<td>( \text{cm} )</td>
<td>Select size that gives as close as possible to one lidar point per grid cell. Check point density of dataset to get an idea.</td>
</tr>
<tr>
<td>Base of Exponential Function</td>
<td>( b )</td>
<td>-</td>
<td>Base used in Equation 1. Usually set to 2 to increase window size exponentially and reduced iterations.</td>
</tr>
<tr>
<td>Maximum Window Size</td>
<td>( w_{\text{max}} )</td>
<td>( \text{m} )</td>
<td>Set to be just larger than the largest building.</td>
</tr>
<tr>
<td>Slope</td>
<td>( s )</td>
<td>-</td>
<td>Set according to average terrain slope in the area.</td>
</tr>
<tr>
<td>Initial Elevation Threshold</td>
<td>( dh_0 )</td>
<td>( \text{cm} )</td>
<td>Set to be close to lidar measurement error.</td>
</tr>
<tr>
<td>Maximum Elevation Threshold</td>
<td>( dh_{\text{max}} )</td>
<td>( \text{cm} )</td>
<td>For buildings set to lowest building height and trees to largest elevation different in the area.</td>
</tr>
</tbody>
</table>
Row-Column Order Modifications

Modifications have been made to the existing algorithm to test the effects of the ordering of the erosion and dilation functions in Step 2 of Figure 1. The morphological functions in the algorithm can include Row Erosion (RE), Column Erosion (CE), Row Dilation (RD) and Column Dilation (CD). The four possible orderings of these functions are shown in Figure 2.

Zhang et al. (2003) originally only used Order 2 of Figure 2, with column direction first, followed by row direction for both the erosion and dilation functions. The code has been modified to include the other three orders. The function order follows from left to right as given in each row of Figure 2. For example, Order 1 begins with the original gridded surface as the input for the RE function. The output of that function becomes the input of the CE function which then feeds to the RD function which feeds the CD function which then produces the final dilated surface. No other changes to the algorithm were made.

<table>
<thead>
<tr>
<th>Orders</th>
<th>Erosion</th>
<th>Dilation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>RE</td>
<td>CE</td>
</tr>
<tr>
<td>2</td>
<td>CE</td>
<td>RE</td>
</tr>
<tr>
<td>3</td>
<td>RE</td>
<td>CE</td>
</tr>
<tr>
<td>4</td>
<td>CE</td>
<td>RE</td>
</tr>
</tbody>
</table>

Figure 2. The different four row-column orders tested with variables defined as follows: Row Erosion (RE), Column Erosion (CE), Row Dilation (RD) and Column Dilation (CD).
Grid Rotation Modifications

A second set of modifications to the existing algorithm were made in order to evaluate the potential effects of the grid bias on classification results. This was done by implementing incremental rotation of the grid into the algorithm. Figure 3 gives the flowchart of the grid rotation algorithm, with changes given in a bold font.

The first change to the algorithm was the addition of a rotation angle parameter. This parameter is defined as the incremental rotation angle at which the lidar points are rotated around the z-axis.

The next addition to the algorithm was the RotateXYPoints function or Step 3 of the flowchart found in Figure 3. The actual raw lidar points are rotated instead of the grid itself. This was done because of the simplicity of coding the function and the fact that the same results were generated whether the points or the grid were rotated. The function’s input values are the rotation angle, original raw lidar x, y, z points and the minimum and maximum values of the x and y coordinates.

The minimum and maximum x and y coordinates are used to find the center of the lidar dataset or, in other words, the pivot position for rotating the points. A direction cosine matrix (DCM) given by Equation 3 is used to rotate the points where (a) is the rotation angle.

\[
DCM = \begin{bmatrix}
\cos(a) & -\sin(a) & 0 \\
\sin(a) & \cos(a) & 0 \\
0 & 0 & 1
\end{bmatrix}
\] (3)
Import raw x, y, z coordinates and produce surface grid

For i = 1 to n

RotateXYPoint Function

Erosion & dilation functions (1D Column)

Elevation difference < Elevation threshold (dhT)

False
Remove points from surface grid

True
Classify ground points and generate filtered surface grid

Increase window size (wk) and calculate elevation threshold (dhT)

Window size (wk) < maximum window size (wmax)?

True

False

Extract ground points and develop DTM at given rotation angle

Increase angle of rotation

End

Figure 3. The grid rotation algorithm flowchart. The bolded features show the changes made from the existing progressive morphological filter.
This matrix is multiplied by the original x, y, z coordinates to rotate them about the z-axis. The function returns the rotated lidar points. The points are then used for constructing the grid surface. This function was inserted into the existing code as Step 3 of Figure 3.

To accommodate grid rotation, Step 2 in Figure 1 was changed to Step 4 in Figure 3. The algorithm was also changed from a 2D rectangle window shape to a 1D line window shape. This was done based on the observations found in Zhang and Whitman’s (2005) comparison paper. They found that when the algorithm was run perpendicular to straight and long elevated ground terrain features, the tops of these features were removed from the DTMs. Linear line windows in one direction would maintain the terrain surface while rectangle windows would remove the tops of the terrain surface due to the perpendicular direction filtering. Therefore, the 1D window in the column direction was used in the grid rotation algorithm in order to characterize and compensate for this bias.

The new algorithm rotates the lidar points in multiple directions from 0 to 360 degrees. This was implemented into the algorithm by adding a loop as shown in Step 2 of Figure 3. The number of iterations (n) for the loop is calculated using Equation 4:

\[ n = \frac{360}{a} \quad ; \quad 0 < a \leq 360 \]  

(4)  

where (a) is the rotation increment. The loop takes into account all of the direction angles from 0 to 360 degrees at the specified increment. Each time through the loop, the angle size was increased by the rotation angle parameter (a) until it reached or exceeded 360 degrees. After each rotation, the 1D column erosion and dilation functions were run,
the entire lidar dataset was classified, and DTMs generated as shown in Step 8 of Figure 3.
EXPERIMENTAL SETUP

This section explains the experimental setup used to evaluate the effects of the algorithm modifications. The experimental methods, including details for both the row-column and grid rotation experiments are first explained. Then details regarding representative datasets for both rural and urban areas are described.

**Methodology**

The beginning processing steps are the same for both the row-column and the grid rotation algorithms. Datasets from lidar projects can consist of millions of data points. Software programs can have memory problems when dealing with such large datasets. In the 32-bit MatLab programming environment, it was found that the data needed to be broken up into blocks and then processed one at a time. This step is defined as the “blocking” of the dataset. The block size was determined according to the amount of data MatLab could efficiently process at a given time. This was found to be approximately 38,000 points. The creation of these relatively small blocks created some issues associated with the block edges that had to first be solved before the rest of the analysis could proceed. This is explained below.

When first running the algorithm on a test data block, a block-related classification error was discovered along the edges of the block. This error was observed on several different processed blocks. Further investigation determined that the edges affected by the error were influenced by the terrain slope next to the edge. When the slope was positive or uphill at the edge, the algorithm would misclassify the points within
a certain distance from the edge. Figure 4 shows an example of a processed block and associated profile section. The orange points were classified as ground points and white points were classified as non-ground points. The green rectangle through the block shows where the profile at the bottom was taken from.

Figure 4. Edge misclassification error block and profile example. Orange and white lidar points are classified ground and non-ground points. The green rectangle through the blocks show where the profile at the bottom was taken from.
The misclassification error runs along the bottom edge of the block and, as seen in the profile, is on the uphill slope. This edge problem results when the window of the morphological filter reaches the edge of the block of the extracted profile. The removal of the highest elevation points at the edge is done when the elevation difference test is performed. At the top of the slope, the points were removed which resulted in the edge misclassification error. It was also observed that the width of the misclassified area next to the edge of the block increased as the maximum window size (wmax) parameter was increased.

To resolve this problem a buffer area surrounding the original block was added to each block. This so-called “buffered block” was then used for processing and then clipped using the original block area, thus removing the edge error misclassification. The amount of buffering was determined according to the window size parameter selected. The larger the window size, the larger the buffer needs to be to remove the adverse edge effects.

The same error analysis was used for both algorithms. In testing for the row-column and grid orientation biases along with their improvements, a measure of success included the Type I, Type II and total correctly classified criteria. Type I errors occur when actual ground points are classified as non-ground points. Type II errors occur when actual non-ground points are classified as ground points. The total correctly classified was the percentage of points classified correctly in the dataset. Table 2 summarizes the definition of Type I and Type II errors in the context of lidar point classification. The actual point class is known from a manually classified “truth” dataset created for this
Table 2. Type I and II error analysis with the algorithm classification at the top right side and the actual or true point classification at the bottom left side

<table>
<thead>
<tr>
<th>Error Analysis</th>
<th>Algorithm Point Class</th>
<th>Ground</th>
<th>Non-Ground</th>
</tr>
</thead>
<tbody>
<tr>
<td>Actual Point Class</td>
<td>Ground</td>
<td>Correct</td>
<td>Type I error</td>
</tr>
<tr>
<td>Non-Ground</td>
<td>Type II error</td>
<td>Correct</td>
<td></td>
</tr>
</tbody>
</table>

study. The algorithm point class referred to in Table 2 is generated by the filtering algorithms tested in this study.

It is hypothesized that Type II errors have a greater effect on the quality of the generated DTM because the surface is corrupted by incorrectly classified non-ground points. Type I errors also have an effect of removing ground points and leaving voids in the surface. However, it is hypothesized that this type of error does not tend to have as great of an impact on DTM quality as the Type II errors. The balancing or weighting of the two types of errors was discussed by Sithole and Vosselman (2004) in their filter comparison paper. They explained the balancing this way:

The question of which error to minimize depends on the cost of the error for the application that will use the filtered data. From a practical point of view, it will often depend very much on the time and cost of repairing the errors manually, which is done during quality control. Experience with manual filtering of the data showed that it is far easier to fix Type II errors than Type I errors. (p. 99)

Type II errors are usually focused on and minimized solely in the literature but from Sithole and Vosselman’s statement it was decided that the Type I errors also needed to be considered. A weighted scored was used to balance the errors and decide the accuracy of the algorithm results. For these reasons the Type II errors were given a weight of three
and Type I errors were given a weight of one. The weighted score (W) is therefore calculated as shown in Equation 5:

\[ W = T1 + 3 \times T2 \]  

(5)

where \( T1 \) is the number of Type I error and \( T2 \) is the number of Type II errors.

The percentage of the two different error types and total correctly classified were calculated by comparing the filtered dataset to the highly edited ground “truth” dataset. By using these four types of criteria the results of the improvements of the progressive morphological filter are presented for both the row-column and grid rotation algorithms.

Row-column order

Once the data was blocked and buffers set, the four row-column algorithms were run separately on all of the blocks in the dataset. Classified block files generated from each row-column algorithm were combined into one file. This yielded one file containing the filtered dataset for each row-column order tested.

The error analysis was then performed by comparing the “truth” dataset to the four filtered datasets. For each of the four orderings of the row-column algorithms, a table was developed showing the number of Type I and II errors, the percentage correctly classified, and weighted scores.

Further analysis of the row-column bias was done by looking at one block in the dataset and tracking the elevation value of a specified point after each erosion and dilation function. Four graphs with the elevation values were developed for each row-column order to visually see the progression of the functions and to analyze the potential ordering bias. Profiles were also plotted after each function to see the change in surface
around the selected point. This resulted in 16 profile graphs for the tracking of the four row-column orders, four graphs per order.

All four algorithms were processed on a single dataset to evaluate the row-column bias problem. The single dataset provided the analysis results necessary to evaluate the bias.

**Grid rotation**

After the data was blocked and buffers were set, the grid rotation algorithm was run on all the blocks. In the beginning, the algorithm was planned to run from angles 0 to 180 degrees because of the repetition of directions. For example the angles of 90 and 270 are on the same line and seemingly should yield the same result. However, it was wrongly assumed that classifying the data along the same line in different directions would be the same. After testing angles on the same line it was observed that the orientation of the terrain relative to direction in which the algorithm was run affected the results of the classification performed by the filter. Therefore the algorithm was eventually run from 0 to 360 degrees in 72, 5-degree increments. For each rotation angle, each buffered block was classified and output to a file. As a result, each of the buffered blocks was processed at all of the different angle rotations.

The next step was to combine all of the block files with the same angle rotation into one file. For example, each of the processed block files for an angle of say 50 degrees was clipped then combined with the others to form one file for the entire dataset.

An error analysis was then run on the results for each of the angles. A table was generated from the analysis to compare the different classification results for the various
angles. Graphs were developed from the table to show the change of Type I, Type II, percent correctly classified and weighted scores compared to the different angle rotations. An example of these graphs is given as Figure 22 which is presented and discussed in the next section. Best angle directions corresponded to the lowest weighted scores. Colored point cloud plots of the results showing where the Type I and II errors occurred in the dataset were also created during the error analysis procedure. Point colors differentiate Type I error points, Type II error points, and correctly classified points; colored yellow, red, and blue, respectively. An example is given as Figure 23 which is presented and discussed in the next section. These colored point clouds enable evaluation of the terrain areas where the algorithms had problems and to better explain bias affects. Problem areas associated with the different angles were subsequently highlighted in order to analyze the grid bias affects at a point by point scale and discover new methodologies for optimally combining classifications from all angles.

Test Datasets

USTAR LASSI Service Center provided the opportunity to select from lidar datasets with various types of terrain surfaces and man-made features. The decision of which datasets to pick was guided by Sithole and Vosselman’s (2004) study and comparison of filter algorithms. They observed that the effectiveness of filters can be greatly dependent on whether terrain is flat, urban, or steep and mountainous. Along with Sithole and Vosselman, the literature points out strengths and weaknesses of filters according to terrain type. For these reasons the following two datasets were selected.
The first dataset is from the T.W. Daniel Experimental Forest. The site is located in Cache National Forest of Northern Utah, about seven miles east of the southern end of Bear Lake. The terrain is covered with rolling hills, thick groves of trees, dirt roads and a few open areas. Figure 5 shows a colored lidar point cloud of the area from a birds-eye view. The selected subset of the Daniel Forest dataset contains one 1300 m flight line with about 1.31 million lidar points and was collected on July 8, 2009. The airplane was flying at a speed of 50 m/s at an altitude of about 550 m above ground level. The laser scanner, consisting of a Reigl Q560 lidar transceiver, was set at a pulse repetition rate of 70 kilohertz (kHz) and a scan sweep of 50° resulting in a swath width of about 500 m. Given these settings, the average lidar point density turned out to be about 2 points per square meter.

The second dataset is an urban site located on the Utah State University (USU) Campus in Logan, Utah. There is a mixture of vegetation and urban terrain objects in this dataset. The area is made up of multiple shapes and sizes of buildings and residential homes with parking lots and low vegetation. There are also several streets and one steep hill with tall trees towards the bottom of the hill. The data subset selected for analysis consists of one 1200 m long flight line and about 1.43 million lidar points. This data was collected on April 9, 2010 from a helicopter at an altitude of about 300 m above ground level. The laser scanner was set at a pulse repetition rate of 70 (kHz) and a swath width of about 320 m. Figure 6 shows a colored lidar point cloud of this dataset.

In both datasets, point colors are derived from a boresighted Canon 5D Mark II color digital camera.
Figure 5. Colored lidar point cloud of a portion of the T.W. Daniel Experimental Forest.
Figure 6. Colored lidar point cloud of a portion of the USU campus.
RESULTS AND DISCUSSION

Input Parameter Selection

The subdivision of Daniel Forest and the USU Campus datasets into blocks was dependent on the amount of data the MatLab software could process in memory at a given time. The Daniel Forest and USU Campus required 66 blocks at 110 m square and 75 blocks at 80 m square, respectively. The layout of the blocks over the Daniel Forest and USU Campus datasets are shown in Figures 7 and 8.

The buffer width chosen was dependent upon the selected maximum window size parameter. The larger the window size the larger the edge misclassification error distance and the bigger the needed buffer width. After the window size was selected for the datasets the buffer size was then determined. This was done by manually viewing the results of the algorithms. Referring to Figure 4, the buffer size was selected by scaling the original block size by at least the edge error distance as seen in the bottom of Figure 4. The buffer scale for Daniel Forest was 1.59 times the original blocks and the USU Campus dataset was scaled by 1.72 times due to the larger window size parameter required for classifying buildings. Figures 8 and 9 show the buffered blocks for each dataset.

Normally, a user would follow the suggestions explained in Table 1 of the Existing Algorithm section to set the input parameters. For this study, a more detailed review of the parameters was undertaken. Sensitivity plots were used to determine the effects of changing the input parameters on the results for the given datasets. The
weighted score from the error analysis was then used as the deciding variable for the sensitivity plots. The lower the weighted scores the better the classification. The input parameter values were plotted on the x-axis and the weighted scores on the y-axis. An example is given as Figure 10. The goal was to rank the parameters from the most to the least sensitive. Fine tuning of each input parameter was done by using this process.

Figure 7. Blocking of Daniel Forest dataset at 110 m block sizes.
Figure 8. USU Campus blocks at 80 m and 1.79 scaled buffer maps.
Figure 9. Daniel Forest 1.59 scaled buffer blocks map.

The most accurate way of tuning the parameters would have been to use the whole dataset to develop the sensitivity plots. The processing of the entire datasets would have taken a prohibitive amount of time, so a sample area from each dataset was selected. Sample areas were required to have a good representation of the terrain objects included in each of the different datasets.
Daniel Forest parameters

Block 19 found in Figure 7 was selected as the sample area for the Daniel Forest dataset. This area included part of a mountain slope, thick trees and a piece of open terrain. “Central values” were selected for each of the input parameters required by the progressive morphological filter. All of the parameters were held constant at the “central values” except for the parameter which was being tested. Values around the “central values” were tested to find the sensitivity ranks and optimum input parameter values. Sensitivity plots were then developed for each of the parameters by following this process.

A logical explanation for the selection of “central values” for each parameter is given as follows and is summarized in Table 3. A “central value” of 0.2 was selected for the slope which is close to the average slope in the area. The grid cell size (c) was set to 50 cm. Having a point density of about 2 point per square meter, it was desired to place as close as possible 1 lidar point per grid cell. The base (b) for the exponential formula in Equation 1 was set to 2. This was done to give several window sizes but to also reduce the number of iterations of the algorithm. The maximum window size (wmax) was set to 20 m for the removal of the large groves of trees in the area. Initial elevation difference threshold (dh0) was set to 1 cm due to the low vegetation in the area. The high vegetation in the area resulted in setting the maximum elevation difference threshold (dhmax) to 100 cm.

The slope of the sensitivity plots determined the ranking of the parameters, 1 being most sensitive and 6 being least sensitive. The slope compares the change in
weighted score over the change in parameter selection. The higher the slope the more sensitive the parameter is to change.

The parameters slope, grid cell size and maximum window size were found to be the most sensitive parameters for the Daniel Forest dataset as shown in Table 3. The sensitivity plots for these three parameters of slope, maximum window size and grid cell size are shown in Figures 10 through 12. The black bar along the x-axis in Figures 10 through 12 shows the location of the “central values” for each parameter. The red arrows are placed at the minimum weighted score locations and represent the selected optimum input parameters.

Table 3. Central values and sensitivity ranks for the Daniel Forest and USU Campus datasets with 1= Most Sensitive and 6=Least Sensitive

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Variable (units)</th>
<th>Central Values</th>
<th>Sensitivity Ranks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slope</td>
<td>s</td>
<td>Daniel Forest</td>
<td>USU Campus</td>
</tr>
<tr>
<td>Maximum window size</td>
<td>wmax (m)</td>
<td>20</td>
<td>30</td>
</tr>
<tr>
<td>Grid cell size</td>
<td>c (cm)</td>
<td>50</td>
<td>40</td>
</tr>
<tr>
<td>Initial elevation threshold</td>
<td>dh0 (cm)</td>
<td>1</td>
<td>2</td>
</tr>
<tr>
<td>Base of exponential function</td>
<td>b</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>Maximum elevation threshold</td>
<td>dhmax (cm)</td>
<td>100</td>
<td>300</td>
</tr>
</tbody>
</table>
Figure 10. Sensitivity plot for Daniel Forest slope parameter with the black bar set to the “central value” of 0.2 and the red arrow set to optimum selected input parameter of 0.13.

Figure 11. Sensitivity plot for the Daniel Forest maximum window size parameter with the black bar set to the “central value” of 20 m and the red arrow set to optimum selected input parameter of 35 m.
Figure 12. Sensitivity plot for the Daniel Forest grid cell size parameter with the black bar set to the “central value” of 50 cm and the red arrow set to optimum selected input parameter of 26 cm.

USU Campus parameters

The USU Campus dataset with the different terrain types, vegetation and variety of building sizes required careful selection of a sample area for tuning the parameters. The first sample area chosen was Block 42 located in Figure 8. This area included several medium sized homes, a couple of large trees surrounding them and a parking lot. This sample area was used for selecting the central values found in Table 3. The cell size (c) was a little smaller than the value selected for Daniel Forest at 40 cm because of the somewhat denser point cloud. The base (b) was set to 2 for the same reasons as the other dataset to give several window sizes and reduce iterations. The maximum window size (wmax) was set to 30 m because the dimensions of buildings in the area were close to that value. The slope parameter was set to the average slope of the terrain at, 0.2. The
initial (dh0) and maximum (dhmax) elevation difference thresholds were set to 2 cm and 300 cm, respectively. The initial threshold was increased in the urban area with less low vegetation and the maximum threshold was set close to the lowest building heights to make sure they were removed from the ground surface (Zhang et al., 2003).

The sensitive ranking for the USU Campus dataset results in the same top two parameters as the Daniel Forest dataset: slope and maximum window size. The third ranking parameter was the base of exponential function as shown in the Table 3. Sensitivity plots of these three parameters are shown in Figures 13 through 15 with the black bars set at the “central values” and the red arrows placed at the selected optimum input parameters.

![Sensitivity plot for USU Campus slope parameter](image)

Figure 13. Sensitivity plot for USU Campus slope parameter with the black bar set to the “central value” of 0.2 and the red arrow set to optimum selected input parameter of 0.25.
Figure 14. Sensitivity plot for USU Campus maximum window size parameter with the black bar set to the “central value” of 30 m and the red arrow set to optimum selected input parameter of 32 m.

Figure 15. Sensitivity plot for USU Campus base of exponential function parameter with the black bar set to the “central value” of 2 and the red arrow set to optimum selected input parameter of 3.
The large variation in building sizes from the campus buildings to the residential homes was observed. Selection of the size of buildings found in the sample area affected the sensitivity plots generated for the maximum window size (wmax) parameter. Blocks of 38, 39, 53 and 54 from the USU Campus dataset indexed in Figure 8 were selected for the second sample area. This contiguous group of blocks includes one large campus building, several smaller buildings and some low vegetation. The sensitivity plot of the maximum window size for the second sample area was generated and is found in Figure 16. This was compared with the first sample area plot in Figure 14. The optimum maximum window size (wmax) for the first area was 32 m and 90 m for the second area.

Tests were done on other blocks of the dataset with maximum window (wmax) set to 90 m. Results yielded edge misclassification errors covering more than half of the block areas. The larger the window size the bigger the buffer scale had to be to remove the edge errors. It was not possible to increase buffer width to more than half the dimension of the block areas.

It was found that ground points underneath trees were completely removed from the DTMs due to the large window size. Only the areas where the large buildings were located did the larger window size perform well. Overall, this classification did not perform as well for a given window size as the first sample area did. The sensitivity plot from the first area was therefore used to select the input parameter for this dataset. Therefore, the maximum window size selected for this dataset had the greatest impact on the building classification results.
The sensitivity plots presented in this section were all used to select the parameters based on the lowest weighted score. The selected parameters are summarized in Table 4.

Figure 16. USU Campus sample area 2 sensitivity plot of maximum window size. Black bar set to the “central value” of 30 and the red arrow set to the optimum input parameter of 90.

Table 4. Selected input parameters from the sensitivity plots for both Daniel Forest and USU Campus datasets

<table>
<thead>
<tr>
<th>Selected Parameters</th>
<th>Daniel Forest</th>
<th>USU Campus</th>
</tr>
</thead>
<tbody>
<tr>
<td>c (cm)</td>
<td>26</td>
<td>26</td>
</tr>
<tr>
<td>wmax (m)</td>
<td>35</td>
<td>32</td>
</tr>
<tr>
<td>s</td>
<td>0.13</td>
<td>0.25</td>
</tr>
<tr>
<td>dh0 (cm)</td>
<td>10</td>
<td>4</td>
</tr>
<tr>
<td>dhmax (cm)</td>
<td>50</td>
<td>125</td>
</tr>
<tr>
<td>b</td>
<td>2</td>
<td>3</td>
</tr>
</tbody>
</table>
Results of Row-Column Order Analysis

The error analysis of the Daniel Forest dataset for the four ordering algorithms is summarized in Table 5. Surprisingly, the classification of all of the row-column orderings was exactly the same. Comparing the results of each order visually also showed that all of the orders classified the data the same.

A grid cell containing a lidar point in the center of Block 13 of the Daniel Forest dataset was used for elevation tracking. In Figure 17 the four order graphs, A through D, show the elevation values of the selected lidar point after each of the erosion and dilation functions. Each order was tracked and the ending elevation of each order was the same although the arrangements of elevations previous were different. The elevation value after the erosion functions was the same in all four orders. This was also true for the ending elevation value after the dilation functions were complete.

Table 5. Error analysis results of the four row-column orders of the Daniel Forest dataset

<table>
<thead>
<tr>
<th>Order</th>
<th>Type I</th>
<th>Type II</th>
<th>Type I %</th>
<th>Type II %</th>
<th>Total Errors %</th>
<th>% Correctly Classified</th>
<th>Weighted Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>31,231</td>
<td>2,596</td>
<td>2.38%</td>
<td>0.20%</td>
<td>2.58%</td>
<td>97.42%</td>
<td>39,019</td>
</tr>
<tr>
<td>2</td>
<td>31,231</td>
<td>2,596</td>
<td>2.38%</td>
<td>0.20%</td>
<td>2.58%</td>
<td>97.42%</td>
<td>39,019</td>
</tr>
<tr>
<td>3</td>
<td>31,231</td>
<td>2,596</td>
<td>2.38%</td>
<td>0.20%</td>
<td>2.58%</td>
<td>97.42%</td>
<td>39,019</td>
</tr>
<tr>
<td>4</td>
<td>31,231</td>
<td>2,596</td>
<td>2.38%</td>
<td>0.20%</td>
<td>2.58%</td>
<td>97.42%</td>
<td>39,019</td>
</tr>
</tbody>
</table>
Figure 17. Elevation tracking of a single lidar point after each function: Row Erosion (RE), Column Erosion (CE), Row Dilation (RD) and Column Dilation (CD). The four orders are shown in A through D.

Profiles containing 15 lidar data points in the center column of the grid were also tracked after each function. The red circles in the graphs symbolize the same lidar point that was tracked in Figure 17. Figures 18 through 21 show the profiles for each of the row-column orders. Each figure shows the results after each of the four steps, A through D. These steps are labeled according to the RE, CE, RD and CD functions used. The elevation and column direction coordinate values shown in the graphs are in meters. The profiles start at the center of Block 13 and include 5 meters worth of lidar data points from the center column.
Figure 18. Order 1 profile tracking from the center of Block 13 in the Daniel Forest Dataset. The x-axis starts at the center of Block 13 as zero. The profile includes 5 meters of lidar points in the column direction. A through D steps show the profiles after each erosion and dilation function.

From these results, it was observed that regardless of what order was used, the same elevation, 2574.35 m was computed. Both the elevation tracking graphs in Figure 17 and the profiles in Figure 18 through 21 show this. Step two in each of the profile graphs also result in the same contour surface. Therefore, all of the elevation values of the 15 points were classified the same. The rest of the classification throughout the entire dataset followed this pattern for each of the four orders.

The erosion functions extract two profiles one in row and the other in column direction. A segment of each profile the length of the window size is examined. Combining these two segments together generates a rectangular area in which the
Figure 19. Order 2 profile tracking from the center of Block 13 in the Daniel Forest Dataset. The x-axis starts at the center of Block 13 as zero. The profile includes 5 meters of lidar points in the column direction. A through D steps show the profiles after each erosion and dilation function.

minimum elevation is searched for. Despite which direction was looked at first or second, the minimum elevation was always returned from the combined segment area. The final eroded elevation was used as the initial elevation for the two different dilation function orders. The two dilation orders of column or row first resulted in the final maximum elevation value of 2577.34 m as seen in Figure 17. The same rectangular area was used but this time the maximum elevation was returned. Therefore, the rectangular area specified by the window size becomes the location by which minimum and maximum values are selected. The area viewed determines the elevation values selected not the order of the row or column functions. For these reasons it is concluded that no
Figure 20. Order 3 profile tracking from the center of Block 13 in the Daniel Forest Dataset. The x-axis starts at the center of Block 13 as zero. The profile includes 5 meters of lidar points in the column direction. A through D steps show the profiles after each erosion and dilation function.

bias is involved with the ordering of the row and column functions.

Results of Daniel Forest Grid Rotation Analysis

The error analysis of the grid rotation for Daniel Forest has been conducted. The results are represented in graphical form to show the changes of accuracy over all the different angles of rotation from 0 to 360 degrees at 5-degree increments. The Type I errors, Type II errors, percent correctly classified, and weighted scores are plotted against the different angles of rotation as shown in Figure 22. It is found that the percentage of classified correctly ranges between 98% and 99%. The top four classifying angles for
Figure 21. Order 4 profile tracking from the center of Block 13 in the Daniel Forest Dataset. The x-axis starts at the center of Block 13 as zero. The profile includes 5 meters of lidar points in the column direction. A through D steps show the profiles after each erosion and dilation function.

minimizing Type I errors occurred at angles near 70, 125, 250, and 310 degrees as seen in the troughs of Figure 22A. Type II errors were minimized in the two troughs of Figure 22B at 25 and 205 degrees. The maximum values for the percent correctly classified give the best classifying angles, while the minimum values for the weighted scores give the best classifying angles. Therefore, the peaks of Figure 22C and the troughs of Figure 22D line up very close to one another and give the best classifying angles. Angles near 70, 125, 240, and 310 degrees were found to have the maximum percent values for the correctly classified criteria. Finally, weighted scores were found to have troughs at approximately 70, 170, 240, and 320 degrees. These angles were the most accurate
classification directions for the dataset. Note the 180 degree differences as shown in Figure 22A through 22D.

The Type I error and weighted score graphs in Figure 22A and 22D have the same profile shape. This is due to the large amount of Type I errors compared to the Type II errors in the filtered dataset. The ratio of Type II to Type I errors on average was 1 to 2.6. Therefore, the weighting of three for the Type II errors did not have a large influence on the measure of success in this dataset.

The Type II errors are spread randomly throughout the dataset while the Type I errors are concentrated in a couple of areas. One of the areas with high Type I errors follows a ridgeline. The location of the ridge in the dataset is marked with a red box around Blocks 51, 52, 56, and 57 in Figure 23. The next area was along a road with a ditch on one side and steep slope on the other side. The lidar points along the road were misclassified as the window moved across the raised middle part of the road. This road’s location is marked with a blue box around Blocks 59, 60, 64, and 65 in Figure 23. Both areas are examined at different classification grid angles.

Figures 24 through 26 show the classification results for the filtered data at the ridgeline area with angles 0, 120, and 240 degrees, respectively. The yellow, red, and blue points represent Type I, Type II, and correctly classified points, respectively. These three angles are illustrative of the differences in the classification of the ridgeline. The top of the ridge was misclassified at angles of 0 and 120 degrees as seen in Figures 24 and 25. More Type I errors occurred at 120 degrees when the algorithm was run perpendicular to the ridge. The ridgeline runs through this area of the dataset near an
Figure 22. Error analysis graphs showing the effect of rotation on the A) Type I Errors, B) Type II Errors, C) Percent Correctly Classified and D) Weighted Scores from angles of 0 to 355 degrees for the Daniel Forest dataset.
Figure 23. A ridgeline and road were found to have high numbers of Type I errors. The red box shows the location of the ridgeline and the blue box the road location.

angle of 240 degrees. When the grid rotation angle is parallel with the ridgeline, as it is at 240 degrees in Figure 26, it is classified correctly. This is consistent with the results reported by Zhang (2003).

Figures 27 through 29 show the results of the filtered data at a road area in the T.W. Daniel Experimental Forest with grid angles 0, 30, and 320 degrees, respectively. Depending on the grid angle relative to the orientation of the sloped surfaces along the roads, the actual road surface and edge of the dataset are misclassified as Type I errors to
Figure 24. Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis grid angle of 0 degrees azimuth in the Daniel Forest. Computational blocks are delineated in green. A ridgeline runs through this area of the dataset near an angle of 240 degrees azimuth.
Figure 25. Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis grid angle of 120 degrees azimuth in the Daniel Forest. Computational blocks are delineated in green. A ridgeline runs through this area of the dataset near an angle of 240 degrees azimuth.
Figure 26. Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis grid angle of 240 degrees azimuth in the Daniel Forest. Computational blocks are delineated in green. A ridgeline runs through this area of the dataset near an angle of 240 degrees azimuth.
varying degrees. Figure 27 shows that the slopes along the road in Block 65 are misclassified given a grid angle of 0 degrees. When the grid angle reached 30 degrees the actual road surface is removed from the ground surface and the edge error misclassification is seen in Block 60 of Figure 28. At a grid angle of 320 degrees both the road and the edge of the dataset are classified correctly as shown in Figure 29.

Just as in the case of the ridge, when the orientation of the grid roughly coincides with the orientation of the road, fewer misclassifications occur. Thus the grid bias associated with the Progressive Morphological Filter, as implemented by Zhang et al. (2003), has been demonstrated.

Development of Methods to Remove Grid Bias

Description of methods

Given the obvious biases in the above results, finding ways to merge the effective angles together and to remove the bias angles was desired in order to generate a better overall filtered dataset. After some analysis, two new combination methods were developed.

The first new method is called the “Best Two Angle” (BTA) method. This method uses the best two angle directions, i.e. the angles that resulted in the least error and combines them together. Given the prior results of comparing the filter against the truth data, it is possible to select the two most successful directions. These directions were found to be mostly dictated by specific terrain shape and building orientations as discussed in the previous section. This method cannot be used in general practice
Figure 27. Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis grid angle of 0 degrees azimuth in the Daniel Forest. Computational blocks are delineated in green. A road runs through this area of the dataset.
Figure 28. Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis grid angle of 30 degrees azimuth in the Daniel Forest. Computational blocks are delineated in green. A road runs through this area of the dataset.
Figure 29. Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis grid angle of 320 degrees azimuth in the Daniel Forest. Computational blocks are delineated in green. A road runs through this area of the dataset.
because of the absence of the truth data needed to determine which angle is best. However, it does provide insight into how much classification improvement could be made in theory if the two preferred orientation angles (e.g., NW-SE and NE-SW angles in a city) were known prior.

Two different BTA combination techniques have been used with this method. The first technique requires both angles to classify a given point as ground for it to be added to the combined surface. This is referred to as the best two angle “both” (BTAB) technique. On the other hand, the second technique only requires a ground point classification at only one angle before it is added to the combined surface. This technique is referred to as the best two angle “either” (BTAE) technique. By using the best two angle directions and these two techniques the Type I or Type II errors were able to be decreased. The results were compared with the “truth” dataset and figures presented to visually show the filtering of the same selected problem areas.

The second method for removing grid bias, one that does not require a prior truth data, is called the “Multiple Angle” (MA) method. This method uses all 72 of the angle directions (5-degree increments from 0 to 360 degrees) to generate a combined result. The requirement for adding a point to the combined surface is based on the number of times the point is classified as ground over all of the angle directions. This is converted to a percentage of angles included and is used as the parameter for combining the different angle results. For example, if this so-called MA% parameter was set to 60%, it is required that 60% of the 72 angle directions have to result in a ground classification in order for that individual point to be added to the combined surface. Therefore, at an
MA% of 100, all of the angles must classify the point as ground before the point can be counted as ground in the combined surface.

Objective optimization of the MA% parameter would require a truth dataset in practice. In this analysis, the parameter has been varied from 5% to 100% in 5% increments to test for the best combined result as determined by comparisons with the truth dataset. An error analysis table was created for each tested MA% parameter. Graphs have then been created from these tables showing how (1) the Type I and Type II errors, and (2) the percentage of correctly classified and weighted scores change over the percentages of angle files included. A figure showing the MA% that yields the highest weighted score has been generated along with a table comparing the two best angle directions and the two combining methods. These table and figures have then been used to find the best classification method for a given dataset as discussed in the next section.

Results

The two techniques associated with the BTA method were used to combine the two “best” (lowest weighted score) angle directions of 240 and 320 degrees. Figures 30 and 31 show the “both” (BTAB) technique results of the ridgeline and road areas respectively. The “either” (BTAE) technique results for the ridgeline and road areas are found in Figures 32 and 33, respectively. These results, along with the results of the MA method are discussed later in this section.

Type I errors and Type II errors, as graphed in Figures 34A and 34B, increase and decrease respectively, as the percentage of angles included approach 100%. When 100% is reached the Type II errors are minimized and the Type I errors are maximized. It was
found that the maximum percent correctly classified takes place at about 45\%, when Type I errors are small and right before the Type II errors start to increase as shown in Figure 34C. The best weighted score as shown in Figure 34D occurs at about 55\% were the Type I and II errors are balanced.

The point cloud was examined to see how combining the classifications from the different angle directions affected the distribution of errors in the ridgeline and road areas. Figure 35 and 36 shows the results of combining 55\% of the angles together at the ridgeline and road areas, respectively. A section of the ridge remains misclassified as seen in Figure 35. All of the roadway surfaces and almost all of the slopes along the roads are classified correctly in Figure 36.

The comparison of the two best single angle classifications of 240 and 320 degrees against the three combining methods tested are found in Table 6.

The analysis in Table 6 concludes that grid angles near 70/240 degrees result in two of the four lowest weighted scores. These angles are nearly 180 degrees a part and represent roughly the same azimuth direction. In other words when the algorithm was run in opposite directions but along the same azimuth, similar scores resulted. The ridgeline as shown in Figure 23 runs in about the same direction as this azimuth. When the algorithm was run parallel to the ridgeline it was classified correctly with the windows opening in the same direction of the ridge as seen in Figure 26.

The grid angles of 170 and 320 degrees were the next highest classifiers. The road area at the bottom right side of the dataset as shown in Figure 23 included a ditch on one side and steep slopes on the other. The road and edge of the dataset are positioned at
Figure 30. Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis of the Best Two Angle Both (BTAB) combination method of the Daniel Forest. Computational blocks are delineated in green. A ridgeline runs through this area of the dataset near an angle of 240 degrees azimuth.
Figure 31. Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis of the Best Two Angle Both (BTAB) combination method of the Daniel Forest. Computational blocks are delineated in green. A road runs through this area of the dataset.
Figure 32. Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis of the Best Two Angle Either (BTAE) combination method of the Daniel Forest. Computational blocks are delineated in green. A ridgeline runs through this area of the dataset near an angle of 240 degrees azimuth.
Figure 33. Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis of the Best Two Angle Either (BTAE) combination method of the Daniel Forest. Computational blocks are delineated in green. A road runs through this area of the dataset.
roughly 320 degrees and was best classified when the grid angle was parallel to them as seen in Figure 29. Therefore, this angle removed the edge error misclassifications at the top and bottom of the dataset. The left and right edges of the dataset were corrected with the 170 degree grid angle lining up parallel with them.

Therefore, the angles at 70/240, 170, and 320 degrees appear to be the best angles for classification because of the orientation of the terrain, specifically the ridgeline, road and edges of the dataset. The improvement of the classification results over the different angle directions confirms there is a marked grid bias within the algorithm.
Figure 35. Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis of the Multiple Angle (MA) combination method that uses 55% of angles from the Daniel Forest. Computational blocks are delineated in green. A ridgeline runs through this area of the dataset.
Figure 36. Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis of the Multiple Angle (MA) combination method that uses 55% of angles from the Daniel Forest. Computational blocks are delineated in green. A road runs through this area of the dataset.
Table 6. Comparison of the best two angles 240 and 320 degrees, Best Two Angle Both (BTAB), Best Two Angle Either (BTAE) and Multiple Angle (MA) combining methods for the Daniel Forest dataset

<table>
<thead>
<tr>
<th>Result Type</th>
<th>Type I</th>
<th>Type II</th>
<th>Type I %</th>
<th>Type II %</th>
<th>Total Errors %</th>
<th>% Correctly Classified</th>
<th>Weighted Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>240°</td>
<td>12,171</td>
<td>5,281</td>
<td>0.93%</td>
<td>0.40%</td>
<td>1.33%</td>
<td>98.67%</td>
<td>28,014</td>
</tr>
<tr>
<td>320°</td>
<td>10,351</td>
<td>5,436</td>
<td>0.76%</td>
<td>0.41%</td>
<td>1.20%</td>
<td>98.80%</td>
<td>26,659</td>
</tr>
<tr>
<td>BTAB</td>
<td>20,146</td>
<td>2,941</td>
<td>1.54%</td>
<td>0.22%</td>
<td>1.76%</td>
<td>98.24%</td>
<td>28,969</td>
</tr>
<tr>
<td>BTAE</td>
<td>2,376</td>
<td>7,776</td>
<td>0.18%</td>
<td>0.59%</td>
<td>0.77%</td>
<td>99.23%</td>
<td>25,704</td>
</tr>
<tr>
<td>MA @ 55%</td>
<td>3,481</td>
<td>5,711</td>
<td>0.27%</td>
<td>0.44%</td>
<td>0.70%</td>
<td>99.30%</td>
<td>20,614</td>
</tr>
</tbody>
</table>

The ditches along both sides of the road caused the linear road surfaces to be removed from the terrain surface at different angles as seen in Figures 27 and 28. The direction in which the profile crossed the road affected whether it was classified correctly or not. The shorter the distance of the profile across the linear road surface the more likely it was to be misclassified. As seen in Figure 28, at 30 degrees, the short profile lengths in block 65 can be seen running across or perpendicular to the road and resulting in Type I errors. On the other hand in Figure 29, the angle of 320 degrees runs parallel to the road and classifies it correctly. Therefore, when the direction of the profiles run parallel to the roadway and slopes the ground surface is maintained, but when perpendicular to them they are removed from the ground surface.

It was found that the BTAE technique highly reduced the Type I errors as shown in Figures 32 and 33. However, at the same time it increased the Type II errors by enough to raise the weighted score above the MA analysis of 55% as seen in Table 6. This was because a greater variety of grid angles were needed to properly classify the different orientations of the roads, ridge and other ground terrain objects in the dataset.
It was found that the MA method using a MA% of 55% yielded the best overall weighted score. This percentage gave the best classification because it allowed many of the Type I error to be corrected while balancing the number of the Type II errors. This was suitable in this dataset with the ratio of Type II errors to Type I errors at 1 to 2.6. By combining the angles at different directions the Type II errors are increased when compared to the best single angle direction from the error analysis as seen in Table 6. Nevertheless, due to the small number of Type II errors, the overall classification of the dataset was improved. The combination of the many different angle directions permitted the profiles to line up along the many road surfaces, slopes, ridges and dataset edges in the dataset. It was thereby able to classify them correctly and add them to the final combined result. It was found that the MA method with an MA% of 55% decreased the weighted score by 26% over the best classified angle at 240 degrees as seen in Table 6. This method combining of angles proved to successfully add more correctly classified ground points to this forested rural dataset and improve the detail of the DTM.

**USU Campus**

The error analysis for the USU Campus dataset, as shown in Figure 37A-D, gives the trends of the Type I errors, Type II errors, percent correctly classified and weighted scores. Type I errors were minimized at grid angles near 85 and 265 degrees. It was found that the Type II errors, percent correctly classified and weighted scores all have maxima/minima at grid angles of 50, 140, 230, and 315 degrees. The angles are the same for these three criteria because the ratio of Type I to Type II errors is 1 to 1.99. Type II errors, being weighted at three, also increased the impact on the weighted scores. This is
why the Type II error and weighted score graphs in Figures 37B and 37D follow the same pattern. Note the 180 degrees differences in the peaks and troughs of the graphs in Figure 37A through 37D.

It was found that a large amount of Type II errors occurred in the misclassification of both large and small buildings. An area located in the center of the dataset was selected to analyze the misclassification patterns of large and small buildings as shown in Figure 38. Grid angles of 0, 230 and 315 degrees were selected to analyze the misclassification patterns and are shown in Figures 39 through 41. The filter window size was set at 32 meters in all cases. This length is delineated as a white scale bar in the figures. In Figure 39 it can be seen that the small building in Block 52 and the larger buildings were not removed from the ground surface due to their dimensions being greater than the window size and being parallel with the direction of the filter. At the angle of 230 degrees the smaller building in Block 52 and the corners of the large buildings are removed as seen in Figure 40. When the grid angle is changed to 315 degrees, the small building and the opposite corners of the large buildings are removed as shown in Figure 41. The two angles of 230 and 315 degrees are 90 degrees a part and cross the long sides of the buildings near a 45 degree angle.

The best classified angles of 230 and 315 degrees were combined in the BTA method using the BTAB and BTAE techniques, and are shown in Figures 42 and 43. Type I errors are minimized in the BTAE method and Type II errors are minimized in the BTAB method.
Figure 37. Error analysis graphs showing the effect of rotation on the A) Type I Errors, B) Type II Errors, C) Percent Correctly Classified and D) Weighted Scores from angles of 0 to 355 degrees for the USU Campus dataset.
Figure 38. The location of the misclassification analyze for the USU Campus dataset is specified by the red box. Large and small buildings are found within this area.
Figure 39. Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis grid angle of 0 degrees azimuth in the USU Campus. Computational blocks are delineated in green and the filter window length of 32 meters in white, shown in the direction of the azimuth. Large and small buildings are found in this area of the dataset.
Figure 40. Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis grid angle of 230 degrees azimuth in the USU Campus. Computational blocks are delineated in green and the filter window length of 32 meters in white, shown in the direction of the azimuth. Large and small buildings are found in this area of the dataset.
Figure 41. Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis grid angle of 315 degrees azimuth in the USU Campus. Computational blocks are delineated in green and the filter window length of 32 meters in white, shown in the direction of the azimuth. Large and small buildings are found in this area of the dataset.
Figure 42. Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis of the Best Two Angle Both (BTAB) combination method of the USU Campus. Computational blocks are delineated in green. Large and small buildings are found in this area of the dataset.
Figure 43. Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis of the Best Two Angle Either (BTAE) combination method of the USU Campus. Computational blocks are delineated in green. Large and small buildings are found in this area of the dataset.
Figure 44. The Multiple Angle (MA) combination method error analysis graphs. Showing the effect of the percent of angles included on the A) Type I Errors, B) Type II Errors, C) Percent Correctly Classified and D) Weighted Scores from 5 to 100 percent for the USU Campus dataset.

The MA method was applied to this dataset using various values of MA%, the results of which are presented in the graphs shown in Figure 44. The Type I and Type II errors followed the similar pattern as the Daniel Forest dataset of increasing and decreasing, respectively as 100% of the angles included were approached as seen in Figure 44A and 44B. By having more Type II errors in the dataset, the best weighted score was found to be closer to the right of the graph at 80% and the percent correctly classified at 70%. In the point cloud of Figure 45, the combined result at 80% shows a large amount of Type II errors compared to Type I errors. The amount of Type II errors was minimized at higher MA% values.
The comparison of the two best single angle classifications of 230 and 315 degrees against the three combining methods tested are found in Table 7. The combining of the angles 230 and 315 degrees with the BTAB method yielded the best overall weighted score by decreasing the Type II errors as seen in Figure 42. However, a large number of misclassifications remain.

The Type I errors in the dataset were best corrected along the 85/265 degrees azimuth direction. A large number of these errors occurred along a road at the bottom of the dataset. The road runs along a canal with a drop off into the canal and a steep slope on the other side. The angles mentioned above are parallel to the road and best maintain the ground surface on the road -- just as was found in with the Daniel Forest dataset. Also several hilltops and ridgeline areas produced Type I errors.

The Type II errors in this dataset had the greatest impact on the results due to the large numbers of buildings. Almost all of the buildings in the dataset were positioned with the longest dimension of the structures facing the east-west direction. Profiles along those long linear buildings would not remove them from the ground surface because of the window size being smaller than the buildings. When the profiles were run at angle of 50/230 and 140/315 degrees the corners of the large buildings were able to be removed. For the smaller buildings many of them were completely removed because of the profiles running along the smaller length sections of the roofs. These angles gave the two best directions for increasing correctly classified and decreasing both the Type II errors and the weighted scores.
Figure 45. Type I (yellow), Type II (red) and correctly classified points (blue) for an analysis of the Multiple Angle (MA) combination method that uses 80% of angles from the USU Campus. Computational blocks are delineated in green. Large and small buildings are found in this area of the dataset.
Table 7. Comparison of the best two angles 230 and 315 degrees, Best Two Angle Both (BTAB), Best Two Angle Either (BTAE) and Multiple Angle (MA) combining methods for the USU Campus dataset

<table>
<thead>
<tr>
<th>Result Type</th>
<th>Type I</th>
<th>Type II</th>
<th>Type I %</th>
<th>Type II %</th>
<th>Total Errors %</th>
<th>% Correctly Classified</th>
<th>Weighted Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>230°</td>
<td>24,017</td>
<td>39,108</td>
<td>1.68%</td>
<td>2.74%</td>
<td>4.42%</td>
<td>95.58%</td>
<td>141,341</td>
</tr>
<tr>
<td>315°</td>
<td>24,682</td>
<td>41,236</td>
<td>1.73%</td>
<td>2.89%</td>
<td>4.62%</td>
<td>95.38%</td>
<td>148,390</td>
</tr>
<tr>
<td>BTAB</td>
<td>42,812</td>
<td>25,593</td>
<td>3.00%</td>
<td>1.79%</td>
<td>4.79%</td>
<td>95.21%</td>
<td>119,591</td>
</tr>
<tr>
<td>BTAE</td>
<td>5,887</td>
<td>54,751</td>
<td>0.41%</td>
<td>3.83%</td>
<td>4.25%</td>
<td>95.75%</td>
<td>170,140</td>
</tr>
<tr>
<td>MA @ 80%</td>
<td>11,412</td>
<td>53,733</td>
<td>0.80%</td>
<td>3.70%</td>
<td>4.56%</td>
<td>95.44%</td>
<td>172,611</td>
</tr>
</tbody>
</table>

The large buildings corners were removed at the above mentioned angles but the center of the buildings were not removed. When the window size was increased and tested in the input parameter section of the report, large numbers of Type I errors began to occur in the surrounding dataset. The edge misclassification error also increased and caused greater errors in the overall dataset. For better classification, the dataset may need to be broken up into small and large building areas. These areas would need to be processed separately with different maximum window sizes.

The MA method at 80% of the angle included had a weighted score of 18% greater than the 230 degree grid angle weighted score. Therefore, the combining of multiple angles in this dataset with high Type II errors did not improve the overall classification of the dataset. Combining the angles this way does not improve results when Type II errors are high and are needing to be decreased.

On the other hand, the combining of 230 and 315 degrees in the BTAB method resulted in a weighted score of 15% less than the 230 degree angle. This was improved because the Type II errors were decreased by adding only the points that were classified as ground in both of the angles. The small buildings and the corners of the large
buildings were best removed as seen in Figure 42. Even though the Type I errors increased it was not as significant as the decrease of the Type II errors. In this dataset the best angle directions were located at the crossing of the short length of the buildings near 230 and 315 degrees. These angles are nearly 90 degrees a part from each other and cross at an angle of 45 degrees relative to the dominant N-S, E-W orientation of the buildings. Figure 40 shows the removal of the Type II errors at the top-left and bottom-right corners of the buildings when the grid angle is set at 230 degrees. Using a grid angle 315 degrees as shown in Figure 41, the opposite corners of the top-right and bottom-left are removed. When combining these two results using the BTAE methods, the Type II errors were combined, leaving all of the corners of the buildings misclassified. However, the results of BTAB method given in Figure 42 show the removal of all the errors from the corners of the buildings. Using this BTAB combining method gave the best removal of the small buildings and the corners of all the larger buildings. The use of only two angles also proved more accurate because almost all of the buildings were positioned and facing the same direction along the same azimuth direction 0/180 degrees. If more buildings had been placed at a variety of angles, the MA method may have worked better. The BTAB method works best at removing Type II errors. The BTAE method gives the best results if Type I errors are desired to be removed.

These results suggest that the selected directions of the grid have a great affect on the ability to remove buildings and roads with steep slopes along them. This further illustrates that there is a significant grid bias in the progressive morphological filter that is
difficult to remove. However, it has also been shown that the approach of performing a classification at different grid angles and combining those results in optimal ways can significantly improve the accuracy of the generated DTMs.
CONCLUSIONS

This thesis has reviewed the different algorithms that have been developed for filtering and classifying lidar data. Because of some of its observed weaknesses, the progressive morphological filter by Zhang et al. (2003) was selected to evaluate and improve. The two weaknesses included the row-column order and grid orientation biases.

The existing algorithm has been explained along with the proposed improvements for overcoming the potential bias. The measures of success of the improvements included Type I and Type II errors where results were compared with a hand-produced “truth” classification dataset. Two datasets were selected for testing the modified filters and the optimum input parameters for the algorithms were calculated using sensitivity plots by trial and error. The results were presented and discussed for each algorithm.

It has been found that the four different row-column orders tests ended up classifying the dataset exactly the same. After the erosion functions were completed, the same resulting surface profiles and elevations were produced regardless of which direction of row or column came first. The same result was found for the dilation functions. The filter windows used by the algorithm were found to create a rectangular area were the minimum and maximum values within that area were always selected. Therefore, the row-column orders did not create a bias in the classification as was hypothesized.

Classification results were found to be greatly influenced by the grid rotation angle and the nature of the surfaces in the two lidar datasets tested. The rural, forested
area of the T.W. Daniels Experimental Watershed resulted in more Type I errors and the urban USU Campus area resulted in more Type II errors. Problem areas in both datasets for Type I errors occurred along roads with ditches and steep slopes running along them as well as ridgelines. The edges of the datasets also gave Type I errors. However, this misclassification was due to the edge error problem caused by the window reaching the edge of the data on uphill slopes. This problem has been largely corrected using buffer zones around each block. The best angles for removing the Type I errors in both datasets were found to be parallel with the centerline of roads, the axes of slopes and along the edges of the datasets. Type II error problems were found to be mostly associated with the misclassification of large and small buildings. Angles that crossed the short length of the buildings at 45 degrees from the building roof best removed the Type II errors. The smaller buildings were almost entirely removed and the corners of the larger buildings were removed with these grid angles. The selected angle directions significantly affected the classification results. Therefore, the grid orientation bias was verified in the existing progressive morphological filter.

Two methods of combining the angle results were developed and used. The Best Two Angle (BTA) method used the two top angle direction classifiers from the Type I and II error analysis. Two different techniques of “both” (BTAB) and “either” (BTAE) were used to combine these angles. The BTAB technique required that both of the angles classify a point as ground before it was added to the combined surface. The BTAE technique required only one of the angles to classify a point as ground in order to add it to the combined surface.
The second combining method developed and tested was the Multiple Angle (MA) method. Ground points were added to the combined surface according to the number of times they were classified as ground in all of the 72 angle directions. The percent of angles required to classify a point as ground for the combined surface was tested from 5% to 100% at 5% increments. The optimum percentage of angles to be included in the combined surface was selected based on the error analysis results.

For the Daniel Forest dataset the MA method performed best in classifying the terrain objects which were in many different orientations. The optimum percentage of angles included was 55%. This decreased the high number of Type I errors in the dataset. In the USU Campus dataset, the BTAB method gave the best combining results. High numbers of Type II errors occurred in the dataset with the large number of buildings. All of the buildings were positioned in similar directions. The best two angles that were used in the method crossed the short length of the building’s roofs at angles 230 and 315 degrees. These angles crossed near 45 degrees in comparison to the roof alignment. The Type II errors were decreased by using this method.

One of the most pressing problems with lidar classification continues to be the changing success of algorithms according to terrain type (Sithole and Vosselman, 2004). This has been shown through the classification results of these two datasets. A variation of each combining method gave improved classification results in both urban and rural types of terrain. Therefore, this method holds the promise of being able to improve the classification of a variety of different terrain types. The two methods of combining a variety of grid angles developed in this thesis improved classification results, but had to
be adapted to the differing surface geometries represented by the lidar data. To minimize the Type I errors the BTAE method and the MA method with a MA% parameter closer to zero gave the best results. When minimizing the Type II errors the BTAB method and MA method with a MA% parameter closer to 100 gave the best results. Also, in datasets with terrain objects oriented in numerous directions, the MA method performs best. In contrast, the BTA methods perform best when terrain objects are positioned in similar directions. By combining the angle results using these two methods, the overall classification improved and more accurate DTM's were generated.
RECOMMENDATIONS

Several basic guidelines have been constructed for implementation of the grid rotation algorithm. Those desiring to use this algorithm for lidar classification in practice without a “truth” dataset can better gain quality DTMs from filtered datasets by following these rules. These guidelines are to direct the user in selecting which combining method to use along with its parameters.

It is recommended to use aerial photographs, or colored lidar data points wherever possible, to better distinguish the type of land terrain features and their orientations. For datasets, whether rural or urban, with numerous terrain objects at different grid orientations it is recommended to use the MA combining method. To obtain a better overall classification with balanced Type I and II errors it is suggested to require 50% of the angles to classify a point as ground before it is added to the DTM. If the user desires a lower amount of Type I errors it is suggested to decrease the MA% parameter. If a lower amount of Type II errors are preferred then the MA% should be increased.

The BTA combining method is recommended for datasets with terrain objects orientated in a consistent direction. An example is an urban area with the majority of the buildings facing the same direction. It is best to select the two grid angles that cross the short length of the building roofs at 45 degrees and use the BTAB method for the urban area. This will decrease the amount of Type II errors in the filtered dataset. A rural area with only two or three large linear features such as ridges and roads would also benefit from using this combining method. For this type of area, the selected grid angles need to be parallel with the long linear terrain features. Combine these two angles using the
BTAE method to decrease the amount of Type I errors that are associated with this type of dataset. By using these guidelines better DTMs can be developed in a variety of different terrain types.

The progressive morphological filter algorithm and the two combining methods need continued improvement and testing. Several areas and ways to accomplished this are recommended. First, it is recommended that further testing of the row-column orders could include two more orders that were not tested in this research. The first order could be the erosion and dilation in the row direction, followed by erosion and dilation in the column direction. The second order could be reversed with column first, then row second. It is hypothesized that by eroding and dilating in one direction and then in the other direction will change the input of erosion functions and therefore change the ending DTM. Comparisons could then be generated to test for bias of these orders as using the methodology from this research.

The second suggestion for improvement deals with window size parameter selection. The window parameter may not have been increased to the degree that might be useful for classifying the larger buildings in the area. Further analysis with different window size parameters could better explore the filtering effects on both small and large buildings. These results should also look at the Type I errors and edge errors increases that may occur with a larger window size. If the window size parameter was not changed a possible alternative would be to remove the buildings before running the grid rotation algorithm. Specific algorithms have been developed for solely removing buildings, but have not been reviewed in this research. It is suggested that a building algorithm be
found and used before the classification of the grid rotation algorithm to compare the filtering results.

Lastly, it is suggested that combining of the grid angles be improved. The MA method used the input parameters that were generated from the sensitivity plots using only one grid angle direction. To test the full capacity of the MA method, the input parameters need to be selected using the combining of multiple grid angles. It is suggested that better optimized input parameters could be found best from developing sensitivity plots using MA combining method. It is suggested that the sensitivity plots for each of the input parameters be generated by processing all of the angle directions and combining the results using the MA method. It is predicted that better input parameters can be found from this procedure and possibly increase the overall classification results.

The two combining methods of the grid angles decreased only Type I errors or Type II errors. The BTA method nor the MA method decreased both of the error types at the same time. It is suggested that a method be developed to remove less effective grid angle directions and combine remaining angles in a way that will decrease both Type I and Type II errors at the same time.
REFERENCES


Soininen, A. 2010. TerraScan user's guide. Terrasolid: Helsinki, Finland.


Zhang, K., and Z. Cui. 2007. Airborne LIDAR data processing and analysis tools. International Hurriance Research Center, Department of Environmental Studies, Florida International University. 1-81.