An Improved Simple Morphological Filter for the Terrain Classification of Airborne LIDAR Data

Article *in* ISPRS Journal of Photogrammetry and Remote Sensing · March 2013

DOI: 10.1016/j.isprsjprs.2012.12.002

CITATIONS
275

READS
12,748

3 authors:

Thomas James Pingel
Binghamton University
53 PUBLICATIONS **21,408** CITATIONS
SEE PROFILE

Keith Clarke
University of California, Santa Barbara
284 PUBLICATIONS **17,870** CITATIONS
SEE PROFILE

William A. Mcbride
University of California, Santa Barbara
2 PUBLICATIONS **274** CITATIONS
SEE PROFILE

ISPRS Journal of Photogrammetry and Remote Sensing 77 (2013) 21–30

Contents lists available at SciVerse ScienceDirect

ISPRS Journal of Photogrammetry and Remote Sensing

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

An improved simple morphological filter for the terrain classification of airborne LIDAR data

Thomas J. Pingel [a,*], Keith C. Clarke [b], William A. McBride [b]

[a] Northern Illinois University, Department of Geography, DeKalb, IL 60115, USA
[b] University of California, Santa Barbara, Department of Geography, Santa Barbara, CA 93101, USA

ARTICLE INFO

Article history:
Received 22 September 2011
Received in revised form 20 December 2012
Accepted 21 December 2012
Available online 27 January 2013

Keywords:
LIDAR
Classification
Algorithms
DEM/DTM
Virtual reality

ABSTRACT

Terrain classification of LIDAR point clouds is a fundamental problem in the production of Digital Elevation Models (DEMs). The Simple Morphological Filter (SMRF) addresses this problem by applying image processing techniques to the data. This implementation uses a linearly increasing window and simple slope thresholding, along with a novel application of image inpainting techniques. When tested against the ISPRS LIDAR reference dataset, SMRF achieved a mean 85.4% Kappa score when using a single parameter set and 90.02% when optimized. SMRF is intended to serve as a stable base from which more advanced progressive filters can be designed. This approach is particularly effective at minimizing Type I error rates, while maintaining acceptable Type II error rates. As a result, the final surface preserves subtle surface variation in the form of tracks and trails that make this approach ideally suited for the production of DEMs used as ground surfaces in immersive virtual environments.

1. Introduction

One important application of LIght Detection and Ranging (LIDAR) technology is the creation of high-resolution Digital Elevation Models (DEMs) that capture the shape of the terrain in great detail. DEMs are used for a variety of purposes, but one of the most recent is the employment of their derived ground surfaces in immersive geographic virtual environments. While large-extent DEMs, such as those produced from the Shuttle Radar Topography Mission (SRTM) are useful as ground surfaces in global and other large-extent views of the environment (e.g., NASA's World Wind or Google Earth), immersion at a locality requires a higher-fidelity model to convey the correct sense of the terrain within highly varying environments. This need can be met through ground surfaces derived from LIDAR datasets. The most basic form of LIDAR data commonly utilized by researchers is the "point cloud" – a cluster of three-dimensional points and, often, associated attributes like the intensity of the return or color information when laser scanners are integrated with digital cameras.

The challenge with such point clouds is that they are rarely useful in themselves, but must instead be processed and transformed into ground models and representations of objects such as buildings and trees. Dozens of algorithms have been published on the extraction of terrain from the point cloud alone (Meng et al., 2010; Shan and Toth, 2008; Vosselman and Maas, 2010). While most algorithms perform tolerably well on unbuilt and undifferentiated terrain, the task has proven difficult for complex urban and highly rugged environments. There have been many attempts to categorize ground filtering algorithms based on their methodology (e.g., Liu, 2008; Sithole and Vosselman, 2004). Meng et al. (2010) helpfully and systematically identify key dimensions (or attributes) on which algorithms typically differ, including whether they use the first, last, or full LIDAR return set, and whether the raw LIDAR data are used to interpolate a surface or whether they are initially fitted to a gridded data structure.

Algorithms are typically tested against computer simulated datasets for which the "true" ground is known, or else tested ad hoc against available LIDAR scans. In the latter case, qualitative interpretation of the resulting hillshaded surface is common, as is manually verifying accuracy against a selected representative sample of the LIDAR returns (Zhang et al., 2003). In all of these cases, test datasets are idiosyncratic and thus meaningful comparison of performance between algorithms is difficult. To mitigate this problem, the International Society of Photogrammetry and Remote Sensing (ISPRS) commissioned a study in which eight algorithms were tested against seven high-resolution LIDAR datasets, four of which were urban landscapes and three of which were forested landscapes (Sithole and Vosselman, 2003, 2004). Fifteen samples from these larger datasets were selected, and the returns within each sample were manually coded as either ground/bare-

* Corresponding author. Tel.: +1 815 753 0631.
E-mail addresses: tpingel@niu.edu (T.J. Pingel), kclarke@geog.ucsb.edu (K.C. Clarke), wmcb8367@gmail.com (W.A. McBride).

0924-2716/$ - see front matter © 2013 International Society for Photogrammetry and Remote Sensing, Inc. (ISPRS) Published by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.isprsjprs.2012.12.002

earth (BE), or non-ground/object (OBJ) observations. The publication of the findings, and more importantly the datasets used in the study, has enabled more direct comparisons of accuracy against the original eight algorithms and to all subsequent algorithms that utilize the data. Table 1 lists the fifteen ISPRS samples along with several relevant characteristics (Sithole and Vosselman, 2003).

There are multiple metrics of performance for ground filtering algorithms. One metric of accuracy is the Type I error rate, which is equal to the number of BE points mistakenly classified as OBJ divided by the true number of BE points. Another metric is the Type II error rate, which is equal to the number of OBJ points mistakenly classified as BE, divided by true total number of OBJ points. The total error rate is equal to the sum of all mistaken classifications divided by the total number of points in the dataset. The results of the original test were given in these values, and some recent algorithms tested against the ISPRS dataset utilize these metrics (Chen et al., 2007). Other recently published algorithms tested against the ISPRS samples (e.g., Jahromi et al., 2011; Meng et al., 2009; Shao and Chen, 2008; Silván-Cárdenas and Wang, 2006) utilize Cohen's Kappa (Cohen, 1960) statistic as a measure of accuracy. Kappa measures the overall agreement between two judges, while taking into account the possibility of chance in the observed frequencies and is commonly used in its native domain of psychology as well as in remote sensing (Congalton, 1991; Jensen, 2005). The range of Kappa values extends from positive to negative one, with positive one indicating strong agreement, negative one indicating strong disagreement, and zero indicating chance-level agreement.

One of the most robust algorithms for ground filtering is Axelsson's adaptive triangulated irregular network (TIN) model (1999, 2000). While many algorithms require the data to first be fitted to a gridded data structure, Axelsson's algorithm begins with a collection of seed points taken from the original data set, and iteratively adds points from the set that meet criteria related to elevation and directional changes. Of the eight algorithms originally tested, Axelsson's performed the best on twelve of the fifteen samples, had the lowest mean total error rate of 4.82%, and the highest mean Kappa score of 84.19% (Table 2). The second best performing filter originally tested, the Hierarchical Robust Interpolation algorithm of (Kraus and Pfeifer, 1998; Pfeifer et al., 1999), had a mean total error rate of 8.03% and an overall mean Kappa score of 75.27%. It performed the best on two of the fifteen samples (Samples 2–1 and 3–1), both of which were drawn from urban landscapes. The Active Contours algorithm of Elmqvist et al. (2001) performed best on the remaining sample (Sample 4–1), though poor performance on a number of other samples led to a comparatively high mean total error rate (20.73%) and low Kappa score (57.78%).

The excellent performance of Axelsson's filter is a strong argument in favor of retaining and processing all of the original data points before interpolating a Digital Terrain Model (or DTM – used here synonymously with Digital Elevation Model, or DEM). Recent attempts to surpass the performance of Axelsson's adaptive TIN densification method have had mixed results. Jahromi et al. (2011) report preliminary results using an algorithm based on artificial neural networks. They tested their algorithm against four samples, finding improvement over Axelsson's in three cases. Meng et al. (2009) produced a directional ground filtering algorithm that had an overall mean Kappa score of 79.93% (compared to Axelsson's mean Kappa score of 84.19%) but performed better on eleven of fifteen individual samples. Shao's Climbing and Sliding Algorithm (CAS) performed quite well against the ISPRS dataset, though it employed a "pseudo grid" rather than a true grid (Shao, 2007; Shao and Chen, 2008). It improved on Axelsson's performance on nine samples, and reduced mean total error to 4.42% (compared to Axelsson's mean total error of 4.82%). These results indicate that grid-based algorithms can have commensurate performance, and typically do so at far less computational expense. Grid-based approaches also have the advantage of tying into widely available and robust image processing algorithms, and thus cut down on the overhead involved in algorithm development and testing.

Table 1
Study site features, after Sithole and Vosselman (2003, 2004).

Site	Density (m^{-2})	Site	Sample	#	Features
Urban	0.67	1	1–1	1	Mixed vegetation and buildings on hillside
			1–2	2	Mixed vegetation and buildings
		2	2–1	3	Road with bridge
			2–2	4	Bridge and irregular ground surface
			2–3	5	Large, irregularly shaped buildings
			2–4	6	Steep slopes with vegetation
		3	3–1	7	Complex building
		4	4–1	8	Large gaps in data, irregularly shaped buildings
			4–2	9	Trains in railway yard
Rural	0.18	5	5–1	10	Data gaps, vegetation on moderate slopes
			5–2	11	Steep, terraced slopes
			5–3	12	Steep, terraced slopes
			5–4	13	Dense ground cover
		6	6–1	14	Large gap in data
		7	7–1	15	Underpass

Table 2
Best reported performance of top algorithms, total error (%) and Cohen's Kappa (%) (in italics) against ISPRS samples (Axelsson, 1999; Chen et al., 2007; Elmqvist et al. 2001; Jahromi et al., 2011; Meng et al., 2009; Pfeifer et al., 1999; Shao, 2007; Sithole and Vosselman, 2003). Values from Meng et al. (2009) are optimized, while values from Shao (2007) are the best performance out of the six parameter sets tested.

Site	Axelsson	Chen	Elmqvist	Jahromi	Meng	Pfeifer	Shao
1 (1–1)	10.76	13.92	22.40	15.90	–	17.35	11.88
	78.48	–	*56.68*	*68.69*	*70.96*	*66.09*	–
2 (1–2)	3.25	3.61	8.18	4.31	–	4.50	4.02
	93.51	–	*83.66*	*91.37*	*93.12*	*91.00*	–
3 (2–1)	4.25	2.28	8.53	0.40	–	2.57	4.67
	86.34	–	*77.40*	*98.83*	*95.40*	*92.51*	–
4 (2–2)	3.63	3.61	8.93	–	–	6.71	5.51
	91.33	–	*80.30*	–	*88.75*	*84.68*	–
5 (2–3)	4.00	9.05	12.28	–	–	8.22	4.80
	91.97	–	*75.59*	–	*87.56*	*83.59*	–
6 (2–4)	4.42	3.61	13.83	–	–	8.64	4.97
	88.50	–	*68.89*	–	*83.39*	*78.43*	-
7 (3–1)	4.78	1.27	5.34	1.32	–	1.80	1.21
	90.43	–	*89.31*	*97.34*	*97.45*	*96.37*	–
8 (4–1)	13.91	34.03	8.76	–	–	10.75	4.91
	72.21	–	*82.46*	–	*88.58*	*78.51*	–
9 (4–2)	1.62	2.20	3.68	–	–	2.64	2.14
	96.15	–	*90.86*	-	*97.25*	*93.67*	–
10 (5–1)	2.72	2.24	21.31	-	–	3.71	3.60
	91.68	–	*52.74*	–	*87.20*	*89.61*	–
11 (5–2)	3.07	11.52	57.95	–	–	19.64	2.80
	83.63	–	*9.36*	–	*65.57*	*41.02*	–
12 (5–3)	8.91	13.09	48.45	–	–	12.60	5.27
	39.13	–	*7.05*	–	*31.25*	*30.83*	–
13 (5–4)	3.23	2.91	21.26	–	–	5.47	2.74
	93.52	–	*55.88*	–	*92.71*	*88.93*	–
14 (6–1)	2.08	2.01	35.87	–	–	6.91	1.38
	74.52	–	*10.31*	–	*52.43*	*47.09*	–
15 (7–1)	1.63	3.04	34.22	–	–	8.85	3.12
	91.44	–	*26.26*	–	*67.36*	*66.75*	–
Mean	4.82	7.23	20.73	–	–	8.03	4.20
	84.19	–	*57.78*	–	*79.93*	*75.27*	–
Median	3.63	3.61	13.83	–	–	6.91	4.02
	90.43	–	*68.89*	–	*87.56*	*83.59*	–

Kilian et al. (1996) proposed a progressive morphological filter based on a series of opening operations applied to a gridded surface model. This model was later developed into a working algorithm by Zhang et al. (2003). At each grid node, the nearest, lowest value was selected, thus creating a "minimum surface." To this minimum surface (ZI_{min}), an opening operation was applied. Opening is an image processing technique whereby the algorithm searches for relative highs within a neighborhood defined by a structuring element (usually shaped like a disk or square) and pulls these high values down to the included low (or background) values. In Zhang's algorithm, if the difference in elevation between the original image and the opened image is above a threshold, it is flagged as a non-ground or object (OBJ) point. This process continues, with progressively larger structuring elements, until the window size is larger than the largest feature to be removed (e.g., a very large building). As the window size increases, the permissible elevation difference threshold also increases at a rate governed by a supplied slope parameter and the difference in size between the current and last window. The final DEM is interpolated using kriging, a geostatistical method of estimation (Wackernagel, 1998). Although Zhang's filter performed well in both urban and rugged forested environments, it performs poorly in high-relief urban areas. In these cases, the filter can mistake the tops of hills for buildings and remove them, thus yielding high Type I error rates and distorted bare earth models. Unfortunately, the publication of Zhang et al. (2003) was coincident with Sithole and Vosselman (2003), and so a direct measure of performance of their algorithm against the ISPRS dataset was never made.

Chen et al. (2007), in a technique reminiscent of Vosselman (2000), improved upon the basic technique by adding a condition whereby the edge pixels of large features selected for potential removal were evaluated to see if elevation changes were gradual or sudden. If the changes on the periphery were gradual, the feature was retained, whereas if the elevation differences were large, the feature was identified as non-ground and removed in much the same way as in Zhang et al. (2003). In both of these filters, the ground points that remained from the original minimum surface were used to interpolate a new ground surface, and any original LIDAR point lying within a specified distance (often 0.5 m) was flagged as ground, while all other points were flagged as non-ground. The description of the algorithm also contained a modified minimum surface generation scheme in which large discontinuities in the data set are filled according to the lowest value along their boundary. This feature allows for a better infill of water bodies which tend not to reflect LIDAR pulses. Were this addition not included in minimum surface generation, trees and other high features near the edges of water bodies could severely distort the local area of the minimum surface and make ground identification quite challenging.

Chen et al. (2007) published the results of their algorithm run against the fifteen ISPRS samples (Sithole and Vosselman, 2003), improving on Axelsson's algorithm for seven of fifteen samples. Although mean total error was not notably low (7.23%), this was largely the result of markedly poor performance on Sample 4–1, where total error was 34.03%. Median total error, which tends to discount the effects of such large outliers, was 3.61%, and was commensurate with Axelsson's progressive TIN densification method (median 3.63%).

2. Algorithm

2.1. Parameters

In order to gauge the baseline performance of a progressive morphological ground filtering algorithm, we developed a simplified model to determine how each part of the algorithm contributes to its overall performance. To this end, we designed an algorithm that requires four parameters in addition to the x, y, and z coordinates of the points in the original LIDAR data cloud: the cell size of the minimum surface grid, a percent slope value that governs the grid cell BE/OBJ classification at each step, a vector of window radii that controls the opening operation at each iteration, and a single elevation difference value that governs the ultimate classification of the LIDAR point as bare earth (BE) or object (OBJ) based on interpolated vertical distance to the minimum surface grid. A fifth, optional elevation scaling parameter assists in the identification of ground points from the provisional DTM, and is treated explicitly in the next section.

Both Zhang et al. (2003); Chen et al. (2007) used supplied parameters to generate exponentially increasing window sizes based on the cell size and the largest expected feature to be removed. We essentially adopt this approach, with the exception that the increase in window radius defaults to an increase of one pixel radius per iteration up to the maximum value that the user specifies (i.e., it increases slowly, and linearly), though the implemented algorithm (http://tpingel.org/code/smrf/smrf.html) is capable of taking user supplied vectors as well. In the following results section, we demonstrate the importance of increasing the window radius gradually. Zhang et al. (2003) noted a preference for this approach, though the extra computational expense associated with this method was apparently thought prohibitive at the time.

2.2. Methodology

The algorithm consists of four conceptually distinct stages (Fig. 1). The first is the creation of the minimum surface (ZI_{min}). The second is the processing of the minimum surface, in which grid cells from the raster are identified as either containing bare earth (BE) or objects (OBJ). This second stage represents the heart of the algorithm. The third step is the creation of a DEM from these gridded points. The fourth step is the identification of the original LIDAR points as either BE or OBJ based on their relationship to the interpolated DEM.

As with many other ground filtering algorithms, the first step is generation of ZI_{min} from the cell size parameter and the extent of the data. The two vectors corresponding to *[min:cellSize:max]* for each coordinate – xi and yi – may be supplied by the user or may be easily and automatically calculated from the data. Without supplied ranges, the SMRF algorithm creates a raster from the ceiling of the minimum to the floor of the maximum values for each of the (x,y) dimensions. If the supplied cell size parameter is not an integer, the same general rule applies to values evenly divisible by the cell size. For example, if cell size is equal to 0.5 m, and the x values range from 52345.6 to 52545.4, the range would be [52346 52545].

The SMRF technique is intended to apply to the first and last returns of the point cloud, though the minimum surface described in the following paragraph could be generated nearly as well with only the last returns. However, though the last return of any given pulse is the most likely to be ground, it need not be: one can imagine that the last return of one pulse could happen to hit an object at a given location, while the first return of another pulse might strike more near to the ground at the same location. In this case, the premature removal of the first return from the second pulse would introduce a small error into the DEM that any filter would have difficulty removing. For this reason, it is suggested that both first and last returns be used since the extraneous observations are soon removed during the generation of the initial grid.

The minimum surface grid ZI_{min} defined by vectors (xi,yi) is filled with the nearest, lowest elevation from the original LIDAR

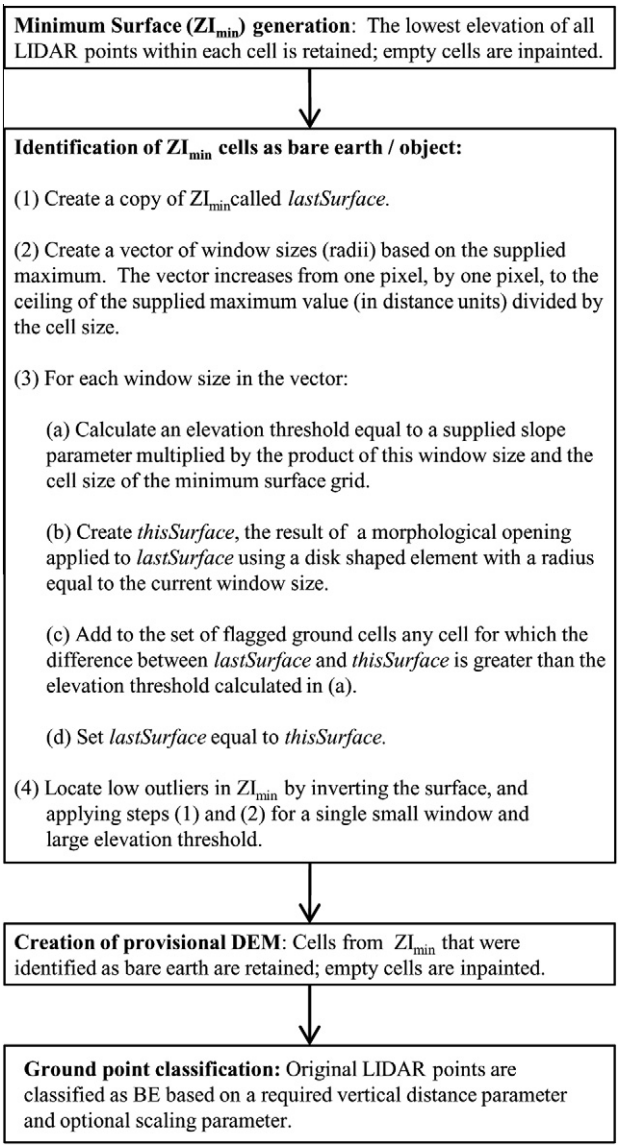

Fig. 1. Workow diagram of the SMRF algorithm.

point cloud (x,y,z) values, provided that the distance to the nearest point does not exceed the supplied cell size parameter. This provision means that some grid points of ZI_{min} will go unfilled. To fill these values, we rely on computationally inexpensive image inpainting techniques. Image inpainting involves the replacement of the empty cells in an image (or matrix) with values calculated from other nearby values. It is a type of interpolation technique derived from artistic replacement of damaged portions of photographs and paintings, where preservation of texture is an important concern (Bertalmio et al., 2000). When empty values are spread through the image, and the ratio of filled to empty pixels is quite high, most methods of inpainting will produce satisfactory results. In an evaluation of inpainting methods on ground identification from the final terrain model, we found that Laplacian techniques produced error rates nearly three times higher than either an average of the eight nearest neighbors or D'Errico's spring-metaphor inpainting technique (D'Errico, 2004). The spring-metaphor technique imagines springs connecting each cell with its eight adjacent neighbors, where the inpainted value corresponds to the lowest energy state of the set, and where the entire (sparse) set of linear equations is solved using partial differential equations. Both of these latter techniques were nearly the same with regards to total error, with the spring technique performing slightly better than the k-nearest neighbor (KNN) approach.

It is worthwhile to note that there are other possible methods of creating the initial surface (e.g., Hollaus et al., 2010). Notably, the maximum value, instead of the minimum, could be calculated for any given grid point. This maximum surface would appear much smoother to the eye when transformed into a hillshaded image, but would make the task of the progressive filter more difficult since permeable objects (like trees) would tend to remain in the image longer. The difference between the maximum and minimum surfaces tends to highlight edges (as well as vegetation), and initial work focused on using this surface to guide later stages of the algorithm. Unfortunately, the ISPRS LIDAR dataset samples were of lower resolution than many of those currently produced, and for this reason the difference surface proved unhelpful in ground classification of the ISPRS datasets at 1 m resolution.

The second stage of the ground identification algorithm involves the application of a progressive morphological filter to the minimum surface grid (ZI_{min}). At the first iteration, the filter applies an image opening operation to the minimum surface. An opening operation consists of an application of an erosion filter followed by a dilation filter. The erosion acts to snap relative high values to relative lows, where a supplied window radius and shape (or structuring element) defines the search neighborhood. The dilation uses the same window radius and structuring element, acting to outwardly expand relative highs. Fig. 2 illustrates an opening operation on a cross section of a transect from Sample 1–1 in the ISPRS LIDAR reference dataset (Sithole and Vosselman, 2003), following Zhang et al. (2003).

In this case, we selected a disk-shaped structuring element, and the radius of the element at each step was increased by one pixel from a starting value of one pixel to the pixel equivalent of the maximum value (wk_{max}). The maximum window radius is supplied as a distance metric (e.g., 21 m), but is internally converted to a pixel equivalent by dividing it by the cell size and rounding the result toward positive infinity (i.e., taking the ceiling value). For example, for a supplied maximum window radius of 21 m, and a cell size of 2 m per pixel, the result would be a maximum window radius of 11 pixels. While this represents a relatively slow progression in the expansion of the window radius, we believe that the high efficiency associated with the opening operation mitigates the potential for computational waste. The improvements in classification accuracy using slow, linear progressions are documented in the next section.

On the first iteration, the minimum surface (ZI_{min}) is opened using a disk-shaped structuring element with a radius of one pixel. An elevation threshold is then calculated, where the value is equal to the supplied slope tolerance parameter multiplied by the product of the window radius and the cell size. For example, if the user supplied a slope tolerance parameter of 15%, a cell size of 2 m per pixel, the elevation threshold would be 0.3 m at a window of one pixel ($0.15 \times 1 \times 2$). This elevation threshold is applied to the difference of the minimum and the opened surfaces. Any grid cell with a difference value exceeding the calculated elevation threshold for the iteration is then flagged as an OBJ cell. The algorithm then proceeds to the next window radius (up to the maximum), and proceeds as above with the last opened surface acting as the "minimum surface" for the next difference calculation.

The end result of the iteration process described above is a binary grid where each cell is classified as being either bare earth (BE) or object (OBJ). The algorithm then applies this mask to the starting minimum surface to eliminate nonground cells. These cells are then inpainted according to the same process described previously, producing a provisional DEM (ZI_{pro}).

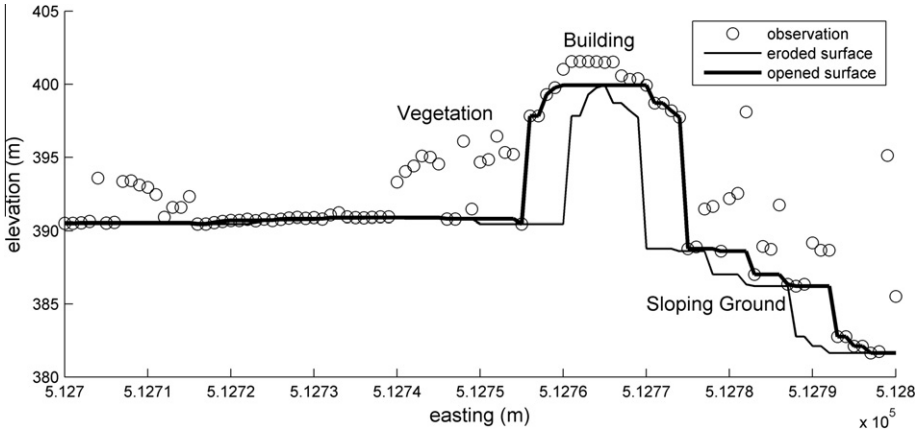

Fig. 2. An open operation applied to a transect [512700 5403805; 512800 5403805] from Sample 1–1 of the ISPRS reference dataset. Opening consists of an erosion operation (effectively pulling high values within a given space down to local minima) followed by a dilation operation (pulling low values to local maxima). This open operation used a ten meter linear structuring element, and removed all of the vegetation, while preserving the structure of the included building as well as the sloping ground surface on the right. It represents an early step in the progressive morphological filter's operation.

The final step of the algorithm is the identification of ground/object LIDAR points. This is accomplished by measuring the vertical distance between each LIDAR point and the provisional DEM, and applying a threshold calculation. While many authors use a single value for the elevation threshold, we suggest that a second parameter be used to increase the threshold on steep slopes, transforming the threshold to a slope-dependent value. The total permissible distance is then equal to a fixed elevation threshold plus the scaling value multiplied by the slope of the DEM at each LIDAR point. The rationale behind this approach is that small horizontal and vertical displacements yield larger errors on steep slopes, and as a result the BE/OBJ threshold distance should be more permissive at these points.

The calculation requires that both elevation and slope are interpolated from the provisional DEM. There are any number of interpolation techniques that might be used, and even nearest neighbor approaches work quite well, so long as the cell size of the DEM nearly corresponds to the resolution of the LIDAR data. A comparison of how well these different methods of interpolation perform is given in the next section. Based on these results, we find that a splined cubic interpolation provides the best results.

It is common in LIDAR point clouds to have a small number of outliers which may be either above or below the terrain surface. While above-ground outliers (e.g., a random return from a bird in flight) are filtered during the normal algorithm routine, the below-ground outliers (e.g., those caused by a reflection) require a separate approach. Early in the routine and along a separate processing fork, the minimum surface is checked for low outliers by inverting the point cloud in the *z*-axis and applying the filter with parameters (slope = 500%, maxWindowSize = 1). The resulting mask is used to flag low outlier cells as OBJ before the inpainting of the provisional DEM. This outlier identification methodology is functionally the same as that of Zhang et al. (2003).

The provisional DEM (ZI_{pro}), created by removing OBJ cells from the original minimum surface (ZI_{min}) and then inpainting, tends to be less smooth than one might wish, especially when the surfaces are to be used to create visual products like immersive geographic virtual environments. As a result, it is often worthwhile to reinterpolate a final DEM from the identified ground points of the original LIDAR data (ZI_{fin}). Surfaces created from these data tend to be smoother and more visually satisfying than those derived from the provisional DEM.

Very large (>40 m in length) buildings can sometimes prove troublesome to remove on highly differentiated terrain. To accommodate the removal of such objects, we implemented a feature in the published SMRF algorithm which is helpful in removing such features. We accomplish this by introducing into the initial minimum surface a "net" of minimum values at a spacing equal to the maximum window diameter, where these minimum values are found by applying a morphological open operation with a disk shaped structuring element of radius ($2^{*}wk_{max}$). Since only one example in this dataset had features this large (Sample 4–2, a trainyard) we did not include this portion of the algorithm in the formal testing procedure, though we provide a brief analysis of the effect of using this net filter in the next section.

The methodology described above shares much in common with previous implementations of progressive morphological filters (PMFs) (Chen et al., 2007; Zhang et al., 2003), but differs in several important details that make a large difference in how well SMRF handles complex terrain. First, it differs from both previous PMFs in the method of interpolating empty cells in ZI_{min}, for while Zhang uses a nearest neighbor approach, and Chen uses a similar approach with an added function to detect and fill large waterbodies, we use image inpainting techniques to produce a smoother grid from the outset. Since the initial grid is the fundamental input for the progressive filtering stage, differences in its constitution are quite important. Zhang's algorithm uses a somewhat more complex system of five parameters to control the growth of the window size and elevation threshold. In contrast, SMRF uses a linearly increasing window up to a maximum specified size, where the elevation threshold is controlled by a single parameter. This has the effect of making exploratory analysis much more simple, since the construction of the provisional DEM requires only two parameters (in addition to the required cell size of the DEM) – a maximum window radius and a slope parameter. Our testing indicates that not only is the SMRF method of controlling window sizes and elevation thresholds more simple, it is also more effective than Zhang's method. Chen's method of differentially controlling the window size and elevation threshold based on whether the filter is targeted at removing vegetation or buildings is somewhat different than either of these, though it is worthwhile to note that the increased number of parameters required for Chen's building filter renders it more difficult to use as an exploratory technique. While both Chen and Zhang utilize kriging (Wackernagel, 1998) to create the provisional DEM, SMRF uses an inpainting solution (D'Errico, 2004) derived from image processing to inpaint only missing values, and retains points from ZI_{min} not excluded by the filter. Finally, SMRF differs from either the Zhang or Chen filter in its inclusion of

a slope dependent elevation threshold for identifying ground points in the LIDAR data set from the provisional DEM. Thus, while SMRF is not a truly novel solution to the terrain classification problem, it represents the sum of a series of critical improvements in implementation.

3. Results

We tested the SMRF algorithm against all datasets using a combination of parameter values to find which single parameter set (applied to all fifteen samples) resulted in the highest mean Kappa score. These parameters define a baseline set of general parameters which could be expected to give a reasonable performance for any given sample, and as such could be used as the starting point to tune the performance of the SMRF algorithm. These parameters were: slope tolerance = 15%, maximum window radius (wk_{max}) = 18 m, elevationThreshold = 0.5 m, and elevationScalingFactor = 1.25. With these general parameters, the mean Kappa score was 85.40% (median, 90.52%) and the mean total error was 4.40% (median, 3.40%). Individual scores were generally quite good with two exceptions: Sample 4–1 had a high total error rate (10.79%), while Sample 5–3 had a low Kappa score (47.24%).

In order to measure the best-case performance of the simple progressive morphological filter, we systematically varied four inputs (slope, maximum window radius, and elevation threshold and scaling factor for ground identification) and measured the result on total error, Type I error, Type II error, and Cohen's Kappa. The cell size was fixed at one meter. Elevation was varied from 5% to 50% slope at 5% increments, and at one percent increments during a second, fine-tune optimization. Maximum window radius was varied from 1 to 40 m. Elevation thresholds were varied from 0 to 6.0 m at 5 cm increments. Elevation scaling factors were varied from 0 to 5 at increments of 0.05. Table 3 shows the optimized values for each of the fifteen samples and the associated Type I, Type II, total error and Kappa score (in percent), where best performance was determined by highest Kappa score. On this criterion, the mean Kappa value across all samples was 90.02 (median 91.81) and the mean total error was 2.97% (median 2.43). The results of the SMRF algorithm applied to Sample 1–1 are shown in Fig. 3.

Optimized parameters varied according to the type of scene, and the mean of each of the optimized parameters generally matched well with its corresponding value from the single parameter set. Correlation analysis using Pearson's coefficient indicates that the slope tolerance parameter generally increased as the maximum window radius decreased for these samples ($r(13) = -0.53$, $p = .042$). Similarly, samples which required large elevation threshold parameters tended to use small scaling factors ($r(13) = -0.82$, $p < .001$). These relationships are illustrated in Fig. 4. Notable outliers were Sample 4–2, which required a maximum window radius of 49, and Sample 5–3 that required a slope tolerance parameter of 45%. Sample 4–2 was a scene of trains in a railyard, featuring very long, linear features that are difficult to remove with iterative applications of disk-shaped elements. The "cut net" feature described above was designed to help in just such cases. Its application for Sample 4–2 with a net cut at 20 m intervals and run with a 20 m radius structuring element (unoptimized values) produced a nearly identical error profile (Kappa = 95.88%). In this case, since the terrain around the railyard is quite flat, a large structuring element did not yield the high error rates as one might normally expect in more differentiated terrain. Sample 5–3 used a very high slope threshold value (45%) and ultimately produced the lowest quality ground classification (Kappa = 68.12%) of any sample tested. Sample 5–3 is a scene taken from a quarry which features steep and highly terraced slopes. Most algorithms tested against this sample perform poorly, including Axelsson's (Kappa = 39.14%). In this case, there are relatively few objects, of typically small size to be removed, resulting in an optimized maximum window radius of only 3 m. Most of the error for this sample is concentrated on the terrace walls in the form of single pixels along the edge misclassified as either OBJ or BE. The large discontinuities in the surface due to terracing are most likely responsible for these misclassifications, since the inpainting routine oversmooths these features.

In order to further characterize parameter sensitivity for SMRF, we used the single parameter set as the basis to observe how small changes in the parameters would change the mean Kappa value across all fifteen samples (Fig. 5). Thus, while we held constant the parameters related to classifying ground points from the provisional DEM, we systematically varied slope threshold and the maximum window radius. In a second test, we held constant the

Table 3
Optimized and single parameter results of the SMRF algorithm when tested against the ISPRS reference dataset expressed in Type I error rate (T.I), Type II error rate (T.II), total error rate (T.E), and Kappa (K, %). Single parameter results were obtained by using values (.15, 18, 0.5, 1.25) for slope tolerance, window radius, elevation threshold and scaling factor, respectively. All results used a one meter cell size to generate the Digital Surface and Elevation Models.

Sample	Optimized								Single parameter	
	Slopetol. (dz/dx)	Window radius (m)	Elevation threshold (m)	Scaling factor	T.I (%)	T.I I (%)	T.E (%)	K (%)	T.E (%)	K (%)
1 (1–1)	0.20	16	0.45	1.20	7.88	8.81	8.28	83.12	8.64	82.40
2 (1–2)	0.18	12	0.30	0.95	2.57	3.30	2.92	94.15	3.10	93.80
3 (2–1)	0.12	20	0.60	0.00	0.26	4.07	1.10	96.77	1.88	94.43
4 (2–2)	0.16	18	0.35	1.30	2.57	5.07	3.35	92.21	3.40	92.07
5 (2–3)	0.27	13	0.50	0.90	3.21	6.17	4.61	90.73	6.48	87.02
6 (2–4)	0.16	8	0.20	2.05	2.25	6.90	3.52	91.13	4.19	89.49
7 (3–1)	0.08	15	0.25	1.50	0.39	1.52	0.91	98.17	2.48	95.00
8 (4–1)	0.22	16	1.10	0.00	3.64	8.17	5.91	88.18	10.79	78.41
9 (4–2)	0.06	49	1.05	0.00	0.27	1.98	1.48	96.48	2.93	93.07
10 (5–1)	0.05	17	0.35	0.90	0.59	4.44	1.43	95.76	3.00	90.74
11 (5–2)	0.13	13	0.25	2.20	3.09	10.08	3.82	81.04	4.17	78.80
12 (5–3)	0.45	3	0.10	3.80	1.18	31.97	2.43	68.12	7.41	47.24
13 (5–4)	0.05	11	0.15	2.30	2.51	2.05	2.27	95.44	3.67	92.65
14 (6–1)	0.28	5	0.50	1.45	0.51	10.70	0.86	87.22	2.02	75.38
15 (7–1)	0.13	15	0.75	0.00	0.99	6.84	1.65	91.81	1.85	90.52
				Mean	2.13	4.47	2.97	90.02	4.40	85.40
				Median	2.25	6.17	2.43	91.81	3.40	90.52
				Min	0.26	1.52	0.86	68.12	1.85	47.24
				Max	7.88	31.97	8.28	98.17	10.79	95.00
				Std	1.99	7.37	2.07	7.85	2.70	12.34

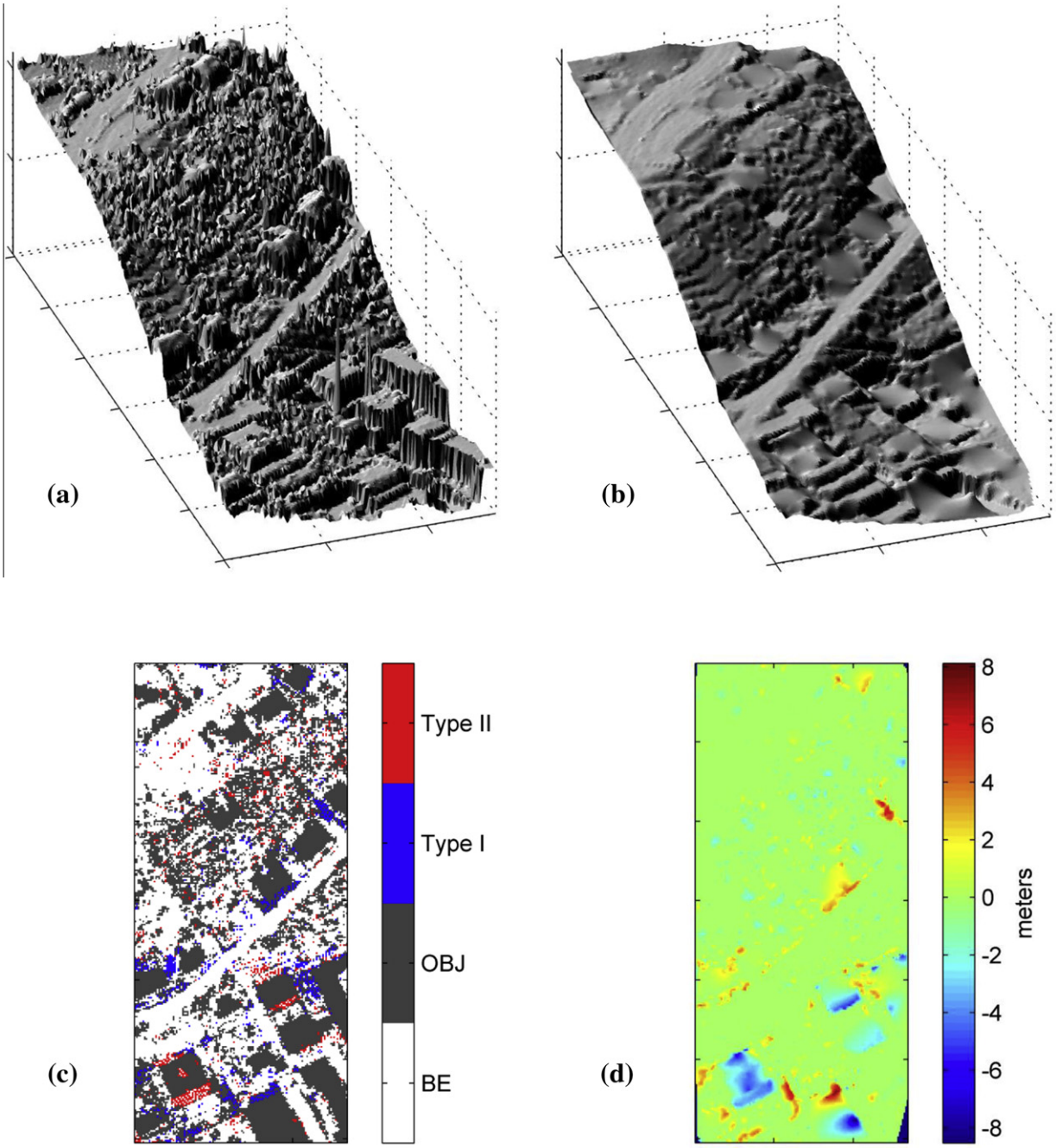

Fig. 3. Performance of SMRF filter on Sample 1–1: (a) ZI_{min}, (b) produced DEM, (c) spatial distribution of Type I and Type II errors, (d) difference between DEM estimated from reference data and the produced DEM. Tick marks and grid lines on all subfigures are at 50 m intervals.

parameters related to identifying cells in the minimum surface as BE/OBJ while systematically varying the elevation threshold and the slope scale factor. The pattern of error reveals that maximum window radius should typically be greater than 10 m and that slope should typically be above 10% to expect a mean accuracy above 80%. Accuracy rates proved much more sensitive to the elevation threshold (in meters) than to the scale factor, indicating that the slope dependent threshold adds only marginal value.

These two methods (i.e., a single parameter set and an optimized parameter set) of evaluation provide a good means to compare the performance of the SMRF algorithm with other published algorithms. Meng et al. (2009) cite a mean Kappa score of 76.7% when using a two parameter-set solution, and 79.9% when using an optimized solution. Axelsson's algorithm (1999) effectively uses seven parameter sets to achieve a mean total error of 4.82% (Kappa, 84.19%). The algorithm of Chen et al. (2007) also used seven parameter sets to achieve a mean total error of 7.23%. Shao and Chen (Shao, 2007; Shao and Chen, 2008) did not optimize their algorithm, but found that the best performing single parameter set achieved a mean total error rate of 4.42%; if the best mean parameter set is used for each sample, the mean overall total error was 4.20%, though this represents only a weak optimization metric. The single parameter set evaluation method produced error rates that were below Axelsson (1999) for nine of fifteen samples, and better for thirteen samples when the fully optimized parameter sets were used. Whether evaluated by single parameter set or full

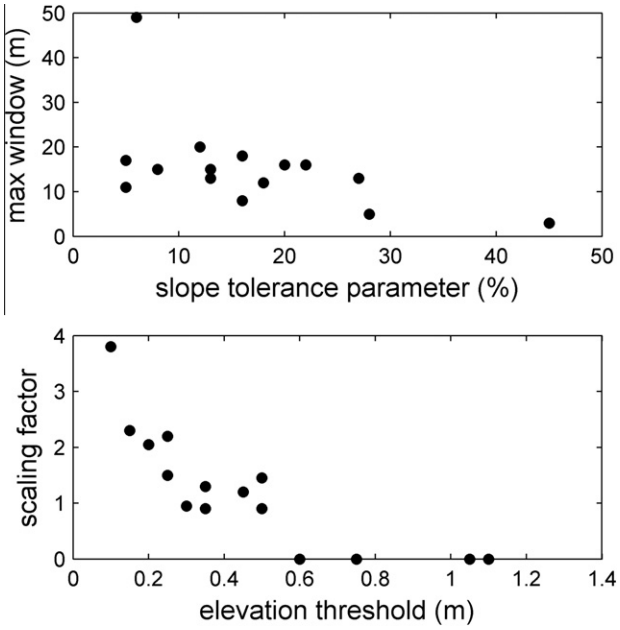

Fig. 4. Relationships between parameters for the fully optimized samples. Slope tolerance was inversely related to maximum window radius (top) and elevation threshold was inversely related to scaling factor (bottom).

optimization, SMRF compares favorably to these results, and establishes that a well-implemented progressive morphological filter can be a useful tool for terrain classification of airborne LIDAR data.

3.1. Analysis of alternative subroutines

The creation of the initial grid is the first step in many ground filtering algorithms, and the previous section described several methods of generating such surfaces. In our development of SMRF, we found that creating a minimum surface improved ground filtering over the creation of a surface where the nearest, highest value was used (i.e., a maximum surface, which had a mean Kappa score of 89.29%). The same pattern held true for mean (89.15%) and median (89.55%) surfaces. The infilling technique described by Chen et al., 2007, in which the boundaries of large unfilled areas are filled according to the lowest value along the periphery also had a lower Kappa score than simple inpainting (87.61%, when infilling was applied to gaps larger than 250 m^2), though the authors note that the approach is only intended to apply to gaps caused by water bodies and not to other gaps in the data.

The progressive nature of the algorithm, in which a series of opening operations are applied with successively larger structuring elements (or windows), traces back to Kilian et al. (1996) and Zhang et al. (2003). The value of using progressively larger windows, as opposed to a single value, is shown in Fig. 6. As the window radius gets larger, Type II errors are reduced at a slight cost of an increased number of Type I errors. This approach works well not only to reduce quantitative error, but to greatly improve the visual product of the final DEM. This is because Type II errors tend to be quite visually disruptive: any point on a building mistaken for ground will severely distort its local area in the final appearance of the DEM.

We advocate that progressive morphological filters use a linearly increasing window in which the opening operation starts with a disc-shaped structuring element of one pixel radius and where the radius increases by one pixel per iteration until the maximum window radius is reached. One alternative strategy is an exponentially increasing window radius (Zhang et al., 2003). There are two ways in which such a strategy might reasonably be implemented. First, the series might be multiplied at each step by a given factor starting from one, and where the maximum window radius is appended at the end of the series if not already present (e.g., [1 2 4 8 16 20] for a factor of two and maximum window radius of 20). Alternatively, the series might be generated in reverse by halving the size starting from the maximum, and rounding toward positive infinity (e.g., [1 2 3 5 10 20]).

We tested each of these two strategies by using both the optimized parameter set and the single parameter set for each sample. Window radii were generated as above, with a base of two in each case. The reverse exponential generation strategy outperformed the forward generation strategy with respect to total error, Kappa, and number of samples with performance exceeding that of the slow linear opening strategy. We therefore confine our continued discussion of the exponential opening strategy to the reverse generation variety. The results for this test are given in Table 4.

For both the optimized and single-parameter tests, the exponential opening (EO) strategy was worse than the slow linear opening (SLO) strategy with respect to mean total error and Kappa. The exponential opening on the optimized set (EO_{opt}) featured a mean Kappa score of 88.64% (compared to 90.02% for SLO_{opt}), a mean total error of 3.32% (compared to 2.97% for SLO_{opt}), and outperformed SLO_{opt} on only 3 of 15 samples. Comparisons with respect to Kappa and total error were similar for the single parameter test, although the EO_{sp} set outperformed SLO_{sp} on 7 of 15 samples. The improve-

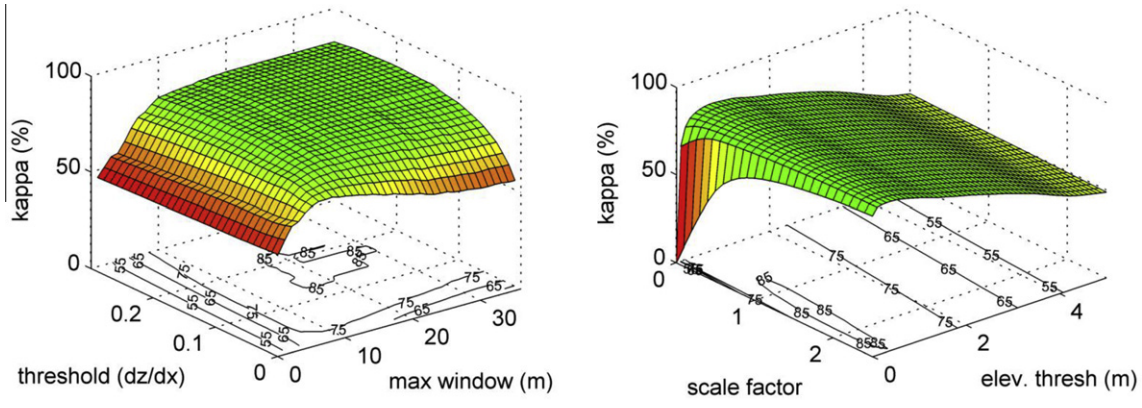

Fig. 5. Parameter sensitivity surfaces, where Kappa score is equal to the mean performance on all fifteen samples. Overall performance drops precipitously when the slope threshold is below .1 and when the maximum window radius is less than 10 m. The highest Kappa values are located when the elevation threshold is set to. 5 m, and perform marginally better with the inclusion of a modest scale factor (i.e., transforming the threshold to a slope-dependent value).

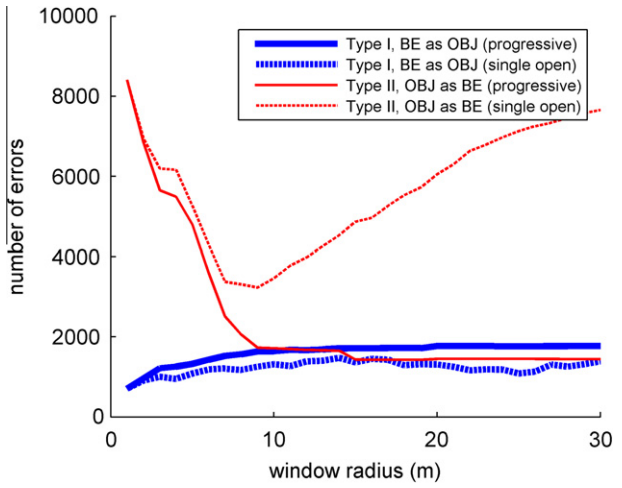

Fig. 6. Type I (BE as OBJ, blue) and Type II (OBJ as BE, red) errors shown for progressive versus single opening on Sample 1–1 of ISPRS data set. Progressive opening greatly reduces Type II error at a slight cost of increased Type I error as window sizes get larger. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

Table 4
Optimized and single parameter results of the SMRF algorithm when using exponentially increasing window radii in place of the slow linear opening method. All other parameters were equal to the values described in Table 3. Exponential opening was superior to slow linear opening in three samples (3,4,8) for the optimized set, and in seven samples (2,3,4,6,7,8,10) for the single parameter set, though mean and median performance on total error and Kappa across all samples were worse.

Sample	Optimized				Single parameter			
	T.I	T.II	T.E	K	T.I	T.II	T.E	K
1 (1–1)	8.59	9.11	8.81	82.04	9.61	8.19	9.00	81.72
2 (1–2)	2.67	3.23	2.94	94.11	1.89	4.10	2.97	94.05
3 (2–1)	0.26	3.97	1.08	96.84	0.23	5.77	1.46	95.70
4 (2–2)	2.30	5.11	3.17	92.61	2.09	5.70	3.22	92.48
5 (2–3)	4.51	6.17	5.30	89.36	9.51	5.22	7.48	85.03
6 (2–4)	2.52	7.14	3.79	90.47	3.35	6.32	4.16	89.63
7 (3–1)	0.42	1.56	0.95	98.10	0.12	4.63	2.20	95.56
8 (4–1)	3.77	4.67	4.22	91.56	15.42	3.07	9.23	81.53
9 (4–2)	0.29	2.02	1.51	96.39	1.74	3.59	3.05	92.79
10 (5–1)	0.97	4.57	1.76	94.82	0.08	12.91	2.88	91.16
11 (5–2)	5.25	8.51	5.59	74.38	5.59	10.46	6.10	72.14
12 (5–3)	1.18	31.97	2.43	68.12	11.07	6.98	10.91	36.81
13 (5–4)	2.91	2.21	2.53	94.91	0.93	6.27	3.80	92.39
14 (6–1)	0.53	10.78	0.88	86.99	3.48	4.23	3.51	63.62
15 (7–2)	4.87	4.80	4.86	78.84	4.51	7.12	4.81	78.67
Mean	2.74	7.05	3.32	88.64	4.64	6.30	4.99	82.89
Median	2.52	4.80	2.94	91.56	3.35	5.77	3.80	89.63
Min	0.26	1.56	0.88	68.12	0.08	3.07	1.46	36.81
Max	8.59	31.97	8.81	98.10	15.42	12.91	10.91	95.70
Std	2.35	7.39	2.18	8.94	4.68	2.63	2.89	15.76

ment on these samples were typically small, while the differences on samples for which it underperformed were typically larger, resulting in the same overall pattern for total error and Kappa observed for the optimized set. These results, taken with the illustration of the difference between slow linear opening and single opening described in Fig. 6 demonstrate that in most cases, slow linear opening will result in improved performance.

The algorithm proceeds from the initial minimum surface (ZI_{min}) to create an opened surface at the first iteration. From this point onward, until the maximum window radius is reached, the SMRF algorithm uses the opened surface from the previous iteration to create a new opened surface at the present iteration. One alternative to this method is to use the original minimum surface as the basis for the opening operation at each iteration. While this did not greatly impact performance, it did result in a lower mean Kappa score (89.79%, median 92.00%) and greater mean total error (3.06%, median 2.44%), where the change was due almost entirely to increased Type I error.

The final step in the algorithm is the identification of the original LIDAR points as either BE or OBJ depending on the vertical distance to the provisional DEM. Because these LIDAR points do not lie on the grid, the provisional DEM must be interpolated at each LIDAR point to calculate the elevation and slope of the DEM at that location. A splined cubic interpolation provided the best quantitative performance, while Kappa scores for nearest neighbor (89.49%), linear (89.36%), and non-spline cubic (89.32%) interpolations were slightly lower.

The strength of the SMRF algorithm lies in its ability to retain a large number of BE points by minimizing Type I error. Fig. 3c illustrates the distribution of Type I and Type II errors on Sample 1–1. For this sample, Type I error was 7.88%, while Type II error was 8.81%. Over all fifteen samples, SMRF had a mean Type I error rate of 2.13% (median, 2.25%) and a mean Type II error rate of 7.47% (median 6.17%). In effect, the SMRF algorithm retains fine details in the terrain, such as footpaths, without exacting high cost in the form of Type II errors. This characteristic of the SMRF algorithm makes it highly useful as the basis of terrain for immersive virtual environments. The retention of detail means that surfaces developed from SMRF DEMs are more likely to be useful with less additional post-processing required. In contrast, Type II errors will tend to be hidden in the virtual environment, at least for urban landscapes, since other models will likely be placed on the terrain at these locations.

As a matter of comparison, Axelsson's algorithm achieved a slightly better overall mean Type II error rate (7.46%) but had a significantly higher Type I error rate (5.55%). Chen et al. (2007) reported Type I and II error rates for Sample 1–1 only, showing an improvement over SMRF on Type II error (6.85%) but with a much higher Type I error rate (19.18%). Shao (2007), in contrast, had a balanced distribution between Type I error (4.77%) and Type II error (6.35%).

4. Discussion

The Simple Morphological Filter (SMRF) algorithm was developed to solve two problems. First, it was designed to be competitive with other ground filtering algorithms for LIDAR data, particularly with regard to urban environments on highly differentiated terrain. It was successful in that it improved on previous work with regard to quantitative performance, achieving the highest mean Kappa and lowest mean total error scores of any published algorithm run against all fifteen ISPRS samples of which we are aware. The SMRF algorithm is successful not only when optimized, but even when using a single set of parameters against all samples, suggesting that novice users can achieve good results with it.

The second contribution of SMRF, perhaps more important than the first, is that it establishes a baseline performance for a progressive morphological filter implemented in its simplest form. The essence of the SMRF algorithm requires the input of a minimum surface and two parameters – a maximum window radius that corresponds to the largest feature to be removed, and a single slope parameter that governs the cell-based ground / non-ground flagging at each iteration. With these two parameters and a supplied minimum surface, the central subroutine of SMRF produces a provisional ground surface (DTM) that is then used to classify the original LIDAR points as bare earth (BE) or object (OBJ). The real contribution of SRMF is that it provides a conceptually and compu-

tationally simple basis to achieve good results, while establishing a baseline performance for progressive morphological filters against which future proposed enhancements can be measured.

One important limitation of the results presented here involves the comparison to other terrain classification algorithms. The algorithms originally tested against the ISPRS dataset are at a distinct disadvantage in that subsequent authors have been able to develop algorithms against a reference dataset where BE/OBJ classifications have been made manually. This means that such filters, including SMRF, can be optimized to produce better quantitative results. Our approach has been to establish not only maximum expected performance (as the result of optimization) but also the results of a single parameter test that establish a meaningful "maximin" performance. Any solution involving non-optimization of parameter values merely indicates sub-maximum performance, without providing a particularly useful estimate as to what the maximum performance might be. Such optimization does not mean that ground-truthing or training is a critical component of the application of the algorithm – in fact the relatively small difference between SMRF's performance on a single parameter set (mean Kappa = 85.4%) versus the fully optimized set (mean Kappa = 90.02%) indicates that the algorithm robustly handles most scenes. This argument does imply, however, that comparisons between optimized and non-optimized results should be made with the knowledge that the results derived from non-optimization methodologies could be slightly, though perhaps not greatly, improved.

While SMRF performs well overall, future work will address Type II errors. Type II errors can be particularly damaging for products that are meant to be used visually, since objects mistaken for ground tend to create bulges and other artifacts that can be distracting. Since our purpose is to use such surfaces for immersive geographic virtual environments, this issue is of particular concern to us. However, the SMRF algorithm performs exceptionally well with regard to Type I error, and thus retains many ground points that give shape and character to the terrain. We have found, for instance, that visually significant but subtle ground features like footpaths tend to be retained very well. In this sense the SMRF algorithm greatly contributes to the potential for automatic generation of virtual environments, since important features for which there may not be any existing data are included in the models at no additional generation cost. Similarly, the retention of such features may significantly add to their value when applied to other problems, such as the modeling of human movement in rugged terrain (Pingel, 2010).

Acknowledgments

This study was supported by the IC Postdoctoral Fellowship Program (Grant #HMN1582-09-1-0013). The authors also wish to thank three anonymous reviewers, each of whom provided comments that directed important improvements in the manuscript.

References

Axelsson, P., 1999. Processing of laser scanner data – algorithms and applications. ISPRS Journal of Photogrammetry and Remote Sensing 54 (2–3), 138–147.

Axelsson, P., 2000. DEM generation from laser scanner data using adaptive TIN models. International archives of photogrammetry. Remote Sensing and Spatial Information Sciences 33 (Part B4), 110–117.

Bertalmio, M., Guillermo, S., Caselles, V., Ballester, C., 2000. Image inpainting. In: Proceedings of the 27th Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH 2000), New Orleans, LA, 23–28 July, pp. 417–424.

Chen, Q., Gong, P., Baldocchi, D., Xie, G., 2007. Filtering airborne laser scanning data with morphological methods. Photogrammetric Engineering and Remote Sensing 73 (2), 175–185.

Cohen, J., 1960. A coefficient of agreement for nominal scales. Educational and Psychological Measurement 20 (1), 37–46.

Congalton, R., 1991. A review of assessing the accuracy of classifications of remotely sensed data. Remote Sensing of Environment 37 (1), 35–46.

D'Errico, J., 2004. Inpaint_nans.m. <http://www.mathworks.com/matlabcentral/fileexchange/4551>. (accessed 7.06.11).

Elmqvist, M., Jungert, E., Lantz, F., Persson, A., Söderman, U., 2001. Terrain modelling and analysis using laser scanner data. International archives of photogrammetry. Remote Sensing and Spatial Information Sciences 34 (Part 3/W4), 219–226.

Hollaus, M., Mandlburger, G., Pfeifer, N., Mücke, W., 2010. Land cover dependent derivation of digital surface models from airborne laser scanning data. International archives of photogrammetry. Remote Sensing and Spatial Information 38 (Part 3A), 221–226.

Jahromi, A.B., Zoej, M.J.V., Mohammadzadeh, A., Sadeghian, S., 2011. A novel filtering algorithm for bare-earth extraction from airborne laser scanning data using an artificial neural network. IEEE Journal of Selected Topics in Applied Earth Observations and Remote Sensing 4 (4), 836–843.

Jensen, J.R., 2005. Introductory Digital Image Processing: A Remote Sensing Perspective. Prentice Hall, New York.

Kilian, J., Haala, N., Englich, M., 1996. Capture and evaluation of airborne laser scanner data. International archives of photogrammetry. Remote Sensing and Spatial Information Sciences 31 (Part B3), 383–388.

Kraus, K., Pfeifer, N., 1998. Determination of terrain models in wooded areas with airborne laser scanner data. ISPRS Journal of Photogrammetry and Remote Sensing 53 (4), 193–203.

Liu, X., 2008. Airborne LiDAR for DEM generation: some critical issues. Progress in Physical Geography 32 (1), 31–49.

Meng, X., Currit, N., Zhao, K., 2010. Ground filtering algorithms for airborne LiDAR data. A Review of Critical Issues 2 (3), 833–860.

Meng, X., Wang, L., Silván-Cárdenas, J.L., Currit, N., 2009. A multi-directional ground filtering algorithm for airborne LIDAR. ISPRS Journal of Photogrammetry and Remote Sensing 64 (1), 117–124.

Pfeifer, N., Reiter, T., Briese, C., Rieger, W., 1999. Interpolation of high quality ground models from laser scanner data in forested areas. International archives of photogrammetry. Remote Sensing and Spatial Information Sciences 32 (Part 3/W14), 31–36.

Pingel, T., 2010. Modeling slope as a contributor to route selection in mountainous areas. Cartography and Geographic Information Science 37 (2), 137–148.

Shan, J., Toth, C.K. (Eds.), 2008. Topographic Laser Ranging and Scanning: Principles and Processing. CRC Press, Boca Raton.

Shao, Y.C., 2007. Ground point selection and building detection from Airborne LiDAR Data. PhD thesis. National Central University.

Shao, Y.C., Chen, L.C., 2008. Automated searching of ground points from airborne lidar data using a climbing and sliding method. Photogrammetric Engineering and Remote Sensing 74 (5), 625–635.

Silván-Cárdenas, J., Wang, L., 2006. A multi-resolution approach for filtering LiDAR altimetry data. ISPRS Journal of Photogrammetry and Remote Sensing 61 (1), 11–22.

Sithole, G., Vosselman, G., 2003. Comparison of filtering algorithms. International archives of photogrammetry. Remote Sensing and Spatial Information Sciences 34 (Part 3/W13), 71–78.

Sithole, G., Vosselman, G., 2004. Experimental comparison of filter algorithms for bare-earth extraction from airborne laser scanning point clouds. ISPRS Journal of Photogrammetry and Remote Sensing 59 (1–2), 85–101.

Vosselman, G., 2000. Slope based filtering of laser altimetry data. International archives of photogrammetry. Remote Sensing, and Spatial Information Sciences 33 (Part B3), 935–942.

Vosselman, G., Maas, H.G. (Eds.), 2010. Airborne and Terrestrial Laser Scanning. CRC Press, Boca Raton.

Wackernagel, H., 1998. Multivariate Geostatistics. Springer-Verlag, Berlin.

Zhang, K., Chen, S., Whitman, D., Shyu, M., Yan, J., Zhang, C., 2003. A progressive morphological filter for removing nonground measurements from airborne LIDAR data. IEEE Transactions on Geoscience and Remote Sensing 41 (4), 872–882.